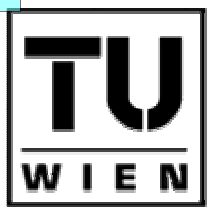


TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

MASTERARBEIT

„Migration zu einer proaktiv betreuten Server-Infrastruktur auf Open Source-Basis“

Ausgeführt am Institut für

Informationssysteme
der Technischen Universität Wien

unter der Anleitung von Dr. Karl M. Göschka

durch

Dipl.-Ing. (FH) Jürgen Neustifter
Gatterburgstraße 9
A-2070 Retz

Wien, 18. Juli 2007

Danksagung

Ich möchte mich an dieser Stelle herzlich bei all jenen Personen bedanken, die mich während meiner Studien und vor allem während dem Verfassen dieser Arbeit tatkräftig seelisch und freundschaftlich unterstützt haben, gerade weil dadurch die Freizeit neben meinem Job noch knapper geworden ist.

Inhaltsverzeichnis

Danksagung 3
Kurzfassung 6
Einleitung 7
1 Grundlagen 1
1.1 Monitoring 1
1.1.1 Definition des Begriffes „Monitoring“ 1
1.2 Reaktiv versus proaktiv 2
1.3 Was ist Monitoring im IT-Bereich? 3
1.4 Standards und Begriffe 5
1.4.1 Simple Network Management Protocol 5
1.4.2 Syslog 11
1.4.3 Arten der Informationsgewinnung 12
1.4.4 Begriffsdefinitionen 16
2 Wozu Monitoring? 22
2.1 Ausfälle 22
2.2 Auswirkungen (ungeplanter) Ausfälle 23
2.3 Ursachen ungeplanter Ausfälle 25
2.4 Nutzen von Monitoring? 27
2.5 Kosten versus Verfügbarkeit - Was kostet Monitoring? 29
2.6 Was kann Monitoring nicht? 30
3 Monitoring-Systeme 31
3.1 Prinzipieller Aufbau 32
3.2 Monitoring-Strategien 34
3.2.1 Blackbox-Monitoring 34
3.2.2 Whitebox-Monitoring 34
3.2.3 Messwerte 34
4 Migrationsprojekt des Unternehmens 35
4.1 Aktuelle Situation im System-Management-Bereich 35
4.1.1 Typischer Ablauf bei Systemausfall ohne Monitoring 35
4.2 Ziel-Definition 36
4.3 Teilprojektziel - Masterarbeit 38
4.4 Grober Projektplan 38
4.5 Ist-Analyse - Inventur 40
4.5.1 Produktions-Systeme 41
4.5.2 Test-Systeme 41
4.5.3 Kategorisierung der Systeme 41
4.5.4 Mögliche Auswirkungen von Ausfällen – Risikoeinschätzung 44
4.5.5 Netzwerkdiagramm – Produktionssysteme 46
4.6 Anforderungsanalyse - Produktionssysteme 46
4.6.1 Zielgruppe - Für wen ist das Monitoring? 46
4.6.2 Aufbereitung der Informationen 47
4.6.3 Welche Systeme und Devices müssen überwacht werden? 47
4.6.4 Welche Objekte der Systeme müssen überwacht werden? 47
4.6.5 Alarmierung bei Problemen 48
4.6.6 Reports und Statistiken 49

4.6.7 Allgemeine Überlegungen 49
4.6.8 Zusammenfassung der Anforderungen und Funktionalitäten des Monitoring-Systems, Entscheidungen 53
4.7 Produktevaluierung 55
4.7.1 Kommerzielle Systeme 55
4.7.2 Open Source-basierende Systeme 58
4.7.3 Fazit – Produktentscheidung 64
4.8 Produkttest – Konfiguration – Customization 65
4.8.1 Konfiguration 67
4.8.2 Reports erstellen und testen 74
4.8.3 Dokumentation erstellen 75
4.8.4 Schulung, Produkt- und Lösungsvorstellung 75
4.8.5 Abschluss und Übergabe an Produktion 76
4.9 Projektsummary 76
4.10 Zusammenfassung der Produkte 77
5 Related Work 77
6 Conclusio und Diskussion 78
6.1 Wichtige Entscheidungs- und Diskussionspunkte für die Umsetzung 81
6.2 Welche konkreten Probleme traten während den Arbeiten auf? 83
7 Next Steps 84
8 Future Work 84
Abbildungsverzeichnis 86
Tabellenverzeichnis 87
Abkürzungsverzeichnis 88
Literaturverzeichnis 92
Anhang A 96
Auswirkungen von Downtimes 96
Anhang B 99
Syslog - Facilities und Severities laut RFC 3164 99
Sourcecode Beispiel Windows-Eventlog-Überwachung - Eventlog-to-Syslog: 99
Konfigurationen 103

Kurzfassung

„Vorbeugen ist besser als heilen“ – dieser Leitsatz ist in der IT-Welt genauso gültig wie in der Medizin. Die IT-Abteilungen müssen sich heute als professionelle Dienstleister präsentieren und serviceorientiert denken, dabei sind höchstmögliche Verfügbarkeit der Systeme und die Einhaltung von Service Level Agreements zwingende Forderungen.

Proaktives (Erkennen von Problemen bevor es zu Fehlern kommt) Monitoring ermöglicht ein rechtzeitiges Agieren bei Nicht-Normzuständen und eine Verminderung von Ausfällen. Durch sinnvolles Monitoring sollen die Anzahl und die Dauer von ungeplanten Ausfällen reduziert bzw. vermindert und dadurch negative Auswirkungen auf das Unternehmen wie geschädigtes Image, finanzielle Einbußen und Verschlechterung der Produktivität so weit wie möglich vermieden werden.

Für ein fiktives Unternehmen, welches auf realen Rahmenbedingungen basiert, werden die Analyse- und die Konzeptionsphase eines Migrationsprojektes zu einer proaktiv betreuten Server-Infrastruktur erarbeitet und aufbereitet. Während der Konzeptionsphase werden verschiedene am Markt befindliche Monitoring-Lösungen analysiert und den Anforderungen gegenübergestellt. Als Ergebnis steht ein fertiges und produktionsreifes Konzept für die Migration zu einem proaktiv arbeitendem Monitoring-System auf Open Source-Basis.

Der erforderliche Umsetzungsaufwand eines solchen Projektes ist nicht zu unterschätzen und tritt bei freien als auch bei kommerziellen Systemen gleichermaßen auf. Allein die Lizenzkosten fallen bei den freien Systemen nicht an – Support ist in den meisten Fällen auch kostenpflichtig. Heute verfügbare Open Source-Produkte erfüllen in vielen Bereichen die Anforderungen an ein Monitoring-System ohne Probleme.

Einleitung

Die Informationstechnologie (IT) wird für Unternehmen zu 100 Prozent geschäftskritisch, wodurch die Anforderungen an die IT-Systeme kontinuierlich steigen:

- maximale Ausfallssicherheit (Verfügbarkeit)
- „stufenlos mitwachsende“ Systeme (Skalierbarkeit)
- optimale Benutzerfreundlichkeit (System-Management)
- Sicherheit in allen Bereichen (Sicherheit)

Jeder Ausfall der IT kostet - ein Ausfall wirkt sich zumeist negativ auf die Produktivität aus. Gerade in Zeiten hohen Konkurrenzdruckes und in denen die verfügbaren Ressourcen optimal eingesetzt werden müssen, ist ein Ausfall nicht nur ärgerlich, sondern in den meisten Fällen auch kostspielig. Wie in der Medizin ist hier der Leitsatz „`Vorbeugen ist besser als heilen`“ anwendbar - einer der Hauptpunkte von Monitoring.

Der Begriff „Monitoring“ wird in aktueller Zeit oftmals im Zusammenhang mit Service Level Agreements (SLAs) genannt. Vielerorts wird es als „Modewort“ gehandelt - was soll man unter „Monitoring“ verstehen? Großteils wird „Monitoring“ mit enormen Kosten in Verbindung gebracht – dabei werden allerdings oft der mögliche und vor allem der erwartete Nutzen eines ganzheitlichen Überwachungssystems außer Acht gelassen.

IT-Abteilungen müssen sich zukünftig verstärkt als professionelle Dienstleister mit starker Kundenorientierung behaupten und positionieren. Dazu müssen effektive und effiziente IT-Service-Prozesse mit hochwertigen IT-Services einhergehen. Bei der Optimierung der IT-Prozesse hilft ITIL (IT Infrastructure Library) – laut computerwoche.de (25.11.2005 – „Fast jeder kennt ITIL“) kennen bereits 82 Prozent der Befragten IT-Entscheider den de facto-Standard für das IT-Service-Management.

Das Hauptziel dieser Arbeit sind die Dokumentation und Begleitung der einzelnen Schritte der Analyse- und der Konzeptionsphase des Migrationsprojektes zu einer proaktiv betreuten Server-Infrastruktur eines fiktiven Unternehmens und deren Ergebnisse bishin zur Produktivsetzung des fertigen Konzeptes. Das betrachtete Unternehmen ist rein fiktiv, basiert allerdings auf Erfahrungen und Ergebnissen eines real existierenden Unternehmens und

dessen IT-Infrastruktur. Das Ergebnis stellt ein produktionsreifes Konzept für eine Monitoring-Lösung auf Basis von Open Source-Produkten dar. Dazu werden die grundlegenden Begriffe abgegrenzt und die Vorteile von Monitoring den Aufwänden grob gegenübergestellt.

Es werden in vielen Fällen die englischen Bezeichnungen verwendet, da diese sich vielerseits in der Fachwelt etabliert haben bzw. als „eingedeutscht“ verwendet werden. Großteils werden die deutschen Fachausdrücke in Klammern angegeben.

Markennamen sind eingetragene Warenzeichen der jeweiligen Unternehmen.

Alle Bezeichnungen sind geschlechtsneutral zu verstehen.

1 Grundlagen

1.1 Monitoring

1.1.1 Definition des Begriffes „Monitoring“

Für ein gemeinsames Verständnis ist der Begriff „Monitoring“ zu definieren.

Die lateinischen Ursprünge des Wortes „monitor“ bedeuten „warnen“.
Warnen bedeutet, jemandem oder etwas über eine bevorstehende Gefahr oder Situation Informationen zukommen zu lassen, um entsprechende präventive Maßnahmen setzen zu können.

Nach [Smith R. F., 2005] (Computer Science) gibt es folgende Definitionen für „monitor“:
„monitor“ is a program that observes, supervises, or controls the activities of other programs.

1. To keep track of systematically with a view to collecting information.
2. To test or sample, especially on a regular or ongoing basis.
3. To keep close watch over; supervise.

Monitoring is defined as the periodic oversight of a process, or the implementation of an activity, which seeks to establish the extent to which input deliveries, work schedules, other actions and targeted outputs are proceeding according to plan, so that timely action can be taken to correct the deficiencies detected. [who.int, 2006]

Monitoring ist das ständige und sorgfältige Untersuchen und Überwachen und Beobachten einer bestimmten Situation oder Gegebenheit. [Wahrig-Burfeind, 2003]

Monitoring ist der Überbegriff für 3 Begriffe im Deutschen:

- Beobachtung (Observation) – Erfassung eines Zustandes
- Überwachung (Detection) – besondere Beobachtung eines Objektes
- Kontrolle (Control) – Vergleich zwischen Ist- und Soll-Zustand

[www.at-mix.de, 2007]

Monitoring allgemein betrachtet ist die systematische Beobachtung über einen bestimmten Zeitraum oder auf Dauer – mit den Zielen der Einhaltung von Vorgabe-Werten und um positive oder negative Veränderungen im Zeitverlauf zu erkennen. Es ist das periodische Überwachen von Objekten und deren Zuständen, sodass ein rechtzeitiges Agieren bei nicht-Normzuständen möglich und ein aktueller Informationsstand über den Zustand bzw. die Zustände im System gegeben ist. Ungeachtet dessen, dass Monitoring ein essentieller Bestandteil der IT-Infrastruktur ist, haben verschiedene Personen verschiedenen Bedarf an Informationen über die Systeme.[olev.de, 2007]

1.2 Reaktiv versus proaktiv

Jede Maschine, jedes System, jeder Server und jede Komponente einer IT-Landschaft wird gemanagt und in Stand gehalten. Der Unterschied zwischen reaktiv und proaktiv besteht darin, ob von der zuständigen Stelle erst eine Aktivität gesetzt wird, wenn der betroffene Personenkreis ein Problem meldet, oder ob der Administrator immer über den Status seiner Systeme Bescheid weiß und bereits vor der Problemmeldung eine entsprechende Behebungsaktivität bzw. sogar eine präventive Maßnahme zur Problemvermeidung setzen kann oder nicht.

> **Reaktion** ist eine Aktion nach dem Eintreten eines Ereignisses.
> **Proaktiv** ist eine Aktion vor dem Eintritt des Ereignisses.

Im ersten Fall **reagiert** der Administrator auf eine Informationsmeldung des Betroffenen. Im zweiten Fall werden eventuelle Probleme versucht zu vermeiden (rechtzeitiger Austausch von Komponenten bei Überschreiten eines definierten Schwellwertes) oder schneller auf Ereignisse zu reagieren.

Kommentar: Im Grunde genommen steht auch im zweiten Fall die Reaktion im Vordergrund, allerdings nicht die Reaktion auf ein bereits bestehendes und eventuell betriebsbeeinträchtigendes Problem, sondern es wird dabei auf einen Indikator für ein eventuell bevorstehendes Problem reagiert. Dieser Indikator ist zum Beispiel ein überlegt gesetzter Schwellwert, sodass bei dessen Überschreiten noch genügend Zeit für fehlervermeidende Aktivitäten zur Verfügung steht, um den tatsächlichen Eintritt des

Problems / Fehlers zu vermeiden bzw. dessen Auswirkungen so gering wie möglich halten zu können.

Eine System-Management-Lösung kann auf zwei Ebenen proaktiv sein:

- Es können dadurch Probleme entdeckt und gelöst werden bevor die Benutzer davon betroffen sind und den Helpdesk kontaktieren.
- Es können Anomalien bereits festgestellt werden bevor sie zu Problemen werden.

Laut [de.wikipedia.org, 2007] ist „Proaktivität“ ein Neologismus aus dem Lateinischen (vor; tätig) und bedeutet wörtlich übertragen „voraushandelnd“. Es ist damit ein frühzeitiges initiatives Handeln im Gegensatz zum abwartenden „reaktiven“ Handeln gemeint. In „Das große Fremdwörterbuch“ von Duden ist das Wort seit der 3. Auflage, 2003 erfasst und bedeutet frühzeitiges Handeln, noch ehe die Umwelt das Unternehmen zu (reaktiven) Maßnahmen zwingt.

1.3 Was ist Monitoring im IT-Bereich?

Wie unter 1.1.1 dargestellt, wird Monitoring im Umfeld der Informationstechnologie (IT) als systematisches, periodisches und proaktives Überwachen von Systemen und deren Komponenten verstanden – der Administrator (Systemmanager) soll Fehler im System erkennen und wenn möglich beseitigen und beheben, bevor der Kunde bzw. der Benutzer es merkt und ein Problem meldet. Im Grunde genommen ist jede Komponente in einem IT-Umfeld dazu geeignet überwacht zu werden. Die Sinnhaftigkeit ist mit Sicherheit durch die verschiedensten Anforderungen von System zu System unterschiedlich. Um eine Komponente bzw. ein Objekt überwachen zu können, muss diese(s) eine Schnittstelle zur Verfügung stellen, über welche die entsprechenden Daten und Informationen abgefragt bzw. selbstständig Informationen über den eigenen Systemzustand verschickt werden können. Je standardisierter diese Schnittstelle implementiert ist, desto einfacher kann das Objekt in eine bestehende Monitoring-Umgebung integriert werden.

Je nach Anforderungsfall bzw. Implementierung kann der Fokus des Monitorings auf verschiedene Bereiche bzw. Ebenen gelegt werden:

Der grundlegendste Bereich zielt auf die Überwachung der **Server-Hardware** ab. Dabei soll sichergestellt werden, dass eventuell eintretende Hardware-Fehler rechtzeitig erkannt und behoben werden können. Heute verfügbare Server-Hardware ist in der Grundausstattung schon weitgehend mit redundanten Komponenten, die teilweise sogar hot-plug (siehe [it-wissen.info, 2006]) ausgetauscht werden können, ausgestattet. Dies gestattet oftmals einen uneingeschränkten Betrieb auch bei Fehlern und Reparaturarbeiten. Zur Hardware-Überwachung gehört auch die Überwachung der Temperatur der einzelnen Komponenten, sowie die Temperatur innerhalb des Systems, z.B. im Server. Der Überwachung der Temperatur kommt ein wichtiger Stellenwert zu, da ungünstige Temperaturverhältnisse die Lebensdauer einzelner (Sub)-Komponenten und damit des Gesamtsystems drastisch verkürzen können. Dieses Kapitel ist somit sehr eng mit der Verfügbarkeit und Funktionstüchtigkeit der installierten Lüfter und der Klimatisierung verbunden. Ein Server-System besteht heutzutage im Normalfall aus dem Server selbst als auch aus einem kleinerem bzw. größeren Speichersystem (Storage). Dieses Storage wird in größeren Rechenzentren nicht mehr durch eingebaute Festplatten-Systeme (Direct Attached Storage (DAS)), sondern durch Storage Area Networks (SAN) bzw. über Network Attached Storage (NAS) bereitgestellt.

Typischerweise steht ein „echter“ Server nicht unter dem Schreibtisch der Sekretärin sondern in einem eigens dafür vorbereiteten Bereich – dem Computer- bzw. Serverraum. Die dafür notwendige **Infrastruktur** wie Klimaanlage, Notstrom bzw. USV-Anlagen (ununterbrochene Stromversorgung (UPS – Uninterruptable Power Supply)) etc sind zusätzliche Komponenten für die eine geeignete Überwachung eingerichtet werden muss.

Aufsetzend auf die Hardware wird die Überwachung der **Systemkomponenten** und deren **Ressourcen** eingerichtet. Dabei soll sichergestellt werden dass alle notwendigen Komponenten für den einwandfreien und performanten Betrieb der Systeme ausreichend Ressourcen zur Verfügung stellen sodass rechtzeitig auf Engpässe reagiert werden kann. Ein Server stellt Dienste im Netzwerk zur Verfügung welche von Benutzern bzw. anderen Systemen genutzt werden – Ressourcen welche verfügbar sein müssen. Gerade bei Beschwerden über die Geschwindigkeit von Diensten (z.B.: bei Web-Services) ist es sehr hilfreich Informationen über die Ressourcenauslastung am System auch in der Historie

verfügbar zu haben. Ähnlich gewichtet sind auch die Bereiche **Security** und **Log Management** für die Einhaltung von Datenschutz und Datensicherheit. Das „National Institue of Standards and Technology“ (NIST) arbeitet unter anderem an einem Entwurf für „Computer Security Log Management“[1]. Die Zukunft der IT-Leistungen liegt in der Orientierung auf **Services.** Diese müssen in geeigneter Weise überwacht werden und stellen mit Sicherheit die größte Herausforderung dar. Einerseits aus dem Grund, dass die Zusammenhänge der verschiedenen Komponenten und Systeme für ein aussagekräftiges Monitoring bekannt sein müssen, andererseits müssen geeignete Mittel für die Überprüfungen vorhanden sein bzw. entwickelt werden.

1.4 Standards und Begriffe

Um eine einfache und weniger problematische Integration der verschiedenen Systeme zu erzielen, ist es sinnvoll auf bestehende und etablierte Standards zu setzen. In den folgenden Abschnitten werden nun kurze Überblicke über Standards und Begriffe in Bezug auf Monitoring und System Management gegeben.

1.4.1 Simple Network Management Protocol

Das Simple Network Management Protocol[2] (SNMP) ist Teil der „Internet Protocol Suite“ der Internet Engineering Task Force (IETF) und heute der de facto Standard bei der Überwachung und Administration von Netzwerken und Systemen.

1.4.1.1 Geschichte und Entwicklung

Version 1 von SNMP wurde 1988 in den ersten Requests for Comments (RFCs) definiert. In SNMPv1 war praktisch keine Sicherheit bzw. Authentifizierung enthalten und wurde stark kritisiert. Die Authentifizierung wurde nur durch einen so genannten „Community String“, welcher noch dazu in Klartext übertragen wird, realisiert. Der Nachfolger SNMPv2 löste Version 1 ab und enthält Verbesserungen in Punkto Performance, Security, Confidentiality (Geheimhaltung) und auch Manager-zu-Manager-Kommunikation. Es wurden Erweiterungen im Bereich der Kommandos (protocol data units (PDU)) eingearbeitet. Aufgrund des „**Party-**

[1] http://csrc.ncsl.nist.gov/publications/drafts/DRAFT-SP800-92.pdf
[2] Für nähere Informationen siehe auch:
http://www.snmp.com/FAQs/snmp-faq-part1.txt
http://www.snmp.com/FAQs/snmp-faq-part2.txt

Based“ Security-Systems wurde SNMPv2 als zu komplex angesehen und daher selten implementiert und wenig verbreitet. SNMPv2c, das „**Community**[3]**-Based** Simple Network Management Protocol“ wurde in den RFCs 1901 bis 1908 definiert und ersetzte SNMPv2. SNMPv2**c** ist SNMPv2 ohne dem vieldiskutierten Sicherheitsmodell, dafür aber mit dem einfachen „community-based“ Sicherheitsmodell aus SNMPv1. Obwohl SNMPv2c offiziell nur als „Draft Standard“ – also nicht als „offizieller Standard“ – definiert ist, hat es sich zum de facto Standard etabliert.

Die ersten Schritte zu einer neuen Version von SNMP wurden gesetzt. Der aktuelle Standard durch die IETF ist seit dem Jahre 2004 als SNMPv3 definiert. Alle vorhergehenden Versionen von SNMP sind als „obsolete“ bzw. „historical“ gekennzeichnet. SNMPv3[4] beinhaltet die Funktionalitäten von SNMPv2. Die Hauptverbesserungen wurden im Bereich Sicherheit (Security) durchgeführt. So wurden drei neue Features eingeführt: Authentifizierung, Access Control und Geheimhaltung (secrecy). SNMPv3 unterstützt eine Zugriffsbeschränkung auf den MIB-Baum, sodass ein weniger privilegierter Benutzer nur Ausschnitte aus den Managed Objects manipulieren kann. Durch die Einführung von SNMPv3 wird auch eine Remote-Konfiguration von SNMP Objekten möglich.

1.4.1.2 Aufbau

Das „Simple“ von „Simple Network Management Protocol“ bedeutet nicht, dass es simple zu verwenden ist, sondern dass es recht einfach aufgebaut ist. SNMP realisiert die Kommunikation zwischen einer Management Station und den Agenten.

Das Konzept der SNMP-Architektur basiert auf dem Client-Server-Prinzip und besteht aus folgenden wichtigen Komponenten:
Management Station (Network Management Station (NMS)): fungiert als Client in der SNMP-Architektur und führt die Requests (lesend als auch schreibend) aus. Weiters empfängt die NMS auch Traps (siehe Punkt 1.4.3.1) und verarbeitet diese entsprechend. Auf der NMS laufen die diversen Management-Applikationen welche die Systeme monitoren bzw. verwalten.

[3] Community ist eine logische Gruppe, welche Geräte umfasst die in der gleichen administrativen Domäne sind. [itwissen.info, 2007]
[4] In den RFCs 2271 bis 2275 definiert.

Agent („Managed Object“): ist der Serverteil zum Client und besteht aus einem Software-Teil wie zum Beispiel eines Daemons (Linux) bzw. Dienstes (MS Windows). Der Agent nimmt die Anfragen (Requests) der NMS entgegen und verarbeitet sie mit Hilfe von Subagenten (siehe auch Abbildung 1-1 bis Abbildung 1-3).

Ein **Subagent** ist hierarchisch unter dem Agenten angesiedelt und auf das zu überwachende Objekt abgestimmt. Der Subagent wird vom Agenten abgefragt und weiß wie er mit dem Objekt kommunizieren kann und liefert dem Agenten die jeweiligen Informationen zurück – ist nicht nach außen hin sichtbar bzw. abfragbar.

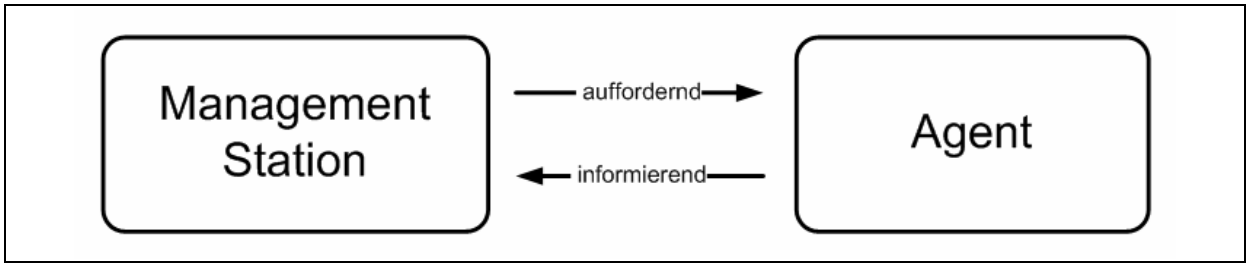


Abbildung 1-1 - Kommunikation zwischen Management-Station und Agent

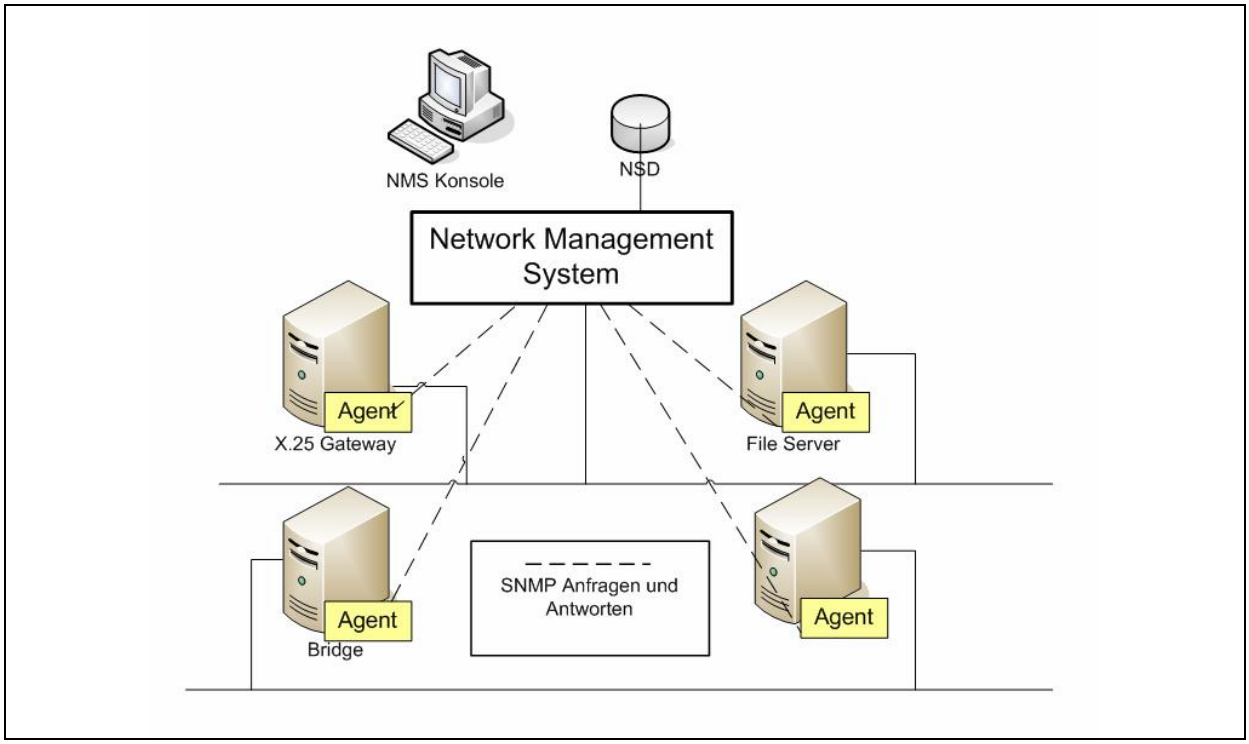


Abbildung 1-2 - Aufbau-Skizze einer SNMP-Konfiguration
(Quelle: [it-wissen.info, 2006])

SNMP setzt auf ein Transport-Protokoll – typischerweise „User Datagram Protocol“ (UDP) - auf und agiert auf dem „Application Layer“ (Layer 7) des OSI Modells. Der Agent horcht auf UDP-Port 161 und verschickt über einen verfügbaren Port, der Manager (Client) horcht auf UDP-Port 162 auf Traps und verschickt Requests über einen verfügbaren Port, auf welchem dann auch die Antwort (Response) auf den Request empfangen wird (siehe auch Abbildung 1-3).

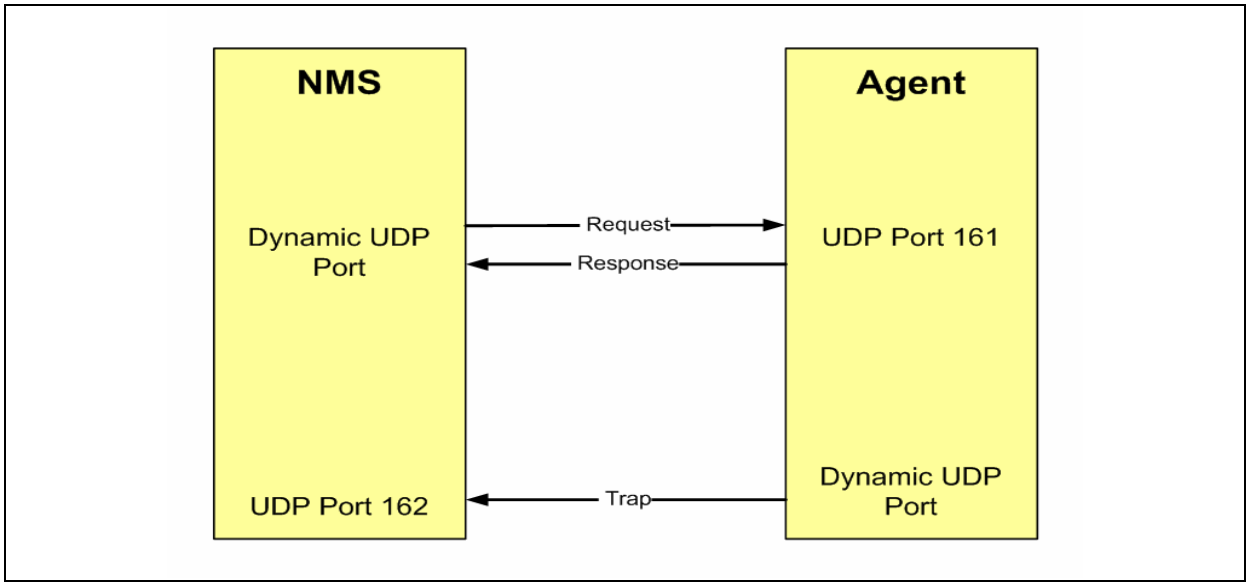


Abbildung 1-3 - vereinfachte SNMP Architektur

1.4.1.3 Management Information Base

Eine Management Information Base (MIB) ist eine Sammlung von Objekten, welche via SNMP gemanagt werden können. Fälschlicherweise wird unter einer MIB oftmals eine Datenbank bzw. ein Datenstore verstanden – eine MIB enthält keinerlei Daten und holt bzw. schreibt auch keine Daten vom/auf das überwachte System.

Um Informationen über ein System zu erhalten muss man (das NMS, der Admin) Informationen haben, welche Daten das System „anbietet“. Dafür ist die MIB da. Sie stellt eine logische Gruppierung der verfügbaren Informationen in einer hierarchischen Struktur dar. Es werden die verfügbaren Objekte beschrieben. Ein MIB-File ist ein reines ASCII[5]-Textfile in SMIv2[6]-Syntax. Eine „Enterprise MIB“ wird durch das Unternehmen des

[5] American Standard Code for Information Interchange
[6] Structure of Management Information (SMI) is an adapted subset of ASN.1

verwalteten Objektes erstellt und definiert Objekte für das Produkt, welche vom Administrator abgerufen bzw. gesetzt werden können. Jedes Objekt wird durch den „Object Identifier“ (OID) identifiziert.

MIB (management information base)

Die ISO[7] beschreibt die zu managenden Parameter in Form von Objekten und Attributen. Ein Objekt ist ein abstrakter Begriff für eine Ressource, die einen bestimmten Teilbereich des Netzes darstellt. Diese Ressource kann über eine Vielzahl von Parametern verfügen. Kann dieses Objekt in ein Netzwerkmanagement-Konzept eingebunden werden, so spricht man von einem managebaren Objekt. Die Gesamtheit der managebaren Objekte wird als Management Information Base (MIB) bezeichnet. Die MIB ist eine Datenbank, die alle oder viele für ein Management-System relevante Daten in relationaler oder objektorientierter Form enthält. Der Begriff MIB wird meist im Zusammenhang mit SNMP benutzt. Die SNMP-MIB ist ein einheitlicher, hierarchisch aufgebauter, protokollunabhängiger Raum für Datenobjekte.

Die Namen der Variablen sowie deren mögliche Werte sind in einer MIB eindeutig festgelegt. Alle Objekte in einer MIB werden einheitlich in ASN.1, der Abstract Syntax Notation One, die ursprünglich für die Definition abstrakter Transfersyntaxen in der Darstellungsschicht des OSI-Referenzmodells entworfen wurde, formuliert, wodurch eine Normung der abstrakten Darstellungen gegeben ist. Objekte in ASN.1 können später auch von anderen Management-Protokollen benutzt werden.

Die aus der TCP/IP-Welt bekannte MIB I reflektiert Protokolle des Internet. Diese MIB wurde 1990 durch die MIB 2 abgelöst und wird seither nicht mehr verwendet. MIB 2 hat wesentlich mehr Elemente, die die verschiedenen Bereiche wie Host-MIB, Bridge-MIB, Router-MIB usw. beinhalten. [...]
[www.it-wissen.info, 2006]

[7] International Standardization Organization

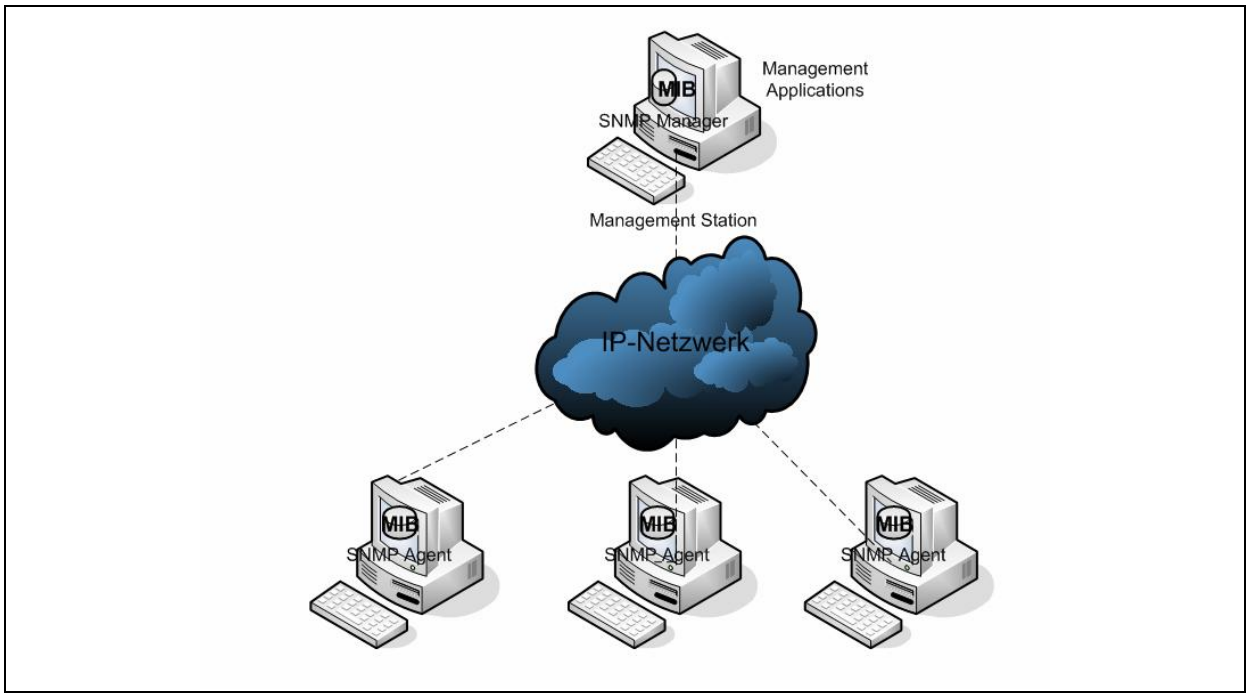


Abbildung 1-4 - Rolle der MIB

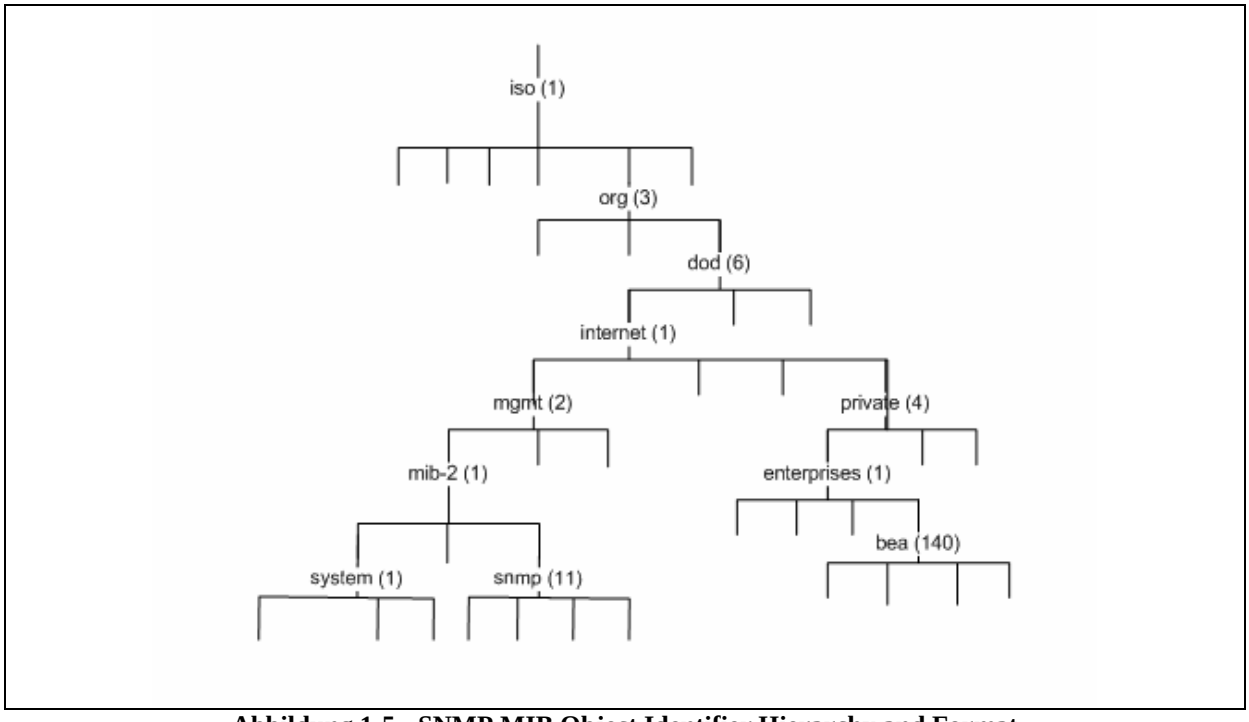


Abbildung 1-5 - SNMP MIB Object Identifier Hierarchy and Format

Abbildung 1-4 zeigt die Rolle der Management Information Base im Network Management System. Der Agent als auch die NMS kennen die MIB. Abbildung 1-5 zeigt Struktur, Hierarchie und das Format einer MIB und der Object Identifier (OID). [Quelle: beide http://e-docs.bea.com]

1.4.2 Syslog[8]

Syslog ist der Audit und Logging Mechanismus für die konsolidierte Verarbeitung und Speicherung von System- und Applikations-Meldungen auf Unix und Linux Betriebssystemen. Syslog ist auf eine einfache Logging-Funktionalität ausgelegt und bietet dem Administrator einen „Single Point of Management“ von Logging Informationen. Dort wird definiert was und wohin die einzelnen Messages gespeichert werden. Das Syslog-Protokoll ist wie SNMP relativ einfach aufgebaut und wurde 1980 als Teil des „sendmail“-Projektes entwickelt. Dabei verschickt der Sender einfache Nachrichten mit einer Länge von weniger als 1024 Bytes an den Syslog-Server. Die Übertragung erfolgt über UDP (Port 514) - selten über TCP - im Klartext. Syslog-Implementierungen existieren heute für die verschiedenen Betriebssysteme.

Das Syslog-Protokoll besitzt einige diskussionswürdige Schwachstellen:

- Überträgt Meldungen im Klartext
- Keinerlei Authentifizierung ➔ Gefahr von Denial of Service-Attacken (DoS)
- Prioritäten werden uneinheitlich verwendet
- Nachrichten werden verbindungslos übertragen (UDP)

Die Übertragung per UDP hat andererseits den Vorteil dass der Overhead von TCP nicht anfällt – vor allem bei einem hohen Log-Aufkommen relevant. Aus obigen Gründen gingen verschiedene Implementierungen hervor, welche nicht immer zu 100 Prozent zueinander kompatibel sind. Sehr verbreitet ist zum Beispiel „syslog-ng[9]“ welcher einige der oben genannten Schwachstellen nicht aufweist. Die Unterteilung und Kategorisierung von Log-Einträgen erfolgt aufgrund von Facilities und Severities (siehe Anhang A - 5 und Anhang A - 6 im Anhang).

[8] RFC 3164, „The BSD Syslog Protocol“, http://www.ietf.org/rfc/rfc3164.txt
RFC 3195, „Reliable Delivery for Syslog“, http://www.ietf.org/rfc/rfc3195.txt
http://www.ietf.org/html.charters/syslog-charter.html
[9] http://www.balabit.com/products/syslog_ng/

1.4.3 Arten der Informationsgewinnung

1.4.3.1 Trapping / Polling

Es gibt verschiedene Arten, wie ein Management-System die notwendigen Informationen über ein gemanagtes System erhalten kann.

Polling:

Unter „Polling“ wird das zentrale Abfragen von Werten, Daten, Informationen und Stati durch das Management-System über das gemanagte Objekt in definierten Intervallen verstanden. So wird zum Beispiel die CPU-Belastung alle 5 Minuten abgefragt.

Anfrage und Abruf von gespeicherten Daten (z.B. Dateien oder Programme) mit Hilfe von Abfragestationen, z.B. Monitoren. Normalerweise erfolgt der Abrufbetrieb in einem Request-Response-Verfahren. Im Dialogbetrieb handelt es sich um einen Modus bei dem das System für die Annahme von Abfragen bereit ist.
[itwissen.info, 2007]

Trapping:

Unter „Trapping“ versteht man, wenn das gemanagte System in einem bestimmten Zustand (meistens Fehlerfall) von sich aus aktiv wird und eine Meldung an das Management-System schickt. Der Empfänger verarbeitet die empfangenen Daten und setzt Aktivitäten, sei es einfach eine Meldung an den Administrator oder das Abfragen von zusätzlichen Informationen.

Meldung über ein Ereignis, welches von einem Agenten selbstständig ausgesandt wird. Eine Trap wird von einem SNMP-Agent zu dem Netzwerkmanagement gesandt und zeigt ein bedeutendes Ereignis an. Der SNMP-Manager, der das Trap empfängt, kann nach weiteren Informationen nachfragen.
[itwissen.info, 2007]

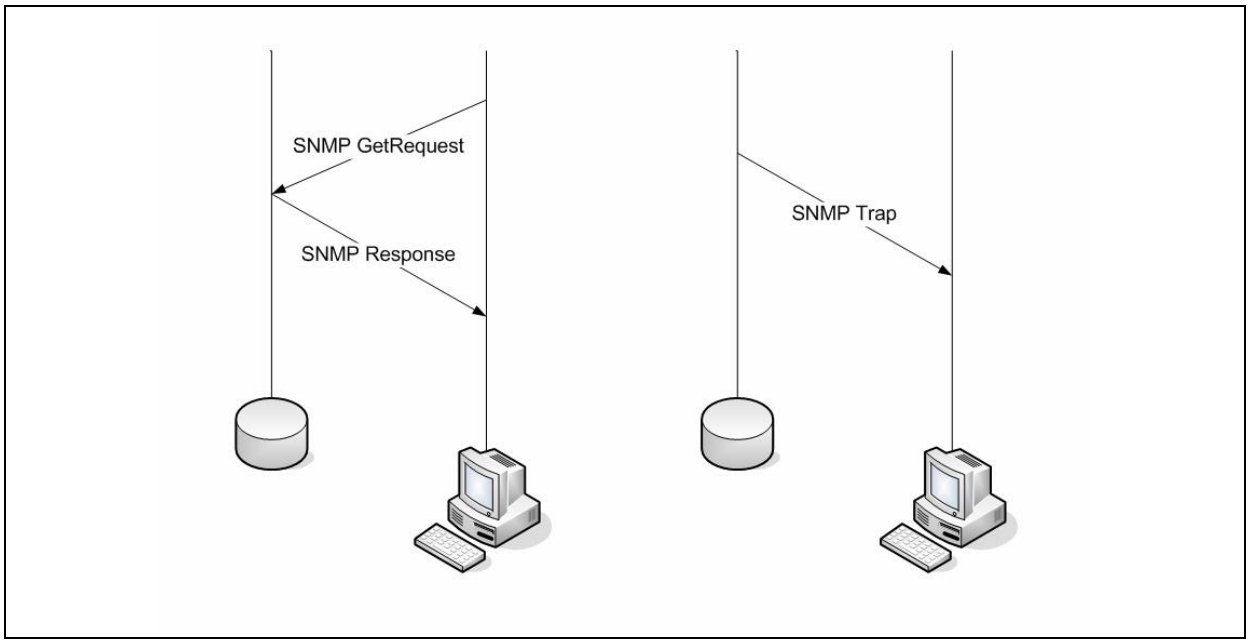


Abbildung 1-6 - SNMP Request (Poll) und Trap

Abbildung 1-6 zeigt deutlich den Unterschied zwischen einem Poll (Abfrage) durch eine NMS (dargestellt durch einen PC) auf der linken Seite und einem Trap auf der rechten Seite.

1.4.3.2 RRD – Round Robin Database / RRDTool

RRD steht für „Round Robin Database“. Die RRD arbeitet mit einer fixen Menge an Daten und einem Pointer auf das aktuelle Element in der Datenbank. Die Datenstruktur kann sich als Ring vorgestellt werden. Die Größe der Datenstruktur wird beim Anlegen festgelegt, sind alle Speicherplätze mit (numerischen) Werten belegt, so wird der jeweils älteste Wert überschrieben. Das Dataset wird also nie größer und dadurch ist auch keine Wartung der Datenbank notwendig.

RRDTool wurde von Tobias Oetiker[10] entwickelt und unter der GNU General Public License[11] (GPL) veröffentlicht. RRDtool ist dafür designed und entwickelt, um eine Serie von Daten im Zeitverlauf zu speichern. Dabei wird bei jedem Datenupdate ein zugehöriger Timestamp (Zeitstempel) in Sekunden seit 1.1.1970 (epoch) mitgespeichert. Mit RRDtool

[10] http://oss.oetiker.ch/rrdtool/
[11] http://www.fsf.org/licenses/gpl.html

werden unter anderem Datenbanken angelegt, Daten abgespeichert, Daten ausgelesen, Grafiken aus den Daten erzeugt, usw.

RRD sind auf Grund des primären Einsatzes für Monitoringzwecke durch eine recht einfache Struktur (zum Beispiel im Vergleich zu relationalen Datenbanken mit Tables) ausgezeichnet. Die abgespeicherten Werte müssen numerisch sein, allerdings nicht notwendigerweise Integer-Werte. RRDtool besitzt keine Mechanismen um Alerts zu generieren.

1.4.3.3 Inband / Outband

Inband:

Beim „Inband“ werden die gemanagten Objekte über das User- bzw. Datennetz geprüft und gemanaged. Als **Vorteil** davon kann genannt werden, dass impliziert sicher gestellt ist, dass das überwachte System über das User- bzw. Datennetzwerk erreichbar und das Usernetzwerk zwischen Monitoring-System und dem überwachten System in Ordnung ist, sobald das System vom Monitoring-System aus ohne Schwierigkeiten erreichbar ist. Als Kritikpunkte können diverse Sicherheits-Überlegungen (z.B.: Trennung von Usernetz und Management-Netz) als auch eine Belastung (und dadurch eine eventuelle Beeinträchtigung der Performance für den Benutzer) des Usernetzwerkes berücksichtigt werden (siehe auch Abbildung 1-7).

Von Inband spricht man auch bei der Übertragung von Protokollen, wenn diese über die gleichen Übertragungsmedien wie die Primärprotokolle übertragen werden. Typische Beispiele für Inband-Protokolle sind Management-Protokolle. [www.it-wissen.info, 2006]

Outband:

Beim „Outband“ werden die Systeme über eigene dezidierte Netzwerkverbindungen überwacht. Der größte Vorteil liegt darin, dass auch überwacht und gegebenenfalls gewartet und konfiguriert werden kann, wenn das Usernetz nicht (mehr) verfügbar ist. Im Gegensatz zu Inband-Monitoring kommt hier eine sicherheitstechnische Trennung vom Usernetzwerk zum tragen. Ein schlagkräftiger Nachteil entsteht durch die zusätzlichen Kosten unter anderem durch eigene Hardware und Netzwerkleitungen, als auch durch die zusätzlich notwendigen Konfigurationsparameter (siehe auch Abbildung 1-8).

Darüber hinaus wird die Außenband-Signalisierung auch für Fehlermeldungen angewendet, um eventuelle Einflüsse oder Probleme auf die Inbandkanäle zu umgehen.
[itwissen.info, 2007]

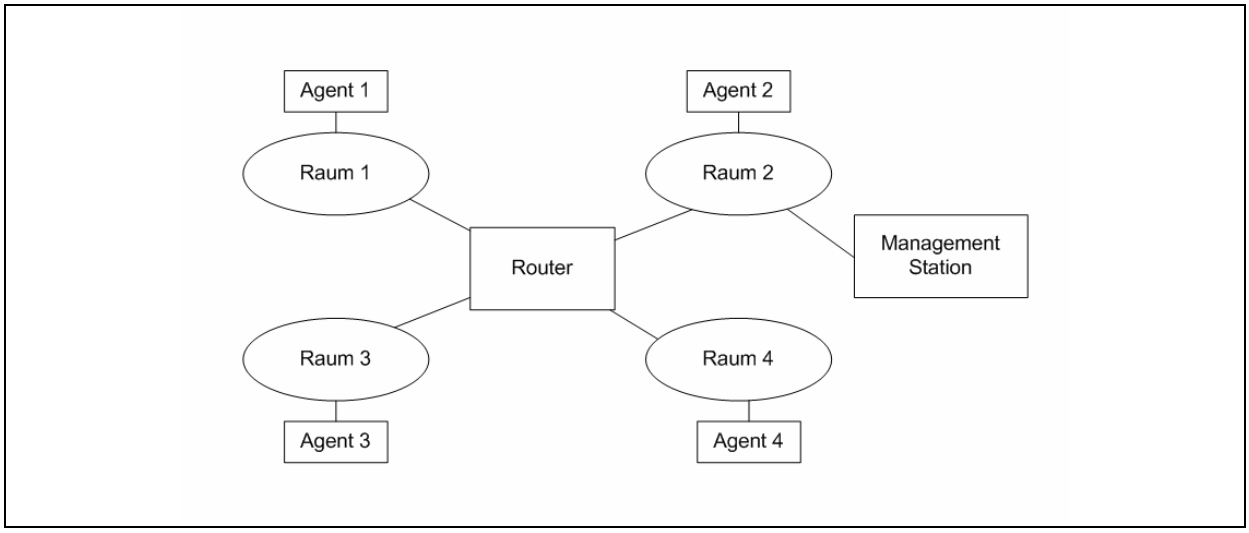


Abbildung 1-7 - Inband Management
(Quelle: [Siegl, 1993])

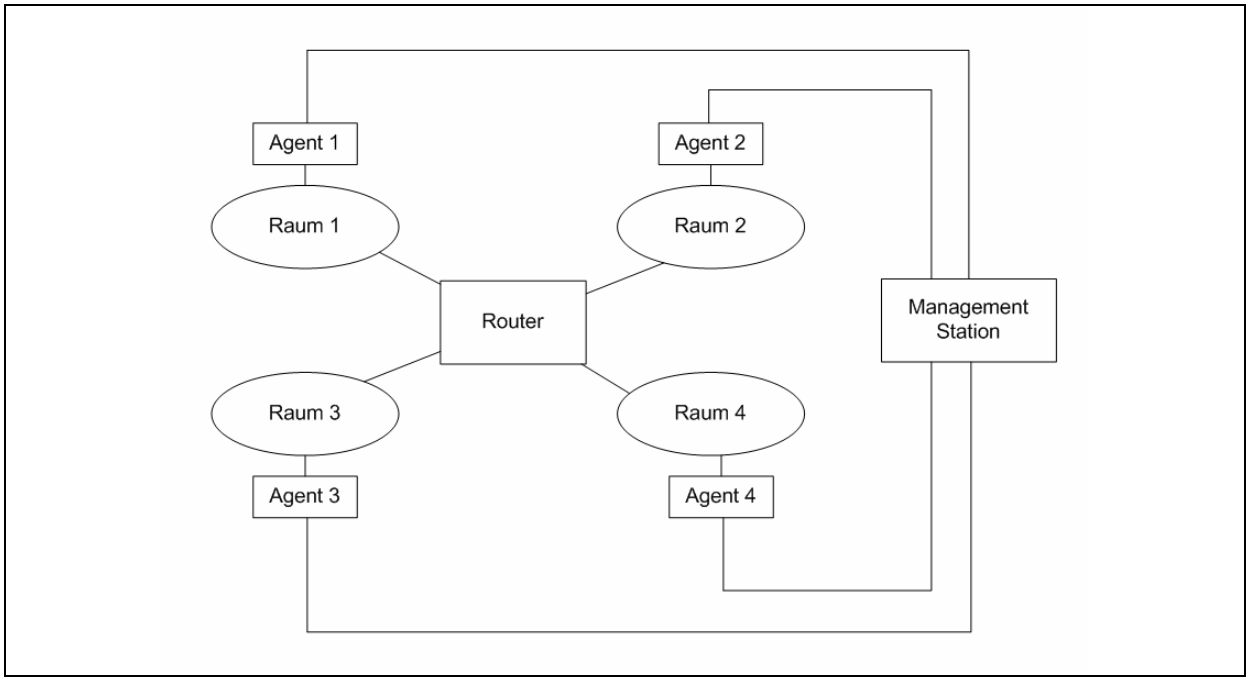


Abbildung 1-8 - Outband Management
(Quelle: [Siegl, 1993])

1.4.4 Begriffsdefinitionen

Quellen:

[de.wikipedia.org, 2007], [itwissen.info, 2007],
[ee.ntu.edu.tw, 2007], [nagios-flapping],
[en.wikipedia.org, 2006], [Souppaya, 2006],
[iicm.tugraz.at, 2007] [uni-erlangen.de, 2007]

Event (Ereignis):

Ein Ereignis ist eine plötzliche Veränderung (positiv als auch negativ) in einem System, wobei der Zeitpunkt des Eintretens nicht vorherbestimmbar ist. Es gibt einmalige und wiederkehrende Ereignisse.

Log:

Ein „Log“ ist eine Sammlung von Events (Ereignissen), welche innerhalb eines Systems oder Netzwerkes auftreten. Der „Log“ besteht aus Einträgen, welche detaillierte Informationen zum entsprechenden Event beinhalten.

Fault (Fehlerursache):

Ein Fault ist ein unerwarteter Zustand, der zu einem Fehler (Error) führen kann. Z.B.: durch eine externe Störung oder durch einen Design-Fehler.

Error (Fehler):

Dies ist ein unerwünschter Systemzustand, der durch einen Fault verursacht ist und nicht den Spezifikationen entspricht. Ein Error ist die Manifestierung eines Faults im System.

Failure (Fehlerverhalten, Funktionsausfall):

Für das System ist die erwartete bzw. erwünschte Diensterbringung nicht mehr möglich.

Fehlerfortpflanzung:

Ein Failure kann in einer anderen Ebene einen Fault darstellen. Bei der Differenzierung ist die Systemgrenze ausschlaggebend – ein Failure in dem einen System wirkt sich zum Beispiel im darauf aufsetzenden System als Fault aus.

Problem (Problem):

Das System funktioniert noch, allerdings nicht mehr zu 100 %, da ein anormaler Zustand eingenommen wurde; z.B. kann ein Web-Server nicht mehr alle Request oder nur mit entsprechend höheren Antwortzeiten abarbeiten.

Alert (Alarm):

Dies ist die alarmierende Anzeige einer Bedingung, die eine mögliche negative Auswirkung eines überwachten Objektes zur Folge haben kann. z.B.: wenn ein Schwellwert (Treshold) überschritten wurde, geht bei einer Ampelanzeige die Farbe von grün auf rot.

Availability (Verfügbarkeit):

Eine mögliche Definition (vor allem für technische Systeme) von „Verfügbarkeit“ ist der Anteil einer Zeitspanne, in der das System ohne Fehlverhalten benutzbar ist. Die Verfügbarkeit wird normalerweise mit der beliebten Neuner-Kombination angegeben: z.B.: „eine Verfügbarkeit von 99.99 %“. Verfügbarkeit = Uptime / (Downtime + Uptime)

Was bedeutet eine Verfügbarkeit von 99,99 %? – das wären rechnerisch rund 52 Minuten Nicht-Verfügbarkeit pro Jahr (siehe auch Tabelle 1). Die Verfügbarkeit wird durch die beiden Faktoren „Meantime Between Failure“ (MTBF) und „Meantime To Repair“ (MTTR) bestimmt. Optimal sind eine möglichst hohe MTBF und einer möglichst kleinen MTTR:

Verfügbarkeit	Ausfallszeit pro Jahr	Verfügbarkeitsklasse
90 %	~ 36 Tage und 12 Stunden	Normale Verfügbarkeit
95 %	18,25 Tage	
99 %	3,65 Tage	
99,5 %	~ 43 Stunden	Einfache Verfügbarkeit
99,9 %	8,76 Stunden	Erhöhte Verfügbarkeit
99,95 %	~ 4,3 Stunden	
99,99 %	52 Minuten	Clusterverfügbarkeit
99,999 %	~ 5 Minuten	Hochverfügbarkeit
99,9999 %	~ 30 Sekunden	„six nines“ z.B.: in der Flugsicherung
99,99999 %	~ 3 Sekunden	
100 %	0 Sekunden	Non-Stop-Verfügbarkeit

Tabelle 1 - Verfügbarkeitsklassen und Ausfallszeiten

Eine gewünschte höhere Verfügbarkeit – weniger ungeplante Ausfallszeiten – geht einher mit höheren Kosten in den Bereichen Hard- und Software. Die Kosten für das „Hinzufügen“ einer weiteren „9“ ab 99,99 % muss mit dem zu erwartenden Nutzen gegengerechnet werden.

MTBF:

„Meantime Between Failure“ ist die mittlere Zeitspanne zwischen dem Auftreten zweier Fehler.

MTTR:

„Meantime To Repair“ ist die mittlere Zeitdauer der Problembeseitigung.

Downtime (Ausfallzeit):

Damit ist jene Zeitspanne gemeint, während der ein System oder ein Dienst für den Benutzer nicht verfügbar ist. Es wird zwischen geplanter und ungeplanter Downtime unterschieden.

Outage (Ausfall):

Eine Betriebssituation in der der Benutzer das System / den Dienst nicht nutzen kann (siehe auch Punkt 2.1).

„Flapping“:

zu deutsch: to flap = flattern, mit den Flügeln schlagen.

Beim Monitoring ist dabei das „state flapping“ gemeint; wobei sich der Status für das Monitoring-System von Check-Intervall zu Check-Intervall ändert: z.B. von „unreachable“ auf „ok“ und umgekehrt. Ein „state flapping“ hätte grundsätzlich eine Flut von Error- und OK-Benachrichtigung zur Folge. Siehe auch [nagios-flapping].

Redundancy (Redundanz):

Redundanz bedeutet im Grunde genommen übermäßig und überflüssig. In der IT und in der Technik allgemein benutzt man verschiedene Redundanzverfahren zur Sicherstellung von Daten, Verfügbarkeiten etc. Fehlertoleranz und Redundanz sind miteinander untrennbar verbunden. Im Komponentenbereich bezieht sich Redundanz vor allem auf den Einsatz von mehreren Geräten bzw. Komponenten mit identischer Funktion. Entweder laufen alle Komponenten gleichzeitig (z.B. bei Lüftern) oder eine ist die aktive Komponente und die restlichen laufen im Standby-Modus und übernehmen die Aufgabe wenn die aktive Komponente ausfällt.

Skalierbarkeit:

Unter Skalierbarkeit versteht man die Erhöhung der Leistung eines Systems ohne Änderungen an der Software vorzunehmen – nur durch Hinzufügen von Hardware-Leistung.

Single Point of Failure (SPoF):

Der „Single Point of Failure“ ist der Teil eines Systems, der durch seinen Ausfall oder durch einen Fehler einen Ausfall des Gesamtsystems mit sich zieht. Typische Beispiele für SPoF sind unter anderem Stromversorgung, Klimatisierung, Netzwerk-Anschlüsse von Systemen, etc.

Log-Management:

Unter Log Management werden jene Arbeiten und Vorkehrungen verstanden, die sicherstellen, dass Log-Einträge in ausreichender Qualität für eine vordefinierte Zeitspanne gespeichert und ausgewertet werden können. Die Herausforderung liegt darin, eine Balance

zwischen der notwendigen Detaillierung und der Limitierung der Datenmenge zu finden. [Souppaya, 2006] – siehe auch den Entwurf von NIST.

Event-Management:

Das Event Management ist der sinnvolle Umgang mit den zentral gesammelten Log-Einträgen der diversen Systeme und Applikationen. Die Analyse und gegebenenfalls eine Alarmierung unterscheidet Event-Management von einem einfachen Log-Server. Im Event-Management werden die Einträge kategorisiert, gefiltert und analysiert. Aus den Ereignisdaten können Abweichungen erkannt und eventuell wichtige Rückschlüsse gezogen werden.

Performance:

Dieser Begriff kann mit Leistung gleichgesetzt werden. Es gibt die Unterscheidung in objektiver (messbarer) und subjektiver (wie ein Benutzer die Performance empfindet) Performance.

Performance-Management in der Praxis:

- Niedriger Automatisierungsgrad
- Viel mehr reaktiv statt proaktiv
- Probleme sind schwer zu erklären und meistens mit hohem Aufwand verbunden
- Optimierung erfolgt nur bei Problemen oder bereits bestehenden Defiziten.

Reliability (Zuverlässigkeit):

Zuverlässigkeit ist die Fähigkeit über einen gegebenenen Zeitraum hinweg unter bestimmten Bedingungen korrekt zu arbeiten.

Service Level Agreement

Deutsch: Dienstgütevereinbarung

Ein Service Level Agreement (SLA) ist eine bilaterale, juristische Übereinkunft zwischen Netzwerk-Provider und Kunde, in der die vertraglichen Vereinbarungen zur Qualität der Leistungen spezifiziert sind. Zu den wesentlichen Aspekten einer solchen Qualitätsvereinbarung, in denen die Netz- und Service-Parameter festgelegt sind, gehören die Bandbreite, die Verfügbarkeit, die Netzkapazität und die Netzqualität. Neben den genannten technischen Parametern spielen die Güte der Dienstleistung, die Technik und Messtechnik,

mit der die Dienstleistungen erbracht werden, die Verfügbarkeit, die Ausfall-, Reaktions- und Reparaturzeiten, die Servicequalität beeinflussende Faktoren, eine Rolle. [...] Da es keine Standards für die SLAs gibt, kann jeder Service Provider beliebige Angebote und Leistungen mit dem SLA-Label versehen.

Ein SLA-Vertrag sollte den Gegenstand der Dienstleistung definieren, die Laufzeit und die Kündigungsmodalitäten enthalten, die technische Leistung beschreiben, die Behandlung von Störungen, Vertragsstrafen und das Änderungsmanagement umfassen.[...]
[itwissen.info, 2007]

Service Level Agreements sind einerseits dazu da um eine Preis/Leistungs-Transparenz für Kunde und Auftragnehmer zu schaffen, andererseits bieten sie eine Unterstützung bei der Streitschlichtung bzw. -vermeidung. Der Auftraggeber erhält eine in den SLAs fixierte Leistung (z.B.: Reaktionszeiten des Supports, Restore von Daten, Erledigung von Anfragen, etc.) zu den vereinbarten Bedingungen (Preis, etc) und der Auftragnehmer garantiert, dass er sich an diese Vereinbarung hält. Um nun die Einhaltung der definierten Bedingungen kontrollieren bzw. beweisen zu können, müssen diese Punkte messbar sein und natürlich gemessen – gemonitored – werden. Im Endeffekt sollen IT Services zum vereinbarten Zeitpunkt in der vereinbarten Qualität geliefert werden.

	SLA Level 1	**SLA Level 2**	**SLA Level 3**
Garantierte Mindest-Verfügbarkeit	99,7 %	99,8 %	99,9 %
Garantierte Geschwindigkeit	100 % der vertraglichen Bandbreite	100 % der vertraglichen Bandbreite	100 % der vertraglichen Bandbreite
Support-Zeiten	Montag – Freitag 08:00 – 17:00 Uhr	Montag – Samstag 07:00 – 19:00 Uhr	7x24 Stunden
Support-Type	Online-Forum	E-Mail	Telefon-Support
Überwachung des Systems	permanent	permanent	permanent
Statistik	Update alle 5 Minuten	Update alle 5 Minuten	Update alle 5 Minuten

Abbildung 1-9 - Beispiel eines SLA

2 Wozu Monitoring?

2.1 Ausfälle

Was wird unter einem Ausfall verstanden?

Es ist dabei zwischen geplanten und ungeplanten Ausfällen zu unterscheiden – siehe auch Abbildung 2-1. Geplante Ausfälle werden durch den Administrator und das Betriebsgeschehen „verursacht“. Sie können geplant und deswegen auch in unkritischere Zeitbereiche verschoben werden. Typische Beispiele dafür sind notwendige Wartungsarbeiten und Upgrades, aber auch der nicht zu unterschätzende Bereich des Backups.

Ungeplante Ausfälle fallen in die schwerwiegendere Kategorie, da sie nicht planbar sind und sogar zu den ungünstigsten Zeitpunkten auftreten können. Im Gegensatz zu den geplanten Ausfällen kann hier zusätzlich nicht die Ausfallsdauer vorhergesagt oder bestimmt werden. Die Dauer hängt immer von der Situation und der Fehlerursache ab. Abbildung 2-2 gibt einen Überblick über mögliche Ursachen und deren Auswirkungen von Unterbrechungen.

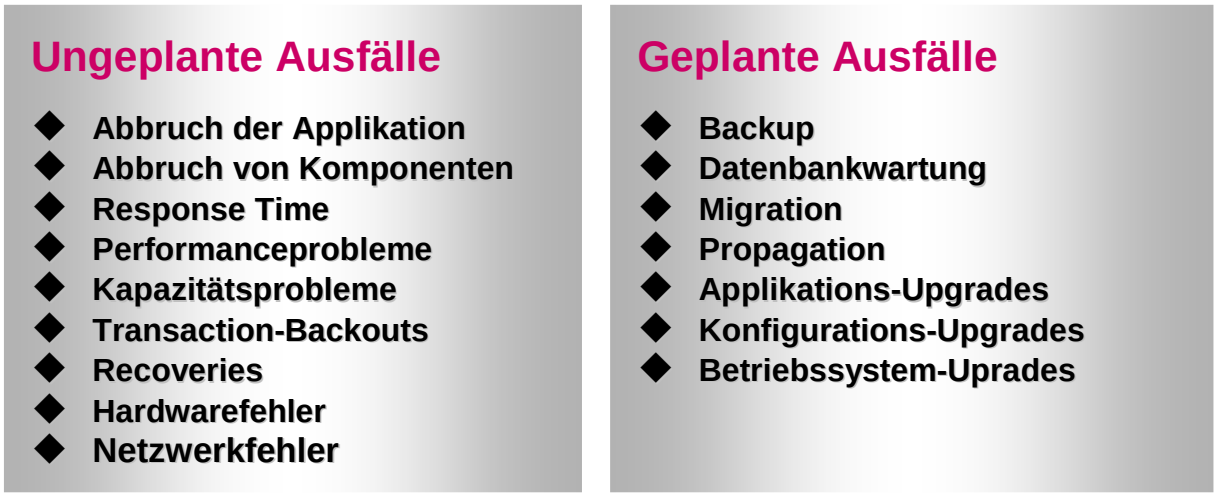


Abbildung 2-1 - geplante und ungeplante Ausfälle

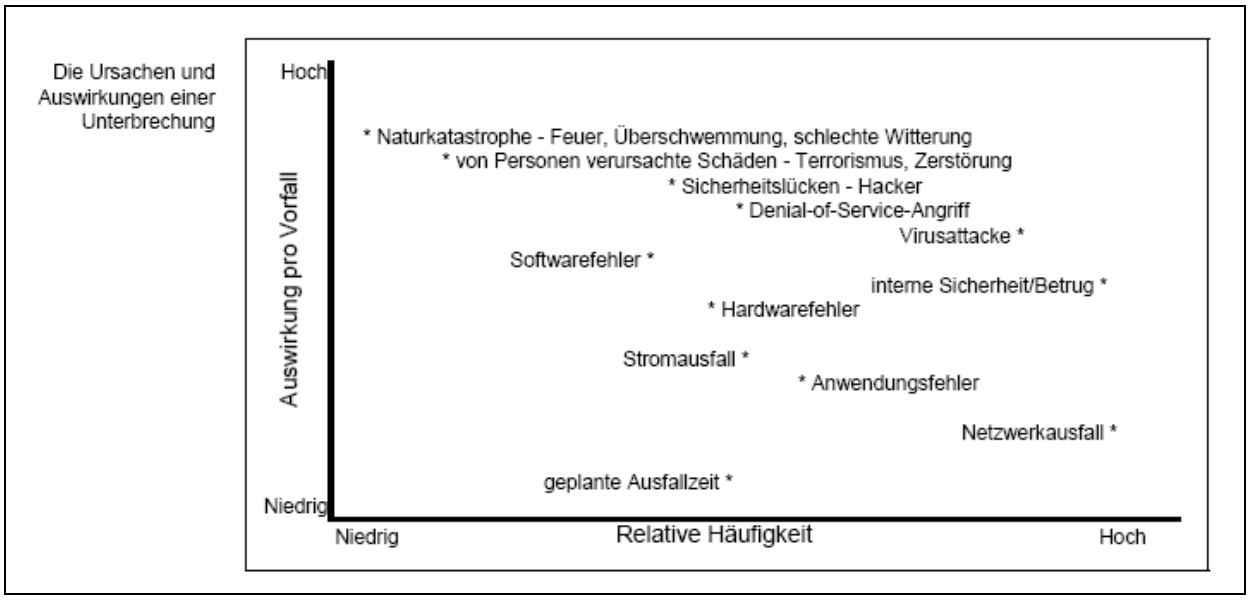


Abbildung 2-2 - Ursachen und Auswirkungen einer Unterbrechung (Quelle: [hp.com, 2007])

2.2 Auswirkungen (ungeplanter) Ausfälle

Die Auswirkungen von Ausfällen - vor allem von ungeplanten - können vielfältig sein und sind von Unternehmen zu Unternehmen unterschiedlich und hängen insbesondere von der Art des Ausfalles als auch vom betroffenen System und dadurch dem betroffenen Benutzerkreis ab. Grundsätzlich können die Konsequenzen temporärer oder dauerhafter Natur sein. Unter temporäre Konsequenzen fallen zum Beispiel Rollbacks von Transaktionen, Performance-Einbußen sowie nicht optimale Ressourcen-Ausnutzung. Dauerhaft sind unter anderem Imageverlust, Verlust von Kunden aber auch der Verlust von Menschenleben.

Die Auswirkungen können grob in folgende Kategorien eingeteilt werden (Quellen: [securitymanager.de, 2007], [ca.com, 2007], [aspectra.ch, 2007], [hp.com, 2007], [keos.de, 2006], [linbit.com, 2006], [bitkom.org, 2007]):

Umsatz:

Ein Ausfall hat in den meisten Fällen direkte Auswirkungen im Bereich des Umsatzes. z.B.: direkte Verluste, weil bei einem Web-Shop beispielsweise keine neuen Geschäfte getätigt wurden oder werden konnten.

Produktivität:

Was machen die Mitarbeiter wenn das EDV-System nicht verfügbar ist? In vielen Bereichen sind die Workflows der Unternehmen bereits bis zu 100 % in der EDV abgebildet. Sollte das System bzw. ein Teil des Systems eine Zeit lang nicht zur Verfügung stehen, so können die Mitarbeiter nicht alles erledigen, Arbeit bleibt liegen und es leidet die Produktivität.

Ruf / Image:

Ein einzelner kurzer Ausfall eines Systems, vielleicht sogar noch außerhalb der Betriebszeiten hat kaum Auswirkungen auf den Ruf bzw. das Image. Sollte das System aber immer wieder und vielleicht auch noch zu den Hauptgeschäftszeiten ausfallen, so bewirkt dies einen geschädigten Ruf bei den Kunden (auch bei den innerbetrieblichen Kunden). Es kann zu einer Abwanderung der Kunden zu einer dadurch gestärkten Konkurrenz kommen.

Finanzielle Auswirkungen:

Je nach Typ des Unternehmens und Kategorie des Ausfalles kann es teils zu gewaltigen Auswirkungen auf die finanzielle Zukunft haben. In manchen Branchen kann es zu Verlusten der Kreditwürdigkeit kommen, der Aktienwert eines börsennotierten Unternehmens kann fallen oder es werden Vertragsstrafen auf Grund von Terminüberschreitungen fällig.

In den meisten Fällen werden nur die direkten und offensichtlichen Auswirkungen eines Ausfalls erkannt und beachtet. Allerdings können indirekte Auswirkungen der Ausfallszeit wesentlich schwerwiegender und unvorhersehbarer sein – siehe auch Abbildung 2-3.

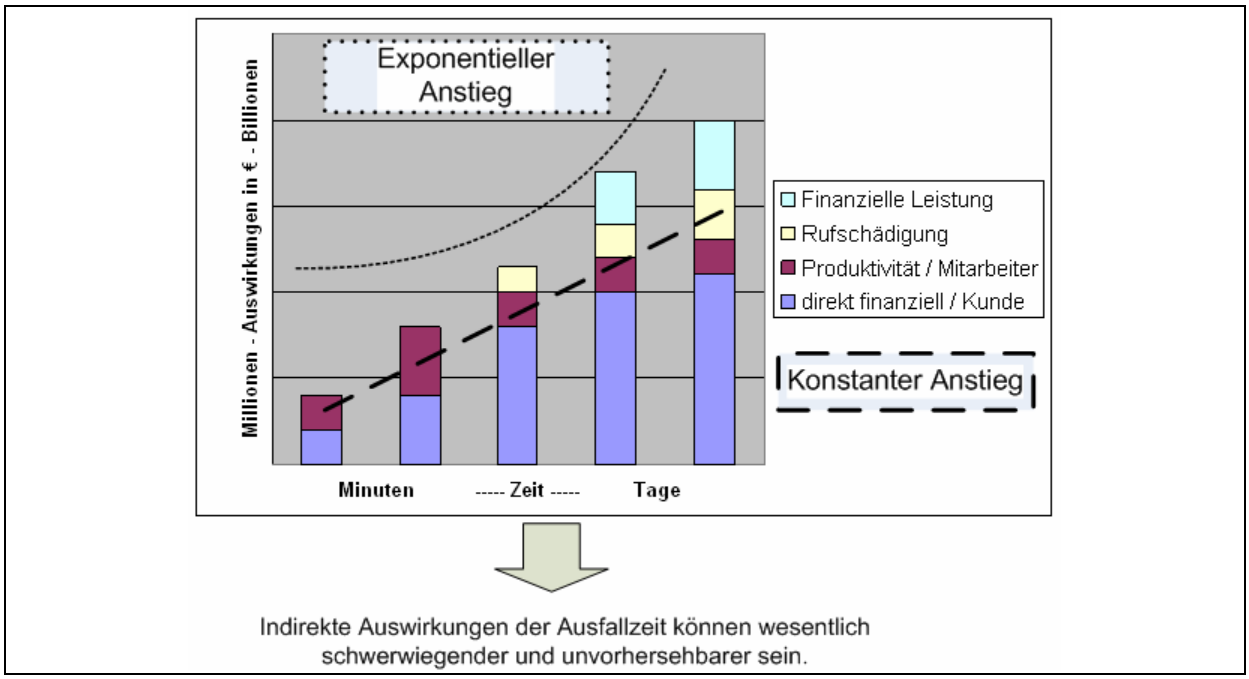


Abbildung 2-3 - Auswirkungen eines Ausfalls
(Quelle: [hp.com, 2007])

2.3 Ursachen ungeplanter Ausfälle

Laut [Gartner-HP, 2006] werden 80 Prozent aller ungeplanten Ausfallzeiten durch Bedienfehler oder mangelhafte Prozesse verursacht. In vielen Fällen wohl durch mangelnde Achtsamkeit der Verantwortlichen – ein falscher Klick oder Parameter und schon kann ein System nicht mehr verfügbar sein. Bis ein System wieder verfügbar ist vergeht wertvolle Zeit, dazu kommt noch die Dauer für den Start einer Applikation, eines Dienstes und die eventuell notwendigen Recovery-Aktionen auf Grund des unvorhergesehenen „Restarts“. Die Komplexität erhöht sich mit dem Einsatz von verteilten Systemen.

Auf der anderen Seite stehen sicherlich **Fehler** im Bereich der eingesetzten **Software**. Sei es nun ein neuer Patch auf der Betriebssystemseite, ein neuer Treiber, ein Patch der Applikation oder eine „einfache“ Umkonfiguration an einer der beteiligten Komponenten – es ist empfehlenswert, Software oder Softwareteile erst nach einem angemessenen Qualitätssicherungsprozess in der Produktivumgebung einzusetzen und jede Änderung entsprechend zu dokumentieren. Die Anzahl der möglichen Ursachen für ungeplante Stillstände liegt wohl in einem unendlichen Bereich.

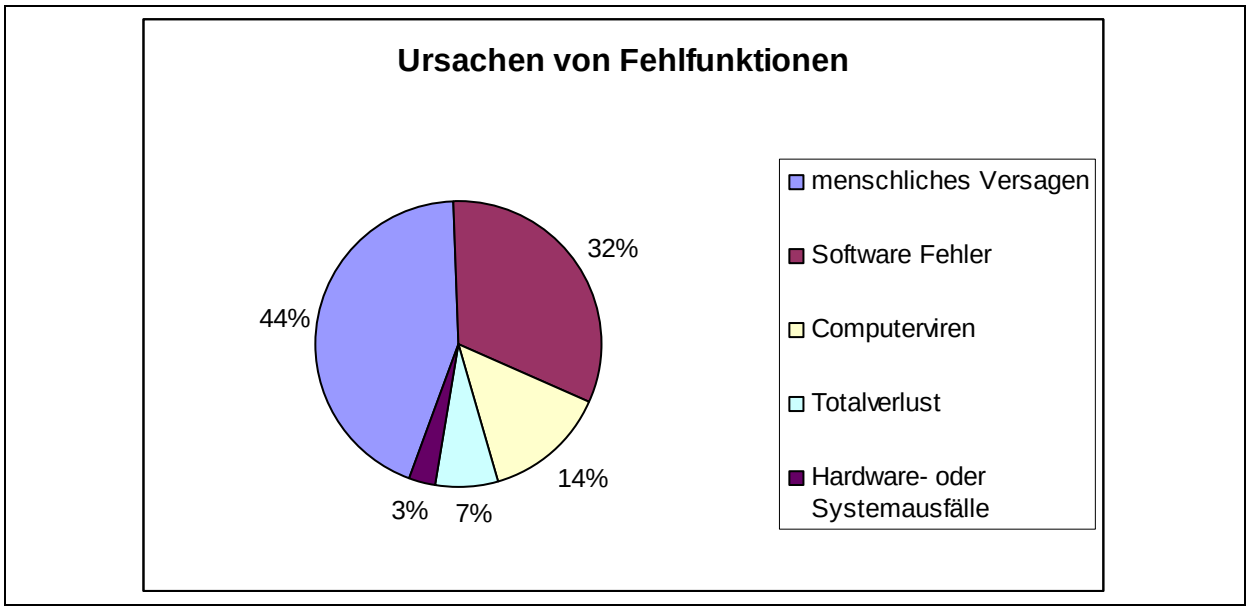


Abbildung 2-4 - Ursachen von Fehlfunktionen
[datac-gmbh.de, 2006]

Abbildung 2-4 zeigt deutlich, dass gegen rund 53 % der Ausfallsursachen (menschliches Versagen, Software Fehler und Computerviren) keine noch so hochverfügbare Hardware-Lösung helfen kann. Zu diesen Ausfällen kommen dann noch die geplanten Ausfälle auf Grund von Wartungs- und Upgrade-Arbeiten. Im Vergleich dazu zeigt Abbildung 2-5 eine Aufstellung von [hp.com, 2007] wonach weniger als 50 % der ungeplanten Ausfälle durch Systemfehler verursacht werden.

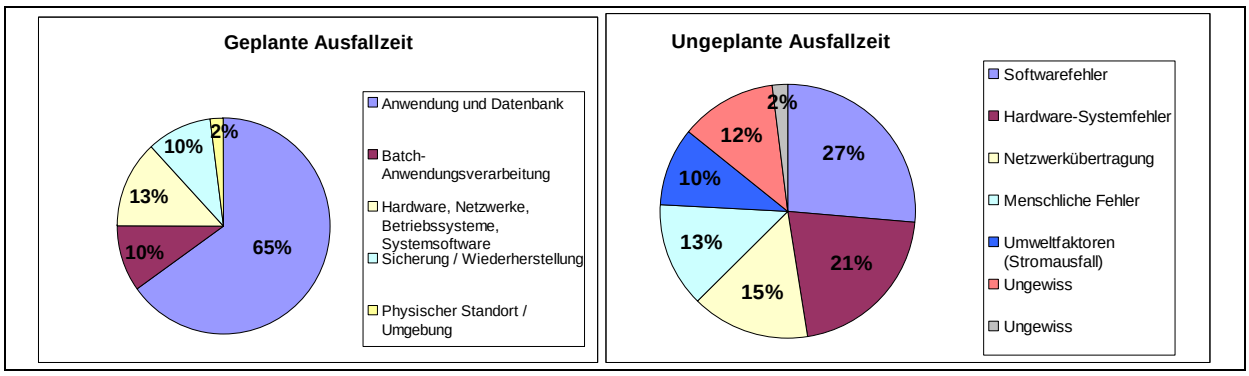


Abbildung 2-5 - Risiken / Ursachen von Ausfällen

2.4 Nutzen von Monitoring?

Im nächsten Kapitel werden die Kosten von Monitoring betrachtet bzw. kurz diskutiert. Aber was ist der Nutzen von Monitoring?

Wie vielfach schon in vorhergehenden Kapiteln erläutert, bringt Monitoring vor allem den Vorteil, jederzeit über **die Zustände der überwachten Systeme** Bescheid zu wissen. Ohne Monitoring könnte dem Vorgesetzten erst nach mehr oder weniger hohem Aufwand und entsprechender Zeitdauer ein Statusbericht gegeben werden. Mit Monitoring kann sich dieser Vorgesetzte ohne Aufwand selbst ein Bild davon machen.

Andererseits stellt Monitoring die Voraussetzungen her, um schneller und vor allem **rechtzeitig** über Probleme und eventuelle Engpässe **informiert** zu werden. Dazu liefert das System zusätzliche Informationen über das jeweilige Problem bzw. unterstützt bei der **Fehlerlokalisierung**. Es werden erste **proaktive Aktivitäten** ermöglicht. Dadurch, dass das System 7x24 Stunden läuft, werden Fehler auch außerhalb der Arbeitszeit erfasst und es kann eine rechtzeitige Verständigung (zum Beispiel eines Bereitschaftshabenden) erfolgen. Durch die Verfügbarkeit von Informationen über Probleme bzw. deren Lokalisierung bietet eine umfangreiche Monitoring-Lösung den Grundstein für ein umfassendes **Incident Management** zum Beispiel auf Basis von ITIL.

Durch eine überlegte Archivierung der gesammelten Daten (z.B. Performance-Daten) kann eine Übersicht über die **historische Entwicklung** abgerufen werden. Durch Analyse historischer Daten und Einberechnung der aktuellen Situation und Erfahrungswerte können **Trends** vorausgesagt und dadurch das **Planning** erleichtert und unterstützt werden. Es können Aussagen über Soll- und Ist-Zustände getätigt werden. War das Planning richtig? Oder wurden bestimmte Parameter und Entwicklungen falsch eingeschätzt?

Ein Messen und Auswerten diverser Parameter ermöglicht die Einhaltung und Überprüfung verschiedener Anforderungen. Seien dies nun auf Grund von **vertraglichen Vereinbarungen** mit Kunden (Service Level Agreements) oder **gesetzliche Anforderungen** (z.B.: gemäß

Basel II, Sarbanes-Oxley-Act (SOX), EUROSOX[12]). Durch die Automatisierung des Monitoring-Prozesses wird die Möglichkeit für die IT-Verantwortlichen geschaffen sich auf das **Kerngeschäft** zu konzentrieren.

Situation vor/ohne Monitoring:

- Reaktives Handeln – Management by „Turnschuh“
- Hohe Risiken für das Unternehmen und die Geschäftsprozesse
- Verletzungen von SLAs

Situation mit Monitoring:

- Proaktives Handeln
- Verringerte Risiken
- Einhaltung von Service Levels
- Optimierung von Ressourcen-Nutzung
- Höhere Verfügbarkeit(en)
- Kenntnis von Problemen vor den Anrufen der Benutzer
- Weniger manuelle Schritte reduzieren Management-Kosten
- Mehr Transparenz

[12] http://www.eurosox.at – Das Management muss die Integrität des Prozesses der Finanzberichterstattung und die dafür bereitgestellte Technologie kontinuierlich prüfen und dokumentieren. Der SOX-Prüfer muss nach Schwachpunkten in der IT suchen.

2.5 Kosten versus Verfügbarkeit - Was kostet Monitoring?

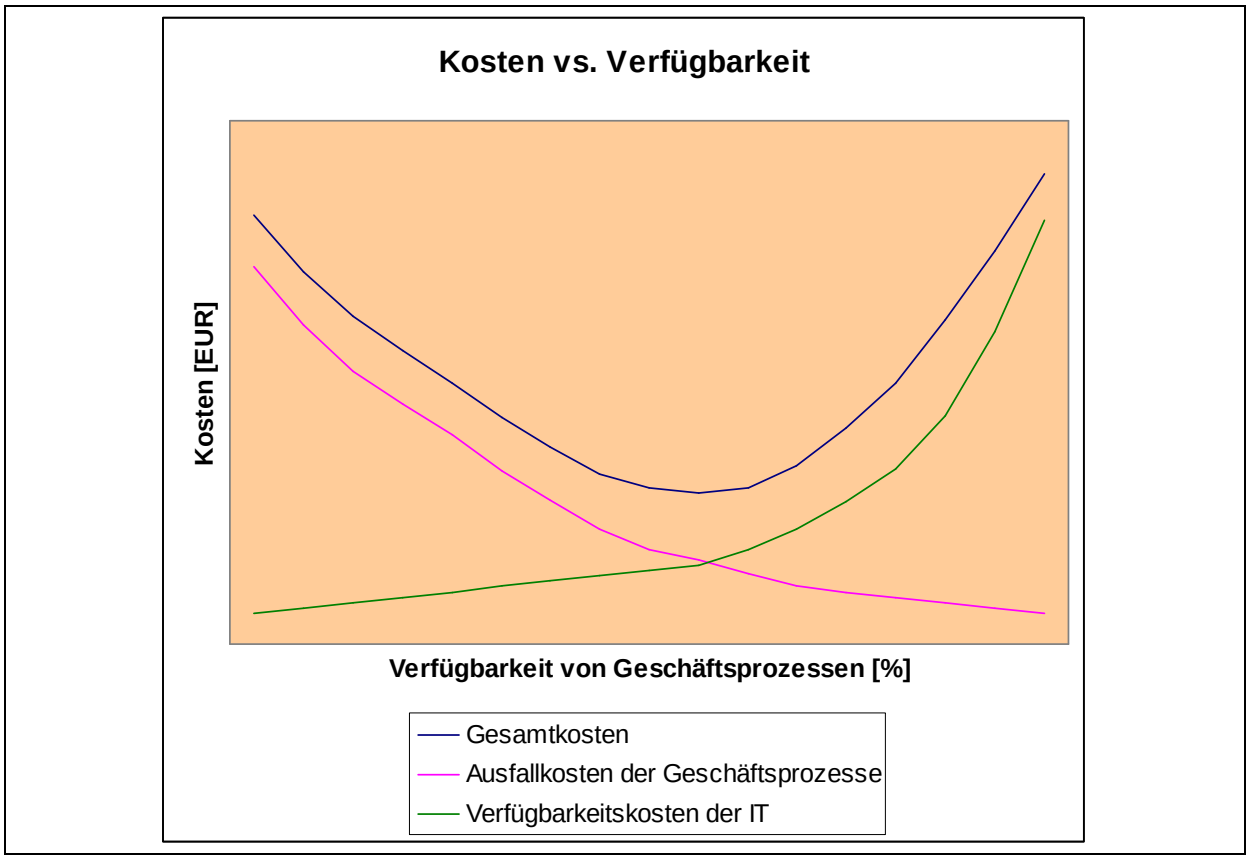


Abbildung 2-6 - Verfügbarkeitskosten der IT vs Ausfallkosten der Geschäftsprozesse [4managers.de, 2006]

Abbildung 2-6 versucht die Zusammenhänge zwischen Kosten zur Erhöhung der Verfügbarkeit und Kosten die durch Ausfälle der IT und dadurch durch Ausfälle von Geschäftsprozessen darzustellen. Je höher die Verfügbarkeit der Geschäftsprozesse auf Grund der Verfügbarkeit der IT sein soll, desto höher sind die Kosten die investiert werden müssen, um Ausfälle der IT zu vermeiden. Die Verfügbarkeit der IT-Infrastruktur kann nicht bis ins Unendliche getrieben werden – es gibt einen Punkt, wo eine hinreichende Verfügbarkeit der Geschäftsprozesse mit einem akzeptablen Aufwand gewährleistet werden kann (siehe auch Punkt 1.4.4). Ab einem gewissen Punkt (kostenoptimaler Punkt) stehen unverhältnismäßig hohe Kosten einem geringen zusätzlichen Nutzen gegenüber.

Als Gegenargument eines umfassenden Monitoring-Systems werden vielfach die Lizenzkosten (die Produktkosten) angegeben. Dies stimmt bei kommerziellen Software-Paketen sicherlich bis zu einem gewissen Punkt. Um dieses Argument zu entkräften ist das

Ziel des Migrationsprojektes die Umsetzung eines Monitoring-Systems für das Unternehmen auf Open Source-Basis, wodurch keine Lizenzkosten anfallen. Auf keinen Fall dürfen die **Implementierungs**- und **Konfigurationsaufwände** außer Acht gelassen werden. Diese sind sowohl bei kommerziellen Produkten als auch bei Open Source-Software (OSS) einzukalkulieren. Es lässt sich kaum eine seriöse Aussage treffen, dass die Aufwände bei OSS höher bzw. geringer sein werden als bei lizenzierten Produkten – siehe die verschiedenen Studien bezüglich Microsoft Produkten und deren OpenSource-Pendants. Auch innerhalb der Produktklassen gibt es mehr oder weniger hohe Unterschiede in den Adaptierungsaufwänden. Diese Aufwände hängen einerseits von der Komplexität als auch der Heterogenität der zu überwachenden IT-Landschaft ab, als auch des gewünschten Anpassungsgrades an firmeneigene Wünsche. In vielen Fällen sind Standardkonfigurationen ausreichend, in anderen Fällen müssen Konfigurationen an die eigene Landschaft Schritt für Schritt angepasst werden – typische Beispiele hierfür wären verschiedene Tresholds (Schwellwerte) beim Ressourcen-Monitoring.

2.6 Was kann Monitoring nicht?

Eine Monitoring-Lösung kann auf keinen Fall eine strukturierte und koordinierte Fehlereingrenzung ersetzen – allerdings unterstützt es bei der Lokalisierung von Fehlern und bei der Sammlung von Informationen. Verbunden mit gut etablierten ITSM-Prozessen (IT Service Management) (siehe ITIL) bietet eine Monitoring-Lösung einen Großteil der notwendigen Informationen für eine umfassende Fehlereingrenzung und –Analyse.

Das Monitoring-System kann keine Dokumentation ersetzen – verbunden mit einem Inventory-System erleichtert es allerdings die Arbeit des System-Administrators. Eine sorgfältige Vorgehensweise der Administratoren ist Voraussetzung jeder IT-Infrastruktur und kann nicht durch eine Software-Komponente ersetzt werden. Wie auch in Abbildung 2-4 aufgezeigt, ist ein Großteil der Ausfälle durch menschliches Verhalten/Versagen verursacht und kann nicht durch eine lückenlose Überwachung vermieden werden. Bei der Umsetzung bzw. bereits bei der Analyse der Ist-Situation werden oftmals fehlerhafte Prozesse erkannt und können dadurch angepasst bzw. korrigiert werden.

Ganz allgemein formuliert: Ein Monitoring-System kann nur jene Systeme und Stati überwachen, welche im System implementiert bzw. durch den Administrator abgebildet und konfiguriert wurden. Ein Monitoring-System ist kein Allheilmittel – geschweige denn ein Wundermittel, das schlagartig alle technischen und auch prozessbedingten Probleme löst und vermeidet.

3 Monitoring-Systeme

Auf dem Markt sind verschiedenste Produkte für die Überwachung von IT-Komponenten verfügbar. Manche Lösungen legen ihren Fokus auf eine Applikation bzw. Technologie, wie z.B. Microsoft Exchange, Microsoft Security Log oder auch Statistiken und Zugriffsanalyse eines Webservers. Andere Produkte haben sich auf das Realtime-Monitoring spezialisiert, andere wiederum auf mittel- bzw. langfristige Trendanalysen. Quer über diese Lösungen gibt es wiederum Produkte, welche eher für Klein- und Mittelbetriebe (KMUs) ausgelegt sind, andere für große Unternehmensnetzwerke.

Um die bestmögliche Monitoring-Lösung auszuwählen, ist es wichtig vorher sorgfältig die Anforderungen (**Requirements**) für das Unternehmen zu definieren. Dazu ist es notwendig zu wissen, welche Probleme und Herausforderungen in Bezug auf Monitoring auftreten können – System Monitoring zeigt sich als technisch komplexes Konstrukt. Nach der Definition der Requirements müssen die wichtigsten **Risiken** in Bezug auf die IT-Infrastruktur identifiziert werden.

Eine eingesetzte Monitoring-Lösung muss alle notwendigen Systeme und Technologien überwachen und überprüfen können – dazu muss das System ausreichend **Kenntnis** über die eingesetzten Technologien aufweisen. So können durch verschiedene Überwachungs-Technologien unterschiedliche Sichtweisen auf das überwachte System erlangt werden (siehe auch Punkt 3.2).

Die Anforderungen an die Monitoring-Lösung in Punkto Technologien wachsen schnell mit den unterschiedlich zu überwachenden Systemen/Technologien (z.B.: verschiedene Betriebssysteme, Netzwerk-Devices, Datenbank-Management-Systeme (DBMS), Mailing-Systeme, Web-Services, usw.). Weitere wichtige Punkte bei der Bewertung und Auswahl von

Monitoring-Systemen sind die **Skalierbarkeit** (Scalability) und die Wartbar- bzw. **Managebarkeit** (Manageability). Eine Monitoring-Lösung muss weiters sinnvolle und praktikable Techniken zur Analyse von Log-Daten (z.B. Log-Files, Eventlog,...) und für das Reporting zur Verfügung stellen. Alternativ können geeignete Schnittstellen (z.B. auch Speicherformate von Logfiles) zu Reporting-Tools (ev. „Crystal Decisions' Crystal Reports[13]“) angeboten werden. Im Gegensatz zum Reporting steht die **Alarmierung** (Alerts). Beim Reporting werden historische Daten analysiert und ausgewertet bzw. die aktuellen Stati der Systeme zu einem bestimmten Zeitpunkt zusammengefasst. Bei der Verständigung sollte möglichst nah an Echtzeit auf Grund von definierten Regeln bei Auftreten eines Ereignisses eine (Verständigungs-)Aktion (z.B.: E-Mail oder SMS) gesetzt werden. Das definierbare Regelwerk sollte für die jeweilige Situation in ausreichender Granularität Konfigurations- und Filtermöglichkeiten bieten. Es müssen soweit wie möglich falsche Alarme und eine zu große Informationsflut vermieden werden. Es soll nicht passieren, dass ein Administrator mehr Alarme zugestellt bekommt als notwendig – ansonsten könnte es zum Phänomen „alarm fatigue“ kommen - dabei kommt es zu unnötigen bzw. falschen Alarmen was früher oder später zu einer Art „Müdigkeit“ führt, welche die Reaktion auf echte und wichtige Fehler/Alarme beeinträchtigt. Fehlalarme machen nachlässig. Beim Einsatz eines Monitoring-Systems werden normalerweise Standard-Schwellwerte und Konfigurationen eingestellt – das System muss schrittweise an die eigenen Systeme und Rahmenbedingungen angepasst werden. Ein Monitoring-System muss einfach hand zu haben und zu verwalten sein und Möglichkeiten zu Erweiterungen anbieten.

3.1 Prinzipieller Aufbau

Die Architekturen von Monitoring-Lösungen können folgendermaßen unterteilt werden:

- Monolitisch
- Agent-basierend
- Hybrid

Die **monolithische** Architektur besteht aus nur einer Komponente, welche die Datensammlung, das Monitoring, das Alerting und womöglich auch die Reporting-Funktionen durchführt. Im einfachsten Fall wird der Server installiert und dann nur noch die

[13] http://www.germany.businessobjects.com/

zu überwachenden Systeme eingetragen. Vorteil dieser Architektur ist mit Sicherheit, dass auf den Systemen keine extra Software installiert werden muss. Aus Gründen der Skalierbarkeit und der meist fehlenden Granularität bei der Konfiguration von unterschiedlichen Systemen sind solche Systeme vor allem für kleinere Unternehmen geeignet. Da die Überwachungs-Arbeit von zentraler Stelle vorgenommen wird, ist auch eine gewisse Load (abhängig von der Anzahl der überwachten Systeme und den definierten Zeitintervallen) am Netzwerk vorprogrammiert und nicht zu unterschätzen.

Bei der **Agenten[14]-basierenden** Architektur wird der Prozess zur Datensammlung in eine eigene eigenständige Komponente – den Agenten – ausgelagert. Dieser Agent muss auf jedem überwachten System installiert werden und überwacht dort lokal das jeweilige System und sammelt Daten. Das Netzwerk wird vom Agenten nur dann benötigt, wenn definierte Schwellwerte über- bzw. unterschritten werden, ein Problem auftritt oder der Agent mit dem zentralen Monitoring-Server in Kontakt treten muss (Übertragung der gesammelten Daten, Keep-alive-Nachrichten,...). Auf diese Weise kann eine große Anzahl an Systemen mit höherer Frequenz überwacht werden. Als eventuelle Nachteile sind allerdings der Installationsaufwand als auch die „Fremdsoftware“ im System zu nennen. Fremdsoftware birgt ein gewisses Risiko (Software ist nicht immer fehlerfrei) in sich wodurch in manchen Fällen der Support durch verschiedene Lieferanten entfällt.

Bei der **hybriden** Architektur kann sich der Administrator die für das Unternehmen beste Lösung zusammenstellen und dabei die Nachteile der einen Variante durch die Vorteile der anderen ersetzen. Bei dieser Variante werden den Anwendern die größtmögliche Flexibilität und Skalierbarkeit geboten.

---

[14] *Ein Agent ist normalerweise eine Software, die Anfragen aktiviert und Antworten bearbeitet.*
1. In Netzwerkmanagement-Systemen residieren die Agenten in allen managebaren Geräten und übertragen die Werte von speziellen Einstellungen und Parameter zur Managementstation. Auf Aufforderung eines Managers oder beim Auftreten eines Ereignisses übermittelt der Agent die Informationen an die Management-Konsole. Die Kommunikation zwischen der Management-Station und dem Agent findet bei OSI-Protokollen mittels CMIP statt, bei TCP/IP-Protokollen mittels SNMP.
2. Im Internet ist ein Agent ein Programm, das im Auftrag des Anwenders Suchprozesse durchführt. [itwissen.info, 2007]

3.2 Monitoring-Strategien

3.2.1 Blackbox-Monitoring

Unter Blackbox-Monitoring wird der Zugriff auf ein System bzw. ein Service von außen wie ein Anwender/Benutzer des Services verstanden. In den meisten Monitoring-Systemen sind Plugins bzw. Funktionalitäten für die Standardprotokolle (z.B.: FTP, HTTP, NFS, CIFS,...) bereits inkludiert. Für eigene Anwendungen bzw. Spezialapplikationen müssen eigene Überprüfungsskripts für die automatische Überprüfung implementiert werden. Für einen Webshop zum Beispiel kann das Skript automatisch einkaufen und so die Funktionalität des Webshops überprüfen. Der Auftrag könnte speziell markiert werden, sodass keine weitere Verarbeitung oder gar Durchführung des Einkaufs angestoßen wird.

3.2.2 Whitebox-Monitoring

Unter Whitebox-Monitoring wird die Überprüfung von Einzelkomponenten einer Anwendung bzw. eines Systems verstanden. Am Beispiel des Webshops könnten dadurch zum Beispiel folgende Teilkomponenten überprüft werden: die notwendigen Datenbanken, Webserver (Frontendserver), Backendserver, Netzwerkkomponenten. Das Whitebox-Monitoring ist gerade für den Administrator für die Problemlokalisierung wichtig – allerdings liefert dies naturgemäß nicht die gleiche Sicht auf das System bzw. die Applikation / Situation wie für den Anwender.

3.2.3 Messwerte

Messwerte liefern laufend Informationen und je nach Applikation, System und Anforderung unterschiedliche Leistungsdaten. Beispiele für Leistungsdaten können CPU-Auslastung, Netzwerkbelastung, Festplattenplatzauslastung, Cache, abgearbeitete Requests (z.B.: beim Webshop) sein. Die Messwerte werden zur Alarmierung bei nicht optimaler Leistung (z.B.: bei der Einhaltung von SLAs notwendig), zur Alarmierung bei bedrohlichen Zuständen (z.B.: Festplattennutzung > 90 %) und auch für Tuning- und Erweiterungsmaßnahmen herangezogen.

4 Migrationsprojekt des Unternehmens

Wie auch in der Einleitung schon kurz skizziert, basiert das in dieser Arbeit betrachtete fiktive Unternehmen auf praktischen Erkenntnissen und Erfahrungen. Das unter 4.5 beschriebene Environment als auch die betrachtete IT-Infrastruktur ist ein Modell eines real bestehenden Unternehmens und könnte in dieser oder ähnlicher Konfiguration ohne weiteres öfter im Einsatz sein oder installiert werden.

Das Unternehmen setzt schon seit längerer Zeit auf Open Source Produkte – vor allem im Bereich der Betriebssysteme der Serverlandschaft, als auch bei Webservern und Datenbank-managementsystemen. Vom obersten Management ist wie in vielen österreichischen Unternehmen der Punkt „Monitoring“ derzeit (noch) nicht als „wichtig“ eingestuft, was sich im verfügbaren Budgetrahmen und den Personalressourcen widerspiegelt.

Das dieser Arbeit zu Grunde liegende Projekt definiert als Ziel ein produktionsreifes Konzept für eine Migration zu einer Monitoring-Lösung der IT-Landschaft des Unternehmens und besteht aus mehreren Teilschritten (siehe auch 4.4) Diese Arbeit dokumentiert das erste Teilprojekt, die Konzeption des Monitoringsystems und Umsetzung im Testenvironment. Nach erfolgreich abgeschlossenen Tests folgt der Einsatz im Produktionsumfeld (nicht mehr Teil dieser Arbeit), anschließend daran sind diverse Verfeinerungs- und Erweiterungsprojekte geplant.

4.1 Aktuelle Situation im System-Management-Bereich

Der erste Schritt in diesem Migrationsprojekt war die Aufstellung und Analyse des aktuellen Ist-Zustandes. Es war dies zudem auch der wichtigste Punkt in Hinblick auf die letztendliche Lösung und deren Erfolg.

4.1.1 Typischer Ablauf bei Systemausfall ohne Monitoring

Die aktuelle Situation bei der Fehlererkennung läuft meist so ab, dass der Administrator durch Zufall den Ausfall oder eine Beeinträchtigung einer Komponente erkennt oder bereits Störungsmeldungen durch die Benutzer erfolgen. Teilweise sind Insellösungen für die Hardware-Überwachung eingesetzt. Andererseits existieren proprietäre Lösungen für einige wenige Applikationen bzw. Dienste. Bei Installationen und Konfigurationen wurden bereits

verschiedene Ausfallsszenarien und Redundanzen bedacht: z.B.: RAID-Systeme, USV-Versorgung, Installation der Services auf mehreren Systemen usw. (siehe auch Abbildung 4-1).

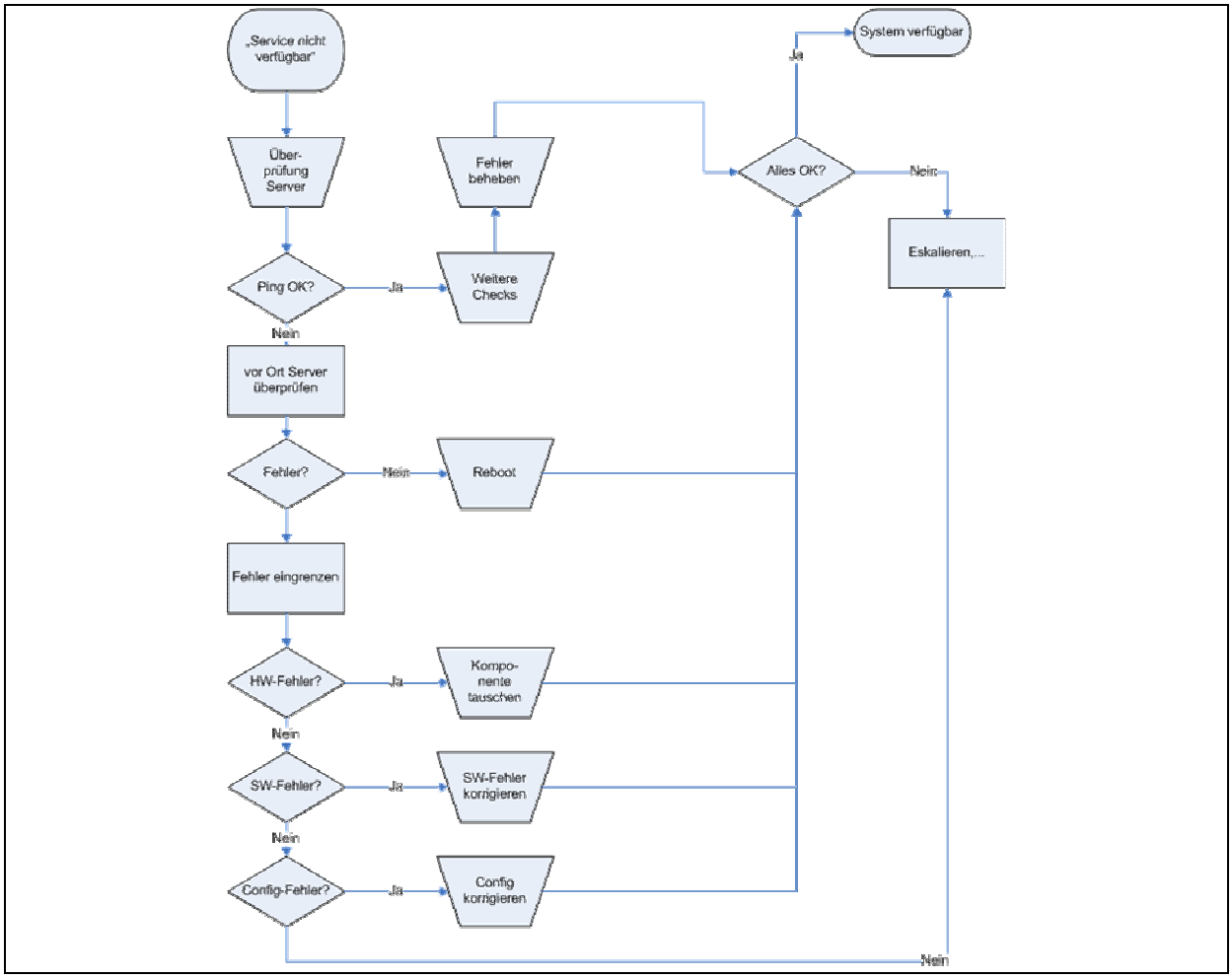


Abbildung 4-1 - typischer Ablauf ohne Monitoring

4.2 Ziel-Definition

Das End-Ziel des Migrationsprojektes ist eine umfassende und automatisierte Überwachung aller IT-Komponenten mit automatischer Verständigung der zuständigen Stellen, sowie einem zentralem Event- und Logmanagement. Eine zusätzliche Forderung wird die Integration und Überwachung von Service Level Agreements innerhalb des Unternehmens sein – dafür sollen bereits Überlegungen bzw. Vorbereitungen getroffen werden. Die gesammelten Daten sollen zukünftig für Aussagen über Trends und Zukunftsprognosen und dadurch auch für Beschaffungs- und Aufrüsttätigkeiten die Grundlagen bilden.

Grundziele:

- Das Hauptziel des einzuführenden Monitoring-Systems ist die Verbesserung der aktuellen Situation bei Ausfällen der IT.
- Ein ähnlich gewichteter Punkt ist der oftmals umgangssprachlich als „Management by Turnschuh“ bezeichnete – der Administrator läuft los, sobald er eine Störungsmeldung durch einen Benutzer bekommt – soll soweit wie möglich eliminiert werden.
- Die Administratoren wollen über eventuelle Probleme schon vor einem Anruf des Benutzers informiert sein.
- Die eingesetzte Monitoring-Lösung soll so günstig wie möglich aber doch hinreichend sein → keine Einwände gegen ein Open Source-Produkt (siehe auch Punkt 4.7.2)
- Weniger ungeplante Downtimes
- Risikominimierung von Downtimes
- Sensibilisierung im Unternehmen für Monitoring
- Einführung für die zuständigen IT-Administratoren

Technische Ziele:

- Aktive Überwachung aller Server- und Netzwerkkomponenten, welche für die Verfügbarkeit der Basisdienste (siehe Punkt 0) notwendig sind.
- Prinzipielle Überwachung der Erreichbarkeit via IP (via ICMP)
- Einfache Ressourcenüberwachung der Serversysteme und Netzwerkkomponenten
- Jederzeitige Übersicht über den aktuellen Status / Zustand der überwachten Objekte (im ersten Schritt für die Administratoren, ev. lesenden Zugriff für Guests)
- Automatische Verständigung des Administrators bei bestimmten Events bzw. Problemen via E-Mail
- Zugriff via Web-Interface für eine gewisse Ortsunabhängigkeit ➔ auch in Hinblick auf Bereitschaftsdienste
- Es soll eine kostengünstige, aber hinreichende technische Lösung umgesetzt werden.
- Bei der Auswahl der Produkte soll auf technische Standards gesetzt werden.
- Die Lösung soll weitgehend flexibel gestaltet und für weitere Umsetzungs- und Erweiterungsschritte vorbereitet werden.
- Gesammelte Daten sollen für historische Übersichten als auch für Zukunftsprognosen und Trends gespeichert und ausgewertet werden.

- Eine Integration in bestehende Systeme (z.B.: Dokumentationen) soll in die Überlegungen einbezogen werden.
- Für eine bessere Verfügbarkeit der Dienste sollen generelle Überlegungen zur Erhöhung dessen angestellt werden.
- Überlegungen zu Komponenten, welche aus technischen Gründen nicht überwacht bzw. nicht in das gewählte System einbindbar sind.
- Zentrales Log- und Eventmanagement
- Einheitliche und übersichtliche Darstellung der Informationen
- Ampelprinzip bzw. Ansichten für das Management
- Ressourcen-Engpässe frühzeitig erkennbar machen
- Automatische Dokumentation über Events, Downtimes, Änderungen,...

4.3 Teilprojektziel - Masterarbeit

Das Ziel dieser Arbeit ist die Dokumentation des Konzeptions- und Realisierungprozesses des Monitoring-Systems für die Testsysteme, bevor der erste Produktiv-Schritt, welcher ein Basis-Monitoring zur Sicherstellung der Grund- und Basisdienste zur Verfügung stellt, umgesetzt wird.

4.4 Grober Projektplan

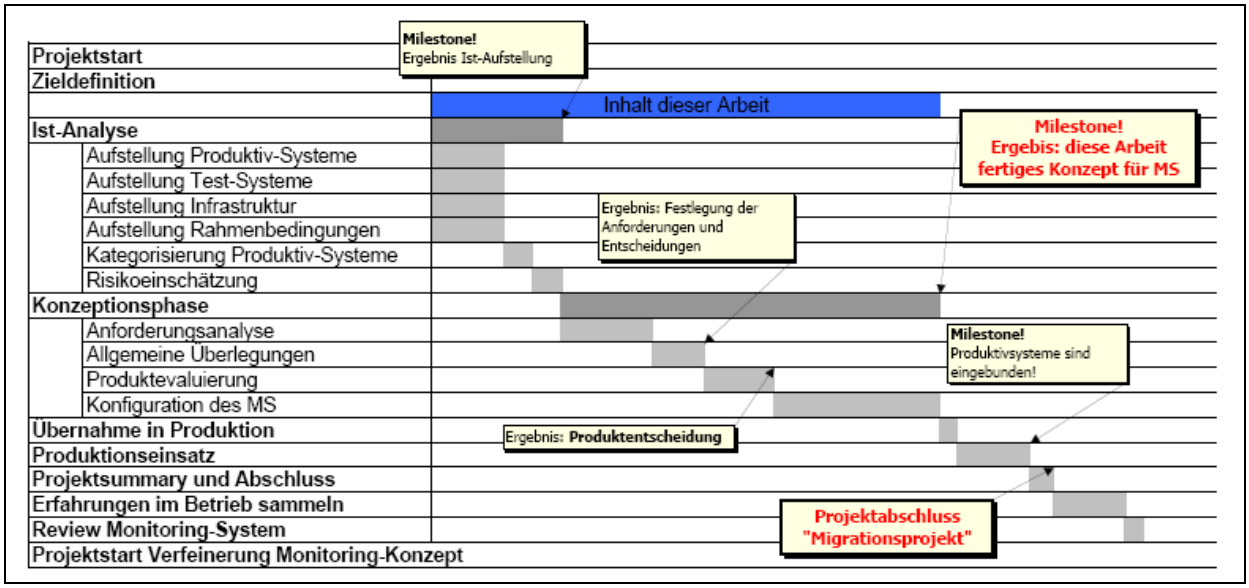


Abbildung 4-2 - grober Projektplan

Abbildung 4-2 zeigt skizzenhaft den groben Projektplan für das gesamte Migrationsprojekt. Es sollen damit vor allem die Abhängigkeiten und die durchgeführten bzw. geplanten Teilschritte verdeutlicht werden. Der Inhalt dieser Arbeit ist in der Skizze blau markiert und stellt am Abschluss der Konzeptionsphase mit dem fertigen Konzept gemeinsam die Ergebnisse für den zweiten Meilenstein im Migrationsprojekt dar. Die Länge der Balken gibt keine Aussage über die geplante bzw. aufgebrachte Zeitdauer, sondern soll nur einen groben Eindruck über die Zeitverhältnisse vermitteln und die notwendigen Reihenfolgen bzw. möglichen Überschneidungen wiedergeben.

Projektphasen:

- **Projektstart**: dazu sind im optimalen Fall die notwendigen Verantwortlichen als Projektteilnehmer eingeladen
- **Zieldefinition**: umfasst die Punkte 4.1 bis 4.3
- **Ist-Analyse** – Milestone! Artefakt ist eine komplette Aufstellung der Systeme und Infrastruktur
    - Aufstellung der Produktiv-Systeme
    - Aufstellung der Test-Systeme (Qualitätssicherung)
    - Aufstellung der Infrastruktur (USV, Klimaanlage, etc.)
    - Definition und Aufstellung der Rahmenbedingungen
    - Kategorisierung der Produktiv-System, Services, etc. (siehe Punkt 4.5.3)
    - Risikoeinschätzung bezüglich eines Ausfalles und Aufzeigen von Abhängigkeiten der Systeme untereinander (siehe Punkt 4.5.4)
- **Konzeptionsphase** – Milestone! Artefakte sind diese Arbeit und ein fertiggestelltes Konzept für das Monitoring-System.
    - Anforderungsanalyse (siehe Punkt 4.6), als Ergebnis entsteht eine Auflistung der Forderungen und Wünsche von Funktionaltitäten an das System, Pflichtenheft
    - Allgemeine Überlegungen und Entscheidungen (siehe Punkt 4.6.7)
    - Produktevalierung – Betrachtung verschiedener Software-Produkte, als Ergebnis steht die Produktentscheidung (siehe Punkt 4.7)

    - Konfiguration des MS auf Grund der Anforderungsanalyse (siehe Punkt 4.8), Artefakt ist das produktionsreife Konzept des MS
- Nach Abschluss der Vorbereitungsphasen – nicht mehr Teil dieser Arbeit – wird das Konzept inklusive Softwarepaketen und Dokumentationen in den Produktionsbetrieb übernommen.
- **Produktionseinsatz** – Milestone! Hier sollen bereits alle Produktionssysteme eingebunden sein und überwacht werden.
- Projektsummary und **Abschluss** – Milestone! Die Ergebnisse und Erfahrungen des gesamten Projektes sollen einem kurzem Review unterzogen werden. Als krönenden Projektabschluss ist ein kleiner Umtrunk aller Projektteilnehmer angedacht.
- Erfahrungen im **Produktionsbetrieb** sammeln
- Im **Review** des Monitoring-Systems sollen allen Erfahrungen und Probleme während des Produktionsbetriebes diskutiert und das Konzept gegebenenfalls überarbeitet werden. Als Ergebnis könnte die Entscheidung über Nachfolgeprojekte stehen.
- Start der **Nachfolgeprojekte**

4.5 Ist-Analyse - Inventur

Derzeit ist im Unternehmen kein 7x24-Stunden-Betrieb eingeführt. Dies bedeutet, dass eine Verfügbarkeit der IT-Dienste nur in der Zeit von Montag bis Freitag, von 07:00 Uhr bis 17:00 Uhr gefordert ist. Bei eventuell auftretenden Problemen und Ausfällen werden die Verantwortlichen erst nach zufälligen Informationen außerhalb der Betriebszeiten aktiv, während der Arbeitszeit auf Grund von Anrufen durch Mitarbeiter (Helpdesk). Test- und Qualitätssicherungs-Systeme fallen generell nicht in den Aufgabenbereich der Bereitschaft.

4.5.1 Produktions-Systeme

2 Windows-Server (Windows 2000 Server) mit Active Directory Services[15] (DCs)
2 Windows-Server (Windows Server 2003) mit MS Exchange[16] (Version 2003)
7-15 Windows-Server – Virtual Machines (Windows 2000 Server und Windows Server 2003)
7-10 Linux-Server – Virtual Machines (RedHat Linux[17])
1 Windows-Server (Windows Server 2003) mit Internet Information Services und MS SQL-Server (Version 2003)
4 Linux-Server mit Samba[18] (File- und Printservices)
2 Linux-Server – DNS und DHCP
2 Linux-Server mit Apache[19] und Timeservices
1 Linux-Server mit MySQL[20]
1 Linux-Server mit ProstgreSQL[21]
2 AIX[22]-Systeme (mit Oracle[23]-DB)
3 VMware ESX3-Server[24]
1 USV-Anlage
1 Temperatur-Überwachung
4 Cisco[25]-Switche
1 TapeLibrary von HP[26]
2 SAN FC-Switche (HP)[27]
1 EVA8000 – SAN-Storage[28]
1 Monitoring-Server

~ 59 Systeme

4.5.2 Test-Systeme

Die Test- und Qualitätssicherungsumgebung stellt ein Abbild der Produktionssysteme – wenn auch in abgespeckter Form - dar.

4.5.3 Kategorisierung der Systeme

Für die Integration ins Monitoring-System ist eine Kategorisierung / Gruppierung der Systeme sinnvoll bzw. notwendig. Die Kriterien für eine solche Kategorisierung sind in dem

[15] http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/activedirectory/default.mspx
[16] http://www.microsoft.com/exchange/
[17] http://www.redhat.com
[18] http://www.samba.org
[19] http://www.apache.org
[20] http://www.mysql.org
[21] http://www.postgresql.org
[22] http://www.ibm.com/aix
[23] http://www.oracle.com
[24] http://www.vmware.com/vinfrastructure/
[25] http://www.cisco.com
[26] http://welcome.hp.com/country/us/en/prodserv/storage.html
[27] http://h18006.www1.hp.com/storage/entrystorage.html?jumpid=reg_R1002_USEN
[28] http://h18006.www1.hp.com/products/storageworks/eva/index.html

Unternehmen bereits durch verschiedene organisatorische Rahmenbedingungen wie Aufteilung der Verantwortung und Betriebsführung auf verschiedene Organisationseinheiten (OE) nach Betriebssystemen bzw. Applikationen festgelegt bzw. eingeschränkt. Die weitere Kategorisierung für das Monitoring kann nach verschiedenen Gesichtspunkten erfolgen. Wünschenswert ist eine Abbildung im System nach mehreren Gesichtspunkten: z.B. nach den laufenden Diensten als auch nach dem Betriebssystem, wenn möglich zusätzlich nach Lokationen.

4.5.3.1 Gruppierung nach Funktionalität

Im Folgenden wird eine grundlegende Kategorisierung auf Grund der angebotenen Dienste definiert. Die Wichtigkeit von verschiedenen Diensten ist unter anderem unter Punkt 4.5.4 ersichtlich.

Produktionssysteme:

Infrastrukturdienste / Basisdienste:

- Domain Name System (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Domain Controller (DC) – Active Directory
- Time-Service

Mailing:

- Exchange-Server (+ Server in DMZ)

File- und Printerserver

Datenbank-Server:

- MySQL
- PostgreSQL
- Oracle DBMS
- MS-SQL

Web-Server:

- Proxy-Server

- Apache-Webserver
- Server mit MS-Internet Information Services

Restliche Applikationen:

- Restliche virtuelle Server

Network-Devices:

- Switche

SAN:

- EVA
- FC-Switche
- Tape-Library

Infrastruktur:

- USV
- Temperatur-Sensoren

4.5.3.2 Gruppierung der Systeme nach Betriebssystem

- Microsoft Windows
- Linux RedHat
- AIX
- VMware ESX
- Andere: USV, HP EVA, Tape Library,...

Eine gut überlegte und sinnvolle Kategorisierung bringt hohen Nutzen wenn die Überwachungsdaten ausgewertet werden und ein Bericht einen Überblick über die aktuelle Situation geben soll. Auch bei der Frage „wer wird bei welchem Problem verständigt?“ sind diese Informationen wichtig; aber auch bei geplanten Ausfällen (z.B. HW-/SW-Upgrades bzw. Umbauten) ist es wichtig zu wissen, welche Komponenten bzw. Services von diesen

Arbeiten betroffen sind – essentielle Bestandteile von ITIL[29] (IT Infrastructure Library). Heutzutage sind die verschiedenen Dienste nicht mehr auf einen Server beschränkt, sondern sind über mehrere Systeme verteilt bzw. gegenseitig abhängig.

4.5.4 Mögliche Auswirkungen von Ausfällen – Risikoeinschätzung

Objekt	Mögliche Auswirkung(en)	Mögliche Ursache(n), Risiken
AD, zentraler Authentifizierungs-Dienst	Keine Authentifizierung der internen Benutzer an der Workstation; keine Benutzung von Exchange / Mail-Client → AD, Global Catalog	Ausfall aller DCs (eher unwahrscheinlich); Ausfall zentraler Netzwerkkomponenten; logischer Fehler im AD: z.B.: Aufgrund von Konfigurationsänderungen, Upgrades,...
Exchange	Kein Zugriff auf Mails, Free-Busy-Informationen, Termine, Kalender	Korrupte Exchange-DB (tw möglich); Ausfall eines oder beider Exchange-Server
Mail-Server	Mailing nach extern oder von extern nicht möglich	Ausfall des Server bzw. der Internetverbindung, Probleme mit externem DNS
ESX-Server	Ev. teilweiser Ausfall von VMs, eher Performanceeinbussen da restliche ESX die VMs übernehmen (HA-Features)	HW-Fehler am ESX-Host; Risiko eher gering
Oracle-Server (DB)	2 wichtige Applikationen nicht verfügbar	Auswirkungen hoch
DB-Server (MySQL)	Intranet-Anwendungen nicht funktionsfähig	Kurze Ausfälle sind zu verschmerzen, Datenverlust darf nicht entstehen.
Printserver	Drucken für die Benutzer nicht möglich	Ausfall des Servers; keine Auswirkungen nach extern, für Benutzer unangenehm,
Fileserver	Zugriff auf zentrale Daten nicht möglich	Ausfall des Fileservers; je nach Daten bzw. Fileserver und Benutzer unkritisch bis hochkritisch

[29] (siehe [itil.org, 2006], [itil.co.uk, 2006] und [en.wikipedia, 2007] bzw. [de.wikipedia, 2007])

Timeserver	Bei Ausfall eines Timeservers wenig bis gar keine Auswirkungen, bei Ausfall beider Timeserver Auswirkungen auf verschiedenen Ebenen: z.B.: Authentifizierung (Kerberos); Datenbanken,...	Verursacht durch einen Serverausfall. Bei kurzen Ausfällen wenig kritisch, bei längeren Ausfällen (>1 Tag) beider Timeserver kritisch.
Webserver – Intranet	Siehe DB-Server (MySQL)	Siehe DB-Server (MySQL)
Webserver – Applikationen	Nichtverfügbarkeit ein oder mehrerer Applikationen	Je nach Applikation von eher unkritisch bis unternehmenskritisch, siehe auch Oracle-Server
DB-Server MS-SQL	Siehe Webserver - Applikationen	Siehe Webserver - Applikationen
DNS-Server	Beim internen DNS Auswirkungen auf alle Systeme (vor allem auch ADS), beim externen DNS Auswirkungen auf Mailing und externe Webseiten	Der Ausfall von DNS ist eher hochkritisch bis unternehmenskritisch einzustufen.
DHCP-Server	Keine Vergabe von IP-Adressen an interne Clients und Devices. Server arbeiten mit statischen IP-Adressen.	Durch eine sorgfältig gewählte Time-To-Live eher unkritisch, da die Devices mit den aktuellen Adressen weiterarbeiten können.
Proxy-Server	Bei Ausfall des Forward-Proxys kein Internet-Zugriff (Web, HTTP, HTTPS, FTP) für die Benutzer möglich. Bei Ausfall des Backward-Proxys Ausfall aller externen Webseiten und Applikationen.	Beim Forward-Proxy je nach Benutzertyp unterschiedlich – kurzzeitig eher unkritisch. Beim Backward-Proxy bis unternehmenskritisch.

Tabelle 2 - Ausfälle, Risikoeinschätzung, Auswirkungen

In Tabelle 2 wurde eine erste Einteilung der Systeme nach Diensten und deren Auswirkungen bei Nichtverfügbarkeit durchgeführt. Zusätzlich wurden grob mögliche Ursachen für den Dienst-Ausfall angegeben – bzw. welche Komponenten verfügbar sein müssen damit der gesamte Dienst verfügbar ist. Die Liste ist bei weitem noch nicht vollständig und wird im Zuge der Umsetzung der neuen Monitoring-Lösung noch angepasst und erweitert.

4.5.5 Netzwerkdiagramm – Produktionssysteme

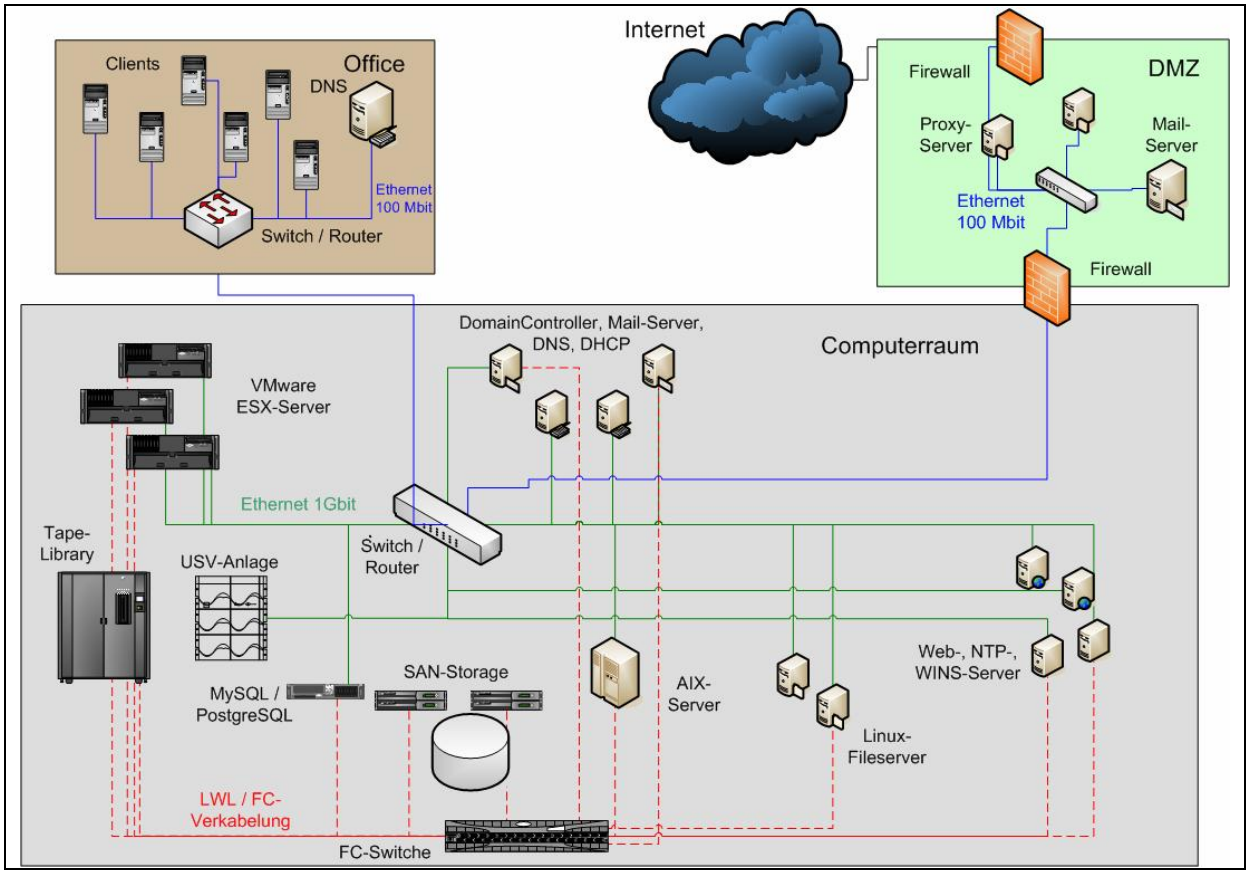


Abbildung 4-3 - Skizze Netzwerkdiagramm und Konfiguration

4.6 Anforderungsanalyse - Produktionssysteme

Für eine optimale Produktentscheidung und Konfiguration des Monitoring-Systems, sowie zur Abdeckung aller Anforderungen werden die Anforderungen an das System, die technischen Rahmenbedingungen und die organisatorischen Bedingungen analysiert und definiert. Die Testsysteme wurden bei der Anforderungsanalyse nicht näher betrachtet, da diese die Produktionsumgebung abbilden und daher impliziert betrachtet werden.

4.6.1 Zielgruppe - Für wen ist das Monitoring?

In erster Linie ist das einzuführende Monitoring-System (erster Schritt) für die System-Manager gedacht – globaler betrachtet natürlich für das gesamte Unternehmen, denn es sollen durch das System die Ausfälle reduziert, die Verfügbarkeit erhöht und die Qualität für die Benutzer gesteigert werden. Wie auch in den Projektzielen definiert, soll das System den

Administratoren helfen, Probleme und Fehler so schnell wie möglich zu erkennen und zu lokalisieren.

4.6.2 Aufbereitung der Informationen

Wie die Informationen aufbereitet bzw. dargestellt werden sollen, hängt von mehreren Faktoren ab. Einerseits ist die gewünschte bzw. sinnvolle Darstellung von der Zielgruppe abhängig. Andererseits kommt es darauf an, welche Informationen vermittelt werden sollen. So sind zum Beispiel beim Performance Monitoring überblicksmäßig die Stati der Systeme sinnvoll, bei Problemen muss dann Zugriff auf die Detailinformationen gegeben sein. Je nach Anforderung sollten die Daten beinahe beliebig aufbereitet und dargestellt werden können. Im Dashboard, dem Portal für die Benutzer ist das Ampelprinzip sinnvoll – wenn alles in Ordnung ist grünes Licht, bei leichten Problemen gelb und bei gravierenden Problemen das rote Licht. Ob statische oder dynamische Reports erstellt werden, hängt von der jeweiligen Anforderung ab. Für Statistiken und Auswertungen sind allerdings nicht nur Werte, sondern auch Grafiken gewünscht.

4.6.3 Welche Systeme und Devices müssen überwacht werden?

Die Auflistung der zu überwachenden Devices, sowie deren Kategorisierung sind unter den Punkten 4.5.3 und 4.5.4 ersichtlich.

4.6.4 Welche Objekte der Systeme müssen überwacht werden?

Im ersten Realisierungsschritt soll auf alle Fälle ein zentrales Event-Management umgesetzt werden – teilweise sind dabei bereits Ansätze eines Log-Managements inkludiert. Dazu sind die systemspezifischen Logs zu überwachen bzw. auszuwerten. Soweit SNMP-Traps von den Systemen unterstützt werden, so sind diese zu integrieren.

- Windows-Systeme: Eventlog
- Linux-Systeme: Syslog
- Netzwerk-Devices: Syslog

Für das Ressourcen-Management sind die grundlegenden System-Ressourcen zu überwachen:

- CPU
- Memory
- NICs
- Storage

Soweit bereits mit den zur Verfügung stehenden Mitteln umsetzbar sollen auch verschiedene System-Services und Applikationen integriert werden.
z.B.:

- FTP und HTTP-Services
- Prozesse
- Filesysteme – freier Speicherplatz, Quotas, etc.
- Filesysteme – Existenz von Files, Directories,...
- Antwortzeiten von HTTP-Services

4.6.5 Alarmierung bei Problemen

Bei den Möglichkeiten der Verständigung (Alarmierung) gibt es technisch gesehen keine Einschränkungen. Grundsätzliche Möglichkeiten sind unter anderem:

- E-Mail
- SMS
- Pager
- NewsTicker
- Dashboard
- Client-Server-Applikation
- Telefon/Handy ➔ Anruf

Im konkreten Fall sind die Anforderungen nicht so hoch gesteckt. Anfangs soll die Verständigung überwiegend per E-Mail[30] – erst im nächsten Schritt per SMS - erfolgen. Die Möglichkeiten von Anruf, Pager und einer Client-Server-Anwendung werden als nicht sinnvoll angesehen. Welche Person bzw. welcher Personenkreis zu benachrichtigen ist, hängt

[30] Auch im Hinblick darauf, dass keine Verständigung erfolgen kann / wird, wenn das Mailing-System oder der Netzwerkpfad ausgefallen ist, wird im ersten Realisierungsschritt auf E-Mail-Verständigung gesetzt.

vom System und den betroffenen Services ab. Grundsätzlich sind während der definierten Betriebszeiten die Administratoren zu benachrichtigen. Außerhalb dieser Zeiten werden auch die Administratoren benachrichtigt, diese allerdings auf Grund der Abwesenheit erst am nächsten Arbeitstag aktiv werden. Beim Vorhandensein einer Rufbereitschaft wird in Zukunft dann der Bereitschaftshabende außerhalb der Betriebszeiten verständigt – per SMS.

4.6.6 Reports und Statistiken

Das Reporting soll zu Beginn vor allem grundlegende Reports über Verfügbarkeit, Anzahl und Konfiguration der Systeme und Events / Fehler liefern. Diese Reports müssen nicht automatisiert generiert, sondern können in definierten Intervallen durch einen Benutzer händisch abgerufen werden – die Templates für die Reports sollen allerdings existieren. Es ist kein besonderes Format der Reports gefordert – es wird angestrebt, dass die Reports ebenfalls über den Webbrowser abgerufen und dann in ein PDF[31]-Dokument exportiert werden können. Individuelle Reports könnten Inhalte der weiterführenden Spezialisierungs-Projekte sein.

4.6.7 Allgemeine Überlegungen

Für die Auswahl der optimalen Lösung sowie die Definition der geforderten Funktionen wurden im Vorfeld einige allgemeine Überlegungen angestellt und diskutiert:

- Auf welcher Plattform soll das Monitoring-System implementiert werden?
- Ist es sinnvoll plattformunabhängig zu bleiben? Ist es besser wenn das System unter anderem auch auf Grund der Heterogenität der IT-Landschaft auf verschiedene Systeme verteilt und über ein zentrales System gemanagt wird? Existiert bereits eine Komplettlösung bzw. ein von einem System-Integrator abgestimmtes System am Markt?
- Wie schaut es mit der Skalierbarkeit des Systems aus? Wie viele Systeme sollen jetzt überwacht werden? Wie viele in nächster Zukunft?
- Es sollen keine unnötigen Redundanzen bei der Datenmessung entstehen – bestehende Lösungen sollen durch das neue System abgelöst werden. Bereits in Betrieb befindliche Insellösungen sollen abgelöst werden – bietet das neue System auch mindestens die gleichen Funktionen wie die bestehenden Lösungen? Wie kann eine Migration (im Grenzfall eine Integration) erfolgen? Bei Migrationen ist oftmals

[31] Portable Document Format

externer Support notwendig – welcher Support wird für das neue System angeboten? Steht dahinter ein kommerzielles Unternehmen oder basiert die Unterstützung „nur“ auf einer Community?

Begründung: Einer Integration der bestehenden Überwachungssysteme stehen hauptsächlich die proprietären Lösungen entgegen. Großteils sind die Lösungen auf einen sehr eingeschränkten bzw. speziellen Fokus (teilweise Eigenentwicklungen) zugeschnitten bzw. abgestimmt und bieten keine Schnittstelle zu einer gesamten Monitoring-Lösung. Der Aufwand für eine Adaptierung der meisten Lösungen scheint im Vergleich zu einer kompletten Ablöse durch das neue System unverhältnismäßig hoch. Im Grunde genommen ist ein Ziel des gesamten Projektes die optimale Ausnützung vorhandener Ressourcen und die Geringhaltung der notwendigen Aufwände. Das Rad neu zu erfinden sollte durch die Strategie „re-use before buy“ vermieden werden. Soweit möglich und sinnvoll sollen bereits erfolgreich erprobte Lösungen integriert – nötigenfalls mit Adaptierungen – integriert werden. Allerdings wird dies nicht nicht komplett möglich sein.

- Für Wartungsarbeiten muss es möglich sein, so genannte Maintenance-Windows im System zu definieren während denen zwar die Systeme überwacht werden, aber keine Verständigung erfolgt.
- Welche Schnittstellen sind für die Integration anderer bzw. in andere Systeme notwendig bzw. sinnvoll? Das vorhandene Inventory-System soll mit dem Monitoring-System Daten austauschen: z.B. sollen alle im Inventory-System eingetragenen Systeme automatisch nach gewissen Regeln ins MS übernommen werden. Welche Schnittstellen existieren zu anderen Systemen? Können welche selbst implementiert werden? Wenn ja, mit welchem Aufwand?
- Inwieweit können existierende SLAs im System abgebildet werden?
- Wie lange sollen die gesammelten Daten aufbewahrt werden? Für die Analyse und Trendprognosen sind historische Daten wichtig. Die Aufbewahrung der Daten hat Auswirkung auf die zu speichernde Datenmenge und dadurch auf verschiedene andere Parameter.
- Es sollen zukünftig auch die Systeme in der DMZ überwacht werden – kann das System über Firewalls hinweg überwachen?

- Wenn nicht alle notwendigen Funktionen im System integriert sind, macht dies Eigenentwicklungen vor allem für proprietäre Applikationen oder exotische Systeme notwendig. Sind Eigenentwicklungen möglich? Wenn ja, mit welchem Aufwand? Ist das notwendige Know-How (z.B. Programmiersprache) im Unternehmen vorhanden?
- Sind Aktionen auf Grund von Ereignissen bzw. Fehlern sinnvoll? Wenn ja, inwieweit besteht die Möglichkeit diese in dem System zu implementieren?
- Wie komfortabel sind Managebarkeit und Parametrierbarkeit des Gesamtsystems? In bestimmten Fällen wird es notwendig sein, dass ein einzelnes Device mit anderen Parametern konfiguriert sein soll als alle anderen – ist dies möglich bzw. mit welchem Aufwand?

Für die Aufbereitung der gesammelten und aggregierten Daten sowie für die Verständigung mussten im Vorfeld unter anderem folgende Fragen beantwortet werden:

- Wer (welcher Benutzerkreis) benötigt welche Informationen?
- Für welche Personengruppe sind welche Informationen interessant bzw. notwendig?
- Wer darf auf welche Informationen Zugriff haben? – Auf Security-Informationen sollte nicht jeder Benutzer zugreifen dürfen.

Angestellte Überlegungen in diesem Bereich waren unter anderem:

- Sollen alle Netzwerk-relevanten Daten an den Netzwerk-Administrator weitergeleitet werden?
- Sollen die Security-relevanten an den Sicherheitsbeauftragten (Security-Manager) weitergeleitet werden??
- Wäre es sinnvoll, HW-Fehler gleich auch direkt zum Hardware-Hersteller bzw. –Lieferanten weiterzuleiten?
- Sollen Applikationsmeldungen an den jeweiligen Applikationsverantwortlichen weitergeleitet werden?
- Performance-Informationen darf der System-Administrator auslesen.

4.6.7.1 Geplante Erweiterungen in Folgeprojekten

- Integration der Klimaanlage des Serverraumes
- Integration der gesamten DMZ
- Zugriffsstatistiken der Webserver
- Statistiken von DNS und DHCP
- Blackbox-Monitoring von DNS und DHCP
- Überprüfung der Websites via HTTP-Requests, Responsetimes
- Überwachung der Zeitdifferenzen der einzelnen Systeme / Server
- Integration der einzelnen Datenbank-Managementsysteme (DBMS)
- Integration der Forward- und Backward-Proxies (Statistiken,...)
- Integration der Fileservices (Quotas, Statistiken, Zugriff-Tests,...)
- Active Directory-Authentifizierung
- Ressourcen-Monitoring des SAN-Storages
- Überprüfung des Mailing-Systems mit Blackbox-Tests
- Überwachung der Virenscanner (Statistiken, Versionen,...)
- Auswertung und automatische Kontrolle diverser Logfiles
- Integration der proprietären Applikationen (Blackbox- als auch Whitebox-Tests)
- Eventuell Migration zu einem verteilten Monitoringsystem - Skalierung
- Erweiterung der Verständigungstechnologien (z.B.: auf SMS per GSM)
- Eventuell teilweiser Aufbau redundanter Netzwerkwege fürs Monitoring (Stichwort: outband-Monitoring – siehe Punkt 1.4.3.3)
- Detaillierte Auswertung und Überwachung von Ressourcen und Einzelapplikationen
- Erweiterte Verständigungsregeln an fein definierte Personenkreise
- Vermeidung bzw. Ausfilterung von Fehlerfortpflanzungen z.B. nur eine Alarmierung bei Ausfall eines Netzwerkknotens anstatt aller dadurch nicht erreichbaren (aber dennoch funktionierenden) Systeme

4.6.8 Zusammenfassung der Anforderungen und Funktionalitäten des Monitoring-Systems, Entscheidungen

Nach der Anforderungsanalyse lassen sich die unbedingten Anforderungen als auch die Funktionalitätswünsche zusammenfassen.

Forderungen:

- Integration aller definierten Systeme
- Skalierbarkeit (das System soll noch weitere Systeme überwachen können)
- Einsatz von erprobten Standards in den Bereichen Monitoring und (Netzwerk-)Management
- Web-Oberfläche
- Empfang und Verarbeitung von SNMP-Traps
- Zentrales Event- und Logmanagement
- Automatische Verständigung per E-Mail auf Grund definierbarer Regeln
- Definition von Maintenance-Windows
- Grundlegende Reporting-Funktionalitäten
- Performance-/Ressource-Monitoring
- Verfügbarkeits-Monitoring
- Übersichtliche Benutzerverwaltung inkl. Gruppenmanagement
- Sinnvolles Berechtigungssystem
- Customizable Dashboard
- Möglichkeit der Funktionserweiterung
- Verteiltes Monitoring
- Manageability
- Übersichtlichkeit der Funktionen und Informationen
- Integrierbarkeit in bestehende (geforderte) Systeme
- Möglichkeit eines professionellen Supports
- Firewall-Tauglichkeit
- Grundlegende SLM-Funktionen
- Schnittstellen zu definierten Systemen

- Zukunftssicheres System – Produkt darf nicht in naher Zukunft End-of-Live sein ➔ Updates
- Funktionen gehen konform mit aktuellen Standards für Prozesse, sprich ITIL

Wünsche:

- Erweiterbarkeit für neue Systeme
- Möglichkeit für zusätzliche Verständigungsmöglichkeiten
- Export von Daten für Import in andere Systemen (z.B.: Reportingsysteme)
- Exportfunktion für Reports
- Dynamische Reportfunktionalitäten
- Komplettlösung
- Open Source-Produkt(e)
- Automatische Aktionen auf Grund von Events
- Etablierte und erprobte Lösung
- Professionelle SLM-Funktionen
- Einsatz moderner / aktueller Technologien
- Ansprechende Grafiken und Statistiken
- Übersichtliche Funktionen
- Intuitive Bedienoberfläche
- Übersichtliche und erschöpfende Produktdokumentation

4.7 Produktevaluierung

Für eine optimale Auswahl einer Monitoring-Lösung werden auf Grund der Ergebnisse der Ist-Analyse und der Anforderungsanalyse verschiedene am Markt befindliche Produkte und Systeme näher betrachtet. Obwohl das Ziel der Realisierung eine Open Source-basierende Lösung sein soll, werden auch kommerzielle Produkte der vier großen Anbieter („The Big 4“) einer kurzen Betrachtung unterzogen.

4.7.1 Kommerzielle Systeme

Die kommerziellen Systeme sind eigentlich Frameworks welche aus verschiedenen Einzelprodukten bestehen und je nach Bedarf der Funktionen (siehe auch Tabelle 3) implementiert (und lizenziert) werden können. Die Lizenzierung erfolgt meistens auf Grund der Anzahl der zu überwachenden Server / Systeme und der eingesetzten Produkte bzw. Module.

Einige kommerzielle Produkte stellen Funktionen zur Verfügung, die es dem Monitoring-System erlaubt, selbstständig vordefinierte Aktionen auf Grund eines bestimmten Events zu setzen. Der Administrator wird dann per E-Mail (oder eine andere Verständigungsart) über die Durchführung informiert und die Ereignisse inklusive der Aktivitäten werden protokolliert. Ein Beispiel dafür könnte eine bestimmte Applikation sein, deren Server-Prozess in regelmäßigen Abständen immer wieder „verstirbt“ – die durch den Administrator zu setzende Maßnahme wäre das Service neu zu starten. Diesen Restart (Recovery Action) übernimmt das System selbstständig.

4.7.1.1 CA Unicenter TNG

CA Unicenter TNG[32] ist eine Enterprise-Lösung von Computer Associates (CA) im Bereich des IT Managements. Unicenter besteht aus verschiedenen Produkten welche je nach Anforderung angeschafft und implementiert werden können.

[32] http://www3.ca.com/solutions/Solution.aspx?ID=315

4.7.1.2 HP OpenView

HP OpenView[33] stellt eine Produktfamilie von HP für eine komplette System-Management-Lösung dar. Je nach Anforderung können einzelne Produkte gekauft und integriert werden.

4.7.1.3 IBM Tivoli

IBM Tivoli[34] ist der Oberbegriff von Softwareprodukten um Informationssysteme zu verwalten und durch ITIL beschriebene Prozesse zu unterstützen. Mit Tivoli ist es zum einen möglich, Rechner zu überwachen, Software zu verteilen, Systeme zu inventarisieren oder Datensicherung durchzuführen. Zum anderen werden Prozesse laut ITIL wie Release-, Change- und Storage Management mit Applikationen unterlegt.

4.7.1.4 BMC Performance Manager

BMC Performance Manager[35] (früher BMC Patrol) besteht aus drei verschiedenen Produkten (BMC Infrastructure Management, BMC Application Management, and BMC Database Management) und ermöglicht das proaktive Management von Verfügbarkeiten, Performance und den „Business Impact“ verteilter Systeme. Unterstützt werden z.B.: Netzwerk, eine breite Palette von Applikationen, Datenbanken und Betriebssystemen.

[33] http://www.openview.hp.com/
[34] http://www-306.ibm.com/software/tivoli/ und http://de.wikipedia.org/wiki/Tivoli_(IBM)
[35] http://www.bmc.com/

	HP OpenView	IBM Tivoli	CA Unicenter TNG	BMC Performance Manager
Status via Webseite, Dashboard	X	-	X	Client
WAP-Support	kA	kA	kA	kA
Client-Server Architecture	kA	kA	X	X
Redundantes System / Skalierbarkeit	X	X	kA	kA
Support von SNMP Traps	X	X	kA	X
Plattform-support Server	Windows	Windows, Linux, AIX, HP,...	Windows	Windows
Plattform-support Client	Windows, Linux, HP, Sun, OpenVMS, AIX, ...	Windows, AIX, HP, Linux, Sun,...	Windows, Redhat, VMware,...	Windows Desktops + Server, UNIX, Linux, iSeries, OpenVMS
Service-Tests	X	X	X	X
Performance-Monitoring	X	X	X	X
Notification	X	X	X	X
History	X	X	X	X
Reporting	X	X	X	X
Plug-ins	SPIs[36]	kA	kA	Knowledge-Module
Support	kommerziell	kommerziell	kommerziell	X
Agentbased / agentless	beides	kA	Agent	Beides, hybrid
Recovery Actions	kA	kA	kA	X
Event Management	X	X	X	X
Application Level Monitoring	X	X	X	X
Integration in andere / anderer Systeme	in andere HP OpenView Produkte	X	X	X

Tabelle 3 - Funktionsvergleich kommerzielle Systeme ("Big Four")

Legende:
X ... verfügbar
kA ... keine Angabe, keine Informationen

[36] Smart Plugins (siehe auch: http://www.openview.hp.com/products/spi/prod_spi_0002.html)

4.7.2 Open Source-basierende Systeme

Open Source Software (OSS) ist schon länger allgegenwärtig – die bekanntesten Open Source-Produkte sind Apache, OpenOffice und verschiedene Linux-Derivate. Es wäre nicht seriös zu behaupten, dass Open Source Software besser wäre als ein kommerzielles Produkt oder umgekehrt. Es soll hier keine Bewertung von Open Source versus kommerzieller Lösungen abgegeben werden. Diskussionen welche der beiden Arten besser bzw. billiger/günstiger ist, wurden in den letzten Monaten immer wieder in Medien und Fachkreisen geführt. Aufgrund der Firmenpolitik ist ein quelloffenes System zu bevorzugen für den Fall, dass es die notwendigen Anforderungen erfüllen. Gegen den Einsatz eines quelloffenen Produktes ist laut neuesten Studien und Erfahrungsberichten nicht wirklich etwas einzuwänden – es kommt auf das Einsatzgebiet und die jeweiligen Anforderungen an. Unter [optaros, 2007] wurde ein „Open Source Catalogue“ für das Jahr 2007 mit über 260 Open Source Produkten veröffentlicht – ein Beweis dafür, dass Open Source Produkte sich in den Unternehmen etabliert haben. Laut einer EU-Studie[37] ist freie Software funktionell ebenbürtig. Über lange Sicht betrachtet soll freie Software ein Einsparungspotenzial von 36 Prozent bieten, wenngleich Investitionen in Mitarbeiterschulungen und Migrationsprozesse zunächst für Mehrkosten sorgen.

4.7.2.1 Cacti

Cacti[38] (aktuelle Version: 0.8.6j) ist ein komplett in PHP geschriebenes graphisches Frontend für RRDTool (siehe Punkt 1.4.3.2). Cacti ist unter der GNU GPL veröffentlicht und komplett frei verfügbar. Die zusätzlichen Daten für Graphiken u. dgl werden in einer MySQL-Datenbank gespeichert. Da Cacti nur ein Frontend zu RRD ist, können auch nur dessen numerische Werte graphisch aufbereitet und dargestellt werden. Cacti kann perfekt für die Visualisierung von Performance-Daten eingesetzt werden. Weitere notwendige Funktionen eines Monitoring-Systems wie das Event-Management u. dgl sind nicht verfügbar.

4.7.2.2 Nagios

Nagios[39] (aktuelle Version: 2.7) ist unter der GNU GPL v2 veröffentlicht und ein webbasierter Service- und System-Monitor um bei Problemen entsprechend den

[37] http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf
[38] http://cacti.net/
[39] http://nagios.org/

Administrator zu benachrichtigen. Nagios wurde zwar für Linux entwickelt, läuft aber auch auf anderen *nix-Systemen. Das Konzept von Nagios basiert auf externen Plugins welche von den Monitoring-Daemons angestoßen und abgefragt werden. Standardmäßig können von Nagios eine Vielzahl an Services wie HTTP, FTP, SMTP überprüft werden. Mit Plugins kann die Funktionalität von Nagios fast beliebig erweitert werden. Es stehen verschiedene Notifikationsmöglichkeiten wie SMS, E-Mail, Pager zur Verfügung. Im Laufe der Zeit wurden einige Extensions hinzugefügt, so z.B. eine Statusmap, 3D-Status.

Nagios ist per Definition nicht dafür designed um eine vollfunktionale SNMP-Management-Lösung zu ersetzen. Es können allerdings mit Nagios SNMP-Traps empfangen und Alerts generiert werden. Dazu wird NET-SNMP[40] (früher UCD-SNMP) eingesetzt. Die Lösung ist nicht sehr komfortabel und daher nur in kleinen Lösungen sinnvoll.

4.7.2.3 Big Brother System and Network Monitor

„Big Brother System and Network Monitor“[41] ist in vielen Fällen lizenzfrei, dazu existiert auch eine professionelle Version. Die aktuellen Lizenzbedingungen besagen, dass eine „commercial license“ zu erwerben ist, sobald Big Brother dazu verwendet wird um finanziellen Nutzen daraus zu ziehen (Commercial purposes include any activity engaged in for the purpose of directly generating revenue or in support of activity that generates revenue.). „Educational, Government and Non-profit“ müssen keine kommerzielle Lizenz erwerben, außer sie überwachen damit eCommerce-Sites oder andere gewinnbringende Seiten. Auf Grund dieser Lizenzbedingungen könnte Big Brother in einigen Fällen als kommerzielle Lösung eingeordnet werden und kommt daher als Produkt nicht in Frage.

4.7.2.4 GroundWork Monitor Open Source

„GroundWork Monitor Open Source“[42] (GW-OS) (aktuelle Version: 5.0) ist eine komplette Monitoring-Lösung einer IT Infrastruktur. Groundwork ist ein Framework erprobter und etablierter Open Source-Tools: Nagios, Nmap, SNMP TT, PHP, Apache, MySQL, Groundwork Guava, etc auf PHP/AJAX[43]-basierter Oberfläche. GW-OS ist laut Definition für

[40] http://net-snmp.sourceforge.net/
[41] http://www.bb4.org
[42] http://www.groundworkopensource.com/
[43] Siehe Abkürzungsverzeichnis

mittelgroße Unternehmen geeignet. GW-OS ist im Grunde genommen ein Überbau zu Nagios inklusive zusätzlicher Funktionen (siehe auch Abbildung 4-5). Groundwork Monitor Open Source ist frei verfügbar – zusätzlich gibt es „Groundwork Monitor Business“ und „Groundwork Monitor Professional“ mit zusätzlichen Funktionen und Support.

Function	Open Source	Small Business	Professional
Availability Monitoring	●	●	●
Monitor servers, devices and applications	●	●	●
Web-based configuration	●	●	●
Alarms, notifications, escalations	●	●	●
Documentation	●	●	●
Integrated performance monitoring	●	●	●
Profiles encapsulate monitoring "best practices"		●	●
Integrated multiple monitoring data (traps, logs)		●	●
Reporting and exceptions analysis		●	●
Integrate Event Console views		●	●
Role-based dashboards		●	●
SLA Reports w/custom report creation		●	●
Support with regular maintenance/updates		●	●
Monitor more than 50 devices	●		●
Deployment options and services			●

Abbildung 4-4 - Versionsvergleich Groundwork Monitor
(Quelle: http://www.groundworkopensource.com/products/comparison.html)

Wie in Abbildung 4-4 und Abbildung 4-5 ersichtlich, sind in der Open Source-Version kein Reporting sowie keine mehrfachen Datenquellen (außer Nagios) möglich. Zusätzlich fehlen noch das notwendige Performance-Monitoring und der Empfang/Verarbeitung von SNMP-Traps.

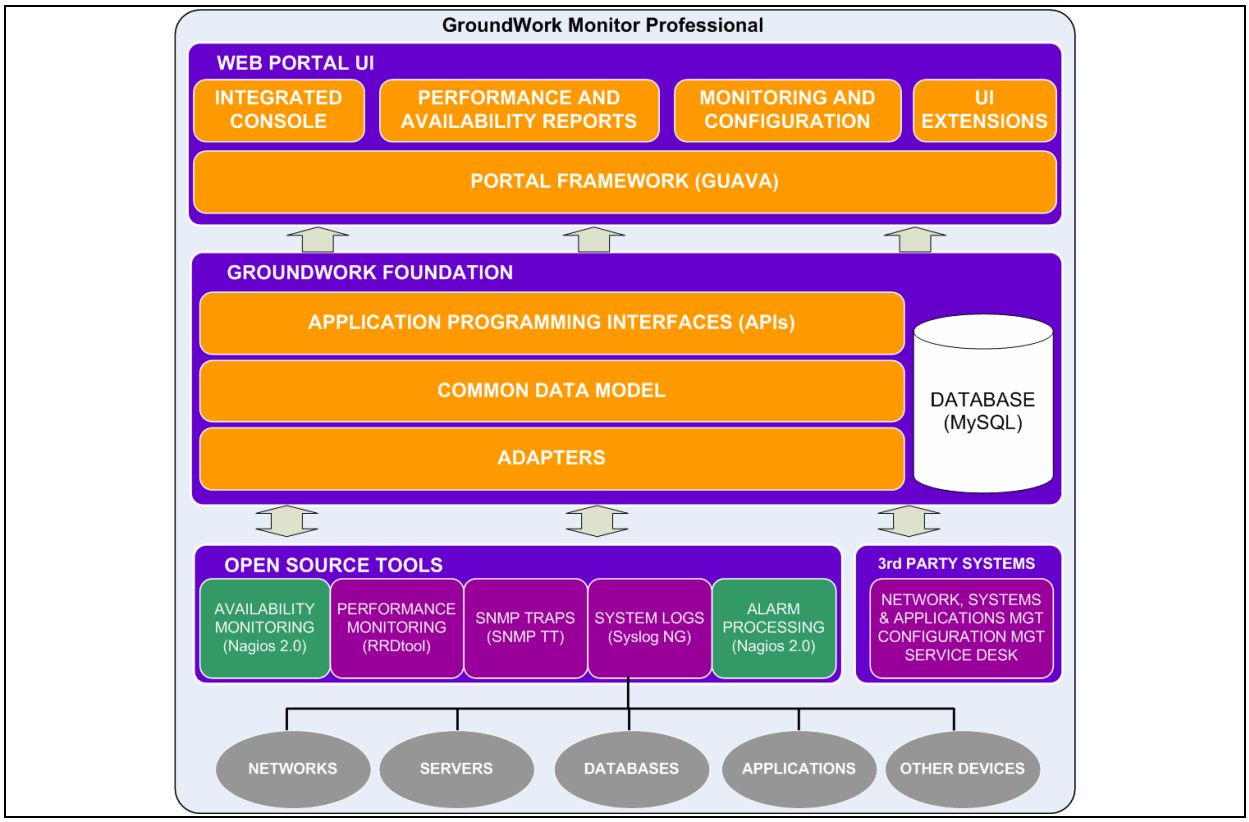


Abbildung 4-5 - GroundWork Extensible Architecture
(Quelle: [groundworkopensource.com, 2006])

4.7.2.5 OpenNMS

OpenNMS[44] (aktuelle Version: 1.2.9) ist laut eigener Webseite die erste unternehmensweite Network-Management-Plattform unter der GNU GPL v2 und besteht aus einem Community-supported Open Source-Projekt, als auch aus kommerziellen Services, Trainings und Support. OpenNMS ist eine in Java geschriebene, verteilte, skalierbare Plattform zur Abdeckung aller Aspekte von FCAPS[45] Network Management. Es fokussiert auf drei Eckpunkte: Service Polling, Data Collection, Event Management. Die Daten werden in einer PostgreSQL-DB gespeichert. OpenNMS kann SNMP-Traps empfangen und verarbeiten.

[44] http://www.opennms.org/
[45] FCAPS ist ein Modell der ISO für Netzwerkmanagement. FCAPS definiert die Eckpunkte des heutigen Netzmanagements. FCAPS ist die Abkürzung für die unterschiedlichen Aufgaben in die Netzmanagement aufgeteilt wird: Faults, Configuration, Accounting, Performance, Security.

4.7.2.6 Zenoss

Zenoss Core[46] (aktuelle Version: 1.1) bietet eine umfassende Sammlung an Funktionalitäten für ein komplettes Monitoring-System unter der GNU GPL. Wie in Abbildung 4-6 skizziert, besteht Zenoss aus folgenden Grundkomponenten: Inventory & Configuration, Availability Monitoring, Performance Monitoring sowie ein durchdachtes Event Management, als auch Alerts (E-Mail, Pager) und Reporting Funktionalitäten etc.

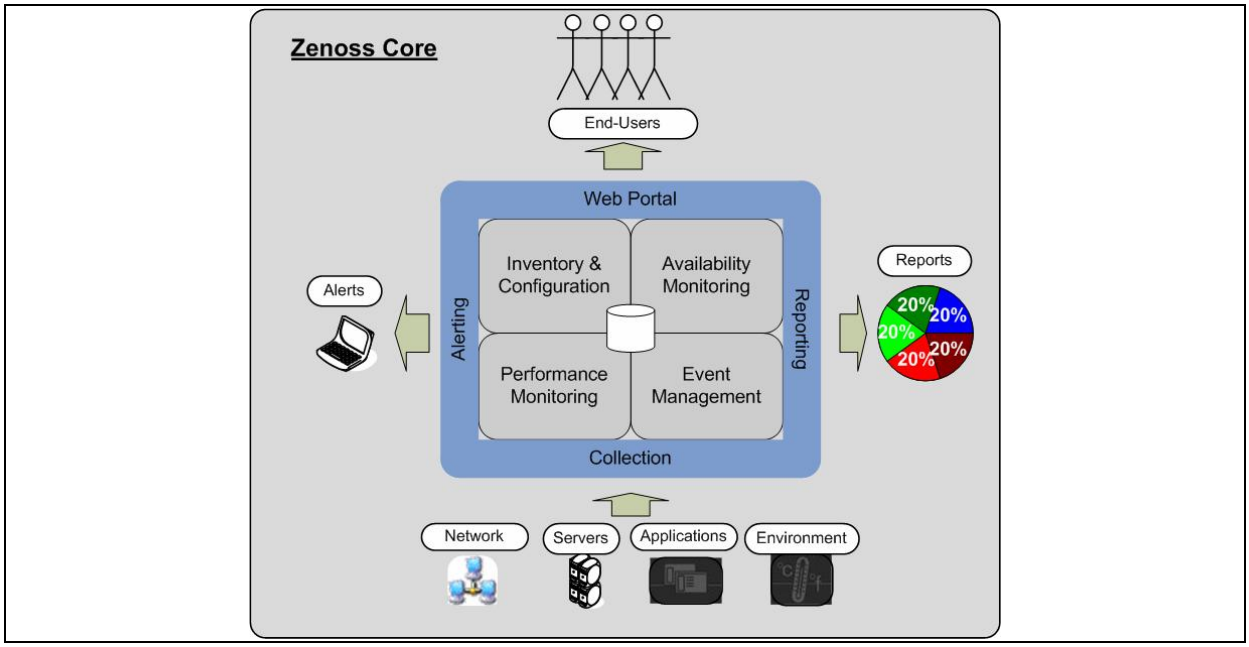


Abbildung 4-6 - Zenoss Core 1.1 Features
(Quelle: [zenoss.org, 2007])

Die Konfiguration als auch das zu customizende Dashboard sind AJAX-basiert und via Webbrowser (Mozilla Firefox[47] bevorzugt) verfügbar. Die Inventur- und Konfigurationsdaten werden in einer Zope Enterprise Object (ZEO)[48] database gespeichert. Für die Event-Persistierung wird MySQL eingesetzt. Zenoss Core sowie die diversen Skripts sind in Python[49] programmiert. Für eine automatisierte Konfiguration und Verarbeitung können die Devices und Daten in Zenoss via REST[50] manipuliert werden.

---

[46] http://www.zenoss.org
[47] http://www.mozilla.com/en-US/firefox/
[48] http://www.zope.org
[49] „Python is an *interpreted*, *interactive*, *object-oriented* programming language. It is often compared to Tcl, Perl, Scheme or Java.“, http://www.python.org/doc/Summary.html
[50] (REST) Representational State Transfer bezeichnet einen Softwarearchitekturstil für verteilte Hypermedia-Informationssysteme wie das World Wide Web von Roy Fielding aus dem Jahr 2000.

4.7.2.7 MRTG

MRTG[51] („The Multi Router Traffic Grapher“) (aktuelle Version: 2.15.1) ist vom Entwickler von RRDTool. MRTG ist in Perl (bis auf wenige Ausnahmen in C) geschrieben und unter der GNU GPL veröffentlicht. MRTG läuft unter Linux/Unix, Windows und Netware. Mit MRTG können via SNMP alle möglichen Arten von Network Devices gemonitored werden. Da die gesammelten Daten in RRDs gespeichert werden, können wie bei Cacti nur numerische Daten verarbeitet werden. MRTG kreiert aus den gespeicherten Daten HTML-Seiten mit PNG[52]-Grafiken der gesammelten Werte. Event Management und auch die Verarbeitung von SNMP-Traps können nicht implementiert werden.

4.7.2.8 Zabbix

Zabbix[53] (aktuelle Version: 1.1.5) ist eine all-in-one Monitoring-Lösung, welche unter der GNU GPL veröffentlicht und unter Ubuntu Linux und RedHat AS2 Linux getestet wurde. Laut Webseite sind unter anderem folgende Funktionen verfügbar: Performance Monitoring, Availability Monitoring, Notification Rules, Alerts (E-Mail, SMS), Logging, Mapping und Graphing, Real-time SLA reporting. Es existieren diverse High performance-Agents für alle supporteten Plattformen. Der nächste Schritt der Funktionalitäten wird erst mit Version 1.4, die für Jänner 2007 geplant war, gegeben sein – dafür sind grundsätzliche Verbesserungen und Änderungen angekündigt.

4.7.2.9 Big Sister Network Monitor

„Big Sister Network Monitor“[54] (aktuelle Version: 1.02) ist unter der GNU GPL veröffentlicht und als Ersatz und Verbesserung von Big Brother geschrieben um dessen Einschränkungen zu umgehen. Big Brother war damals in shell-Skripts geschrieben – Big Sister wurde in Perl entwickelt und sollte mit den bei Big Brother fehlenden Funktionen ausgestattet werden. Die Konfiguration erfolgt hauptsächlich über Konfig-Files.

[51] http://oss.oetiker.ch/mrtg/
[52] http://www.libpng.org/pub/png/
[53] http://www.zabbix.org/
[54] http://bigsister.graeff.com/

4.7.3 Fazit – Produktentscheidung

Die **Produktentscheidung** fiel auf **„Zenoss Core“**.

„Zenoss Core“ ist ein quelloffenes System am aktuellen Stand der Technik und wird in naher Zukunft um weitere Funktionen erweitert werden. Die nächste Major-Release (Version 2.0) wurde bereits im Juni 2007 veröffentlicht. Jedes betrachtete System (kommerziell bzw. quelloffen) hat seine Berechtigung sowie seine Stärken (als auch seine Schwächen). Cacti, Nagios und MRTG fallen auf Grund der festgelegten Anforderungen des Unternehmens (siehe 4.6.8) aus der weiteren Betrachtung – vor allem die Möglichkeit des Empfangs und der Verarbeitung von SNMP-Traps kann bei diesen Produkten nicht umgesetzt werden. „Zenoss Core“ bietet im Vergleich zu den kommerziellen Produkten eine Komplettlösung und kein Framework, welches sich bei spezielleren Anforderungen als komplexes Konstrukt darstellt. Es sind bereits alle notwendigen Funktionen für ein Monitoring-System auf Open Source-Basis verfügbar und aufeinander abgestimmt. Im Vergleich zu den kommerziellen Produkten, wo des öfteren die hohen Lizenzkosten abschrecken, entstehen beim „Zenoss Core“ keine Softwarekosten. Stellt man die Funktionen und Features der betrachteten Systeme gegenüber, so zeigen sich Überschneidungen zwischen den kommerziellen Produkten, als auch zwischen kommerziellen Lösungen und frei verfügbaren Produkten. Die Open Source-Angebote sind großteils als Komplettlösung erhältlich und bieten dabei bereits Funktionalitäten, welche bei den kommerziellen Produkten erst durch Einsatz weiterer Produkte bzw. Module gegeben sind.

„OpenNMS“ hätte laut Funktionsbeschreibungen auch einen Großteil der definierten Anforderungen erfüllt. Die verfügbare Version stammte allerdings aus dem Jahre 2006. Auf der Projekt-Webseite war bereits für Jänner 2007 eine neue Version mit zusätzlichen Funktionen angekündigt, welche dann allerdings noch nicht verfügbar war. Es waren zum Update keine weiteren Informationen verfügbar. Mittlerweile steht eine Development-Version von 1.3.3 zum Download bereit.

„Zenoss Core“ soll ab der Version 2.0 eine CMDB (Configuration Management Database) - „First Commercial Open Source CMDB - a single repository for your IT assets“ - beinhalten. Die Weboberfläche ist einfach und intuitiv zu bedienen und wird laufend verbessert. Laut

Angaben auf der Projekt-Webseite skaliert die Gesamtlösung bis zu mehrere tausend Systeme und ist somit für den Einsatz in Unternehmen geeignet. Support wird durch die Community als auch durch eine „Zenoss Enterprise Subscription[55]“ zur Verfügung gestellt. „Zenoss Core“ bot den positivsten Gesamteindruck vor allem da die Lösung auf aktuellen Technologien basiert und bei eventuellen Problemen durch die Community in relativ kurzer Zeit eine Problemlösung bzw. ein Workaround zur Verfügung stand. Für neue Ideen und Anregungen stehen die Ansprechpartner grundsätzlich offen und positiv gegenüber.

4.8 Produkttest – Konfiguration – Customization

Für die Tests und Konfigurationsanpassung wird ein Monitoring-System auf einem dezidierten Server aufgesetzt. Für Zenoss wird ein fertiges VMware-Image auf Basis von rPath-Linux[56] zum Download angeboten, wodurch die Installation und Basiskonfiguration großteils entfallen und sogleich mit den ersten Tests begonnen werden kann. Für die Produktivumgebung ist dieses System allerdings nicht unter seriösen Gesichtspunkten einsetzbar.

Bei der Durchführung der Basistests der einzelnen Funktionen und Features wird bereits auf eine zukünftige Produktionskonfiguration hingearbeitet sowie der Fokus auf eine Service-Orientierung und die Kategorisierung der Systeme aus den Punkten 4.5.3 und 4.5.4 gelegt. Im Zuge der Tests werden die einzelnen Konfigurationen schrittweise verfeinert und überarbeitet. Es werden unter anderem die optimalen Intervalle für Polling erarbeitet, die verschiedenen Zeitfenster definiert und verändert, die Verständigung bei Alerts und die einzelnen Devices Schritt für Schritt eingebunden. Um die Kommunikation zwischen Monitoring-System und den Devices zu testen werden unter kontrollierten Rahmenbedingungen verschiedene Fehlersituationen erzeugt und beobachtet ob und wie das Monitoring-System darauf reagiert und die definierten Regeln z.B. für Alerts greifen.

Das „Digital Dashboard“ – die Kommando- und Übersichtszentrale des Monitoring-Systems wird den Firmen-Policies z.B. in punkto Firmenlogo, Firmenfarben etc angepasst. Die

[55] http://www.zenoss.com/support
[56] http://www.rpath.com

folgenden Abbildungen zeigen Beispiel-Seiten der Original-Web-Oberfläche von Zenoss – es wurden dabei bereits Anpassungen für die Systeme vorgenommen und Devices eingetragen.

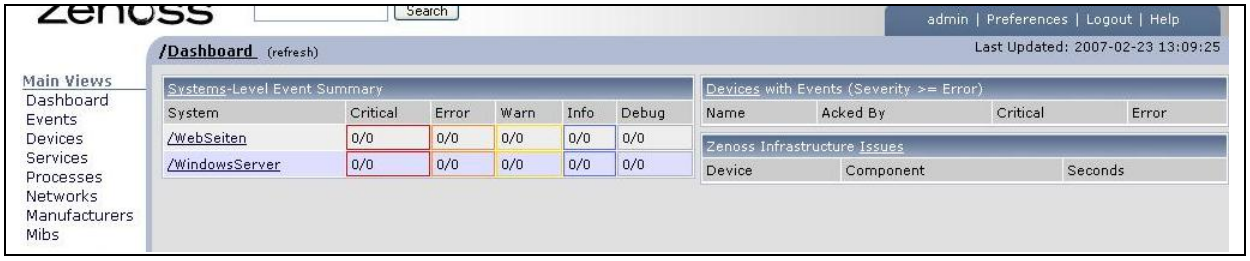


Abbildung 4-7 - Zenoss Dashboard

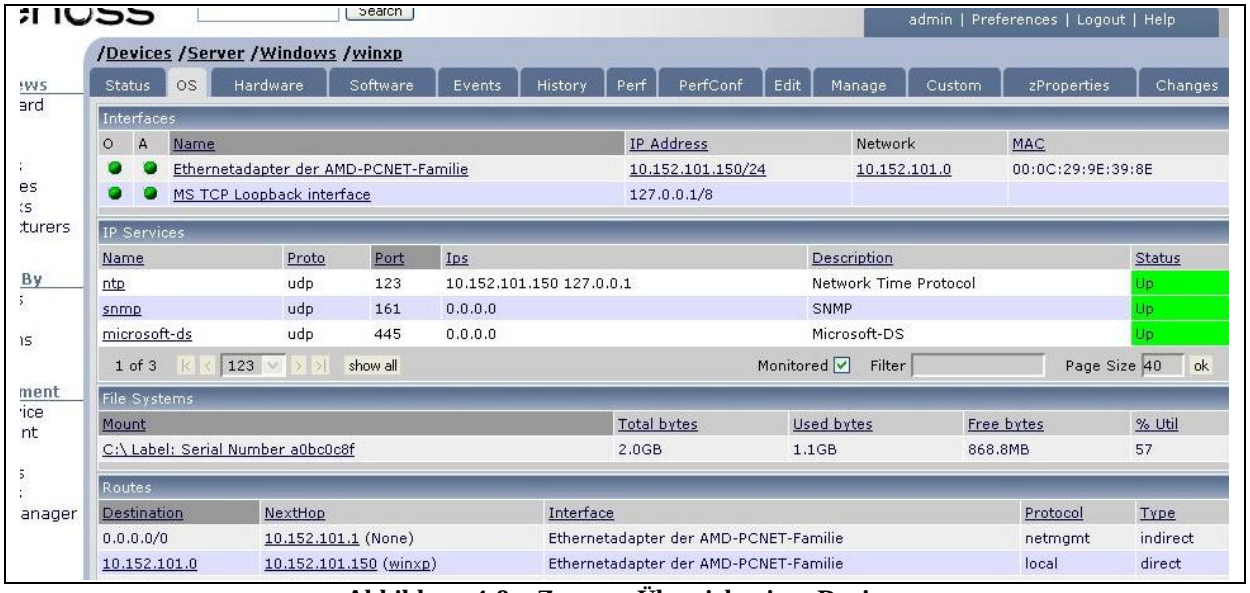


Abbildung 4-8 – Zenoss - Übersicht eines Devices

Abbildung 4-9 – Zenoss - Eventübersicht pro Device

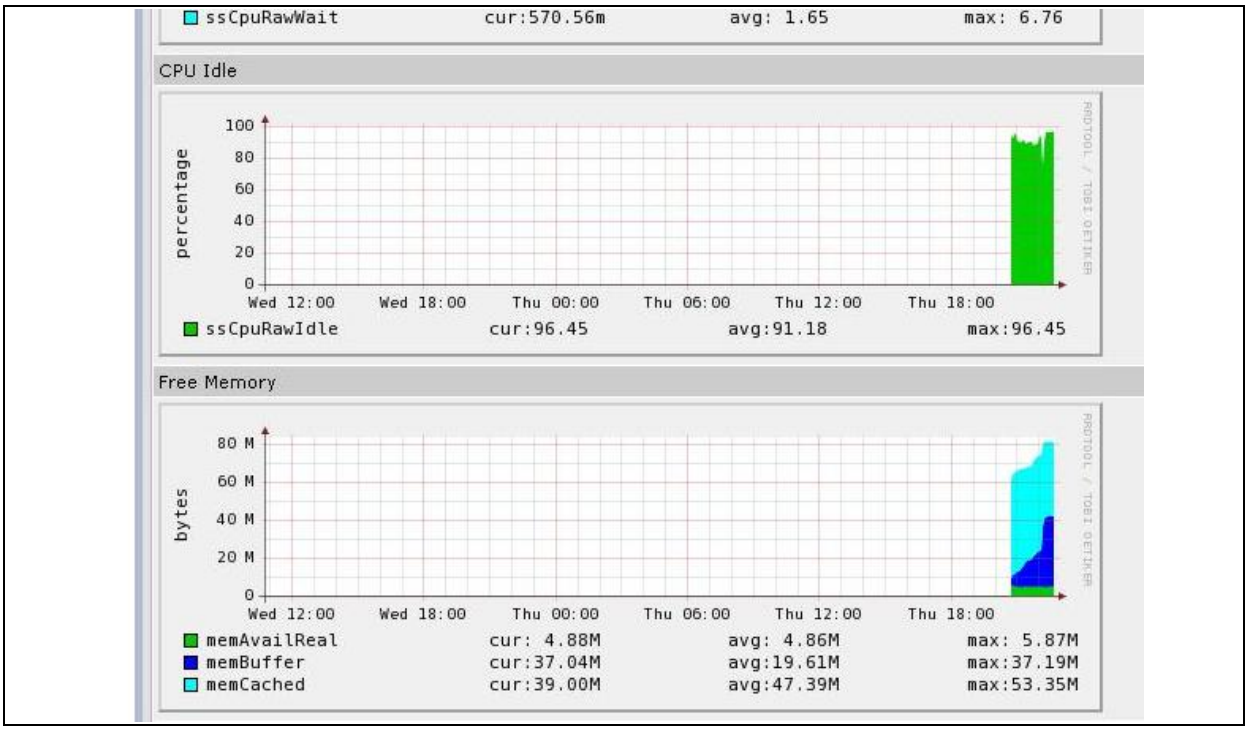


Abbildung 4-10 – Zenoss - Performance-Graphen

4.8.1 Konfiguration

4.8.1.1 Allgemeine Voraussetzungen / Prerequisites

- Gemeinsame Zeitquelle aller Systeme ➔ für Konsolidierung und Auswertungen der Logs und Events
- IP-Adresse und FQDN für den oder die Monitoring-Server
- Hardware und installiertes Betriebssystem für Monitoring-Server
- Installierter „Zenoss Core“ inkl. der dafür notwendigen Voraussetzungen
- Zugriff auf alle Systeme bzw. auf die berechtigen Administratoren

4.8.1.2 Überwachung Systemmeldungen

4.8.1.2.1 Windows

4.8.1.2.1.1 Eventlog

Unter Microsoft Windows (ab Windows NT 4.0) steht als Standard-Protokollierungs-Werkzeug der „Eventlog“ (Ereignisdienst) zur Verfügung. Die System-Dienste protokollieren

hier nach Kategorien und Identifikationsnummern (IDs) unterschiedlich ihre Meldungen. Standardmäßig gibt es einen Log für „Application“ (Applikation), „Security“ (Sicherheit) und „System“. Der Eventlog ist grob betrachtet mit dem Syslog unter Linux (siehe 1.4.2) zu vergleichen, allerdings werden die Logs immer lokal am System gespeichert. Ab einer gewissen Anzahl von Servern und Systemen ist es ein mühsames Unterfangen regelmäßig diese Logs zu kontrollieren und auf Unregelmäßigkeiten und Fehlereinträge zu analysieren und zu filtern (nicht jeder Eintrag ist ein Fehler und zieht eine notwendige Aktion mit sich). In dieser Aufgabe sind entsprechendes Know-How und Praxiserfahrung notwendig, damit die Vielzahl an Meldungen auf die wichtigen Einträge reduziert und entsprechend interpretiert werden kann – dies ist Aufgabe des zentralen Event- und Log-Managements. Es steht standardmäßig keine Funktion zur Verfügung, um Ereignisse an eine zentrale Stelle (Management-Station), womöglich auch noch nach diversen Kriterien gefiltert, weiterzuleiten.

Es standen verschiedene **Lösungsansätze** zur Debatte:

- Versand von SNMP-Traps für bestimmte (wenige) Eventlog-Einträge mit Hilfe von „evtcmd.exe“ (erst mit Windows Server 2003 verfügbar)
- Kommerzielle Eventlog-Monitoring-Tools, welche dann verschiedene Verständigungs- bzw. Weiterleitungsfunktionen anbieten (z.B.: „eventsentry[57]“)
- Eventlog-Monitoring via „zenwin“ von Zenoss Core
- Eventlog-to-Syslog-Tools

4.8.1.2.1.2 SNMP-Traps mit Hilfe von „evtcmd.exe“

Ab Windows Server 2003 steht das Tool „evtcmd.exe“ zur Verfügung. Der Administrator kann damit einen Listener auf den Eventlog registrieren und konfigurieren. Aufgrund von definierten Event-IDs werden dann vom System SNMP-Traps verschickt. Der Pferdefuss dabei ist, dass nur auf bestimmte/konfigurierte Einträge reagiert wird – ein dementsprechend großer Aufwand, um alle wichtigen Einträge zu filtern und konfigurieren. Wie werden neue Event-Einträge behandelt? Als weiteres Manko kann angemerkt werden, dass keine eigene MIB für diese SNMP-Traps existiert, was wiederum die Weiterbehandlung durch die

[57] http://www.eventsentry.com/

Management-Station erschwert und auch dort entsprechenden Konfigurationsaufwand bedeutet.

4.8.1.2.1.3 Kommerzielle Eventlog-Monitoring-Tools

Am Markt existieren natürlich auch fertige (in den meisten Fällen kommerzielle) Produkte, welche Windows-Systeme in den verschiedenen Ausprägungen überwachen – darunter auch den Eventlog.

Ein Beispiel[58] wäre „EventSentry“: EventSentry bietet eine Vielzahl an Monitoring- und Auswertungsmöglichkeiten. Allerdings ist die Software kostenpflichtig. Die frei verfügbare Freeware-Version bietet leider nur die Verständigung via einen SMTP-Eintrag. Syslog- und SNMP-Unterstützung sind in der Vollversion vorhanden.

4.8.1.2.1.4 „Zenwin“

Für „Zenoss Core“ gibt es speziell für die Integration von Windows-Systemen „zenwin[59]“, welches auf einem eigenen (zusätzlichen) Windows-Server installiert werden muss wobei in zukünftigen Versionen eine Linux-Portierung angeboten werden wird. Mit Hilfe der WMI[60]-Schnittstelle werden von dem zentralen Server die anderen Windows-Systeme überwacht – auch die Eventlogs. Kritikpunkte an dieser Lösung sind unter anderem der zusätzliche dezidierte Windows-Server (oder auch mehrere) und der dadurch integrierte Single Point of Failure (SPoF) – fällt dieser Server aus, so werden die Windows-Systeme für diese Zeitspanne nicht weiter überwacht.

4.8.1.2.1.5 Eventlog-to-Syslog

Auf dem gleichen Prinzip wie „evtcmd“ funktionieren auch die folgenden Lösungen – es wird eine Konvertierung von Eventlog-Einträgen auf ein anderes Message-Protokoll (in diesem Fall syslog – siehe 1.4.2) durchgeführt.

Das Open Source-Projekt „ntsyslog[61]“ (aktuelle Version: 1.13) bietet eine einfache Möglichkeit, die verschiedenen Eventlogs auf einem Server zu überwachen. Die

[58] Weitere Beispiele können der Literatur entnommen bzw. über diverse Suchmaschinen gefunden werden.
[59] http://www.zenoss.com/download
[60] Windows Management Interface
[61] http://sourceforge.net/projects/ntsyslog/

verschiedenen Event-Einträge können entsprechend den standardisierten Syslog-Priorisierungen eingestuft werden. Ein detailliertes Ausfiltern außer nach Event-Type (Information, Warning, Error,...) ist nicht möglich. Es können keine eigenen Eventlogs (z.B. durch installierte Applikationen angelegt) direkt überwacht werden.

Auf „ntsyslog“ basierend existiert „SyslogAgent[62]“ (aktuelle Version: 3.3.5 free version). Es bietet zusätzlich zu den vorgenannten Funktionen die Möglichkeit nach verschiedenen Event-IDs zu filtern, außerdem können verschiedene Text-basierte Logfiles überwacht und ausgewertet werden. Im Vergleich zu „ntsyslog“ sind diverse Optimierungen eingearbeitet worden.

Als einfache aber immer noch recht kostengünstige Lösung wurden Eigenentwicklung diskutiert. Ein einfacher Listener lässt sich mit relativ wenig Programmier-Aufwand (abhängig von den entsprechenden Programmier-Kenntnissen) selbstständig erstellen und an die spezifischen Anforderungen anpassen. Im Anhang sind zwei Code-Beispiele für solche Listener aufgelistet:

Das erste Programm wurde in Perl[63] geschrieben. Die Einträge werden via Syslog verschickt und zusätzlich an der Konsole ausgegeben. Das Code-Beispiel ist in dieser Form nicht für einen breiten Einsatz auf Servern geeignet, da es als interaktive Applikation und nicht als Dienst läuft und außerdem fehlen Konfigurationsmöglichkeiten.

Das zweite Code-Beispiel wurde in der Windows-eigenen Script-Sprache „Visual Basic Script“ (VBS) realisiert. Es ist nicht so funktionsreich wie die Perl-Version und soll die Möglichkeiten aufzeigen. Auch dieses Beispiel ist nicht als Produktions-Variante einzusetzen – es könnte allerdings für ein kurzzeitiges Remote-Monitoring bei der Problem-Suche eingesetzt werden. Für nähere Informationen zu VBS bzw. WMI oder auch Eventlog siehe unter anderem unter [microsoft.com, 2006].

[62] http://www.syslogserver.com/syslogagent.html
[63] http://www.perl.com

Alle vier andiskutierten Lösungsvarianten sind auf jedem einzelnen System zu installieren – die Perl- und VBS-Varianten könnten ähnlich wie „zenwin“ als Remote-Lösung installiert werden, hätten dadurch aber die gleichen Nachteile wie zenwin.

Die Entscheidung fiel auf die Variante mit **„SyslogAgent“** – für den ersten Realisierungsschritt ist dies völlig ausreichend.

4.8.1.2.2 Linux, AIX, ESX-Server

In allen derzeit eingesetzten Linux-Systemen ist der Standard-Protokollierungs-Dienst Syslog (siehe Punkt 1.4.2) installiert. In der Datei „syslog.conf“ wird der Syslog-Daemon konfiguriert. Dabei wird aufgrund von Facility und Severity (siehe Anhang) definiert ob und wo das System Meldungen aufgrund von Events hinprotokollieren soll.

Das Virtualisierungsprodukt „ESX Server“ von VMware basiert auf einem RedHat-System (Version 7.2 (ESX 2.5.x) bzw. EL 3.0 (VMware Virtual Infrastructure 3)) und ist in den Grundfunktionen genauso zu überwachen.

AIX-Systeme basieren auf SystemV[64] und sind in einigen Bereichen ähnlich wie Linux-Systeme zu verwalten, laufen allerdings meist auf einer anderen Hardware-Architektur - deswegen werden in dieser Phase nur grundlegende Funktionen wie unter anderem die System-Meldungen überwacht und in die Gesamt-Monitoring-Lösung integriert. Für nähere Informationen und Überwachungsmöglichkeiten (vor allem im Hardware-Bereich) sei hier auf den jeweiligen Hersteller bzw. Produktdokumentation verwiesen – Aufgabe eines der Nachfolgeprojekte.

Der Syslog-Daemon des jeweiligen Systems wird so konfiguriert, dass alle wichtigen Messages zusätzlich via Syslog auf den Zenoss-Monitoring-Server geschickt werden (Konfiguration siehe auch Anhang B).

[64] http://de.wikipedia.org/wiki/System_V

4.8.1.3 Überwachung Hardware

4.8.1.3.1 Windows, Linux und ESX-Server

Für seine Server-Systeme stellt Hewlett Packard (HP) das Software-Paket „HP System Insight Manager[65]“ (HP-SIM) zur Verfügung. Anfangs war der HP-SIM rein für das Monitoring von HP-Server-Hardware gedacht und eingesetzt. Mittlerweile (aktuell in der Version 5.2 verfügbar) wurden die Funktionalitäten um viele Features und Funktionen für ein zentrales System-Management erweitert:

- Performance-Management (Performance Management Pack (PMP) – lizenzpflichtig)
- Management virtueller Maschinen (z.B.: von VMware ESX - lizenzpflichtig)
- Command-Console für HP-UX-Systeme
- Zentrale Verteilconsole für unterstützte Treiber

Der HP-SIM fällt in die Kategorie der agenten-basierenden Monitoring-Systeme (siehe Kapitel 3). Dazu werden auf den unterstützten Systemen[66] entsprechende Agents, welche zum Teil auf Systemkomponenten wie SNMP und SMTP aufsetzen, installiert und entsprechend konfiguriert. Bei richtiger Konfiguration schicken die Agents SNMP-Traps bei auftretenden Events an den HP-SIM-Server. Die Server-Komponente setzt eine konfigurierte Datenbank[67] voraus. Es werden die verschiedenen Systeme eingetragen und protokolliert. Beim Einsatz des PMP werden auch die Performance-Daten in dieser Datenbank abgespeichert. Die zu überwachenden Systeme können einerseits händisch eingetragen, oder automatisch via Netzwerk discovered werden (mit eventuellen Impacts auf die Netzwerkbelastung). Eine andere Möglichkeit besteht darin, dass die Systeme sich automatisch beim Server anmelden. Die Console für den Administrator ist Webbrowser-basierend und kann in ein Single-Sign-On-Konzept mit Authentifizierung gegenüber einer Microsoft Windows-Domain bzw. LDAP-Server eingebunden werden. Für verschiedene Betriebssysteme (u. a. RedHat Linux, Microsoft Windows, VMware ESX-Server) werden von Seiten Hewlett Packard eigene Agents für die Überwachung der Hardware zur Verfügung gestellt, die wiederum eine

[65] http://h18013.www1.hp.com/products/servers/management/hpsim/index.html?jumpid=go/hpsim
[66] Nähere Informationen siehe Homepage
[67] Unterstützte DBMS siehe Hersteller-Webseite

Integration in den HP-SIM bzw. andere SNMP-basierende Systeme ermöglichen. Für die Integration sind die MIBs (siehe auch 1.4.1.3) von HP und VMware notwendig.

Events können unter anderem:

- auf der Console angezeigt werden (Bearbeitung durch den Administrator)
- automatisch verarbeitet werden (z.B.: Weiterleitung via E-Mail oder SNMP, Start eines Skripts, Pager-Verständigung, Acknowledgement,...)

HP-Server verfügen über eine vom Betriebssystem unabhängige Remote-Console mit zahlreichen Features (Remote Insight Board (RIB) bzw. Integrated Lights-Out (iLO) und iLO2). Durch die Konfiguration des iLOs werden zusätzliche Informationen im Monitoring-System protokolliert (siehe auch Anhang B - 1 und Anhang B - 2) – z.B. Reset des Systems durch einen Administrator. Diese Hardware-Komponente fällt in die Kategorie des Outband-Managements, da das Board eine eigene Netzwerkverbindung aufweist und vom jeweiligen Betriebssystem unabhängig verfügbar ist.

4.8.1.3.2 AIX

Die Hardware wird derzeit nur insofern überwacht als Meldungen via Syslog mitprotokolliert werden.

4.8.1.4 Überwachung Performance / Ressourcen

4.8.1.4.1 Windows, Linux, ESX-Server

Werte über CPU-Auslastungen, Netzwerk-Interfaces, Memory und Storage werden via SNMP ausgewertet – dazu ist keine zusätzliche Installation einer Software-Komponente notwendig. Es sind unter anderem nur die Tresholds überlegt zu setzen, ansonsten kommt es zu ungewollten Fehlermeldungen.

Unter Windows wird für die Zukunft der Einsatz von „SNMP Informant Standard[68]“ überlegt um mehr Informationen via SNMP auslesen zu können – Windows bietet von Natur aus nur einen eingeschränkten Informationsbestand via SNMP.

[68] http://www.snmp-informant.com/

Für den ESX-Server werden die Daten standardmäßig vom „Virtual Center Server[69]“ (Management Server für die Virtuelle Infrastruktur von VMware) gesammelt und werden zu diesem Zeitpunkt nicht gesondert überwacht. Es wird allerdings angedacht, die Daten aus dem Virtual Center in das Monitoring System zu übernehmen um alle Informationen in einer zentralen Console im Zugriff zu haben.

Am ESX-Server laufen keine besonderen zusätzlichen Dienste, was eine Überwachung derzeit unnötig macht. Ein HTTP-Server läuft zwar, ist aber für den laufenden Betrieb nicht notwendig – auf den Systemen der anderen Betriebssystem werden die Standard-**Dienste** mit den mitgelieferten Plugins (z.B.: check_http und check_ftp) geprüft. Die Existenz von **Prozessen** kann mit „Zenoss Core“ via SNMP überwacht werden.

4.8.2 Reports erstellen und testen

Für verschiedene Statistiken (das obere Management möchte mindestens eine monatliche Verfügbarkeitsstatistik der gesamten Produktiv-Systeme) und Auswertungen werden für den Produktionseinsatz die ersten Reports erstellt und fertig konfiguriert, sodass sie dann in das Produktivsystem importiert werden können (siehe auch Abbildung 4-11). Der Funktionsumfang für Reports wird in folgenden Versionen von „Zenoss Core“ noch erweitert und flexibler gestaltet werden.

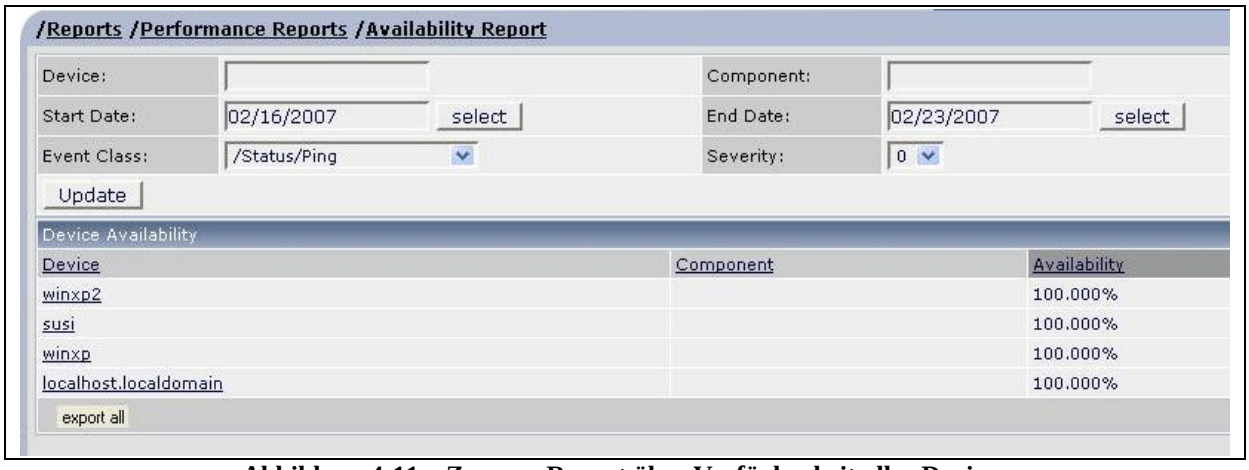


Abbildung 4-11 – Zenoss - Report über Verfügbarkeit aller Devices

[69] http://www.vmware.com/products/vi/vc/

4.8.3 Dokumentation erstellen

Nachdem die Funktionstests abgeschlossen sind, wird auf Grund dieser bestehenden und voll funktionsfähigen Konfiguration die Dokumentation erstellt, die ein essentieller Bestandteil für den Produktionseinsatz ist.

Für die verschiedenen Systeme werden Installationspakete geschnürt. Für die Linux-Systeme sind dies RPM-Packages[70]. Für die Windows-Systeme werden MSI[71]- bzw. Installer-Pakete oder einzelne einfachere Installationsroutinen erstellt. Für die Netzwerk-Devices werden in einfacher Form die durchzuführenden Konfigurationsschritte aufgelistet.

Es wird festgelegt wie und wann die zu überwachenden Systeme ins Monitoring-System kommen wobei es hier verschiedene Ansätze gibt, welche im Endeffekt in einer Mischform in Produktionseinsatz gehen werden. Netzwerk-Devices werden mit (Auto)Discovery eingetragen. Die Windows-Systeme werden durch das Installationspaket automatisch via REST[72] ins Monitoring-System eingetragen. Die Linux-Systeme sind in geringer Anzahl im Einsatz und werden daher händisch über den Web-Browser im Monitoring-System bekannt gemacht.

4.8.4 Schulung, Produkt- und Lösungsvorstellung

Der erste Realisierungsschritt ist hauptsächlich als Unterstützung für die Administratoren gedacht. Es bedarf für die Administratoren eine Einführung bzw. Schulung – wenn auch in geringerem Umfang als für reine Anwender.

Für das mittlere und obere Management sollen Präsentationsveranstaltungen abgehalten werden, um das neue System mit all ihren Vorteilen und Features vorstellen und „verkaufen“ zu können. Je positiver die Einstellung zum neuen System im gesamten Unternehmen ist, desto höher wird die Akzeptanz dessen sein. Wenn in einem nächsten Schritt das Management direkten (lesenden) Zugriff auf das System bzw. des Dashboards bekommt, sind weitere Schulungen sinnvoll und notwendig.

---

[70] Früher: Red Hat Package Manager, heute: RPM Package Manager
[71] Früher: Microsoft Installer, heute: Windows Installer
[72] Siehe Punkt 4.7.2.6

Es soll vor allem auch eine Sensibilisierung für die Grundgedanken von Monitoring stattfinden und die Vorteile eines sinnvoll eingesetzten Monitoring-Konzeptes und der Stellenwert der IT(-Abteilung) für das Unternehmen hervorgehoben und vermittelt werden.

4.8.5 Abschluss und Übergabe an Produktion

Nach Fertigstellung der Dokumentationen sind die Phasen der Analyse und Konzeption abgeschlossen.

Mit dem Abschluss dieser Arbeit ist die Projektdokumentation komplett und die Konzeptionsphase des Migrationsprojektes beendet. Nach Absegnung des Monitoring-Konzeptes durch die für strategische Entscheidungen verantwortliche Abteilung wird der Startschuss für die nächsten Projektphasen gegeben. Das fertige Konzept inklusive Software und Dokumentationen wird für die Produktion freigegeben.

4.9 Projektsummary

Die Anforderungen an das Monitoring-System im ersten Realisierungsschritt (siehe Punkt 4.6) wurden für die Testsysteme umgesetzt. Da die Erfahrungen im Unternehmen mit einem gesamtunternehmerischen Monitoring-System noch recht gering sind, muss vor allem bei der Umsetzung dieses Projektes auf Qualität und Zuverlässigkeit des Systems von Beginn an Wert gelegt werden. Es könnte ansonsten passieren, dass das System nicht allgemein angenommen wird. Es besteht auf alle Fälle noch Verbesserungspotential – vor allem die einzelnen Konfigurationen müssen noch feiner auf die spezifischen Systembedingungen abgestimmt und die Tresholds angepasst werden. Für die Folgeprojekte, welche unter anderem noch zusätzliche Systeme und Devices inkludieren werden, sind die Grundlagen gelegt. Da die Software ständig weiter entwickelt und erweitert wird, sollte es kein Problem sein alle Anforderungen zu erfüllen. Die größte Herausforderung ist anfangs die richtige Behandlung der verschiedenen Events im System und deren weitere Bearbeitung. Zu Beginn werden noch eine große Menge an „unbekannten“ Events aufscheinen, welche richtig eingeordnet und priorisiert werden müssen. Nachdem sich das System eingespielt hat, werden dann nur noch ganz neue unbekannte und die bekannten hoch-priorisierten Events im System aufscheinen. Bei der Integration der einzelnen Systeme im Produktionsbetrieb sollen natürlich nicht deren Verfügbarkeit und Performance beeinträchtigt werden.

4.10 Zusammenfassung der Produkte

- Zenoss Core Version inkl. Requirements:
  - Zenoss Zenoss 1.1.1
  - Zope Zope 2.8.8
  - Python Python 2.4.1
  - Database MySQL 5.0.24 (Ver 5.0.24)
  - RRD RRDtool 1.2.12
  - Twisted Twisted 2.4.0
  - SNMP PySNMP 3.4.3
  - Twisted SNMP TwistedSNMP 0.3.13
- SyslogAgent (Version 3.3.5, 11 January 2007) (bzw. Eigenentwicklung(en))
- Nagios-Plugins
- SNMP-Informant[73] (Verion 1.4) für besseren SNMP-Support unter Windows
- Zenwin (Version 1.1.1) angedacht für Service- bzw. Prozess-Monitoring unter Windows

5 Related Work

Wie auch in den Anfangskapiteln dargelegt, existiert eine Vielzahl an Produkten und Systemen am Markt, einerseits kommerzieller Natur aber auch frei verfügbar. Die kommerziellen Produkte sind nicht ohne weiteres und ohne kommerziellen Support der Herstellerfirma ins Unternehmen zu integrieren.

Whitepapers behandeln in den meisten Fällen weitgehend nur den Bereich des Netzwerk-Monitorings – die bekanntesten Open Source-Produkte im Bereich Monitoring resultieren daher auch aus dem Bereich Netzwerk und sind darauf abgestimmt (Stichwort MRTG). Die meisten Produkte sind auf ein bestimmtes Monitoring-Segment ausgerichtet, aber eine komplette Lösung ist meist nur von kommerziellen System-Integratoren (z.B. Groundwork) verfügbar.

[73] http://www.snmp-informant.com/

[Daxenbichler, 2004] beschäftigt sich mit „Online-Monitoring für die strategischen IT-Systeme einer Universitätsklinik am Beispiel der Tiroler Landeskrankenanstalten“. Für die komplexen IT-Systeme eines Krankenhauses wird zur Erhöhung der Systemverfügbarkeit und zur kontinuierlichen Überwachung das bestehende Überwachungssystem auf Basis eines Freeware-Produktes angepasst.

Auch ein gut umgesetztes und gewartetes Monitoring-System kann nicht alle Probleme lösen und eine Verfügbarkeit eines Systems garantieren – im Vorfeld müssen auch Strategien zu höherer Verfügbarkeit wie zum Beispiel im Bereich Backup, Hardwarelösungen, Redundanzen, Strom- und Klimaversorgung etc. umgesetzt werden.

6 Conclusio und Diskussion

Das Ziel des Migrationsprojektes war die Realisierung eines Monitoring-Systems für die heterogene Server-Infrastruktur, um bei Problemen **frühestmöglich** – am besten bevor der Kunde es bemerkt – auf Probleme reagieren zu können. Damit ein solches System auch effizient funktionieren kann, müssen sich die betroffenen Personen – vor allem die Administratoren – mit dem System **identifizieren** können. Dazu ist ein gewisser **Anpassungs- und Umstellungsprozess** notwendig der einerseits schon vor der Realisierung und andererseits bei der Einführung des Systems (während den Schulungen) angestoßen und in das Unternehmen integriert werden muss. Laut [Computerwelt 01/2007, Seite 9] hat sich in den obersten Geschäftsetagen bisher noch kein ausreichendes IT-Bewußtsein etabliert. Aufgrund dessen ist es notwendig das Management bei der Umsetzung des Monitoring-Konzeptes rechtzeitig einzubinden und von dessen Notwendigkeit und Nutzen zu überzeugen.

Das Ergebnis der Monitoringlösung ist mit Sicherheit noch kein perfektes – schon deswegen nicht, da bisher nur der erste Projektteil abgeschlossen wurde und durch den noch nicht erfolgten Produktionseinsatz kein Proof der Lösung durchgeführt werden konnte. Die weiteren Realisierungsschritte setzen auf die Ergebnisse der Umsetzung im Produktionsumfeld auf.

Wie auch schon anfangs angesprochen, kann ein Monitoring-System nie grundlegende Probleme im System lösen bzw. vermeiden. Wenn keine Dokumentation über die installierten Systeme und den verschiedenen Arbeiten und Tätigkeiten geführt wird, so kann dies auch nicht durch ein Monitoring-System ersetzt werden – ein gewisser Dokumentationsgrad muss in jeder IT-Abteilung erreicht werden. Eine Monitoring-Lösung schaut im Endeffekt von Unternehmen zu Unternehmen anders aus – die grundlegenden Funktionen sind aber immer die gleichen. In welcher Tiefe Monitoring betrieben wird hängt von den jeweiligen Anforderungen und Bedürfnissen ab. Es muss für die jeweiligen **Rahmenbedingungen** das optimale System bzw. Konfiguration konzeptioniert und umgesetzt werden. Ein zu grobes Monitoring liefert vielleicht nicht alle notwendigen Informationen und Daten (z.B.: beim Performance-Monitoring) – andererseits können bei einem übertriebenen Monitoring zwar jede Menge an Informationen gesammelt und gespeichert werden, auch wenn für die Anforderungen eventuell schon die Hälfte des Aufwandes ausreichen würde.
Ein Monitoring-System liefert die notwendigen Daten für die Überprüfung von Service Level Agreements. Diese Informationen müssen im erforderlichen Detaillierungsgrad ermittelt und persistiert werden. Das MS liefert jedoch nicht nur Daten für SLAs bzw. SLM, sondern kann in andere Management-Systeme integriert werden. Andererseits ist es unter bestimmten Rahmenbedingungen **sinnvoll andere Systeme zu integrieren** – z.B. bei proprietären Monitoring-Insellösungen deren Umstellung derzeit noch nicht möglich ist bzw. zu aufwändig wäre.

Aufgrund des Moves von einer technologisch-orientierten zu einer **service-orientierten** IT-Infrastruktur ist es notwendig über die eingesetzten Applikationen und Services Bescheid zu wissen und nicht mehr in „Systemen“, sondern **in „Services“ zu denken**. Dabei muss die Frage „was muss wo wie laufen damit der Dienst X verfügbar ist?“ beantwortet werden – für ein effizientes Monitoring dieses Dienstes muss die Antwort auch dementsprechend im Monitoring-System abgebildet werden. Bei Ausfall einer Komponente könnten entsprechende Reparaturmaßnahmen auf Grund von Erfahrungen automatisch durch das System in die Wege geleitet werden. In wieweit sind solche Maßnahmen möglich und sinnvoll? Bei bekannten Problemen die immer wieder auftreten und die durch relativ einfache Maßnahmen behoben werden können, ist dies mit Sicherheit eine effektive Möglichkeit Fehler zu umgehen bzw. deren Auswirkungen so gering wie möglich zu halten! Was tun bei komplexeren Problemen?

Wie weit ist so eine Art von einem Self-Healing-System möglich? Je komplexer Services sind desto aufwändiger sind im Normalfall auch die Überprüfungsroutinen und im Endeffekt auch die Reparaturfunktionen.

Das Monitoring-System kann nur jene System überwachen, welche durch den Administrator eingetragen und konfiguriert wurden. Sollte einmal ein System oder eine Konfiguration vergessen werden, so wird dies nie überwacht werden und so einen falschen Eindruck von Fehlerfreiheit vermitteln. Dieser Konfigurations- und **Administrationsaufwand ist auf keinen Fall zu unterschätzen** – dabei ist es völlig irrelevant, ob eine kommerzielle Lösung oder ein Open Source Produkt eingesetzt wird. In beiden Fällen sind die Anpassung an die jeweiligen Anforderungen im Unternehmen vorzunehmen und laufend auf Aktualität zu überprüfen. Bei der Installation von neuen Systemen, neuen Applikationen oder auch bei Upgrades von Applikationen müssen die Monitoring-Parameter überprüft und gegebenenfalls angepasst werden. Bei Änderungen an der Infrastruktur und den Systemen werden auch die meisten „Fehlalarme“ produziert – die Änderungen wurden im Monitoring-System oftmals nicht nachgezogen und werden deswegen als Fehler interpretiert. Ein gewisses Maß an Fehlalarmen kann ohne gröbere Probleme verarbeitet werden. Werden diese Alarme und Events aber zum Regelfall so stellt dies die Sinnhaftigkeit bzw. Effizienz des Monitoring-Systems in Frage da es früher oder später zu zwei ungewollten Phänomenen kommen kann: Der Administrator reagiert bei Alarmen dann einfach nicht mehr oder nur mehr verspätet – auch wenn es richtige Alarme sind. Zweitens muss der Administrator jeden Alarm wieder mit übermäßigem Aufwand überprüfen. Fehlalarme machen Menschen einfach nachlässig. Bei Änderungen ist es hin und wieder sinnvoll in einer kontrollierten Situation (z.B. bei Wartungsarbeiten) eine künstliche Fehlersituation herzustellen (z.B. ein Service stoppen) und die Kontrolle im Monitoring-System durchführen, um zu sehen ob die Überprüfungsalgorithmen einwandfrei funktionieren und inwieweit man sich auf das System verlassen kann. Das System soll unwichtige **Events ausfiltern** und nur auf bestimmte sowie auf unbekannte Events entsprechend reagieren. Beim Auftreten von neuen bzw. unbekannten Events muss der System-Administrator entsprechend die Konfiguration des Monitoring-Systems anpassen. Zu Beginn des Produktiveinsatzes werden vom System mit Sicherheit eine große Anzahl von Events und Alarmen reportet – die Konfiguration des Systems muss erst **Schritt für Schritt verfeinert** werden. Dies ist ein wichtiger Punkt bei der Umsetzung. Dazu

ist dementsprechendes Know-How über das System bzw. die Applikation (Service) notwendig. Eine wichtige Voraussetzung für die Effizienz einer Monitoring-Lösung ist, dass es jemanden gibt der auf Alarme dessen entsprechend reagiert. Dieser Prozess muss sich im Unternehmen etablieren und auch dementsprechend gelebt werden.

Letztendlich nutzt ein sinnvoll eingesetztes Monitoring dem **Gesamtunternehmen**. Dadurch dass auf Probleme schneller und vor allem rechtzeitig reagiert werden kann und eventuelle Beeinträchtigungen vermieden werden, werden die Geschäftsprozesse nicht beeinträchtigt und das Unternehmen kann sich voll auf das Kerngeschäft konzentrieren.

Positiv bewertet wird die Erkenntnis, dass essentielle Bausteine für ein service-orientiertes IT-Unternehmen wie eine (heterogene) Monitoring-Lösung im Fokus verschiedener Open Source-Projekte und –Communities liegen.

6.1 Wichtige Entscheidungs- und Diskussionspunkte für die Umsetzung

Während der verschiedenen Projektphasen wurden verschieden Ansätze und Fragen aufgeworfen und diskutiert:

Soll eher auf **Trapping oder auf Polling** gesetzt werden? Beide Konzepte haben ihre Vor- und Nachteile. Beim Polling werden in definierten Intervallen Überprüfungsroutinen gestartet und damit Ressourcen benötigt. Beim Trapping werden Messages nur dann verschickt, wenn definierte Situationen eintreten. Allerdings stellt sich die Frage was passiert wenn das System keine Möglichkeit hat eine Message zu verschicken. Der Administrator glaubt alles in Ordnung, dabei ist vielleicht das gesamte System nicht mehr verfügbar. In der Praxis ist es daher notwendig einen sinnvollen Mix aus Trapping und Polling einzusetzen.

Soll man auf **Inband**-Monitoring setzen? Mit der Gefahr dadurch das User- und Datennetz zu beeinträchtigen? Oder auf **Outband**-Monitoring mit erhöhtem Ressourcenaufwand? Die Frage kann nicht allgemein beantwortet werden. Im vorliegenden Migrationsprojekt wurde auf Inband-Monitoring gesetzt. Einerseits weil keine redundanten Netzwerkwege existieren

und für zusätzliche Netzwerkverbindungen kein Budget zur Verfügung steht und andererseits weil die implizierte Überwachung des Usernetzes genutzt werden will und das Risiko der User-Beeinträchtigung als gering eingeschätzt wurde.

Sollen bei der Konfiguration „**Full Qualified Domain Names**“ (FQDN) oder **IP-Adressen** eingetragen werden? Die Verwendung von Namen und der Einsatz von DNS hat den Vorteil dass man sich bei IP-Adressen-Änderungen und System-Umstellungen weniger Gedanken über die Konfigurationen machen muss. Allerdings hat dies den Nachteil dass kein Monitoring funktioniert, sollte das DNS nicht verfügbar sein (außer für gecachte Informationen). Eine generelle Entscheidung für FQDNs kann nicht getroffen werden, da nicht alle Applikationen Namenseinträge akzeptieren (z.B. der SyslogAgent für Windows). Auch bei vielen Netzwerkswitches können nur IP-Adressen für den Syslog-Server bzw. als Trap-Destination eingetragen werden. Die Verwendung von DNS beim Monitoring gibt leider auch zusätzliche Angriffsmöglichkeiten. Zum Beispiel könnte durch „Denial of Service“-Attacken (DoS) das DNS beeinträchtigt bzw. außer Funktion gesetzt werden und zugleich auch das Monitoring-System. Es könnten dadurch wichtige Log-Einträge verloren gehen.

Was passiert wenn **kein Monitoring möglich** ist - z.B. auf Grund von fehlenden Voraussetzungen (z.B. DNS)? Oder wenn das **Monitoring-System** selbst **ausgefallen** oder nicht verfügbar ist? Für diese Situation müssen Überlegungen und Vorkehrungen getroffen werden. Überlegenswert wären zum Beispiel: ein redundantes Monitoring-System (Cluster?), ein verteiltes Monitoring-System (eventuell auch für den Einsatz mit Firewalls zu überlegen), Backup-Lösungen etc.

Was ist die sinnvollste **Verständigungsmöglichkeit**? Im vorliegenden Konzept wurde für den Anfang die Notification via E-Mail gewählt. Diese Variante hat allerdings vor allem beim Ausfall von Netzwerkkomponenten oder des Mailservers seine Nachteile. Hier könnte eine redundante Notificationskonfiguration angedacht werden. Wenn zum Beispiel auf eine Mail-Verständigung innerhalb eines definierten Zeitraumes keine Aktion gesetzt wird, so wird eine Verständigung via SMS versucht. Sollte auch dies nicht zu einer Lösung führen, so könnte eine weitere SMS an einen anderen Personenkreis geschickt oder ein Anruf getätigt werden. Die Entscheidung ist je nach Anwendungsfall dann zu treffen.

Ein weiterer wichtiger Punkt ist **Security**. Vor allem bei der Wahl des **Community-Strings** für die SNMP-Konfiguration sollten entsprechende Überlegungen angestellt werden. Standardmäßig werden die meisten Konfigurationen mit „public“ und „private“ als vordefinierte Community-Strings ausgeliefert. Diese sollten auf entsprechende geheime Strings abgeändert werden. Die Standardkonfiguration zu belassen wäre eine fahrlässige Security-Lücke.

6.2 Welche konkreten Probleme traten während den Arbeiten auf?

Die größten Herausforderungen stellte die Informationsbeschaffung dar. Einerseits mussten Informationen in der Analysephase um eine genaue Aufstellung der installierten und im Einsatz befindlichen Systeme zu erhalten, gesammelt werden. Zusätzlich bedurfte es beträchtlicher Energie die notwendigen Anforderungen zu eruieren, zu formulieren und festzulegen. Andererseits war es nicht immer einfach die notwendigen Informationen über verfügbare Produkte und Monitoring-Systeme zu erhalten. Jedes betrachtete Produkt wird zwar auf einer eigenen Webseite präsentiert, Detailinformationen sind in vielen Bereichen recht spärlich angegeben. Vor allem bei den kommerziellen Produkten wurden von mehreren Seiten die Informationen zusammengetragen. Auf technischer Seite war bei der Umsetzung und Konfiguration in einigen Bereichen Expertenwissen gefragt bzw. musste teilweise nachgelesen und erst angeeignet werden.

Bei der Testphase von „Zenoss Core“ traten durch die Verwendung des fertigen VMware-Images so gut wie keine gröberen Probleme auf. Erst das Upgrade auf die aktuellste Version (Development-Release) erforderte einigen Einsatz. Grund ist die Verwendung von rPath als zugrunde liegende Linux-Distribution und dessen eigenen Update- und Installationsroutinen – außerdem wurde von einem SVN[74]-Repository upgegradet.

Die weitere Integration der verschiedenen Systeme stellte konkret keine Probleme mehr dar nachdem die Basiskonfiguration auf dem jeweiligen System bereits durchgeführt wurde (z.B.: SNMP, etc.)

[74] Subversion, siehe auch http://en.wikipedia.org/wiki/Subversion_(software)

Bei Handling-Problemen von „Zenoss Core“ bzw. dessen GUI wurde eine sehr zufriedenstellende Unterstützung durch die Community per Mailing-Liste bzw. in späterer Folge durch ein eigenes Forum geleistet.

Zusammenfassend kann festgehalten werden, dass die größten Herausforderungen organisatorischer Natur bei der Festlegung von Zielen, Anforderungen usw. waren. Die geringsten Schwierigkeiten gab es bei der technischen Realisierung.

7 Next Steps

Die Konzeptions- und Konfigurationsphasen sind abgeschlossen. Die Gesamttests auf den Qualitätssicherungssystemen wurden bisher positiv absolviert. Dadurch steht einem Produktionseinsatz nichts mehr im Wege. Die nächsten Schritte werden unter anderem die Übergabe an die Produktion und der dortige Einsatz für die gesamten Produktionssysteme sein. Zu Beginn der Umsetzungsphase wird ein verstärktes Augenmerk auf eventuelle Probleme und die Konfiguration gelegt. Nach einer zweimonatigen intensiven Beobachtungsphase soll ein Review über die Ergebnisse und Erkenntnisse gezogen und das Konzept gegebenenfalls angepasst werden.

8 Future Work

Nach Etablierung der Basiskonfiguration des Monitoring-Systems und einer gewissen Einlebungsphase können darauf aufbauend weitere Verbesserungen und Erweiterungen umgesetzt werden.

Für ein umfassendes Monitoring sollen weitere Systeme eingebunden werden:

- Infrastruktur-Systeme (Klimaanlage, Temperaturüberwachung, USV-Anlagen, etc.)
- SAN-Storage
- Feineres Ressourcen-Monitoring
- Serviceorientiertes Monitoring
- Auswertung und Analyse von Logfiles (z.B.: Apache- und IIS-Webserver, Proxy-Logfiles, Application-Logfiles, etc.)

Es sollen weiters Überlegungen in Richtung Verständigung via SMS angestellt werden. SMS ist heute eine der sinnvollsten und effizientesten Verständigungsarten, da so gut wie jeder fast überall ein Handy bei sich trägt. Eine Alarmierung per SMS ist auch während der Nachtstunden möglich.

Im Bereich der automatischen Reparaturaktionen könnte eine automatisierte Verständigung des HW-Vendors implementiert werden. Viele HW-Hersteller bieten heute schon diese Dienste an. HP bietet diese Dienste unter anderem bei Storage-Lösungen an, wo das Storage-System bei Plattenfehlern direkt ein Trouble-Ticket erstellt und der Support automatisch eine Ersatzfestplatte zustellen lässt.

Die gespeicherten und aufbereiteten Daten sollen in Zukunft für Trendanalysen und Zukunftsprognosen vor allem bei Beschaffungsprozessen herangezogen und bewertet werden Das heißt, dass in diesem Bereich Aktivitäten für die Auswertungen und Bewertung der gesammelten Informationen gesetzt werden müssen.

Abbildungsverzeichnis

Abbildung 1-1 - Kommunikation zwischen Management-Station und Agent 7
Abbildung 1-2 - Aufbau-Skizze einer SNMP-Konfiguration 7
Abbildung 1-3 - vereinfachte SNMP Architektur 8
Abbildung 1-4 - Rolle der MIB 10
Abbildung 1-5 - SNMP MIB Object Identifier Hierarchy and Format 10
Abbildung 1-6 - SNMP Request (Poll) und Trap 13
Abbildung 1-7 - Inband Management 15
Abbildung 1-8 - Outband Management 15
Abbildung 1-9 - Beispiel eines SLA 21
Abbildung 2-1 - geplante und ungeplante Ausfälle 22
Abbildung 2-2 - Ursachen und Auswirkungen einer Unterbrechung 23
Abbildung 2-3 - Auswirkungen eines Ausfalls 25
Abbildung 2-4 - Ursachen von Fehlfunktionen 26
Abbildung 2-5 - Risiken / Ursachen von Ausfällen 26
Abbildung 2-6 - Verfügbarkeitskosten der IT vs Ausfallkosten der Geschäftsprozesse 29
Abbildung 4-1 - typischer Ablauf ohne Monitoring 36
Abbildung 4-2 - grober Projektplan 38
Abbildung 4-3 - Skizze Netzwerkdiagramm und Konfiguration 46
Abbildung 4-4 - Versionsvergleich Groundwork Monitor (Quelle: http://www.groundworkopensource.com/products/comparison.html) 60
Abbildung 4-5 - GroundWork Extensible Architecture 61
Abbildung 4-6 - Zenoss Core 1.1 Features 62
Abbildung 4-7 - Zenoss Dashboard 66
Abbildung 4-8 – Zenoss - Übersicht eines Devices 66
Abbildung 4-9 – Zenoss - Eventübersicht pro Device 66
Abbildung 4-10 – Zenoss - Performance-Graphen 67
Abbildung 4-11 – Zenoss - Report über Verfügbarkeit aller Devices 74

Anhang A - 1 - Performance Issues Affect Diverse Applications 96
Anhang A - 2 - hourly downtime cost 97
Anhang A - 3 - Kosten pro Stunde 97
Anhang A - 4 - Auswirkungen von Downtimes auf Kosten 98
Anhang A - 5 - Syslog Message Facilities 99

Anhang A - 6 - Syslog Message Severities 99
Anhang A - 7 - /etc/syslog.conf RedHat EL 103
Anhang A - 8 - /etc/syslog.conf VMware ESX 3 104
Anhang A - 9 - /etc/syslog.conf AIX 105
Anhang A - 10 - /etc/snmp/snmpd.conf RedHat EL 105
Anhang A - 11 - /etc/snmp/snmpd.conf VMware ESX 3 105
Anhang A - 12 - /etc/snmp.conf AIX 105

Anhang B - 1 - Konfiguration iLO 106
Anhang B - 2 - Konfiguration iLO 2 106

Tabellenverzeichnis

Tabelle 1 - Verfügbarkeitsklassen und Ausfallszeiten 18
Tabelle 2 - Ausfälle, Risikoeinschätzung, Auswirkungen 45
Tabelle 3 - Funktionsvergleich kommerzielle Systeme ("Big Four") 57

Abkürzungsverzeichnis

ADS, AD	Active Directory Services
AIX[75]	Advanced Interactive eXecutive
AJAX[76]	Asynchronous JavaScript and XML
BIOS	Basic Input Output System
BSD	Berkeley Software Distribution
CIFS	Common Internet File System
CIM	Compaq Insight Manager
CMDB	Configuration Management Database
CMIP	Common Management Information Protocol
CPU	Central Processing Unit
DAS	Direct Attached Storage
DB	Datenbank, Database
DBMS	Datenbank-Management-System
DC	Domain Controller
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone, ent- oder demilitarisierte Zone
DNS	Domain Name System
DoS	Denial of Service
EL	Enterprise Linux
ERP	Enterprise Ressource Planning
ETE	End-to-End (Monitoring)
EVA	Enterprise Virtual Array
FC	Fibre Channel
FCAPS	Faults, Configuration, Accounting, Performance, Security
FQDN	Full Qualified Domain Name
FTP	File Transfer Protocol
GB	Gigabyte
Gbit	Gigabit
GNU[77]	“GNU's Not UNIX”

[75] http://www.ibm.com/aix
[76] http://de.wikipedia.org/wiki/Ajax_(Programmierung)

GPL[78]	GNU Public License
GSM	Global System for Mobile Communication
GUI	Graphical User Interface
HA	High Availability
HDD	Hard Disk Drive
HP	Hewlett-Packard
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HW	Hardware
IBM[79]	International Business Machines
ICMP	Internet Control Message Protocol
IDC[80]	IDC
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IO	Input / Output
IP	Internet Protocol
ISO	International Standards Organization
IT	Informationstechnologie, Information Technology
ITIL	Information Technology Infrastructure Library
KMU	Kleine und mittlere Unternehmen
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAN	Metropolitan Area Network
MB	Megabyte
Mbit	Megabit
MIB	Management Information Base
MOM	Microsoft Operations Manager
MS	Microsoft; Monitoring System
MTBF	Meantime Between Failure

[77] http://www.gnu.org/
[78] http://www.fsf.org/licensing/licenses/gpl.html
[79] http://www.ibm.com/
[80] http://www.idc-austria.at/

MTTR	Meantime To Repair
NAS	Network Attached Storage
NFS	Network File System
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NMS	Network Management Station
NTP	Network Time Protocol
OE	Organisationseinheit(en)
OID	Object Identifier
OS	Operation System
OSI	Open Systems Interconnection Reference Model
OSS	Open Source Software
PDF	Portable Document Format
PDU	Protocol data unit
PMP	Performance Management Pack (HP)
PNG	Portable Network Graphics
RAID	Redundant Array of Independant Disks
RAM	Random Access Memory
REST	Representational State Transfer
RFC	Request for Comments
ROM	Read Only Memory
RRD	Round Robin Database
SAN	Storage Area Network
SHA	Secure Hash Algorithm
SIM	System Insight Manager
SLA	Service Level Agreement
SLM	Service Level Management
SMS	Short Message Send
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOX[81]	Sarbanes-Oxley Act
SPI	Smart Plugin

[81] http://thomas.loc.gov/cgi-bin/query/z?c107:H.R.3763.ENR:

SPoF	Single Point of Failure
SSH	Secure Shell
SSL	Secure Sockets Layer
SVN	Subversion
SW	Software
TCP	Transmission Control Protocol
TP	Twisted Pair
UDP	User Datagram Protocol
UPS	Uninterruptable Power Supply
USV	Unterbrechungsfreie Stromversorgung, Ununterbrochene Stromversorgung
VAS	Value Added Services
VBS	Visual Basic Script
VM	Virtual Machine
WAN	Wide Area Network
WAP	Wireless Application Protocol
WMI	Windows Management Instrumentation
ZEO	Zope Enterprise Object

Literaturverzeichnis

[4managers.de, 2006]	http://www.4managers.de, IT-Strukturen.pdf, letzter Abruf: 2006-05-12
[activestate.com, 2006]	http://www.activestate.com, letzter Abruf: 2006-05-11
[ajax.sys-con.com, 2007]	http://ajax.sys-con.com/read/284228_p.htm, letzter Abruf: 2007-01-16
[apache.org, 2006]	http://www.apache.org, letzter Abruf: 2007-03-04
[aspectra.ch, 2007]	http://www.aspectra.ch, letzter Abruf: 2007-02-06
[balabit.com, 2006]	http://www.balabit.com/products/syslog_ng/, letzter Abruf: 2006-11-25
[Bayuk, 1999]	Bayjuk J.L., „Infrastructure Monitoring Challenges“, Bear Stearns & Co. Inc, http://csrc.nist.gov/nissc/1999/proceeding/papers/p34.pdf, 1999
[bb4.org, 2006]	http://www.bb4.org, letzter Abruf: 2006-10-25
[bigsister, 2006]	http://bigsister.graeff.com, letzter Abruf: 2006-11-15
[bitcom.org, 2007]	http://www.bitkom.org, letzter Abruf: 2007-07-04
[bmc.com, 2007]	http://www.bmc.com, letzter Abruf: 2007-01-31
[Brutlag, 2000]	Brutlag Jake D., Aberrant Behavior Detection in Time Series for Network Monitoring; Proceedings of the 14th Systems Administration Conference (LISA 2000); http://www.usenix.org; 2000
[ca.com, 2007]	http://www3.ca.com/solutions/Solution.aspx?ID=315, letzter Abruf: 2007-02-01
[cacti.net, 2006]	http://www.cacti.net, letzter Abruf: 2006-05-23
[computerwelt.at-01, 2007]	Computerwelt, 01/2007, Seite 2
[cricket.sourceforge.net, 2007]	http://cricket.sourceforge.net/aberrant/rrd_hw.htm, letzter Abruf: 2007-01-22
[datac-gmbh.de, 2006]	http://www.datac-gmbh.de, Ausgabe 02/03, letzter Abruf: 2006-05-12
[Daxenbichler, 2004]	Daxenbichler St., „Online-Monitoring für die strategischen IT-Systeme einer Universitätsklinik am Beispiel der Tiroler Landeskrankenanstalten“, Februar 2004
[de.wikipedia.org, 2007]	http://de.wikipedia.org, letzter Abruf: 2007-02-06

[ec.europa.eu, 2007]	EU-Studie, „Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU“, November 2006; http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf, letzter Abruf: 2007-01-20
[ee.ntu.edu.tw, 2007]	http://lion.ee.ntu.edu.tw/Class/FT/slide/set01.pdf, letzter Abruf: 2007-07-04
[en.wikipedia, 2006]	http://en.wikipedia.org, letzter Abruf: 2006-03-28
[eurosox.at, 2007]	http://www.eurosox.at, letzter Abruf: 2007-01-31
[eventlogmanager.com, 2006]	http://www.eventlogmanager.com, „UNIX/Linux SYSLOG Event Management“, White Paper, letzter Abruf: 2006-05-23
[eventsentry.com, 2006]	http://www.eventsentry.com, letzter Abruf: 2006-11-15
[fsf.org, 2007]	http://www.fsf.org/licenses/gpl.html, letzter Abruf: 2007-01-15
[genesiscom.ch, 2007]	http://www.genesiscom.ch, letzter Abruf: 2007-02-16
[gnu.org, 2007]	http://www.gnu.org, letzter Abruf: 2007-02-21
[groundworkopensource.com, 2006]	http://www.groundworkopensource.com, letzter Abruf: 2007-02-01
[hp.com, 2007]	http://www.hp.com, letzter Abruf: 2007-03-08
[hp-openview, 2007]	http://www.openview.hp.com/, letzter Abruf: 2007-02-01
[ibm.com, 2007]	http://www.ibm.com, letzter Abruf: 2007-01-31
[ibm-tivoli, 2007]	http://www-306.ibm.com/Software/tivoli/, letzter Abruf: 2007-02-01
[idc-austria.at, 2006]	http://www.idc-austria.at, letzter Abruf: 2006-11-15
[ietf.org, 2006]	http://www.ietf.org/rfc/, letzter Abruf: 2006-12-07
[iicm.tugraz.at, 2007]	http://www.iicm.tugraz.at/home/vgarcia/data/slides/01_Einleitung_students.ppt, letzter Abruf: 2007-07-04
[itil.co.ui, 2006]	http://www.itil.co.uk/, letzter Abruf: 2006-05-07
[itil.org, 2006]	http://www.itil.org/de/, letzter Abruf: 2006-12-29
[itwissen.info, 2007]	http://www.it-wissen.info, letzter Abruf: 2007-02-16
[Jußen, 1999]	Jußen J.; Monitoring unter Windows NT; http://etdv.rurh-uni-bochum.de; letzter Abruf: 2006-11-16
[keos.de, 2006]	http://www.keos.de, letzter Abruf: 2006-05-09
[Kurschl, 2000]	Kurschl W., Monitoring von verteilten System, Dissertation, Johannes Kepler Universität, Juni 2000

[linbit.com, 2006]	http://www.linbit.com, letzter Abruf: 2006-05-13
[linux-magazine.com, 2003]	Schmitz Christian, „Flight Recording Box“, „Syslog-NG“, http://www.linux-magazine.com, December 2003
[microsoft.at, 2006]	http://www.microsoft.at, letzter Abruf: 2006-05-12
[microsoft.com, 2006]	http://www.microsoft.com, letzter Abruf: 2006-04-06
[mit.edu, 2006]	http://web.mit.edu/kerberos/www, letzter Abruf: 2006-04-25
[mrtg, 2006]	http://oss.oetiker.ch/mrtg/, letzter Abruf: 2006-10-25
[mysql.org, 2006]	http://www.mysql.org, letzter Abruf: 2006-11-10
[nagios.org, 2006]	http://nagios.org, letzter Abruf: 2007-03-04
[nagios-flapping]	http://nagios.sourceforge.net/docs/2_0/flapping.html; letzter Abruf: 2007-03-04
[nimsoft.com, 2006]	http://www.nimsoft.com, letzter Abruf: 2007-01-15
[nist.gov, 2006]	http://www.nist.gov, letzter Abruf: 2006-05-23
[oetiker.ch, 2007]	http://oss.oetiker.ch/rrdtool/, letzter Abruf: 2007-01-22
[olev.de, 2007]	http://www.olev.de/m/monitor.htm, letzter Abruf: 2007-07-10
[opennms.org, 2007]	http://www.opennms.org, letzter Abruf: 2007-02-01
[optaros, 2007]	optaros, „Open Source Catalogue 2007 U.S. Version 1.1“; http://www.optaros.com/en/publications/ white_papers_reports/open_source_catalogue_2007; letzter Abruf: 2007-01-16
[perl.com, 2006]	http://www.perl.com, letzter Abruf: 2006-05-11
[postgresql.de, 2007]	http://www.postgres.de/, letzter Abruf: 2007-03-04
[postgresql.org, 2007]	http://www.postgresql.org/, letzter Abruf: 2007-03-04
[python.org, 2006]	http://www.python.org, letzter Abruf: 2006-10-25
[redhat.com, 2006]	http://www.redhat.com, letzter Abruf: 2007-03-04
[reflectent.com, 2006]	http://www.reflectent.com, letzter Abruf: 2006-05-12
[Renner, 2006]	Renner B., Moser U., Schmid D., IT-Service-Management, Transparente IT-Leistungen & messbare Qualität, BPX-Edition, 2006
[rpath.com, 2006]	http://www.rpath.com, letzter Abruf: 2006-11-15
[samba.org, 2006]	http://www.samba.org, letzter Abruf: 2007-03-04
[securitymanager.de, 2007]	http://www.securitymanager.de/magazin/artikel_1281-print_it_ausfaelle_vermeiden.html, letzter Abruf: 2007-07-04
[Siegl, 1993]	Siegl M., Low-Level-Tools im Netzwerkmanagement, EDV-Zentrum der Technischen Universität Wien, sie9311.pdf, 1993

[Smith R. F., 2005]	Smith Franklin Randy, Taking Control: Monitoring the Windows Platform Proactively, ITPro Series, 2005
[snmp.com, 2006]	http://www.snmp.com/FAQs/snmp-faq-part1.txt, letzter Abruf: 2006-05-03
[snmp.org, 2006]	http://www.snmp.org, letzter Abruf: 2006-03-28
[Souppaya, 2006]	Souppaya M., Kent K., „Guide to Computer Security Log Management“ - Draft, National Institute of Standards and Technology, April 2006
[sourceforge.net, 2007]	http://www.sourceforge.net, letzter Abruf: 2007-03-01
[sox, 2006]	http://thomas.loc.gov/cgi-bin/query/z?c107:H.R.3763.ENR:, letzter Abruf: 2006-11-15
[syslogserver.com, 2007]	http://www.syslogserver.com, letzter Abruf: 2007-03-04
[uni-erlangen.de, 2007]	http://www4.informatik.uni-erlangen.de/z/Lehre/SS07/HS_FVSEZS/Themen/Fehlertypen.pdf, letzter Abruf: 2007-06-27
[vmware.com, 2007]	http://www.vmware.com, letzter Abruf: 2007-02-01
[Wahrig-Buhrfeind, 2003]	Wahrig-Burfeind R., Wahrig Fremdwörterlexikon, dtv, 6. Ausgabe, Oktober 2003
[wordreference.com, 2006]	http://www.wordreference.com, letzter Abruf: 2006-03-28
[www.at-mix.de, 2007]	http://www.at-mix.de/monitoring.htm, letzter Abruf: 2007-02-16
[zabbix.org, 2006]	http://www.zabbix.org, letzter Abruf: 2006-10-25
[zenoss.org, 2007]	http://www.zenoss.org, letzter Abruf: 2007-02-27
[zope.org, 2006]	http://www.zope.org, letzter Abruf: 2006-11-15

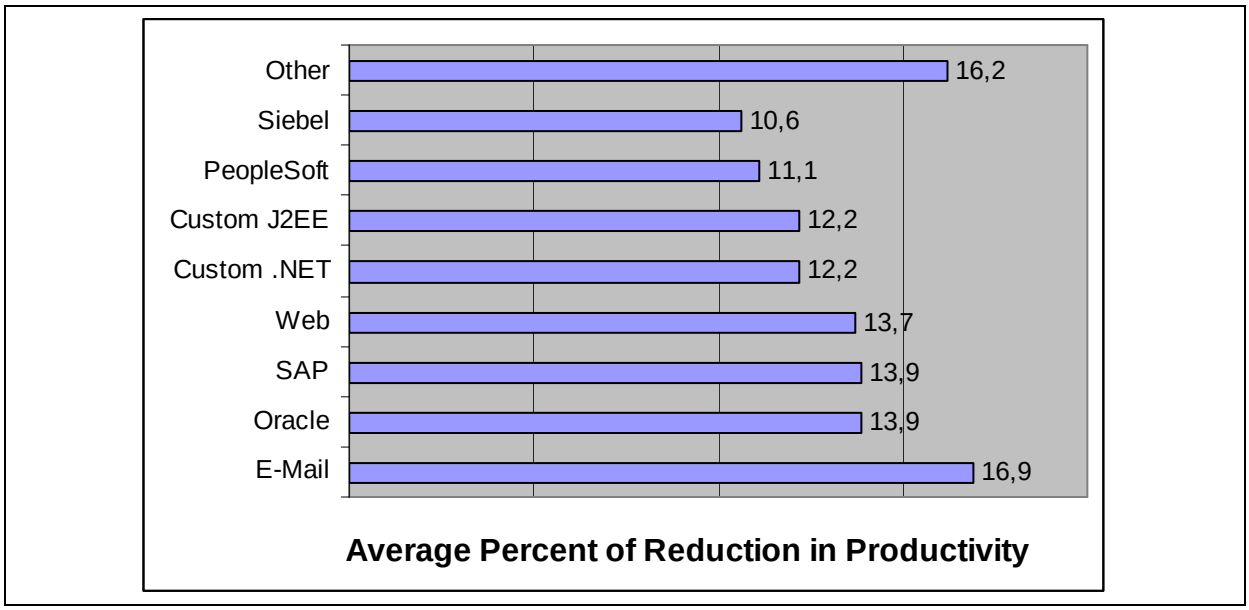
Anhang A

Auswirkungen von Downtimes

In [linbit.com, 2006] wird ein Rechenbeispiel für ein klassisches Netzwerk eines mittelgroßen österreichischen Betriebes aufgezeigt:

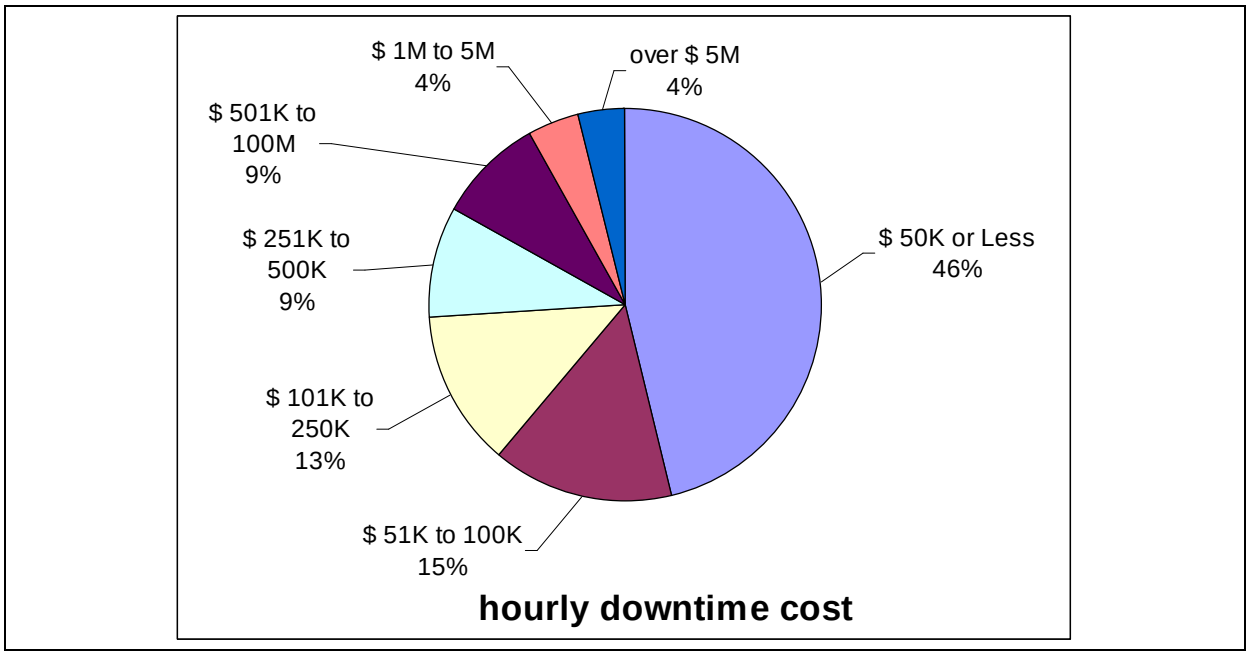
- 150 User mit 3 Servern (Firewall, Web/Mail, File-Server)
- Durchschnittliche Verfügbarkeit der Server im Unternehmen ist 97 %.
- Somit sind 3 % der Zeit, File/DB/Web/Printdienste, nichtverfügbar.
- Mitarbeiter arbeiten im Schnitt 1.700 Stunden pro Jahr.
- Wenn die Arbeitsgrundlage zu 3 % des Jahres nicht verfügbar ist und der Benutzer zu 75 % der Ausfallszeit nichts tun kann, dann: gehen für jeden Benutzer pro Jahr 38,25 Stunden an Produktiv-Zeit verloren und es ergibt sich bei durchschnittlichen Kosten von EUR 3.200,00 pro Mitarbeiter und Monat ein Verlust von EUR 800,00.
- Das bedeutet bei 150 Usern dementsprechend ein Produktivitätsausfall von EUR 120.000,00 pro Jahr.
- Anders formuliert: Jede Stunde Ausfall kostet EUR 3.000,00.
- Und das inkludiert nicht die Verluste durch nicht getätigte Geschäfte oder Pönalen durch verspätete oder verzögerte Lieferungen.

Downtime means idle employees and dramatic productivity loss. According to the Yankee Group 2005 Enterprise Application Management Survey, application performance issues result in an average productivity loss of 14%. Moreover, this loss affects a spectrum of applications, including e-mail, packaged applications and custom .NET and J2EE applications.[refelectent.com, 2006] – siehe Anhang A - 1.

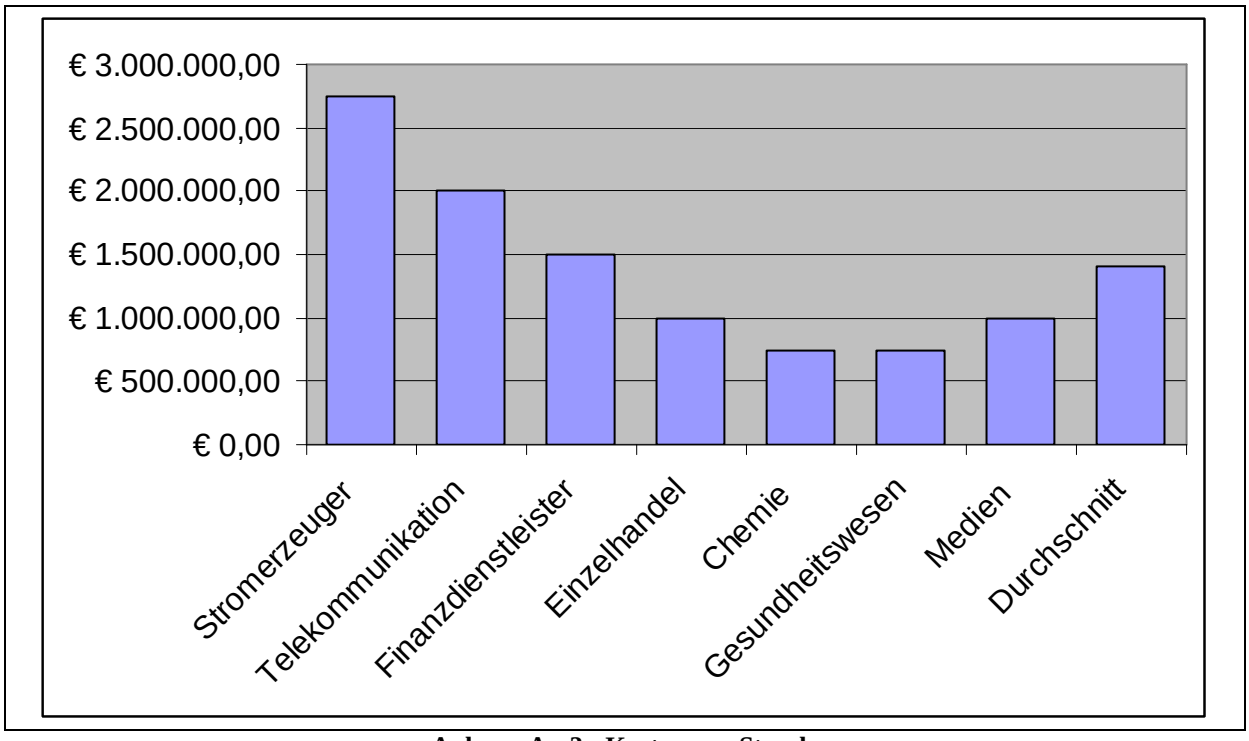


Anhang A - 1 - Performance Issues Affect Diverse Applications

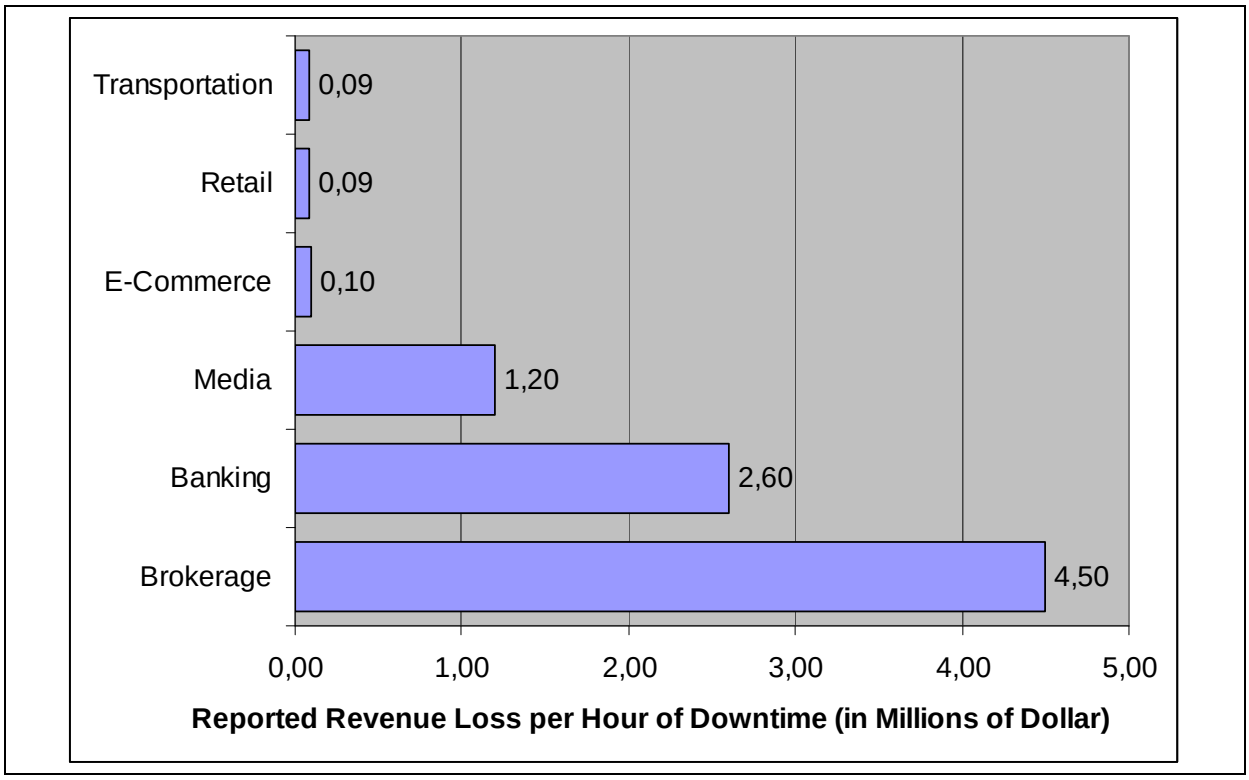
and Impact Employee Productivity
(Quelle: [reflectent.com, 2006])



Anhang A - 2 - hourly downtime cost
(Quelle: [s&t, IDC-Conference, 2006])



Anhang A - 3 - Kosten pro Stunde
(Quelle: [hp.com, 2007])



Anhang A - 4 - Auswirkungen von Downtimes auf Kosten
(Quelle: [reflectent.com, 2006])

Anhang A - 2 und Anhang A - 3 zeigen beispielhaft wieviel eine Downtime auf Stundenbasis anteilsmäßig Kosten verursacht. Die hohe der verursachten Kosten hängt mitunter von der Branche des Unternehmens ab. In Anhang A - 4 werden die verursachten Kosten durch Downtimes verschiedenen Branchen zugeordnet. Man sieht deutlich, dass die Kosten sehr stark steigen je mehr die Branche von der Informationstechnologie abhängig ist.

Anhang B

Syslog - Facilities und Severities laut RFC 3164

Numerical Code	Facility
0	kernel messages
1	user-level messages
2	mail system
3	system daemons
4	security/authorization messages
5	messages generated internally by syslogd
6	line printer subsystem
7	network news subsystem
8	UUCP subsystem
9	clock daemon
10	security/authorization messages
11	FTP daemon
12	NTP subsystem
13	log audit
14	log alert
15	clock daemon (note 2)
16	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

Anhang A - 5 - Syslog Message Facilities

Numerical Code	Severity
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

Anhang A - 6 - Syslog Message Severities

Sourcecode Beispiel Windows-Eventlog-Überwachung - Eventlog-to-Syslog:

Perl:

```
#!.\perl.exe -w
# $Author: juergen $
# $Id: EVENTMON_v01.PL,v 1.3 2007/01/22 21:03:28 juergen Exp $
# $Log: EVENTMON_v01.PL,v $
# Revision 1.3  2007/01/22 21:03:28  juergen
```

```
use vars qw( $Log $Message );
use strict;
use Win32::OLE qw( HRESULT );
use Socket;
my $syslogHost = shift @ARGV;;
my $debug=0;

if(!(defined($syslogHost)))
{
   $syslogHost='zenosshost.localdomain';
}
my $Machine = 'localhost';
my $TIMEOUT = -1;
my %HRESULT = (
  success =>  HRESULT( 0x00000000 ),
  timeout =>  HRESULT( 0x80043001 ),
);
#
my $this_host = Win32::NodeName;
#
use constant SYSLOG_PORT_NO => 514;
use constant   LOG_LOCAL3  => (19<<3); # reserved for local use
# default to LOCAL3;
my $facility = LOG_LOCAL3;
my $SockHandle;
my $portaddr;
########################
sub syslog_open ($)
{
    my $server = shift;

    socket (SockHandle, PF_INET, SOCK_DGRAM, getprotobyname("udp"))
   or return 0;

    my $ipaddr = inet_aton($server);
    $portaddr = sockaddr_in(SYSLOG_PORT_NO, $ipaddr);

    return 1;
}
########################################################################
# logit($message, $host, $facility)
# send a string to syslog server
# timestamp is from this computer
########################################################################
sub logit ($$$)
{
    my $msg = shift;
    my $host = shift;
    my $facility = shift;

    my $log_level = LOG_DEBUG | $facility;

    # strip day of week and year from localtime
    my $timestamp = substr scalar(localtime), 4, 15;

    my $txt = sprintf "<%d>%s %s $0: %s", $log_level,
       $timestamp, $host, $msg;
```

```
    return send(SockHandle, $txt, 0, $portaddr) == length($txt);
}
####################
# src: http://msdn.microsoft.com/library/default.asp?url=/library/en-
us/wmisdk/wmi/win32_ntlogevent.asp
#class Win32_NTLogEvent
#{
#  uint16 Category;
#  string CategoryString;
#  string ComputerName;
#  uint8 Data[];
#  uint16 EventCode;
#  uint32 EventIdentifier;
#  uint8 EventType;
#  string InsertionStrings[];
#  string Logfile;
#  string Message;
#  uint32 RecordNumber;
#  string SourceName;
#  datetime TimeGenerated;
#  datetime TimeWritten;
#  string Type;
#  string User;
#};
#
my $WMIServices = Win32::OLE->GetObject(
"winmgmts:{impersonationLevel=impersonate,(security)}//$Machine/" ) ||
die;
my $Events = $WMIServices->ExecNotificationQuery( "select * from
__instancecreationevent where targetinstance isa 'Win32_NTLogEvent'" );
die unless( $HRESULT{success} == scalar Win32::OLE->LastError() );

print "Listening to $Machine for events...\n";
$~ = "EVENT_RECORD_HEADER";
write;
$~ = "EVENT_RECORD";

my $hResult;
do
{
  my $Event = $Events->NextEvent( $TIMEOUT );
  $hResult = scalar Win32::OLE->LastError();
  if( $HRESULT{timeout} != $hResult )
  {
    $Log = $Event->{TargetInstance};
    $Message = $Log->{Message} || join( "\n", @{$Log-
>{InsertionStrings}} );
    chop($Message);

    # Eventypes:
    # 1 = Error
    # 2 = Warning
    # 3 = Information
    if(($Log->{EventType} eq "1") || ($Log->{EventType} eq "2"))
    {
                syslog_open($syslogHost) or die "Can't open syslog to
$syslogHost: $!";
```

```
                logit "$Log->{Logfile}:$Log->{EventCode}:$Message ($Log-
>{TimeWritten})", $this_host, $facility;

                write;
                print "\n";
   }
  }
}while( $HRESULT{timeout} != $hResult );

format EVENT_RECORD_HEADER =
Machine   Type         Source             Event Message
--------- ------------ ------------------ ------------------------------
--
.

format EVENT_RECORD =
@<<<<<<<< @<<<<<<<<<<< @<<<<<<<<<<<<<<<<<
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$Log->{ComputerName}, $Log->{Type}, $Log->{SourceName}, $Message
~
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                                          $Message
~
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                                          $Message
~
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                                          $Message
~
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                                          $Message
~
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                                          $Message
~
^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                                          $Message
.
###############
# END
###############
```

VBS:

```
' $Author: juergen $
' $Id: eventmon.vbs,v 1.0 2007/01/10 20:13:21 juergen Exp $
' $Log: eventmon.vbs,v $
' Revision 1.0  2007/01/10 20:13:21  juergen
'
' events are only printed to stdout – not sent to syslog-server
'
'
Dim strComputer                       ' computer being monitored
strComputer = wscript.arguments(0) 'first argument is name of
computer
```

```
Set Events = _
GetObject("winmgmts:{impersonationLevel=impersonate,(Security)}\\" &
strComputer & "\root\cimV2").ExecNotificationQuery ("select * from
__InstanceCreationEvent WHERE TargetInstance ISA 'Win32_NTLogEvent'")
’ get instance of ExecNotificationQuery

Set dtmConvertedDate = CreateObject("WbemScripting.SWbemDateTime")
’ get instance of Date-object

Wscript.Echo "waiting for events on " & strComputer & "..."
’ endless loop waiting for new events and print them out to console
Do
      Set NTEvent = Events.nextevent
      WScript.Echo NTEvent.TargetInstance.Message
      dtmConvertedDate.Value = NTEvent.TargetInstance.TimeGenerated
dtmGeneratedTime = dtmConvertedDate.GetVarDate
Wscript.Echo "Written: " & dtmGeneratedTime
Wscript.Echo "****************************"
Loop
```

Konfigurationen

Syslog:

```
# /etc/syslog.conf
# Log cron stuff
cron.*            /var/log/cron
# Everybody gets emergency messages
*.emerg           *
# Save boot messages also to boot.log
local7.*          /var/log/boot
# Send logs to centralized SYSLOG-server (zenoss) – 10.101.100.100
*.err             @10.101.100.100
*.crit            @10.101.100.100
mail.*            @10.101.100.100
auth.*            @10.101.100.100
authpriv.*        @10.101.100.100
```

Anhang A - 7 - /etc/syslog.conf RedHat EL

```
# /etc/syslog.conf
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                 /dev/console
# Log anything (except mail) of level info or higher.
# Don't log private authentication, cron, or vmkernel  messages!
*.info;mail.none;authpriv.none;cron.none;local6.none;local5.none
/var/log/messages
# The authpriv file has restricted access.
authpriv.*                              /var/log/secure
# Log all the mail messages in one place.
mail.*                                  /var/log/maillog
mail.*                                 @10.101.100.100
# Log cron stuff
cron.*                                  /var/log/cron
# Everybody gets emergency messages
```

```
*.emerg                                 *
*.emerg                                @10.101.100.100
# Save news errors of level crit and higher in a special file.
uucp,news.crit                          /var/log/spooler
# Save boot messages also to boot.log
local7.*                                /var/log/boot.log
#send all local6.info messages to special summary log only.
local6.info;local6.!notice              /var/log/vmksummary
#send all vmkernel .warning messages to warnings logs.
local6.warning                          /var/log/vmkwarning
local6.warning                         @10.101.100.100
#send all local6.notice and higher  messages to vmkernel log.
local6.notice                           /var/log/vmkernel
local6.notice                          @10.101.100.100
#send all userworld proxy messages to proxy log
local5.*                                /var/log/vmkproxy
#send all storage monitor related messages to storageMonitor log
local4.*                                /var/log/storageMonitor
local4.*                               @10.101.100.100
#
```

Anhang A - 8 - /etc/syslog.conf VMware ESX 3

```
# /etc/syslog.conf
# <facility> is:
#       * - all (except mark)
#       mark - time marks
#       kern,user,mail,daemon, auth,...
#
# <priority> is one of (from high to low):
#       emerg/panic,alert,crit,err(or),warn(ing),notice,info,debug
#       (meaning all messages of this priority or higher)
#
# <destination> is:
#       /filename - log to this file
#       username[,username2...] - write to user(s)
#       @hostname - send to syslogd on this machine
#       * - send to all logged in users
#
# example:
# "mail messages, at debug or higher, go to Log file.
# File must exist."
# "all facilities, at debug and higher, go to console"
# "all facilities, at crit or higher, go to all users"
#  mail.debug           /usr/spool/mqueue/syslog
#  *.debug              /dev/console
#  *.crit                       *
#
kern.debug                              /dev/console
kern.debug                              /var/adm/syslog.console
kern.debug                              @10.101.100.100
user.debug                              /var/adm/syslog.console
user.debug                              @10.101.100.100
auth.debug                              /var/adm/syslog.console
auth.debug                              @10.101.100.100
mail.debug                              /var/adm/maillog
mail.debug                              @10.101.100.100
lpr.debug                               /var/adm/lpdlog
```

```
syslog.debug                                    /var/adm/daemonlog
daemon.debug                                    /var/adm/daemonlog
syslog.debug                                    @10.101.100.100
daemon.debug                                    @10.101.100.100
#
```

Anhang A - 9 - /etc/syslog.conf AIX

SNMP:

```
# /etc/snmp/snmpd.conf
dlmod cmaX /usr/lib/libcmaX.so
rwcommunity XXX 127.0.0.1
rocommunity XXXpublic 127.0.0.1
rwcommunity  XXX zenosshost.localdomain
rocommunity  XXXpublic zenosshost.localdomain
trapcommunity public
trapsink zenosshost.localdomain public
syscontact admin@localdomain.com
syslocation Zimmer XXX
com2sec notConfigUser  default       public
group   notConfigGroup v2c           notConfigUser
view    systemview    included   .1.3.6.1.2.1.1
view    systemview    included   .1.3.6.1.2.1.25.1.1
access  notConfigGroup ""      any       noauth    exact  systemview
none none
pass .1.3.6.1.4.1.4413.4.1 /usr/bin/ucd5820stat
```

Anhang A - 10 - /etc/snmp/snmpd.conf RedHat EL

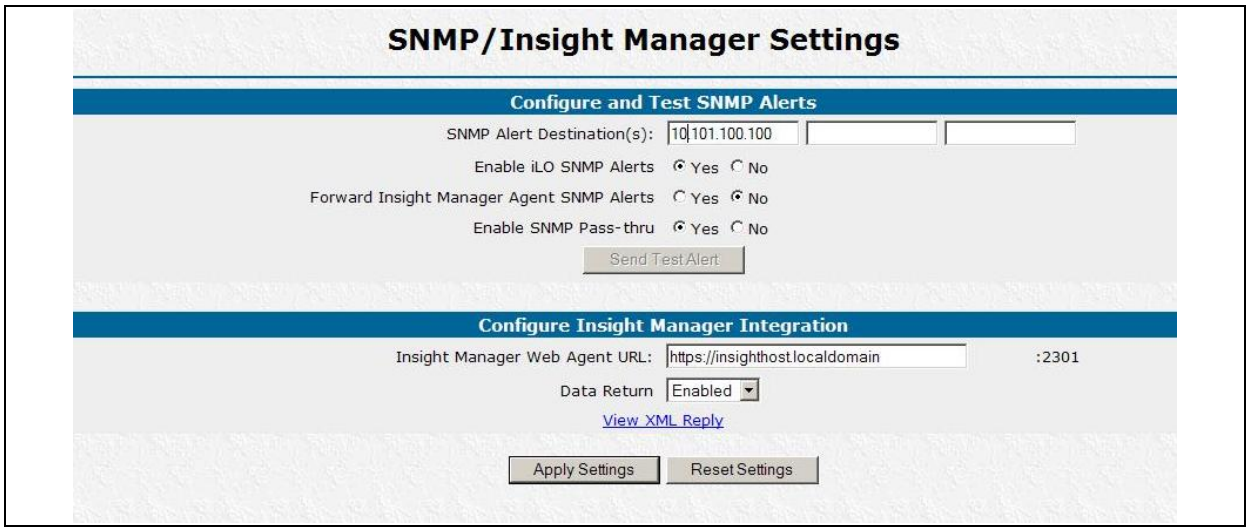
```
# /etc/snmpd/snmpd.conf
dlmod cmaX /usr/lib/libcmaX.so
rwcommunity  private  127.0.0.1
rocommunity  public  127.0.0.1
rwcommunity  XXXprivate zenosshost.localdomain
rocommunity  XXXpublic zenosshost.localdomain
trapcommunity XXX
trapsink zenosshost.localdomain XXX
syscontact  admin@localdomain.com
syslocation  Zimmer XXX
#
```

Anhang A - 11 - /etc/snmp/snmpd.conf VMware ESX 3

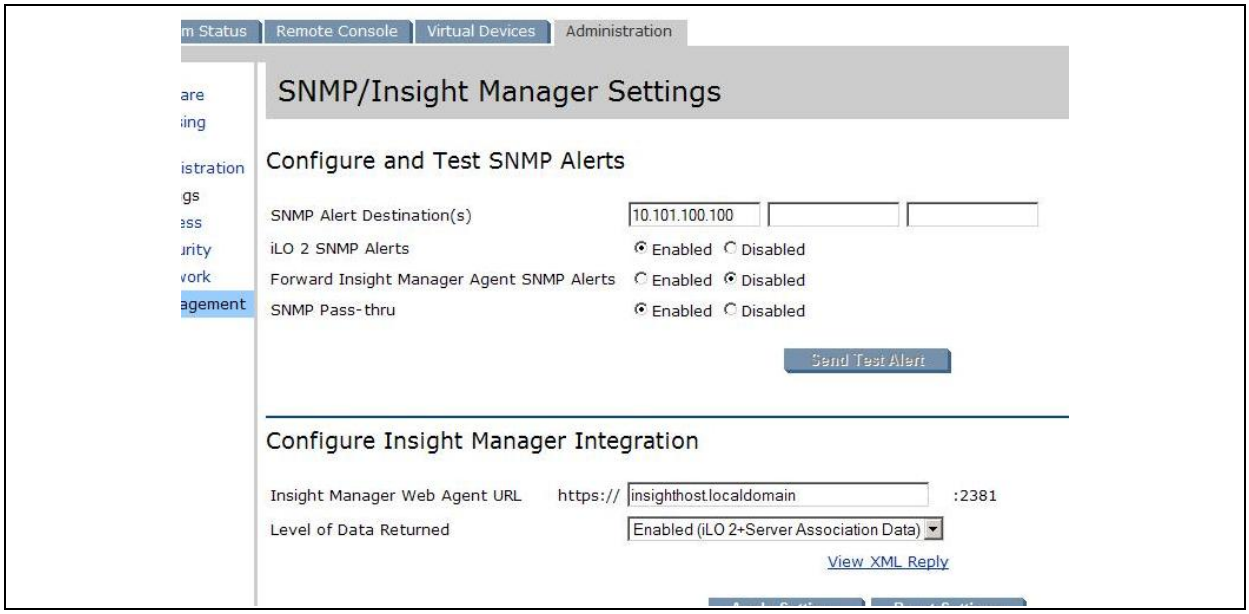
```
# /etc/snmp.conf
logging         file=/usr/tmp/snmpd.log         enabled
logging         size=0                          level=0
community       public
view            1.17.2          system enterprises view
trap            public          127.0.0.1       1.2.3   fe # loopback
trap            private         10.101.100.100  1.2.3   fe # host
#
```

Anhang A - 12 - /etc/snmp.conf AIX

iLO:



Anhang B - 1 - Konfiguration iLO



Anhang B - 2 - Konfiguration iLO 2