Fachhochschule
für Oekonomie & Management

University of Applied Sciences

Diplomarbeit
zur Erlangung des Grads eines
Diplom-Wirtschaftsinformatikers (FH)

- Backup & Restore - Konzepte, Strategien und Anwendungen von Einzelplatz bis Enterprise

Autor:
Dennis Wegner
Matrikelnummer: 178300

Betreuer:
Prof. Dr.-Ing. Torsten Finke

Lizenz- und Versionsinformationen

Namensnennung — Keine kommerzielle Nutzung — Keine Bearbeitung 3.0

Sie dürfen:

- das Werk bzw. den Inhalt vervielfältigen, verbreiten und öffentlich zugänglich machen.

Zu den folgenden Bedingungen:

- **Namensnennung** — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.
- **Keine kommerzielle Nutzung** — Dieses Werk bzw. dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.
- **Keine Bearbeitung** — Dieses Werk bzw. dieser Inhalt darf nicht bearbeitet, abgewandelt oder in anderer Weise verändert werden.

Wobei gilt:

- **Verzichtserklärung** — Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die ausdrückliche Einwilligung des Rechteinhabers dazu erhalten.
- **Sonstige Rechte** — Die Lizenz hat keinerlei Einfluss auf die folgenden Rechte:
  - Die gesetzlichen Schranken des Urheberrechts und sonstigen Befugnisse zur privaten Nutzung;
  - Das Urheberpersönlichkeitsrecht des Rechteinhabers;

– Rechte anderer Personen, entweder am Lizenzgegenstand selber oder bezüglich seiner Verwendung, zum Beispiel Persönlichkeitsrechte abgebildeter Personen.

- **Hinweis** — Im Falle einer Verbreitung müssen Sie anderen alle Lizenzbedingungen mitteilen, die für dieses Werk gelten.

Das Commons Deed ist eine Zusammenfassung des Lizenzvertrags in allgemeinverständlicher Sprache. Um den Lizenzvertrag einzusehen, besuchen Sie die Seite

`http://creativecommons.org/licenses/by-nc-nd/3.0/de/`

oder senden Sie einen Brief an Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Versionsinformationen

Die oben stehende by-nc-nd Creative Commons Lizenz gilt für das vorliegende Werk in der aus SVN-Revision 165 vom 12.01.2010, 11:36 Uhr erstellten Version, sowie zusätzlich für den zugrunde liegenden LaTeX-Quellcode. Diese Revisionsdaten wurden automatisch durch Subversion erzeugt.

Nobody wants backup, everybody wants restore.

- adminzen.org/backup

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listingverzeichnis

1 Einleitung

In diesem Kapitel wird die grundlegende Zielsetzung und Struktur der Arbeit vorgestellt. Außerdem werden die verwendeten typografischen Konventionen und die angesprochene Zielgruppe beschrieben.

1.1 Zielsetzung dieser Arbeit

In dieser Arbeit wird gezeigt, dass jede Datenhaltung auf Computersystemen ohne zumindest eine Sicherheitskopie ständig durch verschiedene Faktoren von Verlusten bedroht ist.

Es wird erläutert, inwiefern eine Absicherung der gespeicherten Daten gegen Verlust für jeden Anwender von Computersystemen ein immanent wichtiger Teil der Benutzung sein muss. Hierbei ist es prinzipiell unerheblich, ob ein einzelner Arbeitsplatzrechner, ein Serversystem oder ein komplettes Rechenzentrum betrachtet wird. Die anzuwendenden Methoden sind dem Grunde nach immer gleich und werden nur dem Anwendungsfall entsprechend skaliert umgesetzt. Bei den vorgestellten Lösungen wird der Fokus auf Open Source Produkte und die mitgelieferten Bordmittel des jeweiligen Betriebssystems gelegt.

Im Gegensatz zur üblichen Betrachtungsweise des Themenkomplexes wird in dieser Arbeit ebenso viel Wert auf den Aspekt der Wiederherstellung, wie auf den der Sicherung der Daten gelegt. Ohne eine erfolgreiche Wiederherstellung ist das Ziel einer jeden Datensicherung, nämlich eben die erfolgreiche Wiederherstellung der Daten im Verlustfalle, verfehlt.

1.2 Zielgruppe dieser Arbeit

Das auf Seite 3 beginnende Grundlagenkapitel dieser Arbeit ist für jeden Anwender gedacht, der ein Computersystem einsetzt und die darauf gespeicherten Daten gegen Datenverlust sichern möchte. Die dort besprochenen Fragen sind für alle Arten von Anwendern wichtig und sollten auch mit reinem Grundlagenwissen über Computersysteme

verständlich sein. Das Kapitel über Konzepte, beginnend auf Seite 17, vertieft die Grundlagen und führt auf ihnen aufbauend tiefer in die Thematik ein. In Kapitel 4 werden ab Seite 35 Backup-Praktiken für verschiedene Szenarios vorgestellt. Hier ist es vor allem in den Abschnitten über Arbeitsgruppen (4.2) und Enterprise-Umgebungen (4.3) sinnvoll, über erweiterte Kenntnisse der Betriebssysteme Windows, Linux und Mac OS X zu verfügen. Insbesondere sollten grundlegende Administrationskenntisse und Basiswissen über Dateisysteme vorhanden sein.

1.3 Konventionen in dieser Arbeit

Für diese Arbeit gelten die folgenden Konventionen:

Abkürzungen werden bei ihrem ersten Auftreten *kursiv* gesetzt und in einer Fußnote ausgeschrieben. Alle verwendeten Abkürzungen sind im Abkürzungsverzeichnis alphabetisch aufgeführt. Auszüge aus Scripten, Datei- und Verzeichnisnamen sowie Befehlseingaben und Adressen von Webseiten werden zu ihrer besseren Erkennbarkeit in `Typewriter` geschrieben. Wann immer es möglich ist, werden Listings direkt im Text benutzt; wo das aus Platzgründen nicht möglich ist, wird auf die entsprechenden Stellen im Anhang verwiesen.

Auf Quellen wird per Fußnote hingewiesen, alle Quellen werden im Literaturverzeichnis aufgeführt. Querverweise werden, ebenso wie erläuternde Hinweise, direkt in den Fußnoten ausgeschrieben.

In der elektronischen Ausgabe dieser Arbeit sind die meisten Verweise, Seitenangaben und Nummerierungen von Kapiteln und Abschnitten außerdem Navigationselemente. Diese führen den Leser per Klick zu der bezeichneten Stelle im Text.

2 Grundlagen

Dieses Kapitel beginnt in Abschnitt 2.1 mit einer Definition des Backup-Begriffs. Hier wird das Backup der Daten als Sicherheitskopie gegen eine Archivierung der Daten abgegrenzt. Außerdem werden Überlegungen vorgestellt, gegen welche Gefahren Daten überhaupt gesichert werden können, und welcher Aufwand dafür zu betreiben ist.

Der nachfolgende Abschnitt 2.2 stellt ab Seite 8 die grundlegenden Fragen vor, die sich stellen, wenn Daten auf einem Computersystem vor Verlust geschützt werden sollen. Diese Fragen sind dabei universell. Sie beschränken sich nicht auf einen speziellen Anwendungsfall sondern sind grundsätzlich auf jedes denkbare Computersystem anwendbar.

Das Kapitel schließt ab Seite 13 mit dem Abschnitt 2.3, der die Historie des Backups anreißt und die Entwicklung bis heute kurz beschreibt.

2.1 Backup: Definition

In diesem Abschnitt wird zunächst der abstrakte Begriff Backup definiert. Seine Verwendung in dieser Arbeit wird erläutert und der Themenkomplex der Sicherheitskopie gegen den der Archivierung abgegrenzt.

2.1.1 Definition und Abgrenzung zur Archivierung

Backup: Definition

Ein Backup bezeichnet üblicherweise *eine Kopie von Daten, die im Falle eines Verlusts der ursprünglichen Daten die Wiederherstellung ermöglicht.* In der Literatur existieren einige andere Wortlaute, auf diese Kernaussage können aber alle reduziert werden.

Problematisch an dieser Definition sind die einander beeinflussenden Begriffe Kopie, Daten, Verlust und Wiederherstellung. Es gibt eine ganze Reihe an unterschiedlichen Datentypen, von denen auf unterschiedliche Arten Kopien gemacht werden können. Der

Verlust kann auf verschiedene Art und Weise eintreten. Eine Wiederherstellung benötigt unter Umständen verschiedene Arbeitsschritte und muss in unterschiedlich großen Zeitfenstern geschehen.

Der Verlust einiger, auf einem privaten Computer als Dateien gespeicherter Musikstücke, aufgrund einer versehentlichen Löschung durch den Anwender, kann beispielsweise durch einen erneuten Import, der als Sicherheitskopie im Regal stehenden CD behoben werden. Der Aufwand und das zur Verfügung stehende Zeitfenster ändern sich unter Umständen beträchtlich, wenn es sich bei den verlustig gegangenen Daten stattdessen um eine Steuererklärung oder eine Diplomarbeit kurz vor der Abgabe handelt.

Ein noch drastischeres Beispiel wäre die versehentliche Löschung einer produktiven Datenbank eines international agierenden Unternehmens, die für Bestellannahmen über das Internet genutzt wird. Hier sind die Datenmengen um einige Größenordnungen höher und das Zeitfenster zur Wiederherstellung praktisch null.

Wie eine gute Backup- und Restore-Lösung für den konkreten Einzelfall aussieht, ist also höchst unterschiedlich. Dennoch kann jede individuelle Lösung durch die in Abschnitt 2.2 ab Seite 8 beschriebenen Fragestellungen gefunden werden. Ohne an dieser Stelle zu weit vorzugreifen, folgen nun die wesentlichen Punkte, die im Kontext dieser Arbeit ein gutes Backup ausmachen:

- Sicherung und Wiederherstellung sind Teile desselben Prozesses
- Die Sicherung ist für sämtliche in Frage kommenden Vorfälle geeignet[1]
- Die Sicherung wird protokolliert und etwaige Probleme angemessen mitgeteilt
- Die Wiederherstellung wurde erfolgreich getestet[2]
- Eine Wiederherstellung der Daten ist innerhalb des nötigen Zeitfensters möglich

Im folgenden werden die Begriffe Sicherung und Backup, sowie Wiederherstellung und Restore als Synonyme benutzt und beziehen sich dabei jeweils auf die vorangegangenen Punkte.

Abgrenzung zur Archivierung

Zunächst ist festzustellen, dass es sich sowohl bei einem Backup, als auch bei einer Archivierung um Kopien von Originaldaten handelt. Die Abgrenzung eines Backups von einer Archivierung ist in dem verfolgten Zweck der Kopie zu sehen.

[1] Siehe dazu auch den Punkt 2.1.2 „Gefahrenarten für Daten" auf Seite 6
[2] Und bestätigt damit gleichzeitig die erfolgreiche Sicherung

Backups halten den aktuellen Stand der Originaldaten vor, um sie beispielsweise bei einer versehentlichen Löschung schnell wiederherstellen zu können. Wenn in einem Verzeichnis eine Datei gelöscht wird, verschwindet sie üblicherweise mit einem der nächsten Backup-Läufe auch aus der Sicherheitskopie. Abhängig von den Rotationseinstellungen des Backup-Systems[3] kann das auch erst in Tagen oder Wochen sein, letztlich wird die Datei aber aus dem System herausfallen.

Das gleiche passiert, wenn eine Datei verändert wird. Die neue, bearbeitete Version wird für eine eventuelle Wiederherstellung bereitgehalten. Die veraltete Version wird aus dem Backup-System herausrotiert. Daten die von einem Backup-System gesichert werden, sind üblicherweise in Bearbeitung befindlich.

Archivierung verfolgt dagegen das Ziel, eine Datei in ihrer endgültigen Form aufzubewahren. Die Zeiträume sind dabei deutlich größer als bei einem Backup, es geht um Jahre bis Jahrzehnte und bei der Langzeitarchivierung auch darüber hinaus. Gleichzeitig muss, häufig auch per gesetzlicher Vorgabe, die Unveränderbarkeit der gespeicherten Daten sichergestellt werden. Man spricht hier von Revisionssicherheit[4].

Durch die lange Aufbewahrungszeit ergeben sich für die Archivierung auch völlig andere Problemstellungen als für das Backup. Insbesondere die Haltbarkeit von Dateiformaten und Datenträgern kann hier zu einem erheblichen Problem werden. Während beispielsweise Festplatten geradezu ideale Medien für ein Backup darstellen, ist ihre Lebensdauer von ca. zwei bis fünf Jahren[5] ungeeignet für eine Speicherung der Daten über Jahrzehnte.

Durch die Notwendigkeit, archivierte Daten sehr lange und unverändert auzubewahren, ergibt sich oftmals ein *WORM*[6]-System als Lösung[7]. Hierbei werden neben einer obligatorischen Dokumentenverwaltung auch spezielle optische Medien namens *UDO*[8] verwendet. Diese ähneln vom Prinzip her einer CD[9] oder DVD[10], sind aber in ihrer zur Archivierung geeigneten Form nur zum einmaligen Beschreiben geeignet. Eine höhere Datendichte (bis zu 28 Gigabyte pro Disk) wird durch den Einsatz eines blauen Lasers erreicht[11].

Im folgenden wird das Thema Archivierung nicht mehr weiter behandelt. Diese Arbeit setzt sich in den folgenden Abschnitten nur noch mit dem Themenkomplex Backup & Restore auseinander.

[3] Siehe dazu auch den Punkt 2.2.2 „Sicherungszeitpunkte und Zyklen" auf Seite 9
[4] [Kampffmeyer 2003]
[5] [Pinheiro et al. 2007]
[6] Write Once Read Multiple (Times)
[7] [Cohasset Associates, Inc. 2005]
[8] Ultra Density Optical
[9] Compact Disc
[10] Digital Versatile Disc
[11] [Gieselmann 2004]

Der interessierte Leser sei jedoch für einen sehr guten Einstieg in das Thema der Archivierung auf die entsprechenden Abschnitte der Grundschutzkataloge des Bundesamtes für Sicherheit in der Informationstechnik hingewiesen[12].

2.1.2 Gefahrenarten für Daten

Um ein erfolgreiches Backup-System zu implementieren, ist es zunächst nötig festzustellen, vor welchen Arten von Datenverlusten das System schützen soll[13]. Erst wenn die Gefahrenlage klar ist, kann das System entsprechend eingerichtet werden. Wichtig ist, die komplette Systemumgebung zu betrachten. Sind die Anwender vertraut mit dem Umgang mit Dateien auf Servern? Gibt es ein abgestuftes Rechtekonzept für die IT-Abteilung? Liegt das Rechenzentrum in einem bekannten Hochwassergebiet? Die Fragen nach den Gefahren für Daten, die sich vor dem Einsatz eines Backup-Systems stellen, sind vielfältig und individuell, sollten aber alle durchdacht werden. Denn nur wenn ein Backup-System den Ansprüchen gerecht wird, die im Ernstfall an dieses gestellt werden, kann ein erfolgreicher Restore-Vorgang stattfinden.

- *Anwenderfehler*
  Mit die häufigste Ursache für Datenverlust sind Fehler durch Anwender. Schnell ist der falsche Ordner oder die falsche Datei gelöscht. Durch ein Berechtigungssystem ist es aber zumindest möglich sicherzustellen, dass ein Anwender nicht die Daten eines anderen löschen kann. Das hilft freilich dem aktuell betroffenem Benutzer nur sehr wenig, vor allem wenn er der einzige Anwender des Computersystems ist. Eine zeitnahe, regelmäßige Sicherung der Daten, die im Zugriff der Benutzer liegen, ist hier der einzige wirksame Schutz.

- *Administratorfehler*
  Wenn ein Mitglied eines IT-Teams einen Fehler macht, sind die Auswirkungen in der Regel schwerwiegender als bei einem Anwenderfehler. Einfache Zahlendreher in einem Befehl können zur Löschung kompletter Partitionen oder Festplatten führen. Auf diesen können mitunter die produktiven Daten von sehr vielen Anwendern liegen. Eine kurze Restore-Zeit ist dann essentiell wichtig. Schützen kann man sich vor diesem Fehler vor allem durch Snapshots und redundant ausgelegte Plattensysteme.

- *Hardwarefehler*
  Hardwarefehler sind getrennt zu betrachten. Es ist ein Unterschied, ob ein komplettes System ausfällt oder eine Festplatte.

[12] [BSI 2005]
[13] [Garfinkel et al. 2003, S. 545 ff.]

  - *Systemausfälle*
    Eine defekte Netzwerkkarte am Fileserver, ein Absturz des darauf laufenden Betriebssystems oder ein Fehler am Netzteil. Alle diese Ursachen können potentiell zu einem Datenverlust führen. Sicher führen sie aber dazu, dass Anwender nicht mehr auf die Daten zugreifen können. Um diese Situation zu vermeiden, sind hochverfügbare Backup-Systeme einzusetzen, die im Fehlerfall innerhalb kürzester Zeit anstelle des ausgefallenen Servers übernehmen können.

  - *Festplattenausfälle*
    Festplatten enthalten viele mechanische Teile und weisen nach einer gewissen Zeit fast zwangsläufig einen Defekt auf[14]. Gegen diese Art von Ausfällen ist ein *RAID*[15]-System der passende Schutz. Der Ausfall einer Festplatte wird durch die redundante Verteilung der Daten auf mehrere Festplatten kompensiert.

- *Softwarefehler*
  Fehler in der zu benutzenden Software sind schlecht in den Griff zu bekommen. Bugs des Betriebssystems oder des verwendeten Datenbanksystems sind nie ganz auszuschließen und können potentiell eine große Menge von Daten in den Abgrund reißen. Der einzige wirksame Schutz sind regelmäßige Sicherungen auf andere Systeme.

- *Diebstahl oder Vandalismus*
  Diese beiden Ursachen von Datenverlust können sowohl rein elektronisch, etwa durch einen Cracker-Angriff oder auch profan physikalisch vorkommen. Wenn ein Privatanwender die externe Festplatte mit dem Backup neben dem Computer aufbewahrt und ein Dieb beides entwendet, kann das Backup noch so gut gewesen sein, die Daten sind trotzdem fort.

- *Naturkatastrophen*
  Ein Rechenzentrum in einem Hochwassergebiet muss gegen Wassereinbruch abgesichert werden. Anwender deren Häuser in einem bekannten Tornado-Gebiet stehen, sollten daran denken, die Backup-Festplatte mit in den Keller zu nehmen oder direkt ein Offsite-Backup vorhalten. Das gleiche gilt für Erdbeben, Springfluten und ähnliches. Idealerweise ist das Gebäude, in dem sich die Computer befinden, gegen die lokalen Unbillen der Natur abgesichert *und* es existiert ein Offsite-Backup. Im Falle von geschäftskritischen Anwendungen sollte dieses auch hochverfügbar ausgelegt sein. Soll das Backup auch gegen verheerende Naturkatastrophen abgesichert sein, empfiehlt sich ein Offsite-Backup in entsprechend großer Distanz von den Original-Daten.

[14] [Pinheiro et al. 2007]
[15] Redundant Array of Independent Disks

- *Andere Ursachen*
  Auch andere Ursachen sollten in Betracht gezogen werden. Beispielsweise kann der Verzicht auf einen Archivierungsprozess dazu führen, dass Daten trotz regelmäßiger Backups verloren gehen. Nämlich genau dann, wenn der Verlust der Datei erst auffällt, nachdem ihre Sicherheitskopie aus dem Backupset herausrotiert wurde. Schon fast in die Richtung der allgemeinen Systemsicherheit gehen die Hinweise auf die Möglichkeit von defekten Gas- und Wasserleitungen sowie Brandvermeidung. Je nach Standort des Computersystems sind dieser Tage auch terroristische Anschläge nicht ausgeschlossen. Mit einem guten Disaster Recovery Plan kann auch eine Katastrophe wie der 11. September 2001 systemseitig durchgestanden werden[16].

Ein ordentlich strukturiertes IT-System ist in jedem Fall eine gute und sinnvolle Grundlage für jede denkbare Backup-Lösung. Im Umkehrschluss ist es sehr schwierig und aufwändig, eine unstrukturierte IT-Landschaft mit einer guten Backup-Lösung auszustatten, ohne grundlegende Strukturen zu schaffen oder bestehende zu verbessern.

Nachdem nun die möglichen Gefahrenarten betrachtet wurden, wendet sich der nächste Abschnitt den Grundfragen eines jeden Backups zu.

2.2 Backup Grundprobleme

In den folgenden vier Unterabschnitten werden die grundlegenden Fragen gestellt, deren Antworten ein Backup-System definieren. Es werden außerdem einige beispielhafte Antworten vorgestellt und erläutert.

2.2.1 Zu sichernde Daten

Wie in Abschnitt 2.1.2 „Gefahrenarten für Daten“ ab der Seite 6 bereits dargelegt wurde, ist eine häufige Ursache für Datenverlust ein Administratorfehler. Darunter fallen aber nicht nur versehentliche Löschungen, sondern auch, und vor allem, versehentliches nicht Sichern von Daten. Ein Backup-System, welches nicht für sämtliche zu sichernden Daten konfiguriert wurde, ist eine Katastrophe, die nur darauf wartet zu passieren. Um dieses Problem auszuschließen, ist es sinnvoll zunächst das komplette System zu sichern[17]. Von der kompletten Sicherung aus können dann Bestandteile des Systems wie Caches, temporäre Dateien und Ordner sowie Swap-Daten ausgenommen werden.

Dieser Ansatz ist unter dem Namen *Exclude-List* bekannt. Es wird alles gesichert, außer den Dateien und Ordnern, die explizit vom Backup ausgenommen werden. Im Gegen-

[16] [Ballman 2001]
[17] [Preston 1999]

satz zur ebenfalls weit verbreiteten *Include-List* hat dieses Vorgehen den großen Vorteil, dass es so viel schwieriger ist, wichtige Bestandteile des Systems zu vergessen. Hier sind beispielsweise Programme zu nennen, die die gesicherten Daten dann auch tatsächlich lesen können. Dies ist insbesondere im Falle von proprietären Dateiformaten oder verschlüsselten Daten äußerst wichtig. Sollte ein System historisch gewachsen sein und die Software zum auslesen der Daten nicht mehr anderweitig zu beschaffen sein, ist das beste Backup wertlos.

Gerade dieses Szenario kann sehr schnell eintreten, wenn die Daten regelmäßig gesichert werden, das Betriebssystem und die Anwendungsprogramme aber nicht oder nur auf manuelle Veranlassung in unregelmäßigen Abständen.

Der einzige Vorteil der *Include-List*-Methode ist der Speicherplatzgewinn[18]. Ein Backup des vollständigen Systems verbraucht natürlich mehr Platz auf dem Backup-Medium als nur Teile des Systems. Außerdem spart man Netzwerkbandbreite, da weniger Daten vom Original zum Backup übertragen werden müssen.

Der Anteil, den installierte Programme und das Betriebssystem an dem gesamten zu sichernden Datenbestand haben, ist jedoch im Laufe der Zeit immer geringer geworden. Heute ist es schon auf privaten Einzelplatzsystemen üblich, gigabyteweise Daten zu speichern. Was auf diesen Systemen die Mediensammlung aus Musik-Dateien und Digitalfotos, ist auf Enterprise-Systemen die Datenbank mit enormem Speicherplatzbedarf. In beiden Fällen ist die Größe der zusätzlich zu sichernden Daten des Betriebssystems und der Anwendungen im Verhältnis zu den reinen Nutzdaten immer weiter gesunken. Daher wird heute, im Gegensatz zu früheren Empfehlungen[19], auch bevorzugt die Methode der kompletten Systemsicherung mit einer Exclude-Liste verwendet.

2.2.2 Sicherungszeitpunkte und Zyklen

Ein weiterer, sehr wichtiger Aspekt eines jeden Backups betrifft die Zeitpunkte der Sicherung. Der Idealfall eines Backuplaufs ist ein ruhendes System, bei dem an keiner der zu sichernden Dateien Änderungen vorgenommen werden. Hier ist vor allem zu beachten, wie schnell sich die Daten ändern und wie schnell nach jeder Änderung eine Sicherung erfolgen soll. Der schlimmste anzunehmende Datenbestand für eine Sicherung ist eine ständig im Zugriff befindliche Datenbank auf einem geschäftskritischem System[20]. In kleineren Firmen ist es üblich, den Backuplauf in der Nacht durchzuführen, wenn die Mitarbeiter nicht an den Datenbeständen arbeiten. Da auch die Administratoren nicht jede Nacht persönlich die Backups überwachen, ist es für eine solche Lösung unabdingbar, automatisiert abzulaufen. So kann ein Backuplauf nicht vergessen werden. Hierbei

[18] Zu den verschiedenen Möglichkeiten Speicherplatz zu sparen siehe auch Abschnitt 3.1 „Backup Methoden“ auf der Seite 17

[19] [Bögeholz 1999]

[20] Vgl. Abschnitt 3.2 „Backup von Datenbanken“ auf Seite 24

muss das Backup-System auftretende Fehler und auch den erfolgreichen Durchlauf aller Backupschritte auf eine Art und Weise an das IT-Team melden, die beachtet und überprüft wird. Ein fehlgeschlagenes automatisiertes Backup ist im Verlustfall der Daten ebenso schlimm wie ein vergessenes manuelles.

Ein anderer Fall sind Backups, die nahezu in Echtzeit ablaufen. Hier wird sofort nach jeder Änderung einer Datei ein Backup erstellt. Für wichtige Projekte mit sehr volatilen Daten kann das eine sehr sinnvolle Ergänzung sein[21], in der Regel ist aber ein Backup mit größeren Zeitabständen, etwa einmal pro Tag oder pro Stunde, ausreichend.

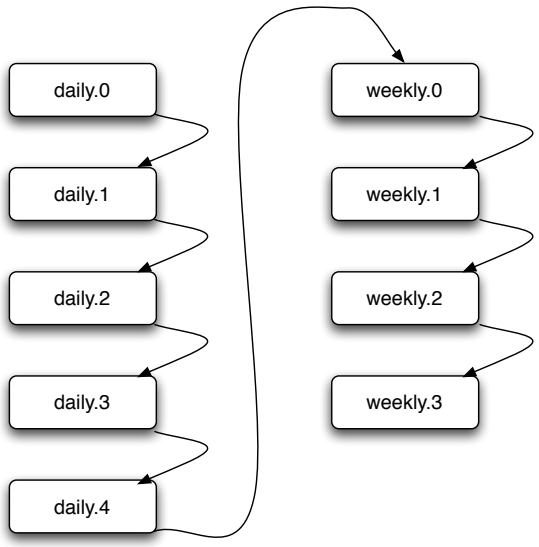

Abbildung 2.1: Beispiel für eine Backup Rotation mit zwei Zyklen

Wenn der Zeitpunkt klar ist, ist die nächste Frage nach den Zyklen und der Rotation der Backups zu beantworten. Es wäre nicht zielführend, das Backup der gestrigen Nacht mit dem der heutigen zu überschreiben. Vielleicht ist genau zwischen den beiden Läufen eine versehentliche Löschung oder Änderung an einer Datei vorgenommen worden, die wiederhergestellt werden muss. Daher ist es stets geboten, einige Versionen der gesicherten Daten nebeneinander zu sichern und vorzuhalten. Wie viele Versionen das im individuellen Fall sind und auf welche Art die Rotation durchgeführt wird, ist pauschal nicht zu beantworten. Es hat sich aber bewährt, mehrere Zyklen ineinander greifen zu lassen.

In der Abbildung 2.1 ist ein Beispiel für fünf Tagessicherungen und vier Wochensicherungen zu sehen. In dem Beispiel wird daily.0 direkt aus dem Dateisystem befüllt. Bei der nächsten Sicherung wird daily.0 zu daily.1 kopiert und neu befüllt. Am Ende der Tagesrotation wird daily.4 zu weekly.0 und anschließend von dem nachrückenden daily.3

[21] [Zivadinovic und Beier 2009]

überschrieben. In der wöchentlichen Rotation wiederholt sich das Prinzip mit größerem zeitlichen Abstand, nämlich wöchentlich statt täglich.

Eine solche Rotation kann natürlich auf individuelle Ansprüche angepasst werden. Sowohl die Rotationszyklen können, beispielsweise durch stündliche oder monatliche, erweitert werden als auch die Rotation selber kann beeinflusst werden. So ist ein tägliches Backup denkbar, das 30 Tage rotiert, ohne in einen weiteren Zyklus überführt zu werden.

2.2.3 Aufbewahrungszeitraum

Die Frage des Aufbewahrungszeitraums ist eng verwandt mit der Frage nach den Zyklen des Backups. Sie spielt ebenso in den Bereich der Archivierung hinein[22]. Gleichzeitig ist der Aufbewahrungszeitraum der Backups sehr von den Ansprüchen und Vorgaben des jeweiligen Anwenders abhängig. Um diesen gerecht zu werden, ist entweder eine entsprechend tiefe Rotationsmethode zu konfigurieren, oder es sind wie unter Abschnitt 2.2.2 ab Seite 9 beschrieben, mehrere ineinander geschachtelte Zyklen zu verwenden.

Außerdem ist der Anspruch des Anwenders auf ein Backup mit der betriebswirtschaftlichen Seite des Backup-Systems in Einklang zu bringen. Jeder weitere Rotationszyklus und jede weitere vorgehaltene Version benötigen Speicherplatz und verursacht damit Kosten. Um eine Backuplösung wirtschaftlich betreiben zu können und außerdem die Ansprüche der Anwender befriedigen zu können, ist also ein Mittelweg notwendig.

2.2.4 Verifikation und Integrität

Backups müssen getestet werden, um eine erfolgreiche Wiederherstellung gewährleisten zu können. Das bedeutet, dass im Prinzip nicht das Backup, sondern das Restore getestet wird. Tests müssen nicht nach jedem Durchlauf des Backups erfolgen, sollten aber idealerweise regelmäßig und automatisiert ablaufen. Als absolutes Minimum sollten Tests durchgeführt werden, wenn ein neuer Server dem Backup-System hinzugefügt wird oder wenn sich Abläufe des Backups ändern. Gerade im letzteren Fall ist ein Unterlassen von Tests fahrlässig zu nennen.

Zu testen ist beispielsweise:

- Die Wiederherstellung einiger einzelner Dateien aus einem großen Datenbestand mit anschließender Checksummenprüfung gegen das Original
- Die Wiederherstellung vieler, unterschiedlicher Dateien auf ein anderes System

[22] Siehe auch den Abschnitt 2.1.1 ab Seite 3 zur Abgrenzung von Archivierung und Backup

- Die Wiederherstellung eines kompletten Dateisystems auf ein anderes Medium und die anschließende Überprüfung der Besitzer und Rechte aller Dateien
- Die Wiederherstellung einer Datenbank zu einem früheren Stand als dem letzten Backup
- Die Wiederherstellung von Teilen der Datenbank zum aktuellen Stand in die gerade wiederhergestellte alte Version

Hier ist die Liste der möglichen Tests noch lange nicht beendet und es ist teilweise sehr schwierig, jeden möglichen Restore-Umstand voraus zu sehen und dann auch zu testen[23].

Eine bekannte Liste beschreibt Tests, die das Verhalten einiger unter Unix-Betriebssystemen gebräuchlicher Backup-Programme unter sehr schwierigen Umständen erprobt[24].

Die vorgestellten Test-Routinen arbeiten mit absichtlich erzeugten Löchern in Dateien, sehr langen Dateinamen mit ungewöhnlichen Zeichen als Namen, mehreren Tausend Verweisen auf ein und dieselbe Datei und weiteren unwahrscheinlichen, aber dennoch möglichen Problemen, die bei einem Backup, und damit auch bei einem eventuell notwendigen Restore für Schwierigkeiten sorgen können. Im Allgemeinen ist es nicht nötig, diese Art von Akribie an den Tag zu legen. Dennoch sollte ein Backup-System es erkennen und passend reagieren, wenn sich beispielsweise eine Datei während des Backuplaufs ändert oder gelöscht wird. Es kann nicht Sinn und Zweck eines Backup-Systems sein, in einem solchen Fall schlicht abzubrechen oder defekte Daten zu sichern.

2.2.5 Restore-Zeiträume

Nach der Verifikation der erstellten Backups steht noch das eigentliche Restore an. In der Literatur[25] findet sich für die Planung des Zeitraumes eines Restores der Begriff *RTO*[26]. Mit diesem Begriff wird der Zeitraum bezeichnet, den ein Restore vom Verlust der Daten oder der darauf basierenden Services bis zur erfolgreichen Wiederherstellung dauern darf. Darin enthalten ist der komplette Zeitaufwand für Versuche das Problem zu lösen, ohne auf ein Backup zurückzugreifen, die Kommunikation mit den betroffenen Anwendern und für den eigentlichen Restore-Vorgang.

Die RTO bezeichnet also die Zeitspanne, die ein System ausfallen darf. Diese kann je nach Service stark variieren und zwischen Null Sekunden und mehren Tagen oder Wochen betragen.

[23] [Preston 1999]
[24] [Zwicky 1991]
[25] [Preston 2007]
[26] Recovery Time Objective

In der Realität sind RTO-Zeiten trotz minutiöser Planung oft nicht einzuhalten. Ihr Wert sollte aber dennoch beibehalten werden, um die Restore-Prozedur mit den gewonnenen Erfahrungen für die Zukunft weiter zu optimieren.

Ein anderer sehr wichtiger Wert ist die *RPO*[27]. Sie hängt sehr stark mit den weiter oben erläuterten Problemen der zu sichernden Daten und der zu verwendenden Zyklen zusammen[28] und bezeichnet die Menge an Daten deren Verlust akzeptabel ist gemessen in Zeit.

Ist es beispielsweise für einen Studenten akzeptabel, die letzten zwei Stunden einer wichtigen schriftlichen Arbeit erneut von Hand zu erstellen, so hat diese schriftliche Arbeit eine RPO von zwei Stunden. Hält ein Unternehmen es für nicht akzeptabel auch nur einen einzigen Datensatz einer Datenbank erneut manuell zu erfassen, so ist die RPO für diese Datenbank null.

2.3 Kleine Geschichte des Backups

Die Geschichte des Backups ist sehr eng verzahnt mit der Geschichte der Computer. In diesem Abschnitt wird vor allem auf die verschiedenen Medien eingegangen, die in den letzten Jahren und Jahrzehnten für Backups eingesetzt wurden[29].

2.3.1 Geschichte der Backupmedien

Bereits die ersten Medien, die aus heutiger Sicht als Wechseldatenträger bezeichnet werden können, die Lochkarten, wurden für Backups eingesetzt. Sowohl Programme, als auch Nutzdaten[30] wurden auf Lochkarten erstellt und auch auf solchen gesichert. Dieses Vorgehen begann bereits 1951 mit dem ersten kommerziellen Computer, dem *UNIVAC I*[31].

Erst in den 1960er Jahren wurden die Lochkarten nach und nach durch Magnetbänder abgelöst. Sie erreichten deutlich höhere Datendichten und waren weniger fehleranfällig als ihre Vorgänger. Durch die Einführung der Magnetbänder entstanden die grundlegenden Backup-Konzepte, die bis heute Auswirkungen auf die moderne Datensicherung und ihre Rotationskonzepte[32] haben.

[27] Recovery Point Objective
[28] Vgl. Abschnitt 2.2.1 ab Seite 8 und 2.2.2 ab Seite 9
[29] [Yurin 2007]
[30] Vgl. Abschnitt 2.2.1 „Zu sichernde Daten“ ab Seite 8
[31] UNIVersal Automatic Computer I
[32] Vgl. dazu Abschnitt 2.2.2 „Sicherungszeitpunkte und Zyklen“ ab Seite 9

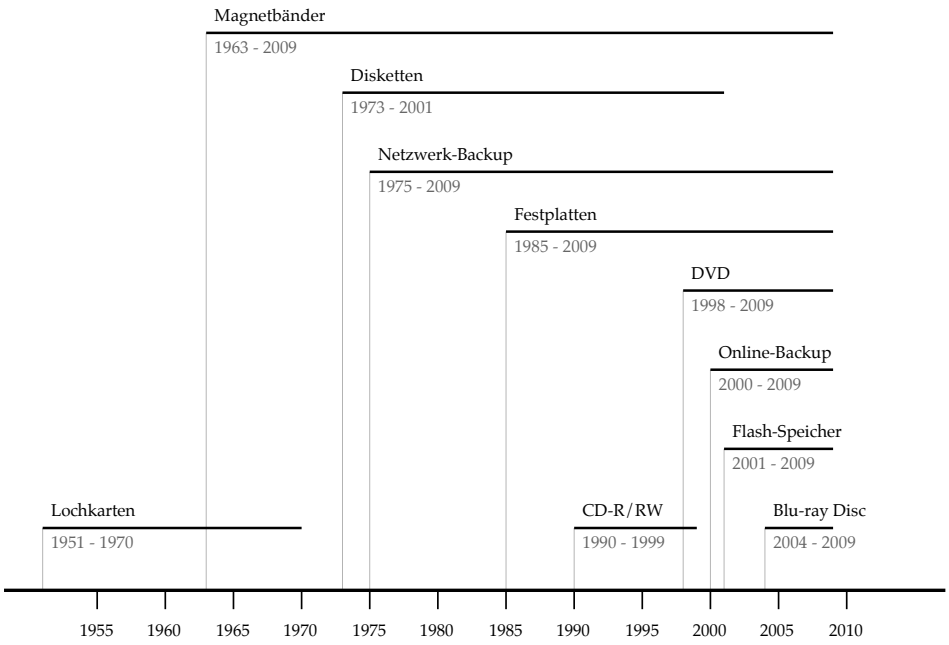

Abbildung 2.2: Verwendungszeiträume verschiedener Backupmedien

Ab den 1970er Jahren bis in die späten 1990er waren Disketten beliebte Medien für ein persönliches Backup. Ihre einfache Handhabung und ihr günstiger Preis machten sie für viele Privatanwender zu dem Medium für den Austausch und die Sicherung der persönlichen Daten. Ihre Beliebtheit sank in den folgenden Jahrzehnten im gleichen Maße, in dem die privaten Daten zunahmen und die einzelnen Dateien größer wurden. Allerdings wurden sie erst ca. 2001 von rapide günstiger werdendem Flash-Speicher in Form von USB-Sticks auf breiter Basis abgelöst. Die erhöhten Ansprüche an Kapazität, Geschwindigkeit und Robustheit konnten von diesen Medien deutlich besser befriedigt werden.

Festplatten wurden erst in den 1980er Jahren als Backupmedien eingesetzt. Vor diesem Zeitpunkt waren sie deutlich zu teuer, zu groß und mit zu wenig Kapazität ausgestattet, als dass sie den Magnetbändern den Rang hätten ablaufen können. Erst seit Mitte der 1990er Jahre können sie mit den Magnetbändern konkurrieren. Seit dieser Zeit sind diese beiden Medien die meist verwendeten für Backups überhaupt. Einzelplatzanwender verwenden externe Festplatten in Größen von hunderten Megabyte bis zu einem Terabyte für eine Sicherung des eigenen Systems. Unternehmen verwenden in ihren Rechenzentren große Festplattenverbände, die unter Umständen viele Terabyte an Daten sichern können.

Bei den optischen Medien sind vor allem die beschreibbaren Versionen der CD und der DVD zu nennen: Die CD-R(W) und die DVD-R(W). Nach der Erfindung der CD im Jahr 1979 dauerte es noch über 10 Jahre bis 1990 die ersten beschreibbaren CD-Rs in den

Handel kamen. Durch ihren anfänglich hohen Preis und die umständliche Handhabung konnten sie sich erst nach und nach als Backupmedium etablieren. Sie ersetzten nach Preissenkungen und technischen Verbesserungen vor dem Siegeszug der USB-Sticks in vielen Anwendungsfällen die zu klein gewordenen Disketten. Einen weiteren Schub gab es für die optischen Medien mit der DVD, die die Kapazität der optischen Medien auf fast 9 Gigabyte steigern konnte. Die Blu-ray Disc ist erst seit dem Jahre 2009 dabei, die Nachfolge der DVD anzutreten. Als Backupmedium hat sie sich bisher allerdings nicht hervorgetan und leidet, trotz respektabler 25 Gigabyte Kapazität pro Schicht, wie ihre Vorgänger am hohen Preis.

2.3.2 Backupübertragungsverfahren

Neben den verwendeten Medien hat sich auch die Übertragungsart der zu sichernden Daten im Laufe der Jahre gewandelt. Zu Zeiten der Lochkarten und der ersten Magnetbänder waren die Backupgeräte unmittelbar an die zu sichernden Maschinen angeschlossen. Erst mit der Entwicklung von Daten-Netzwerken in den siebziger Jahren des letzten Jahrhunderts konnten die zu sichernden Daten über ausreichend schnelle Leitungen an andere Orte transportiert werden. Diese Tatsache ermöglicht es heute Sicherheitskopien im notwendigen Abstand[33] von den Originaldaten aufzubewahren.

Zunächst entstanden technische Lösungen, die sehr dem heute bekannten *NAS*[34]-Konzept ähneln. Speicherplatz wurde über eine Datenleitung und verschiedene Protokollfamilien anderen Computersystemen zur Verfügung gestellt. So konnten die Backups über das bereits vorhandene Netzwerk gefahren werden. Der zusätzliche Traffic und der, die Übertragung weiter verlangsamende, Protokolloverhead sind Probleme, die das konkurrierende *SAN*[35]-Verfahren zu lösen versucht.

Ein SAN ist ein Netzwerk, welches sich komplett der Anbindung von Speicherplatz widmet. So werden andere Übertragungen im normalen Netzwerk nicht belastet. Dazu kommt, dass der angebundene Speicherplatz nicht dateibasiert, sondern blockorientiert verwaltet wird und dadurch eine erhebliche Reduzierung des Overheads ermöglicht. Ein solcher Performancegewinn schlägt sich aber auch im Preis und dem zu treibenden technischen Aufwand nieder. So ist beispielsweise eine zweite Netzwerkinfrastruktur parallel zu betreiben und die blockorientierte Zuweisung von Speicherplatz benötigt selbstverständlich ebenfalls Administration.

[33] Vgl. Abschnitt 2.1.2 „Gefahrenarten für Daten“ ab Seite 6
[34] Network Attached Storage
[35] Storage Area Network

2.3.3 Heutiger Stand der Technik

Die Nachkommen der damaligen Magnetbänder sind bis heute mit das populärste Backupmedium überhaupt. Durch Technologien wie Deduplizierung und Verschlüsselung auf dem Band wird die ursprüngliche Idee immer weiter entwickelt und es erschließen sich neue Anwendungsgebiete. Ihre schärfsten Konkurrenten sind die Festplatten geworden, die sich durch mechanische Verbesserungen und Technologien zur Senkung der Ausfallwahrscheinlichkeit auf breiter Basis etabliert haben. Auch für Einzelplatzsysteme sind Festplatten heute das Sicherungsmedium der Wahl.

Die beiden Netzwerkkonzepte NAS und SAN existieren derzeit parallel, wobei SANs so gut wie ausschließlich im Enterprise-Bereich vorkommen. NAS-Systeme sind dafür mittlerweile in vielen Haushalten zu finden und stellen dort neben dem Speicherplatz für Backups auch viele andere Funktionen zur Verfügung.

Eine interessante Option für alle Gruppen von Anwendern stellen die Online-Backups dar, die sich aus den frühen Anfängen der lokalen Netzwerke und der Entwicklung des Internets ergeben haben. So können heute quasi beliebig große Datenmengen relativ günstig bei spezialisierten Anbietern gesichert werden. Wird hier auf eine vorherige Verschlüsselung der eigenen Daten verzichtet, wird dem Anbieter ein wahrhaft gewaltiger Vertrauensvorschuss gewährt. Gerade Unternehmen sind oft auf die Vertraulichkeit ihrer Daten angewiesen und geben diese nur äußerst ungern aus der Hand, gerade wenn der Anbieter der Datensicherung seinen Sitz im Ausland hat. Desweiteren sind die meisten Internetzugänge nicht darauf ausgelegt Datenmengen in der Größenordnung von mehreren hundert Gigabyte hochzuladen. So werden sich die privaten Anwender, wie zu Zeiten der Diskette, auf eine Sicherung der wichtigsten Daten beschränken und das Gros lokal vorhalten.

Zu den weiteren Erwartungen für die zukünftige Entwicklung siehe Abschnitt 5.1 „Zukünftige Entwicklung“ ab Seite 60.

3 Backup-Konzepte

In diesem Kapitel werden die Grundlagen aus Kapitel 2 weiter vertieft und feiner beschrieben. Desweiteren wird auf die spezielle Problematik der Sicherung von Datenbanken eingegangen und der sichere Betrieb eines Backup-Systems diskutiert. Im Anschluss an dieses Kapitel werden die besprochenen Konzepte in Kapitel 4 praktisch aufgezeigt.

3.1 Backup Methoden

In diesem Abschnitt geht es um verschiedene Rotationsprinzipien für Backups. Während im Abschnitt 2.2.2 „Sicherungszeitpunkte und Zyklen“ auf Seite 9 noch davon ausgegangen wurde, dass jede Rotation jeweils eine vollständige Sicherung beinhaltet, wird hier nun aufgezeigt, wie durch geschicktes Vorgehen Speicherplatz gespart werden kann.

3.1.1 Volles Backup

Ein volles Backup ist der Backup-Typ, von dem bereits in Abschnitt 2.2.2 auf der Seite 9 die Rede war. Er wird in der Literatur häufig als Level 0 Backup bezeichnet. Ein volles Backup wird immer die gesamten zu sichernden Daten auf ein Backup-Medium kopieren. Dabei kann es dennoch rotiert werden. Es ist bei der Rotation darauf zu achten, das jeweils vorhergehende Backup nicht zu überschreiben. Ansonsten wäre es nicht mehr möglich, die eventuell irrtümlicherweise geänderten oder gelöschten Daten wiederherzustellen.

Wenn etwa jeden Tag ein volles Backup angefertigt wird und der Sonntag den letzten Montag überschreibt ergibt sich eine Rotation wie in Tabelle 3.1 dargestellt.

Tabelle 3.1: Volles Backup

Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
0	0	0	0	0	0	0

Ein solches Vorgehen bietet sich jedoch eher für kleinere Systeme an. Durch die Übertragung des vollständigen Datenbestandes muss jeden Tag ein großes Zeitfenster zur Verfügung stehen. Außerdem benötigt eine solche Rotation ständig den siebenfachen Platz der zu sichernden Daten. Ein Vorteil ist, dass man jeden Tag nur das letzte Backup benötigt um einen vollständigen Restore durchzuführen. Das ist mit den folgenden Konzepten unter Umständen anders.

3.1.2 Inkrementelles Backup

Ein inkrementelles Backup ist üblicherweise eine Datensicherung, die alle Daten sichert, die sich seit dem letzten Backup geändert haben. Es gibt aber verschiedene Möglichkeiten zu definieren, was das letzte Backup ist:

> So sichert ein inkrementelles Level 1 Backup alle Daten, die sich seit dem letzen vollen Level 0 Backup verändert haben. Wiederholt man mehrere Level 1 Backups hintereinander, werden dennoch immer die Änderungen seit dem letzten vollen Backup gesichert.
>
> Die inkrementellen Backups der Level 2 bis 9 sichern immer die Daten, die sich seit dem letzten Durchgang des nächstniedrigeren Levels geändert haben. Beispielsweise sichert ein Level 2 Backup alles was sich seit dem letzten Level 1 Backup geändert hat. Sollte kein Level 1 Backup vorhanden sein, wird stattdessen alles gesichert, was sich seit dem letzten vollen Level 0 Backup verändert hat.

Wie aber stellt ein Backup-System nun fest, unabhängig von dem Level des Backups, welche Daten sich geändert haben und daher gesichert werden müssen? Gleichzeitig ist damit festzustellen, welche Daten sich seit dem letzten Lauf nicht verändert haben und daher ausgelassen werden können.

Hierzu gibt es verschiedene Ansätze, die hauptsächlich vom verwendeten Betriebssystem abhängen. Auf Unix basierende Betriebssysteme arbeiten hierzu mit Zeitstempeln für jede einzelne Datei. In `mtime` wird die Zeit festgehalten, zu der die Dateiinhalte zuletzt geändert wurden. In `ctime` dagegen die Zeit, zu der Attribute wie der Besitzer der Datei oder die Zugriffsrechte zuletzt verändert wurden. Neben diesen beiden gibt es noch `atime`, in der die Zeit des letzten Zugriffs festgehalten wird. Für Backup-Zwecke werden aber vor allem die ersten beiden benutzt.

Mit der Kenntnis dieser Datumsangaben können Programme also durch simples Vergleichen der Zeitstempel festlegen, welche Datei im aktuellen Lauf gesichert werden muss und welche nicht.

Unter Windows-basierten Betriebssystemen wird traditionell das so genannte Archivbit[1] genutzt, um festzustellen welche Datei für ein Backup im aktuellen Lauf infrage kommt und welche nicht. Neben der eher unpassenden Bezeichnung[2] gibt es noch ein paar Probleme mit dem Archivbit.

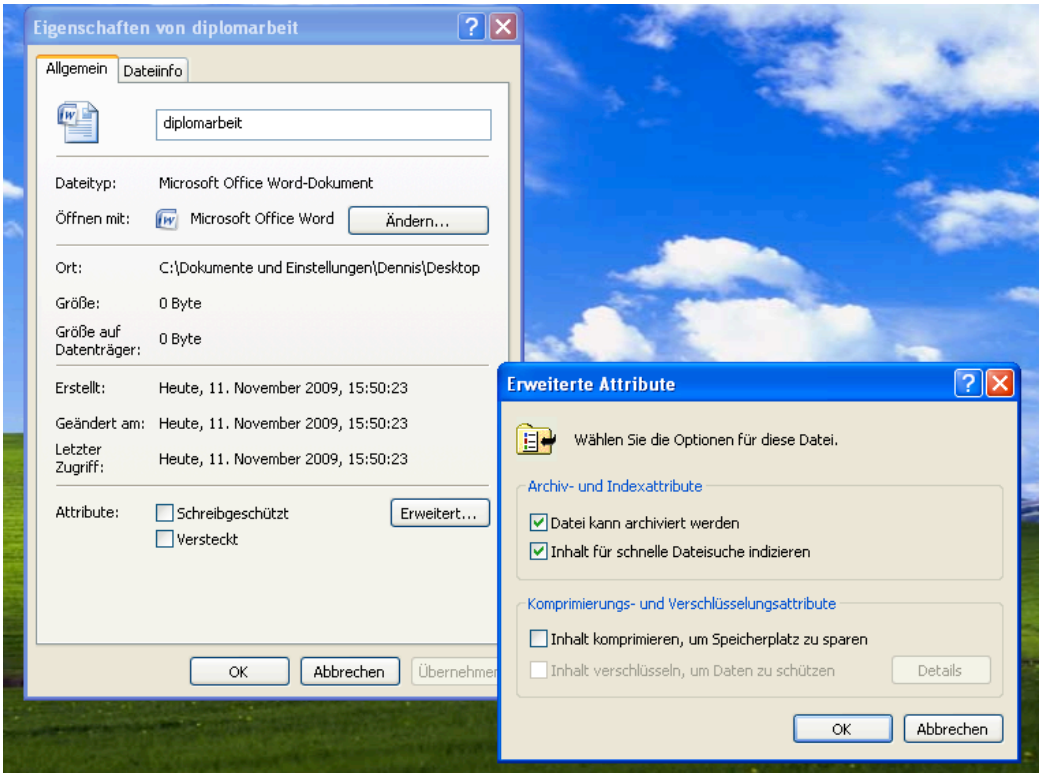

Abbildung 3.1: Anzeige des gesetzten Archivbits

Bei der Erstellung einer Datei unter Windows wird stets das Archivbit gesetzt, um anzuzeigen, dass diese Datei bei dem nächsten Backuplauf mit zu berücksichtigen ist. Nach einem Backuplauf wird dieses Bit dann entfernt und erst durch die nächste Änderung der Datei wieder neu gesetzt, damit die erfolgten Änderungen im nächsten Backuplauf gesichert werden.

Problematisch ist hier beispielsweise der Einsatz von mehr als einer Backup-Software. Sollte ein Anwender seine auf einem zentralen Server gespeicherten Daten mit einem eigenen Programm sichern, so wird das Archivbit von diesem Programm entfernt. Die Dateien wurden ja gesichert. Die Backup-Software, die den zentralen Server sichert, geht nun aber aufgrund des fehlenden Archivbits ebenfalls davon aus, dass diese Dateien nicht zu sichern sind. Auf diese Art und Weise könnten Anwender das komplette Backup-System eines Unternehmens ad absurdum führen. Weitere Programme die das Archivbit setzen und entfernen komplizieren den Stand weiter. Dazu ist das Archivbit, wie auf der

[1] [Leber 1998]
[2] Es gilt eine Entscheidung über ein Backup zu treffen, nicht über eine Archivierung

Abbildung 3.1 auf Seite 19 zu sehen, auch durch den Anwender selbst per einfachem Mausklick manipulierbar.

Als Ausweg aus diesem Dilemma führte Microsoft das *USN Journal*[3] ein[4]. Dieses, auch als Change Journal bekannte Verzeichnis, zeichnet alle Veränderungen auf und versieht sie mit einer eindeutigen Sequenznummer. Backup-Systeme, die das Change Journal unterstützen, können nun, ebenso wie unter Unix, durch einfache Vergleiche von Zeitstempeln feststellen, welche Dateien gesichert werden müssen und welche nicht. Diese Lösung skaliert mit beliebig vielen Software-Lösungen und ist daher auch für den Einsatz in einem sehr heterogenen Umfeld geeignet.

Durch den Einsatz der Technik des inkrementellen Backups kann sehr viel Platz auf den Backup-Medien eingespart werden. Allerdings hat diese Methode den Nachteil, dass für eine erfolgreiche Wiederherstellung immer mehrere Medien benötigt werden. Zunächst wird das letzte Vollbackup eingespielt, darauf folgen dann alle seitdem erstellten inkrementellen Sicherungen. Der Aufwand und damit die Zeit für ein erfolgreiches Restore steigt also im Gegensatz zu einem Vollbackup an. Einen Mittelweg beschreitet die Methode des differentiellen Backups, die im folgenden Abschnitt beschrieben wird.

3.1.3 Differentielles Backup

Ein differentielles Backup sichert alle Dateien, die sich seit dem letzten vollen Backup verändert haben[5]. Dabei lässt es aber, im Unterschied zu dem unter Abschnitt 3.1.2 ab Seite 18 beschriebenen inkrementellen Level 1 Backup, unter Windows das Archivbit unangetastet. Wurde ein volles Backup erstellt, auf das mehrere differentielle Backups folgen, verhält es sich genau so, wie das beschriebene Level 1 Backup. Wird jedoch zwischendurch nur ein inkrementelles Backup erstellt, so wird das Archivbit durch dieses entfernt und das nächste differentielle Backup sichert nur genau die Daten, die seit dem letzten inkrementellen Backup verändert wurden. Dieses Mischen von Vorgehensweisen ist daher nur mit sehr sorgfältiger Planung und gewissenhaftem Check der erstellten Logdateien zu empfehlen.

Das Erstellen von ausschließlich differentiellen Backups nach einem Vollbackup erzeugt eine gewisse Datenredundanz, da die selben Änderungen an den Dateien aufgrund des unveränderten Archivbits immer wieder neu gesichert werden. Dadurch verbraucht ein differentielles Backup mehr Speicherplatz als ein inkrementelles, ist aber im Restore-Fall leichter zu handhaben. Hier wird dann nur das letzte volle Backup und das neueste differentielle Backup benötigt, um den aktuellen Stand der Daten wiederherstellen zu können. Ein Beispiel für ein solches Backup findet sich in Tabelle 3.2 auf der Seite 21.

[3] Update Sequence Number Journal
[4] [Cooperstein und Richter 1999]
[5] [Luther 2004]

Tabelle 3.2: Differentielles Backup

Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
0	Diff/1	Diff/1	Diff/1	Diff/1	Diff/1	Diff/1

Es ist also wichtig auf eine konsistente Strategie der eigenen Backups zu achten, um mögliche konzeptionelle Fehler in den Backups von vornherein auszuschließen. In den folgenden beiden Abschnitten werden nun zwei Rotationsprinzipien vorgestellt, die für eine effektive Medienausnutzung sorgen können.

3.1.4 Großvater-Vater-Sohn Prinzip

Das Großvater-Vater-Sohn Prinzip[6] ist seit Jahrzehnten eines der beliebtesten Rotationsprinzipien für Backupmedien und ist auch unter dem Namen Drei-Generationen Prinzip bekannt.

Mit diesem Prinzip ist es möglich, die Backups für ein ganzes Jahr durch geschickte Rotation auf relativ wenigen Medien zu sichern. Für das auf der Abbildung 3.2 auf Seite 22 abgebildete Verfahren benötigt man ausgehend von einem ersten Vollbackup:

- Vier Medien für inkrementelle Sicherungen von Montag bis Donnerstag. Dies ist die Generation der Söhne.
- Drei Medien für die am Freitag stattfindende vollständige Level 0 Sicherung. Diese Medien gehören der Generation der Väter an.
- An jedem letzten Freitag eines Monats, wird eine Level 0 Sicherung auf eines von zwölf Medien für die Monate des Jahres durchgeführt. Diese bilden dann die Generation der Großväter.

In diesem Verfahren kommt man mit 19 Medien für ein ganzes Jahr aus. Hierbei lassen sich Daten aus der aktuell laufenden Arbeitswoche tagesaktuell wiederherstellen. Wenn erst am Dienstag einer Folgewoche auffällt, dass Daten vom vorhergehenden Montag benötigt werden, liegen diese allerdings nur noch in der Version vom Freitag der vergangenen Woche vor. Das Montagsmedium ist ja schon vom aktuellen Montagsstand überschrieben worden. Daten aus zurückliegenden Monaten lassen sich dann auch nur noch zum Stand des jeweiligen letzten Freitags des Monats wiederherstellen.

Das Vorgehen kann natürlich durch den Einsatz weiterer Medien weiter verfeinert werden. So ist es vorstellbar, dass die Großvater-Sicherung am letzten Freitag des Monats

[6] [Luther 2004]

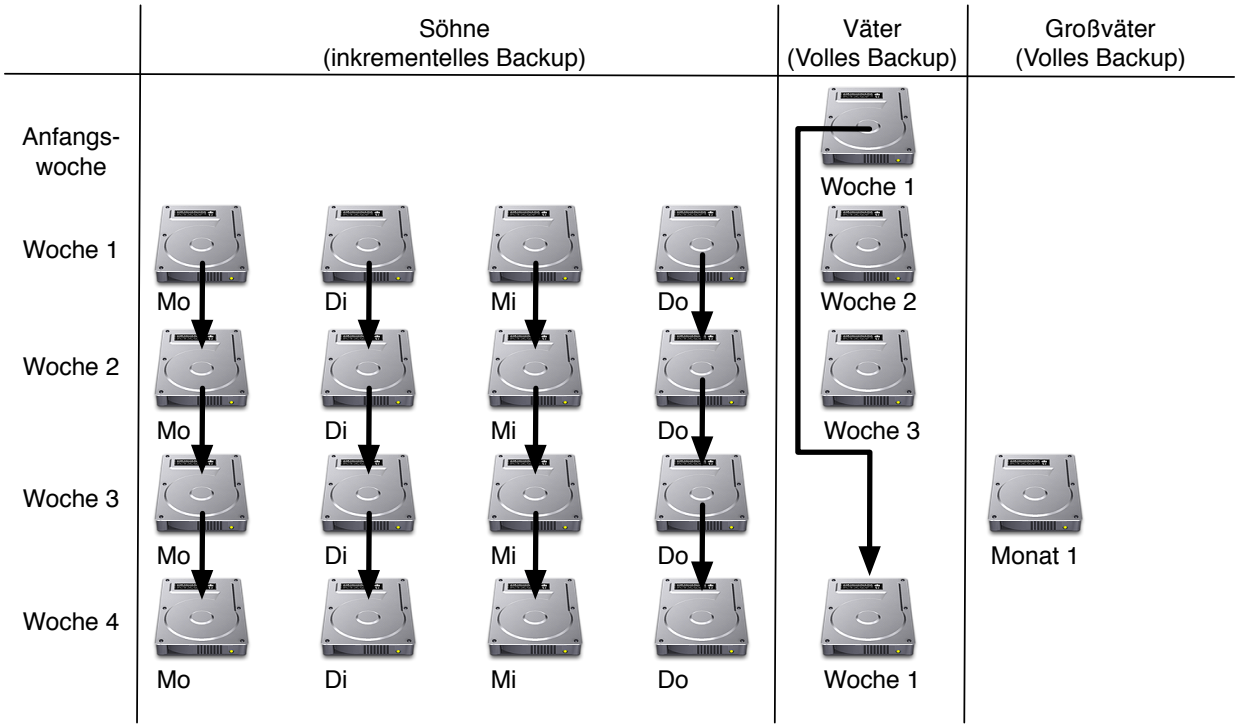

Abbildung 3.2: Dreistufiges Großvater-Vater-Sohn Prinzip

nicht anstelle des Vater-Backups durchgeführt wird, sondern zusätzlich. Wird hierbei die Großvater-Sicherung nicht von der Sohn-Sicherung, sondern aus der Vater-Sicherung erstellt, ist auch an den benötigten Zeitfenstern nichts zu ändern. Prinzipiell stünde für diese Sicherung dann eine komplette Woche zur Verfügung, eben die Zeitspanne, bis das Vater-Medium wieder in die neue Rotation eingebunden wird.

3.1.5 Türme von Hanoi Prinzip

Eine weitere sehr interessante Methode ist die Rotation nach dem Prinzip der Türme von Hanoi[7]. Dieses Konzept beruht auf dem alten mathematischen Knobelspiel, bei dem unterschiedlich große Ringe von einem Stab auf einen anderen gestapelt werden müssen. Dabei ist es nicht erlaubt, einen größeren Ring auf einen kleineren Ring zu legen, außerdem gibt es einen dritten Stab um dort Ringe zwischenzulagern.

Das Prinzip wird für die Rotation von Backupmedien auf die in Abschnitt 3.1.2 auf Seite 18 eingeführten Backup Level übertragen. Diese dürfen analog zu den Ringen des Knobelspiels auch nur ein jeweils niedrigeres Backup Level referenzieren. Eine schematische Darstellung des Konzepts für eine Woche ist in Tabelle 3.3 auf der Seite 23 zu sehen.

[7][Preston 2007]

Tabelle 3.3: Türme von Hanoi Rotation

Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
3	2	5	4	7	6	0

Tabelle 3.4: Erweiterte Türme von Hanoi Rotation

	Woche 1	Woche 2	Woche 3	Woche 4
Montag	0	1	1	1
Dienstag	3	3	3	3
Mittwoch	2	2	2	2
Donnerstag	5	5	5	5
Freitag	4	4	4	4
Samstag	7	7	7	7
Sonntag	6	6	6	6

Ausgehend von einem vollen Backup am Sonntag startet die Woche mit einer Level 3 Sicherung am Montag. Sollte am Montag eine Datei geändert worden sein, wird sie hier mit gesichert, da ja alles seit dem nächst niedrigeren Backup gesichert wird. Wird nun am Dienstag eine weitere Datei verändert, so wird diese von dem Dienstagsbackup zusammen mit der geänderten Datei von Montag gesichert. Das passiert, weil das Level 2 Backup am Dienstag das Level 3 Backup von Montag nicht sichern darf, sondern stattdessen ebenfalls auf das Level 0 von Sonntag zurück greift. Das Level 5 Backup am Mittwoch sichert dann nur die Änderungen, die Mittwoch passiert sind, da es auf das Level 2 Backup von Dienstag zurück greifen kann. Am Donnerstag wird das Level 4 Backup dann wieder auf den Dienstag als nächst niedrigeres Backup zurück greifen. Danach wiederholt sich das Prinzip für den Freitag und den Samstag bevor am Sonntag wieder ein volles Backup erstellt wird.

Bei diesem Vorgehen wird jede Datei zwei mal gesichert. Abgesehen von der Datei die am Dienstag verändert wurde. Hier ist das System also für Hardwaredefekte[8] besonders anfällig.

Um diese Gefahr zu mildern, kann das Konzept um sonntägliche Level 1 Backups ergänzt werden. Eine schematische Darstellung der Erweiterung ist in der Tabelle 3.4 zu sehen. So wird jeden Sonntag eine inkrementelle Sicherung der Dateien erstellt, die seit dem letzten monatlichen Vollbackup verändert wurden. Auf diese Weise ist das System deutlich besser vor Hardwaredefekten geschützt.

[8] Vgl. Abschnitt 2.1.2 „Gefahrenarten für Daten“ auf der Seite 6

Ein System wie die Türme von Hanoi oder auch das Großvater-Vater-Sohn Prinzip ist nur mit viel Disziplin und Konsequenz von Hand zu verwalten. Sollten Magnetbänder als Medium ausgewählt werden, ist über deren Verwendung sorgfältig Buch zu führen, um Datenverlusten durch Verschleiß vorzubeugen. Hier bietet es sich an auf die Automatismen von Backup-Systemen zu setzen, um Datenverluste durch menschliches Versagen auszuschließen. Moderne Tape-Library Systeme bieten ausgefeilte Möglichkeiten, um oft verwendete Bänder automatisiert gegen frische auszutauschen und die Daten umzukopieren. Auch wenn keine Bänder sondern andere Medien zum Einsatz kommen, sollte die Chance auf Automatisierung von Arbeiten nicht vertan werden. Immer dann, wenn das Vergessen einer Tätigkeit zu Datenverlust führen kann, sollte sie automatisiert und mit passenden Einstellungen für die Benachrichtigung eines Administrators versehen werden.

3.2 Backup von Datenbanken

Das Backup von Datenbanken[9] unterscheidet sich vom Backup eines Dateisystems vor allem dadurch, dass vom Dateisystem aus nicht einfach abzusehen ist, wann sich eine Datenbank in einem konsistenten Zustand befindet, der es überhaupt erst ermöglicht ein erfolgreiches Backup durchzuführen. Außerdem ist eine Datenbank in vielerlei Hinsicht deutlich komplexer als eine reguläre Datei in einem Dateisystem. Datenbanken können außerdem gewaltige Ausmaße annehmen und sind alleine deshalb eine Herausforderung für jedes Backup-System.

Neben diesen technischen Problemen gibt es auch personell-organisatorische Herausforderungen, die es zu bewältigen gilt. Oftmals sind Backup-, System-, und Datenbank-Administrator nicht in einer Person vereint und sitzen darüber hinaus auch noch in verschiedenen Abteilungen. Wenn die Datenbank außerhalb des eigenen Unternehmens betrieben wird, sind der direkten Einflussnahme auf erfolgreiches Backup & Restore der eigenen Datenbank eventuell enge Grenzen gesetzt und können nur mittelbar über *SLA*s[10] beeinflusst werden.

In dieser Arbeit fehlt leider der Platz um zu beschreiben, wie die einzelnen Datenbankengines (Etwa *MySQL*[11], Oracle, *DB2*[12] und andere) mit diesen Problemen umgehen und welche Werkzeuge zur Verfügung stehen. Stattdessen werden hier die beiden grundlegenden, universellen Vorgehensweisen für das Backup und Restore von Datenbanken vorgestellt.

[9] [Burleson 1999]
[10] Service Level Agreement
[11] My Structured Query Language
[12] DataBase 2

3.2.1 Cold Backup

Das Cold Backup ist aus Sicht eines Backup-Administrators der Idealfall für das Backup einer Datenbank. Die Datenbankengine wird zunächst komplett gestoppt. So können keine Transaktionen während des Backups verloren gehen. Erst im nächsten Schritt werden dann die Daten der Datenbank gesichert. Teilweise geschieht dies durch die reguläre Sicherung der Dateien, die die Datenbankinhalte im Dateisystem beherbergen. Diese Methode hat den Vorteil, dass außer dem Stoppen der Datenbank keine weiteren Anpassungen des Backup-Systems vorzunehmen sind. Die Dateien, aus denen die Datenbank besteht, werden einfach mit den anderen regulären Dateien gesichert und die Datenbankengine anschließend wieder gestartet.

Der große Nachteil dieser Methode ist, dass die Datenbank während des kompletten Backuplaufs nicht zur Verfügung steht. Im Falle einer sehr großen Datenbank und eines anstehenden Level 0 Backups[13] kann die Zeitspanne erheblich sein und das zur Verfügung stehende Zeitfenster[14] überschreiten.

Darüber hinaus sind viele Datenbanken, vor allem im Bereich Internet, ständig in Benutzung und es gibt überhaupt keine Möglichkeit, die Datenbank auch nur für Minuten herunterzufahren. In diesen Fällen muss ein Backuplauf auch dann erfolgreich durchzuführen sein, wenn die Datenbank produktiv verwendet wird.

3.2.2 Hot Backup

Hier kommt das so genannte Hot Backup ins Spiel. Die zu sichernde Datenbank kann bei diesem Vorgehen weiterhin laufen. In manchen Fällen wird die Benutzung allerdings dennoch stark eingeschränkt. So kann es beispielsweise möglich sein, Informationen aus der Datenbank abzurufen, aber das Löschen oder Erzeugen von Datensätzen ist gesperrt. Das führt prinzipiell zu den gleichen unveränderbaren Verhältnissen wie unter den Bedingungen für ein Cold Backup.

Da dieses Verhalten fast ebenso unerwünscht ist wie ein vollständiger Stop der Datenbankengine, wird oft zu einem kombinierten Verfahren aus Snapshot und Transaktionslog gegriffen.

Die Datenbank kann bei diesem Vorgehen während des Backups in vollem Umfang weiter benutzt werden, sowohl Leseoperationen als auch Schreiboperationen, also Erzeugung und Löschung von Datensätzen, sind möglich. Um diesen Komfort für die Benutzer der Datenbank zu ermöglichen, wird ein ständig fortgeschriebenes Log der aktuellen Operationen benötigt. Dann wird zum Anfangszeitpunkt des Backups der aktuelle Zeitstem-

[13] Vgl. Abschnitt 3.1.1 „Volles Backup“ auf Seite 17
[14] Siehe dazu auch Abschnitt 2.2.2 „Sicherungszeitpunkte und Zyklen“ auf Seite 9

pel des Logs gespeichert und das Backup begonnen. Wenn der Backuplauf durchgeführt wurde, wird erneut der aktuelle Zeitstempel des Logs gelesen. Abschließend werden alle zwischen den beiden Zeitpunkten erfolgten Transaktionen nachträglich auf das Backup angewendet.

Natürlich muss die Backupsoftware bei diesem Vorgehen sehr genaue Kenntnisse des verwendeten Datenbanksystems haben und in der Lage sein, jede Aktion nachzuvollziehen, die in der Datenbank durchgeführt werden kann.

Abhängig davon, welche dieser beiden grundlegenden Vorgehensweisen gewählt wird, ist auch das Restore unterschiedlich durchzuführen.

3.2.3 Restore von Datenbanken

Der eigentliche Zweck eines jeden Backups, nämlich das erfolgreiche Restore im Falle eines eintretenden Datenverlustes, ist bei Datenbanken unterschiedlich vorzunehmen.

Cold Backups, bei denen schlicht die zugrundeliegenden Dateisystemstrukturen gesichert wurden, sind genauso simpel in der Wiederherstellung zu handhaben. Die Datenbank wird, abhängig vom Umfang des eingetretenen Schadens, entweder neu installiert und nicht gestartet oder im laufenden Betrieb gestoppt. Danach wird das Backup an die passende Stelle im Dateisystem zurück gespielt und die Datenbank wird gestartet. Im Anschluss ist die Datenbank mit dem Stand des Backups wieder online und produktiv verwendbar.

Hot Backups bieten im Restore Fall durch die Verwendung eines Transaktionslogs eine erhöhte Flexibilität. So kann gewählt werden, bis zu welchem Punkt genau das Log während des Restores nachvollzogen wird. Waren etwa die letzten drei Transaktionen für eine Fehlfunktion der auf der Datenbank beruhenden Applikation verantwortlich, so werden eben diese Transaktionen nicht wiederhergestellt. Wird das Log der Transaktionen außerdem zusammen mit den Backups der Daten fortlaufend gesichert, ist es prinzipiell möglich, den Stand der Datenbank zu jedem beliebigen Zeitpunkt wiederherzustellen. Der dafür zu treibende Aufwand ist, abhängig von der Kombination aus Daten und Transaktionslog, unter Umständen groß, die Möglichkeit bietet aber ein enormes Maß an Sicherheit für den Betreiber.

Nachteilig an einer solch detaillierten Sicherungslösung ist lediglich das notwendige Zeitfenster, das für eine erfolgreiche Wiederherstellung nötig ist. In punkto Geschwindigkeit ist das Cold Backup deutlich im Vorteil.

Eine Kombination aus beiden Verfahren, in dem die produktive Datenbank laufend per Hot Backup gesichert wird und ein mitlaufender Spiegel für lang andauernde Cold Backups zur Verfügung steht, ist nicht nur aus Sicht eines Backup-Administrators ei-

ne gute Wahl, sondern kann gleichzeitig die Verfügbarkeit des Systems erhöhen. Der Nachteil ist klar in den Kosten einer solchen Lösung zu sehen, die durch den erhöhten Aufwand für Hardware, redundante Vernetzung und Administration entstehen.

Zu diesem Themenkomplex der allgemeinen Datensicherheit folgt nun der Abschnitt 3.3 „Sicherung der Backup-Systeme“.

3.3 Sicherung der Backup-Systeme

Eine zuverlässige Sicherheitskopie ist stets auf einem System zu speichern, welches nach den Grundsätzen[15] der System- und Informationssicherheit eingerichtet wurde und nach denselben Grundsätzen gepflegt wird. Das komplette Thema IT-Grundschutz[16] würde den Rahmen dieser Arbeit mehr als sprengen, daher folgen hier nur drei ausgewählte Unterthemen, welche besonders gut die Bedeutung der Sicherung von Backup-Systemen aufzeigen.

3.3.1 Zugangsberechtigungen

Die physikalische Sicherheit der Backup-Systeme hängt nicht nur von der Prävention vor Naturkatastrophen[17] ab. In den meisten Gegenden der Welt sind sehr viel häufiger unbefugte Personen ein Risiko für die Sicherheit der Backups. Hierbei kann es sich um schlichten Diebstahl eines Mediums, wie etwa ein Magnetband oder eine Festplatte, handeln, oder um Unachtsamkeit eines ungeschulten Mitarbeiters im Umgang mit der Hardware.

Daher ist der Zugang zu den Backup-Systemen mit geeigneten Mitteln zu beschränken[18].

Der Standort der Systeme sollte um so besser gegen den Zugang von Unberechtigten abgesichert sein, je wertvoller oder vertraulicher die gesicherten Daten sind.

Denkbar sind eine Vielzahl von Lösungen. So kann ein privates Backup eines Einzeplatz-Computers schon durch das Lagern der benutzten Festplatte in einem verschlossenem Schrank oder Tresor vor Diebstahl schützen. Im Bereich von Unternehmens-Systemen oder gar Rechenzentren sind dagegen Server hinter stets verschlossenen Türen die Regel. Der Zugang wird oftmals über Magnetkarten gesteuert und besonders sensible Systeme

[15] [BSI 2008]
[16] [BSI 2005]
[17] Siehe auch die Abschnitte 2.1.2 „Gefahrenarten für Daten“ ab Seite 6 und 3.3.3 „Offsite Backup“ ab Seite 29
[18] [Anderson 2008]

sind innerhalb des Server-Raums nochmals durch eine spezielle Sicherheitsschleuse geschützt. Hier werden dann auch teilweise biometrische Systeme wie Fingerabdruck- oder Iris-Scanner eingesetzt.

Daneben ist es ebenfalls wichtig darauf zu achten, dass der Zugang von berechtigten Personen auf angemessene Art und Weise überwacht und protokolliert wird. Hier eignet sich ein Log der verwendeten Magnetkarten kombiniert mit einer Videoüberwachung. Auf diese Art und Weise kann ein Datenleck oder ein Verlust von Sicherheitskopien zeitlich mit dem Zugang von Personen abgeglichen werden.

Wenn der physikalische Zugang gesichert ist, gilt es den logischen Zugriff auf die Daten über das Netzwerk zu betrachten.

3.3.2 Zugriffsberechtigungen

Im Gegensatz zu physikalischen Werten sind Daten auch ohne direkten Zugang zu ihrem Aufenthaltsort zu erreichen. So sollten die Backups niemals ungeschützt und frei im Netz verfügbar sein. Unachtsamkeit oder Vorsatz könnten sonst schnell die beste Sicherheitskopie löschen.

Dazu kommt das Risiko einer Kompromittierung. Backupserver benötigen für die Durchführung des Backups, etwa um Datenbanksysteme vorübergehend zu stoppen[19], tiefgreifende Berechtigungen auf den zu sichernden Maschinen. Gelingt es einem Angreifer einen Backupserver zu übernehmen, so ist es ihm nicht nur möglich auf die Daten in den Sicherheitskopien zuzugreifen, sondern er erlangt auch Zugang zu allen zu sichernden Systemen.

Eine solche Kaskade an kompromittierten Systemen ist der Albtraum eines jeden IT-Sicherheitsbeauftragten. Nicht genug damit, dass das Backup so verloren sein kann; wenn gleichzeitig die Produktivsysteme zerstört werden, könnte das Unternehmen in einen existenzbedrohlichen Zustand gelangen.

Die Bedrohung für Privatanwender spielt sich, wenn auch auf einer kleineren Skala, auf die gleiche Art und Weise ab. Wird etwa das Backup eines privaten Computers auf eine angeschlossene externe Festplatte gesichert und wird dann der Computer von einem Angreifer übernommen, so sind sowohl Originaldaten als auch Sicherheitskopien gefährdet.

Um ein solches Szenario zu verhindern, sind die Backup-Systeme also mindestens eben so gut gegen unbefugten Zugriff zu sichern wie die zu sichernden Systeme.

[19] Vgl. Abschnitt 3.2.1 „Cold Backup“ ab Seite 25

Die Möglichkeiten dazu beginnen bei der Sicherheit des Netzwerkes. Möglichst mehrstufige Firewalls sollten das interne Netz vom Internet und anderen als unsicher betrachteten Netzen trennen und nur explizit zugelassene Verbindungen erlauben. Ein rollenbasiertes Benutzersystem ist ebenfalls essentiell. So sollten User nur Zugriff auf die Backups ihrer eigenen Daten haben, nicht aber auf die anderer User. Die gleiche Logik diktiert, dass die Backups von Datenbanken nicht global und vollständig zur Verfügung stehen, sondern nur den Datenbank-Administratoren. Eventuell ist es auch möglich, Abschnitte der Datenbank den zuständigen Abteilungen oder Benutzern zur Verfügung zu stellen. Diese Lösung bietet großen Komfort, benötigt aber exzellente Vorbereitung der Sicherungen, die bei Änderungen am Datenbanklayout oder an den zugrunde liegenden Geschäftsprozessen auch fortlaufend angepasst werden müssen.

3.3.3 Offsite Backup

Unter Offsite Backup wird die Sicherung der Backups an einem von den Originaldaten entferntem Ort, mindestens in einem anderen Gebäude, verstanden. Im privaten Umfeld kann bereits eine bei Freunden gelagerte externe Festplatte mit den wichtigsten Daten ein probates Offsite Backup darstellen. Im Enterprise Bereich ist es dagegen mittlerweile keine Seltenheit mehr, komplette Rechenzentren redundant auszulegen und global zu verteilen.

Diese Art der Offsite Aufbewahrung von Sicherheitskopien hat eine Reihe von Vorteilen gegenüber dem Onsite Backup vor Ort:

- Wenn die Backups eines privaten Einzelplatzsystems Offsite vorgehalten werden, ist ein Verlust der Daten durch Diebstahl deutlich weniger wahrscheinlich, als bei einem Backupmedium das auf dem Schreibtisch direkt neben den Quelldaten aufbewahrt wird. Der selbe Vorteil gilt natürlich auch für größere Umgebungen und dann sogar in weiterem Umfang.

- Schädliche Umwelteinflüsse haben, eine entsprechende Entfernung vorausgesetzt, keinen Einfluss auf die Backups. Das gilt bei großen Entfernungen selbst für verheerende Umweltkatastrophen und sonstige Desaster.

- Auch der Zugang durch betriebsfremde Personen kann in einem Gebäude, das speziell für die Sicherung von Daten ausgelegt ist, deutlich leichter eingeschränkt werden, als das in einem regulären Bürogebäude mit Publikumsverkehr möglich ist.

- Wenn das Offsite Backup von einem Dienstleister angeboten wird, entfällt der Aufwand für die Administration und die Pflege des Backups durch eigenes Personal.

Diese Vorteile werden allerdings auch mit einer Reihe von Nachteilen erkauft:

- Die geografische Entfernung der Sicherheitskopien ändert nichts an der Notwendigkeit von Zugriffsberechtigungen. Es ist aus der Sicht eines Angreifers im Netzwerk prinzipiell gleichgültig, ob das angegriffene System im gleichen Gebäude oder auf der anderen Seite der Welt steht.

- Für weit entfernte Systeme stehen eventuell nicht die gleichen hochwertigen Netzwerkverbindungen zur Verfügung, die im eigenen Gebäude nutzbar sind. So kann sich die benötigte Dauer des Backup-Fensters[20] signifikant erhöhen.

- Möglicherweise wird es aufgrund zu geringer Bandbreite nötig, die Daten auf physikalischem Wege zu ihrer Offsite Stätte zu verbringen. Hier wartet dann eine völlig neue Problemstellung auf den für das Backup zuständigen Administrator. Der Transport muss, je nach dem Wert der Daten, so sicher wie ein Geldtransport abgewickelt werden. Ein Verlust der Daten kann, neben dem Verlust selber, auch zu negativen Auswirkungen auf das Image der Organisation in der Öffentlichkeit führen[21]. Gleichzeitig ist auf die Unversehrtheit der Datenträger zu achten. Dieser Transport ist dann regelmäßig, abhängig vom eingesetzten Rotationsschema zu organisieren.

- Bei einem von einem Dienstleister angebotenen Offsite Backup muss diesem ein gehöriger Vertrauensvorschuss gewährt werden. Das gilt um so mehr, je wertvoller und vertraulicher die gesicherten Daten sind. Außerdem ist es, mindestens stichprobenartig, erforderlich, die durch den Dienstleister verwalteten Backup Sätze zu testen[22].

Im Falle des oben erwähnten Transports von Medien gilt es, ein besonders geeignetes Medium für den Transport auszuwählen und passend in den eingesetzten Rotationsprozess einzubinden. Hier bietet sich das *D2D2T*[23]-Verfahren[24] an, um alle Vorteile der beschriebenen Prinzipien auszunutzen[25]. Eine schematische Darstellung ist auf der Abbildung 3.3 auf der Seite 31 zu sehen.

Das Backup wird hierbei zunächst auf Festplatten geschrieben. Hier wird die große Geschwindigkeit der Festplatten gegenüber den Magnetbändern ausgenutzt. Ist das Backup dann einmal gesichert wird es auf Bänder kopiert. Diese sind für einen physischen Transport wesentlich besser geeignet als Festplatten, da sie weniger anfällig für mechanische Fehler sind. Vor und nach dem Transport werden die Bänder per Barcode oder einem

[20] Vgl. Abschnitt 2.2.2 „Sicherungszeitpunkte und Zyklen“ ab Seite 9
[21] [Schneier 2005]
[22] Vgl. Abschnitt 2.2.4 „Verifikation und Integrität“ ab Seite 11
[23] Disk to Disk to Tape
[24] [Exabyte 2005]
[25] Die in Abschnitt 4.3.2 ab Seite 54 beschriebene Software Amanda ist ein gutes Beispiel

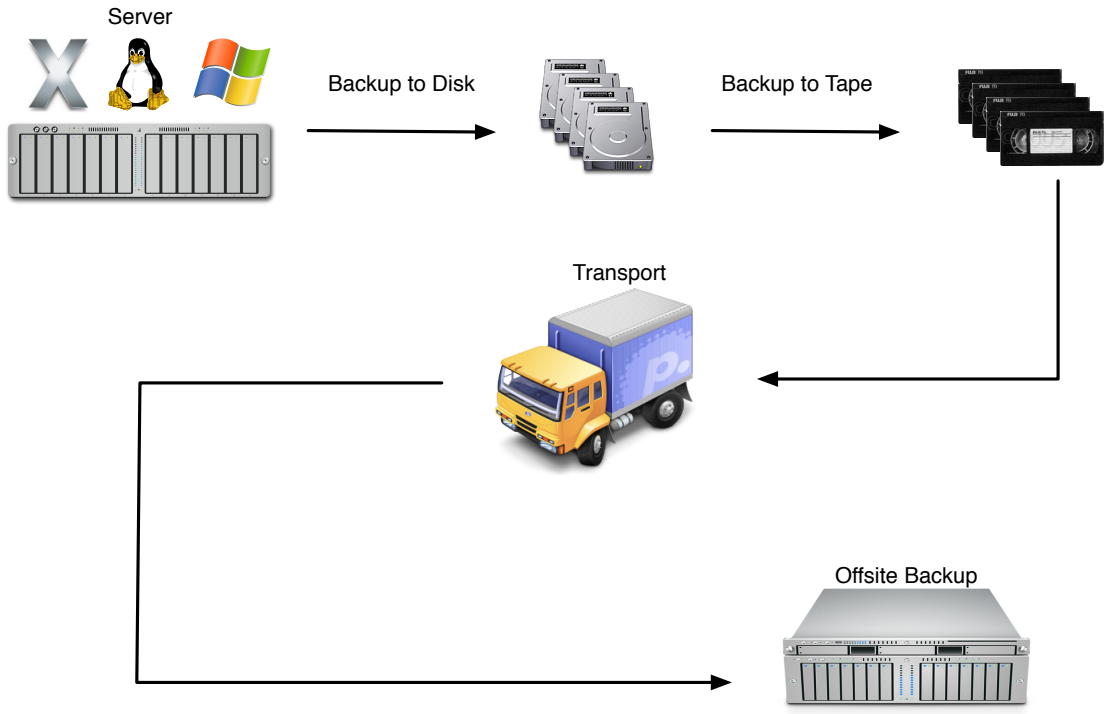

Abbildung 3.3: Offsite Backup mit D2D2T

anderen Verfahren identifizierbar gemacht. Anhand dieser ID kann dann der erfolgreiche Eingang überprüft werden. Die ID kann außerdem genutzt werden, um bestimmte Bänder aus dem Offsite Backup zurück zu ordern, um einen Restore durchführen oder auch testen zu können. Zuletzt kann die ID benutzt werden, um als Teil eines kryptographischen Schlüssels zu dienen. Auf diese Art und Weise kann pro Band ein einmaliger Schlüssel benutzt werden, was die Datensicherheit weiter erhöht.

Zu dem Thema Verschlüsselung von Backups folgt nun der Abschnitt 3.3.4.

3.3.4 Verschlüsselte Backups

Um die Sicherheit der Daten zu verbessern, ist es möglich, die Backups zu verschlüsseln bevor sie auf die Reise zu ihrer Offsite Stätte gehen. Hierbei verbessert sich sowohl die Sicherheit der Übertragung über ein Netzwerk, als auch die des physischen Transports.

Der sehr vereinfachte Aufbau des Prinzips der asymmetrischen Verschlüsselung ist auf der Abbildung 3.4 auf der Seite 32 zu sehen. Das Backup wird mit einem öffentlichen Schlüssel verschlüsselt und lässt sich dann ausschließlich mit dem korrespondierendem privaten Schlüssel wieder entschlüsseln. Dieses Verfahren kann auf unterschiedlichen Ebe-

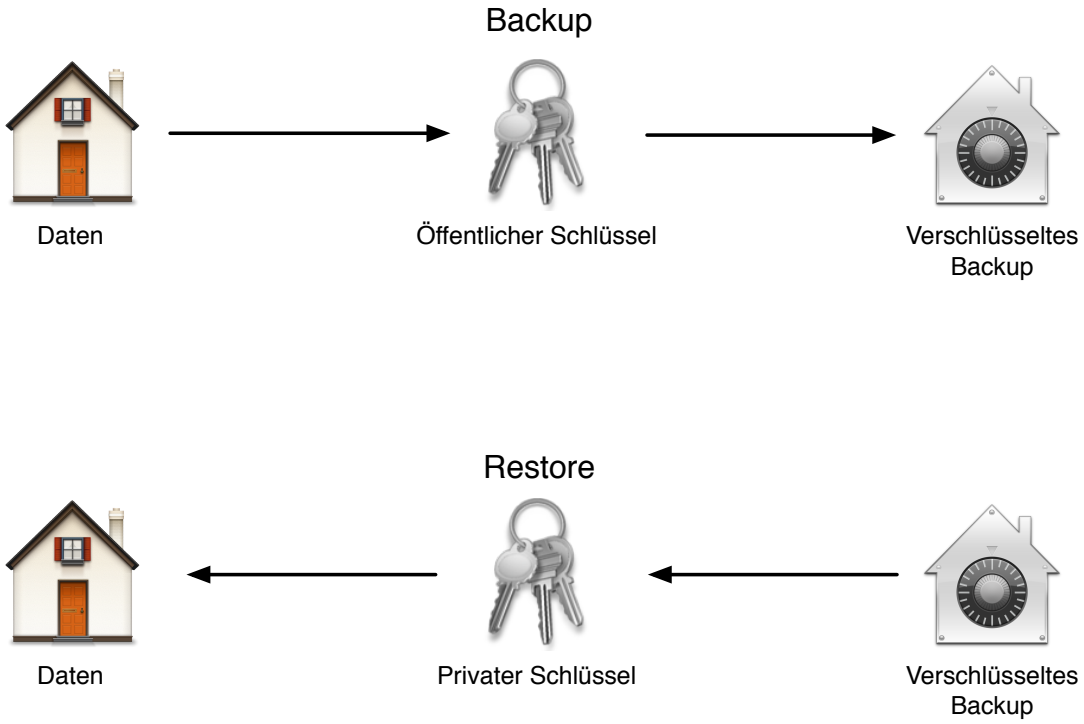

Abbildung 3.4: Backup mit asymetrischer Verschlüsselung

nen eingesetzt werden.

- Das einfachste Verschlüsselungskonzept basiert auf der Verschlüsselung durch das eingesetzte Backup-System. Entweder wird direkt vom Quellsystem aus verschlüsselt, oder die empfangenen Daten werden auf dem Backup-Server direkt nach dem Eingang verschlüsselt. Diese Methode wird von so gut wie jeder aktuellen Backup-Software angeboten, kostet aber eventuell Performance, die für die Verschlüsselung zusätzlich zur Verfügung stehen muss.

  Nachdem die Daten verschlüsselt und gesichert sind, werden sie über ein Netzwerk oder per physischem Transport zu ihrer Offsite Lagerstätte verbracht.

- Eine andere Möglichkeit die Backups sicher über ein Netzwerk zu übertragen, ist die Verschlüsselung des Netzwerkverkehrs. Bei diesem Vorgehen wird nur der Transport der Backupdaten verschlüsselt. Onsite- und Offsite-Backups sind weiterhin unverschlüsselt und müssen entsprechend dem, in den Abschnitten 3.3.1 und 3.3.2 ab Seite 27 beschriebenen Vorgehen, abgesichert werden. Hierfür sind auf den zu sichernden Systemen jedoch keinerlei Einbußen in der Performance zu verzeichnen.

- Eine sehr moderne Art der Backupverschlüsselung ist der Einsatz von Verschlüsselung direkt auf einem Magnetband[26]. Hier werden die Daten in Echtzeit verschlüsselt, während sie auf das Band geschrieben werden. Weder die zu sichernden Systeme, noch der Backup-Server müssen bei dieser Methode Abstriche an der Performance machen. Die komplette Verschlüsselung wird durch das Bandlaufwerk durchgeführt.

  Sofort nach dem Fertigstellen der Sicherung sind die Bänder fertig verschlüsselt und können ohne weitere Verzögerungen transportiert werden.

Backup-Software, die die oben beschriebenen Verschlüsselungsverfahren einsetzt, bietet auch oft die Möglichkeit des Schlüsselmanagements an. Teilweise kann durch das Schlüsselmanagement für jedes Medium ein eigenes Schlüsselpaar erzeugt werden. Diese Schlüsselphalanx von Hand zu organisieren ist eine sehr anspruchsvolle Aufgabe. Sollte diese aber an die Backup-Software ausgelagert werden, so ist penibel darauf zu achten, auch dieses System für den Fall der Fälle zu sichern. Da es sich hier um äußerst vertrauliche Daten handelt, ist auch die größtmögliche Sorgfalt bei der Sicherung der Systeme geboten.

Der größte Nachteil verschlüsselter Sicherheitskopien ist die Erzeugung eines neuen *SPOF*[27]. Geht der private Schlüssel aus irgendeinem Grund verloren, so sind alle Backups die mit dem korrespondierenden öffentlichen Schlüssel erstellt wurden, verloren. Gerät der Schlüssel in falsche Hände, sind alle bisher erstellten Backups als nicht mehr gesichert anzusehen. Erst nachdem alle Backups vollständig entschlüsselt und anschließend mit einem neuen Schlüssel wieder verschlüsselt wurden, ist die ursprüngliche Sicherheit wiederhergestellt.

Damit die Backups nicht durch den Verlust des einzigen Schlüssels unbrauchbar werden, sollte es ein oder mehrere Backups des Schlüssels geben. Diese sind dann entweder gar nicht, oder aber nicht mit sich selbst zu verschlüsseln, da ja diese Sicherheitskopie gerade für den Fall des Verlusts des Schlüssels erstellt wurde. Gleichzeitig ist es wichtig, auch die Backups des Schlüssels nicht in falsche Hände gelangen zu lassen. Hier sind klassische Tresore oder Bankschliessfächer vielleicht die einzige Möglichkeit, eine einigermaßen ausgewogene Balance zwischen Sicherheit und Komfort zu finden.

Von diesen Überlegungen bisher unbeachtet, existiert natürlich auch auf der Seite des Anwenders eines Systems die Möglichkeit seine Daten zu verschlüsseln. Bei diesem, auf Abbildung 3.5 auf der Seite 34 skizzierten Szenario, wird es für Backup-Software schwierig bis unmöglich eine inkrementelle Sicherung[28] zu erstellen. Wenn sich die Dateien innerhalb eines verschlüsselten Containers befinden, ist es der Software nicht mehr möglich, jede Datei einzeln auf Veränderungen zu überprüfen. So muss innerhalb des Con-

[26] [Dorion 2008]
[27] Single Point of Failure
[28] Vgl. Abschnitt 3.1.2 ab Seite 18

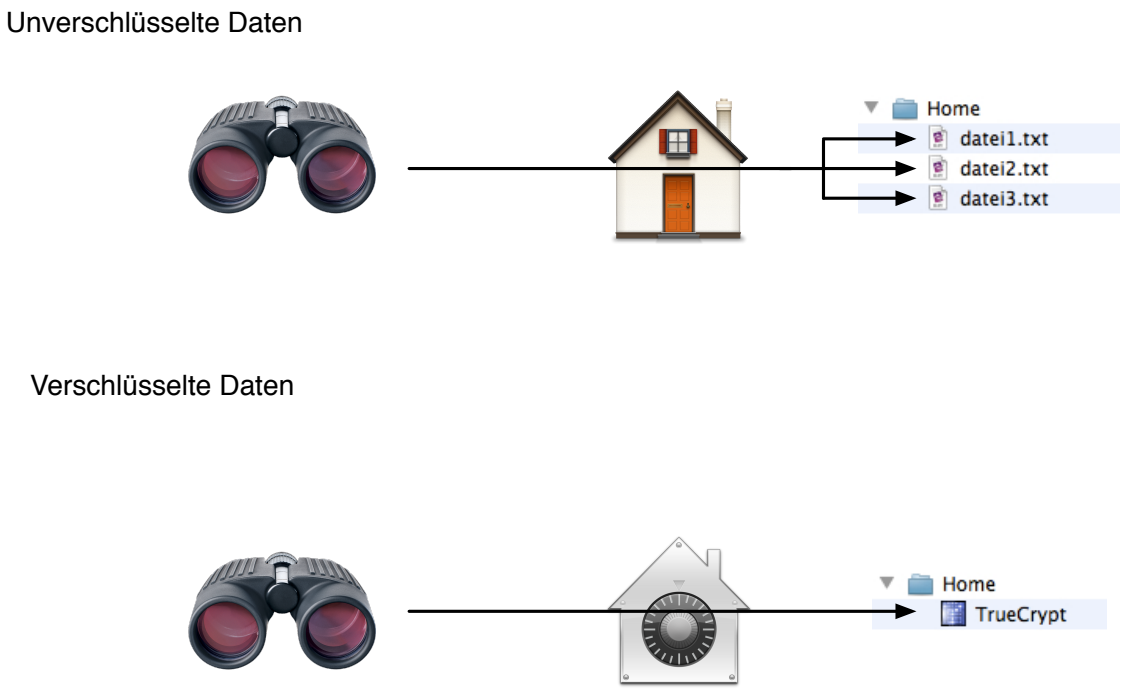

Abbildung 3.5: Verschlüsselte Daten verhindern inkrementelle Backups

tainers nur ein einziges Zeichen innerhalb einer einzigen Datei geändert werden, um ein vollständiges Backup des kompletten Containers nötig zu machen.

Die Verschlüsselung von Backups ist aus den dargelegten Gründen stets sehr intensiv zu betrachten. Der Ablauf eines einmal eingerichteten Backups wird durch die Einführung von Verschlüsselungstechnologien erheblichen Anpassungen unterworfen. Es ist wenig zielführend, wenn die gesicherten Daten zwar vor jedem nicht autorisierten Zugriff geschützt sind, dafür aber auch das Restore unnötig erschwert wird. Die größte Gefahr für die Backups stellt aber der Verlust der kryptographischen Schlüssel für das Restore dar. Sind diese verloren, sind alle damit verschlüsselten Backups nicht mehr wiederherstellbar.

4 Backup-Praxis

In diesem Kapitel werden einige praktische Anwendungen der bisher eingeführten Überlegungen vorgestellt. Dabei entwickeln sich die vorgestellten Lösungen von einem Einzelplatzsystem, über eine Arbeitsgruppe, bis hin zu Enterprise-Lösungen. Die angewendeten Prinzipien und Grundsätze sind dabei stets identisch, nur die Skalierung der eingesetzten Lösung steigert sich entsprechend der Größe der zu sichernden Systeme und der Menge der zu sichernden Daten. Die Auswahl der Lösungen ist mitnichten repräsentativ, sondern beruht auf den praktischen Erfahrungen des Autors.

4.1 Einzelplatz

Einzelplatzsysteme sind heute in einem Großteil der privaten Haushalte vertreten und speichern mehr Daten als jemals zuvor. Von der privaten Korrespondenz über digitale Fotos bis hin zu kompletten Mediensammlungen aus tausenden von Musik- und Film-Dateien werden Festplatten gefüllt.

Immer mehr Selbständige und Angestellte arbeiten außerdem von zuhause aus. Gerade im Bereich von großformatigen Bilddateien, wie sie Grafiker oder Fotografen erstellen, können auch hier schnell viele Gigabyte an Daten anfallen.

Alle diese Daten müssen entsprechend den Überlegungen aus den voran gegangenen Kapiteln gesichert werden und im Falle eines Datenverlustes wiederherstellbar sein. Hierfür bietet der Markt eine unüberschaubare Fülle an Softwareprodukten an. In diesem Abschnitt wird aber lediglich auf zwei Lösungen eingegangen, die ohne Zusatzsoftware direkt vom Betriebssystem, hier Microsoft Windows respektive Apple Mac OS X, mitgeliefert werden und simpel zu bedienen sind.

4.1.1 Backup und Restore Center

Das Backup und Restore Center wurde von Microsoft mit Windows Vista eingeführt, um das rasch alternde und hauptsächlich über eine Kommandozeile einzusetzende NT-

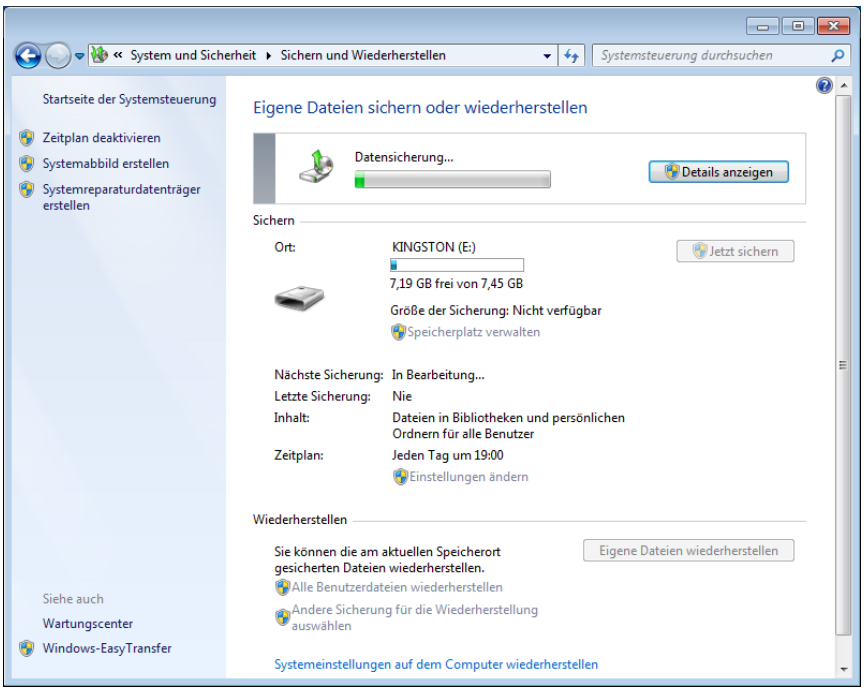

Abbildung 4.1: Das Windows 7 Backup und Restore Center

Backup zu ersetzen. In dieser Arbeit wird die mit Windows 7 Professional ausgelieferte Version behandelt[1].

Mit dem Backup und Restore Center hat der Anwender die Möglichkeit, sowohl ein komplettes, Image-basiertes Backup seines Systems zu erstellen, als auch eine Sicherung seiner eigenen Daten anzufertigen[2].

Beide Sicherungsarten werden über das in Abbildung 4.1 zu sehende Fenster der Systemsteuerung konfiguriert. Für beide kann ein Zeitplan erstellt werden, um die Backups automatisiert durchführen zu lassen. Der kleinste einstellbare Sicherungszyklus ist, siehe hierzu Abbildung 4.2 auf Seite 37, einmal pro Tag zu einer bestimmten Uhrzeit.

Image-basiertes Backup

Das Backup des kompletten Systems erstellt zunächst ein vollständiges Abbild des Systems. Wird als Backup-Medium eine externe Festplatte verwendet, erfolgen die darauf folgenden Sicherungen inkrementell auf Blockebene. Das bedeutet, dass auch Unterschiede innerhalb von Dateien durch das Backup-System erkannt werden und so nur die erfolgten Änderungen erneut gespeichert werden müssen. Dieses Verfahren ist sehr effi-

[1] [Bott et al. 2009]
[2] [Fok 2007]

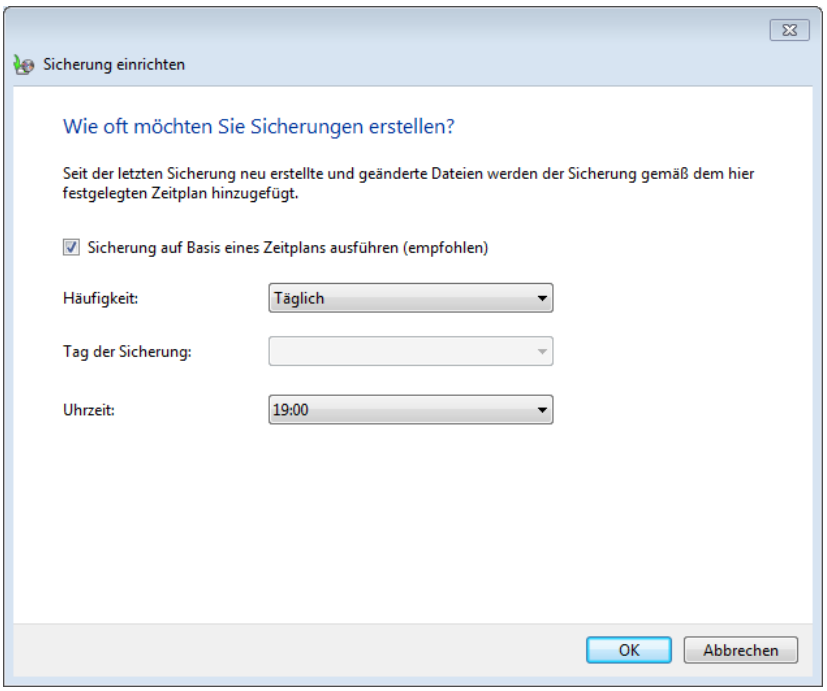

Abbildung 4.2: Windows 7 Backup Zeitplan

zient, wenn der Anwender viel mit großen, sich häufig ändernden Dateien arbeitet. Diese Technik funktioniert nicht, wenn das Backup-Medium ein Netzlaufwerk ist. In diesem Falle können nur vollständige Sicherungen mit einem entsprechend großem Platzverbrauch erstellt werden.

Sollte ein Festplattenfehler das komplette System unbrauchbar machen, ist es möglich, von einer CD oder DVD zu starten und eines der gesicherten Images wiederherzustellen.

Um nur einzelne Dateien aus einem Image wiederherzustellen, ist das Image zunächst per Doppelklick in das System einzubinden. Im Anschluss kann das Image, wie ein reguläres Laufwerk, durchsucht und die gefundene Datei zurück auf die Festplatte kopiert werden. Diese Wiederherstellung von einzelnen Dateien gelingt aber komfortabler, wenn die zweite angebotene Methode der Datensicherung verwendet wird.

Sicherung der eigenen Dateien

Diese andere Methode der Datensicherung, die der Sicherung von eigenen Dateien ohne das Betriebsystem, erfolgt auf eine andere Art. Hier werden von Windows ausgewählte Dateien und Verzeichnisse auf Basis von Schattenkopien in ein komprimiertes Archiv auf ein beliebiges Medium gesichert. Die Technologie der Schattenkopien wird übli-

cherweise *VSS*[3] abgekürzt und existiert seit Windows XP als Eigenschaft des *NTFS*[4]-Dateisystems. Der VSS überwacht fortlaufend die Aktivitäten des Dateisystems und ist in der Lage, konsistente Snapshots[5] des aktuellen Standes zu erstellen, ohne dabei auf Schreiboperationen oder geöffnete Dateien Rücksicht nehmen zu müssen[6]. Dadurch wird ein Anwender während eines Backuplaufs in seiner Arbeit nicht beeinträchtigt. Sollten während eines Backups Fehler auftreten, wird der Anwender über eine ausführliche Windows-Benachrichtigung informiert.

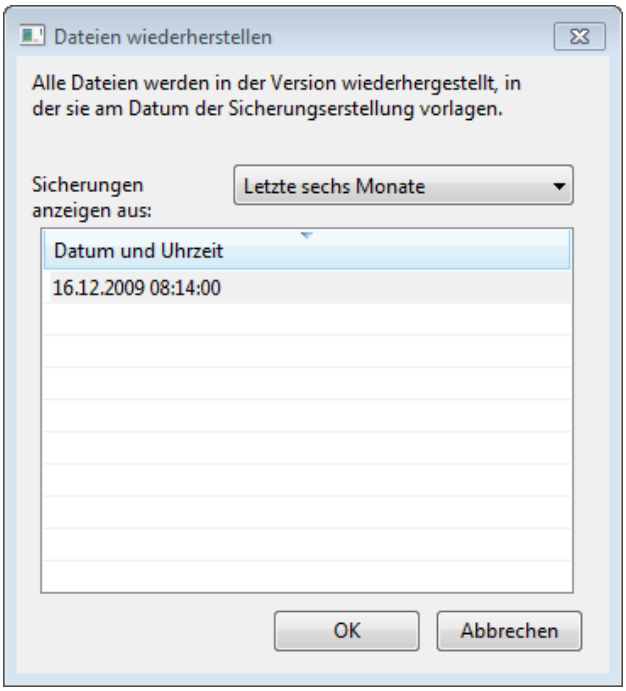

Abbildung 4.3: Windows 7 Restore Dialog

Die zu sichernden Dateien können automatisch von Windows ausgewählt werden. Dann werden unter anderem alle Anwender-Ordner, gespeicherte Suchvorgänge, der Inhalt des Desktops und gespeicherte Kontaktdaten gesichert. Anwender können eine eigene Auswahl treffen und dabei auch komplette Laufwerke berücksichtigen lassen, dabei werden jedoch installierte Programme und Systemkomponenten nicht berücksichtigt. Für diese Art von Backup ist die oben beschriebene Image-basierte Methode vorgesehen. Dateien und Verzeichnisse die nicht gesichert werden sollen, etwa große temporäre Dateien, lassen sich über das Backup und Restore Center auswählen und werden dann zukünftig ignoriert.

Die Wiederherstellung einer vorhergehenden Version einer einzelnen Datei aus ihren Schattenkopien kann dann per simplem Rechtsklick auf die Datei erfolgen. Für größere

[3] Volume Shadow Copy Service
[4] New Technology Filesystem
[5] Vgl. dazu Abschnitt 3.2.2 „Hot Backup“ auf Seite 25
[6] [Microsoft Corporation 2003]

Restore-Vorgänge mit vielen Verzeichnissen und Dateien ist dagegen eher der auf Abbildung 4.3 auf Seite 38 zu sehende Restore-Dialog geeignet. Nach der Auswahl eines der verfügbaren Backups kann der Anwender, über ein Standard Dateiauswahl-Fenster, Dateien und Ordner auswählen und sie an ihrem Ursprungsort wiederherstellen lassen. Alternativ ist auch ein beliebiger anderer Ort auswählbar.

Die in Windows 7 integrierten Backup und Restore Werkzeuge sind für einen Anwender eines Einzelplatzsystems sicherlich als ausreichend zu betrachten. Allerdings ist die Komplexität der Konfiguration nicht zu unterschätzen. Ein Anwender muss wissen, welche der beiden Backup-Arten welchem Zweck dient und sie gegebenenfalls seinen Wünschen anpassen. Diese Komplexität ermöglicht aber auch einen sehr flexiblen Einsatz der Windows 7 Backup-Technologie. Wünschenswert wäre dennoch die Möglichkeit, auch inkrementelle Backups über das Netzwerk zu fahren, sowie eine feiner aufgelöste Steuerung des Zeitplans für automatische Backups.

4.1.2 Time Machine

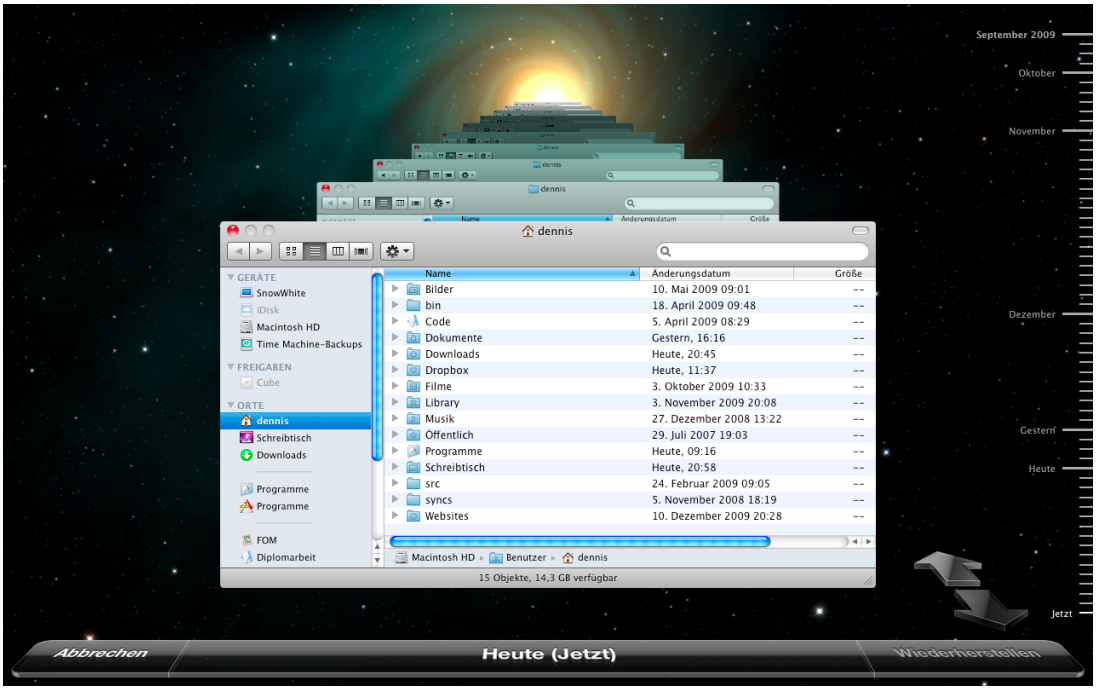

Abbildung 4.4: Time Machine Restore Interface

Seit der Version 10.5 von Mac OS X ist ein Backup und Restore Programm namens Time Machine Bestandteil des Betriebssystems[7]. Time Machine ist ab Werk so konfiguriert, dass der Anwender nach dem Anschluss einer externen Festplatte automatisch gefragt

[7] [Siracusa 2007]

wird, ob er diese für ein Backup verwenden möchte. Danach ist das System bereits in weiten Teilen fertig konfiguriert. Alternativ kann das Backup auch über das Netzwerk auf einen anderen unter Mac OS X laufenden Computer oder auf eine von Apple vertriebene Backup-Appliance namens Time Capsule erfolgen.

Fortan wird jede Stunde ein Backup des Systems angefertigt. Das nicht konfigurierbare Rotationsschema von Time Machine behält alle stündlichen Backups der letzen 24 Stunden, tägliche Backups des letzten Monats und wöchentliche Backups für die vorhergehenden Monate. Sollten während des Backups Fehler auftreten, wird der Anwender durch eine Fehlermeldung informiert. Außerdem schreibt Time Machine seine Aktionen kontinuierlich in die systemweite Logdatei.

Dabei ist nur ein einmaliges, initiales Level 0 Backup nötig[8]. Die nachfolgenden Backups sichern dann nur noch neu hinzugekommene oder geänderte Dateien und entfernen nicht mehr im Original vorhandene Dateien aus dem Backup.

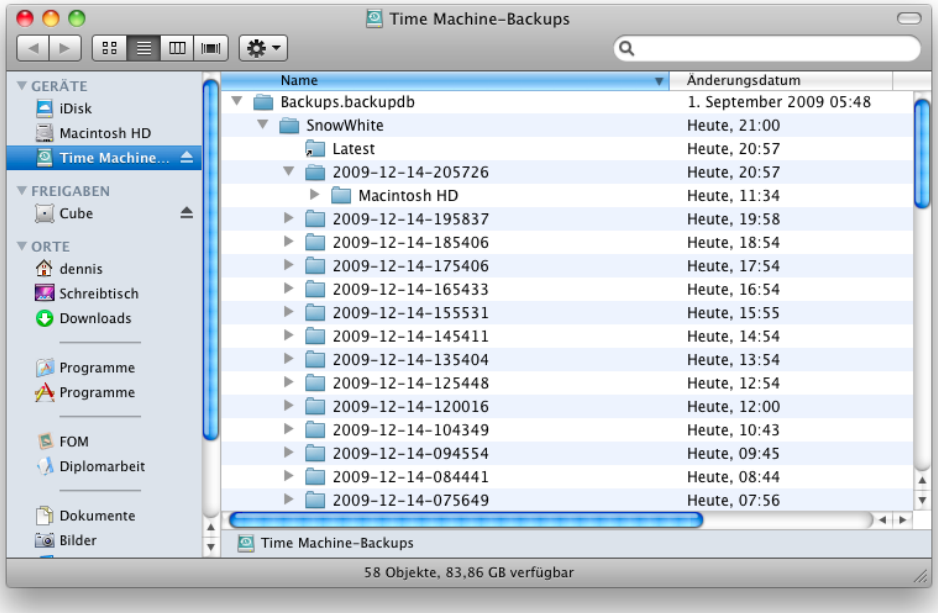

Abbildung 4.5: Ordnerstruktur eines Time Machine Backups

So entsteht eine wie auf Abbildung 4.5 zu sehende Verzeichnisstruktur. Unterhalb eines zentralen Verzeichnisses liegen die Backups in Ordnern, die mit dem Datum ihrer Sicherung in dem Format `YYYY-MM-DD-HHMMSS` benannt sind. Der Ordner `Latest` verweist dabei stets auf das aktuelle Backup.

Dank dieser einfachen Struktur steht es dem Anwender frei, ob er lieber auf traditionelle Weise durch seine Backups navigieren möchte, oder mit dem auf Abbildung 4.4 auf Seite

[8] Vgl. Abschnitt 3.1.1 „Volles Backup“ auf Seite 17

39 zu sehendem Time Machine Interface arbeitet. Letzteres wird über einen prominent platzierten Button aufgerufen und kann neben dem Finder auch mit dem integrierten Adressbuch, der Foto-Verwaltung und einigen anderen Anwendungen direkt zusammen arbeiten. Über die Zeitleiste auf der rechten Seite wird das gewünschte Backup ausgewählt, ein Klick auf *Wiederherstellen* bringt die ausgewählten Daten zurück.

Im Falle eines kompletten Ausfalls des Systems, etwa einem Festplattendefekt, hat der Anwender die Möglichkeit, ein beliebiges Backup während der Installation des Betriebssystems auf eine neue Festplatte einzuspielen. Da es sich hier nicht um ein Image handelt, gelingt die Wiederherstellung auch auf eine größere Festplatte als die, die ursprünglich in dem System ihren Dienst versehen hat.

Hinter der einfachen Anwendung von Time Machine steckt eine Erweiterung des aus der Unix-Welt bekannten Konzepts der Hard Links[9]. Ein Hard Link ist grob mit einem Alias unter Windows zu vergleichen. Bei Hard Links sind jedoch der Name einer Datei und der eigentliche Inhalt verschiedene Dinge. Außerdem kann der Inhalt einer Datei mehrere Namen haben, die auch an unterschiedlichen Stellen im Dateisystem stehen können. Der eigentliche Dateiinhalt ist dabei über jeden Namen gleich ansprechbar. Gelöscht wird die Datei erst dann, wenn alle Namen, die auch Zeiger genannt werden, entfernt wurden.

Listing 4.1: Ansicht einer Original-Datei

```
1 ~/Desktop > ls -lah imac.pdf
2 -rw-r--r--@ 1 dennis  dennis    22M  7 Nov  2008 imac.pdf
```

Die Anzahl der auf einen Dateiinhalt verweisenden Namen kann, wie in Listing 4.1 zu sehen, auf der Kommandozeile mit dem `ls`-Befehl angezeigt werden. Die Anzahl steht nach den Zugriffsrechten und vor dem Eigentümer und beträgt in Listing 4.1 genau eins. Mit dem Entfernen dieses einen Zeigers ist also auch der 22 Megabyte umfassende Inhalt der Datei von der Festplatte gelöscht. Im Gegensatz dazu zeigt Listing 4.2 die gleiche Datei in einem mit Time Machine erstellten Backup.

Listing 4.2: Ansicht einer Datei auf einem Time Machine Volume

```
1 /Time Machine-Backups/.../Desktop > ls -lah imac.pdf
2 -rw-r--r--@ 54 dennis  dennis    22M  7 Nov  2008 imac.pdf
```

Hier beträgt die Anzahl der auf den gleichen Inhalt verweisenden Zeiger 54. Die Datei ist also in 54 unterschiedlichen Backups vorhanden und kann dort jeweils wie das Original behandelt, also beispielsweise wiederhergestellt werden. Die 22 Megabyte werden aber trotz der 54 Hard Links nur ein einziges Mal belegt. Der Platzverbrauch der Hard Links ist dagegen verschwindend gering. Generell benötigen diese nur den Platz für die Anzahl der Buchstaben in ihrem Namen und einen Hinweis auf die Nutzdaten im Dateisystem.

[9] [Schmidt 2006]

Im klassischen Unix können Hard Links nur auf Dateien zeigen. Mit der Einführung von Time Machine wurde diese Beschränkung in Mac OS X aufgehoben. So können Verzeichnisse, deren Inhalt sich zwischen zwei Backups nicht geändert hat, mit einem einzigen Hard Link in das neue Backup-Set eingebaut werden. Im Falle eines unveränderten `/System`-Verzeichnisses, spart das die Erstellung von 58.000 Verzeichnissen und darin enthaltenen 150.000 Hard Links.

Da Time Machine Backups, im Gegensatz zu dem Image Backup von Windows 7[10], auch für das gesamte System auf Dateiebene arbeiten, ist es im Hinblick auf sehr große Dateien, die sich häufig ändern, eher ineffektiv. Große Mailarchive oder die Dateien von Virtualisierungssoftware müssen daher nach jeder Änderung komplett erneut gesichert werden. Zwar existiert die Möglichkeit, bestimmte Ordner oder Dateien vom Backup auszuschließen, dann muss jedoch der Verlust dieser Daten hingenommen werden, oder sie müssen mit einem anderen Verfahren gesichert werden.

Der große Vorteil von Time Machine besteht in der sehr einfachen Konfiguration. Für den Großteil der Anwender eines Einzelplatzsystems ist es völlig ausreichend eine externe Festplatte anzuschließen und die Verwendung durch Time Machine zu bestätigen. Ein Restore ist, aufgrund der simplen Speicherung im Dateisystem ohne Verwendung eines proprietären Backup-Formats, möglich, ohne dabei mit Image-Dateien oder speziellen Restore-Prozeduren hantieren zu müssen.

Verbesserungswürdig ist die Behandlung von großen, sich häufig ändernden Dateien. Außerdem wäre es sehr wünschenswert, die Backups über das Netzwerk auch auf nicht-Apple eigene Geräte zu ermöglichen.

Weder die Lösung von Microsoft, noch die von Apple löst das Problem der Offsite-Backups[11]. Ohne die händische Rotation von Festplatten oder den Einsatz einer zusätzlichen Backup-Lösung sind die Daten nicht vor Diebstahl, Feuer, Wassereinbruch oder ähnlichem geschützt. Über eine Verschlüsselungsmöglichkeit verfügen beide ebenfalls nicht.

4.2 Arbeitsgruppe

Kleine Arbeitsgruppen sind eine sehr heterogene Ansammlung von Nutzungsszenarios. Sie können aus einer Abteilung in einem großen Unternehmen bestehen oder etwa eine Bürogemeinschaft von Ingenieuren oder Rechtsanwälten sein. Auch Wohngemeinschaften oder große Familien fallen unter diesen weiten Begriff.

[10] Vgl. Abschnitt 4.1.1 „Backup und Restore Center“ ab Seite 35
[11] Vgl. Abschnitt 3.3.3 „Offsite Backup“ auf Seite 29

Entsprechend vielgestaltig und individuell verschieden stellen sich die möglichen Backup-Lösungen dar.

In dem nächsten Abschnitt wird der Windows Home Server beschrieben. Er bietet eine praktische Plattform, um das Backup von bis zu zehn Windows-PCs zu übernehmen.

Die im darauf folgenden Abschnitt vorgestellte Lösung ist `duply`. Eine modifizierte Version des Backup-Werkzeugs Duplicity, um verschlüsselte Backups auf nicht vertrauenswürdigen Servern zu sichern.

4.2.1 Windows Home Server

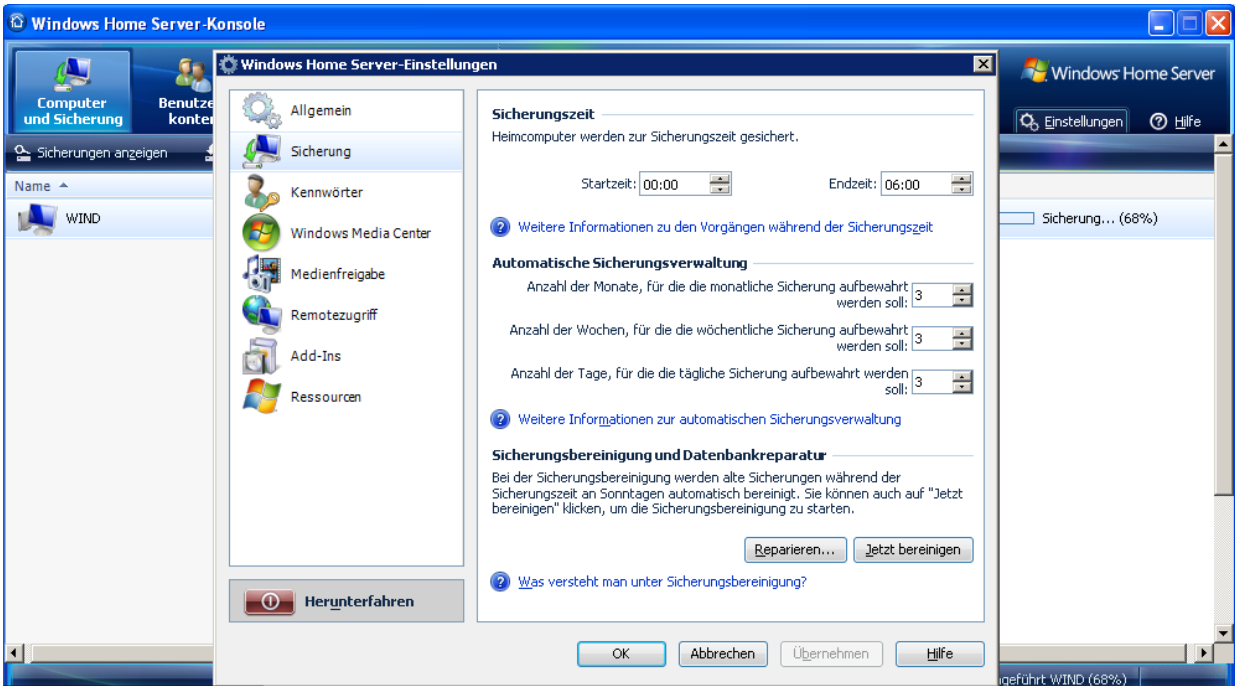

Abbildung 4.6: Windows Home Server Backup Einstellungen

Der *WHS*[12] wurde 2007 von Microsoft vorgestellt und dient als zentraler Server für eine kleine Arbeitsgruppe. Neben vielen anderen Eigenschaften[13] bietet er auch die Möglichkeit eines zentralen Backups über das Netzwerk für bis zu zehn Windows-PCs. Diese zentrale Backup-Funktion wird in diesem Abschnitt betrachtet, die übrigen bleiben außen vor.

Aufgrund der Beschränkung auf zehn Clients ist der WHS nicht die ideale Lösung für große Abteilungen, dafür ist er jedoch wesentlich günstiger als der ebenfalls von Micro-

[12] Windows Home Server
[13] [Endres 2009b]

soft angebotene *SBS*[14]. Aufgrund seiner geschickten Handhabung von Backups, kann der Windows Home Server auch in einer, in allen anderen Belangen von einem SBS verwalteten, Domäne für die Erledigung der Backups sorgen[15].

Nach der sehr einfach zu bewerkstelligenden Installation des Servers selbst, wird eine Client-Anwendung auf jedem zu sichernden PC installiert[16]. Diese stellt, neben einem Fernzugriff auf alle Funktionen des Servers, auch die Basis für die Durchführung der Backups zur Verfügung. Der WHS ist in der Lage, die zu sichernden PCs zu einem durch den Anwender konfigurierbaren Zeitfenster über die *WoL*[17] Funktion einzuschalten, die Sicherung durchzuführen und nach dem Backuplauf wieder auszuschalten. Neben dem Zeitfenster lassen sich über die auf Abbildung 4.6 auf Seite 43 zu sehende Anwendung auch die Rotationszyklen[18] sehr genau konfigurieren.

Die Sicherung selber erfolgt dabei mit Hilfe von Schattenkopien auf Blockebene[19]. Der WHS setzt für die Speicherung der gesicherten Daten eine Technik namens Single Instance Storage ein[20]. Durch ihren Einsatz müssen identische Daten, die von mehreren Client-PCs gesichert wurden, nur ein einziges Mal gespeichert werden, egal wie viele Sicherungen darauf beruhen. Da diese Technologie auf Blockebene arbeitet, ist sie sehr effizient und kann erheblich zu einer sparsamen Nutzung des auf dem Server vorhandenen Speicherplatzes beitragen. Da nur unter Windows laufende Computer mit dem WHS-Backup zusammen arbeiten, sind stets sehr viele Daten auf den Clients identisch. Diese Rate ist am höchsten, wenn auf allen PCs identische Versionen von Windows, also etwa ausschließlich Windows XP oder ausschließlich Windows 7, laufen. In diesem Idealfall müssen alle zum Betriebssystem gehörenden Daten nur ein einziges Mal auf dem Server gespeichert werden. Das selbe Prinzip gilt auch für die auf den Clients installierten Anwendungen und die gespeicherten Daten.

Für jeden Client können einzelne Ausnahmen konfiguriert werden wobei der WHS, wie auf Abbildung 4.7 auf Seite 45 zu sehen, bereits standardmäßig eine sinnvolle Vorauswahl von zu ignorierenden Verzeichnissen trifft.

Sollte ein Client mit defekter Festplatte ausfallen, ist ein komplettes Restore des Systems aus dem Backup auf dem Server möglich. Dazu wird eine Restore-CD angefertigt von der aus der letzte gesicherte Stand des Systems eingespielt werden kann.

Für das Restore von einzelnen Dateien und Ordnern kann ein beliebiges Backup-Set als Laufwerk auf dem Client eingebunden werden und dort wie ein externes Laufwerk

[14] Small Business Server
[15] [Microsoft Corporation 2009]
[16] [Microsoft Corporation 2007]
[17] Wake on LAN
[18] Vgl. Abschnitt 2.2.2 „Sicherungszeitpunkte und Zyklen“ auf Seite 9
[19] Vgl. Abschnitt 4.1.1 „Backup und Restore Center“ auf Seite 35
[20] [Microsoft Corporation 2006]

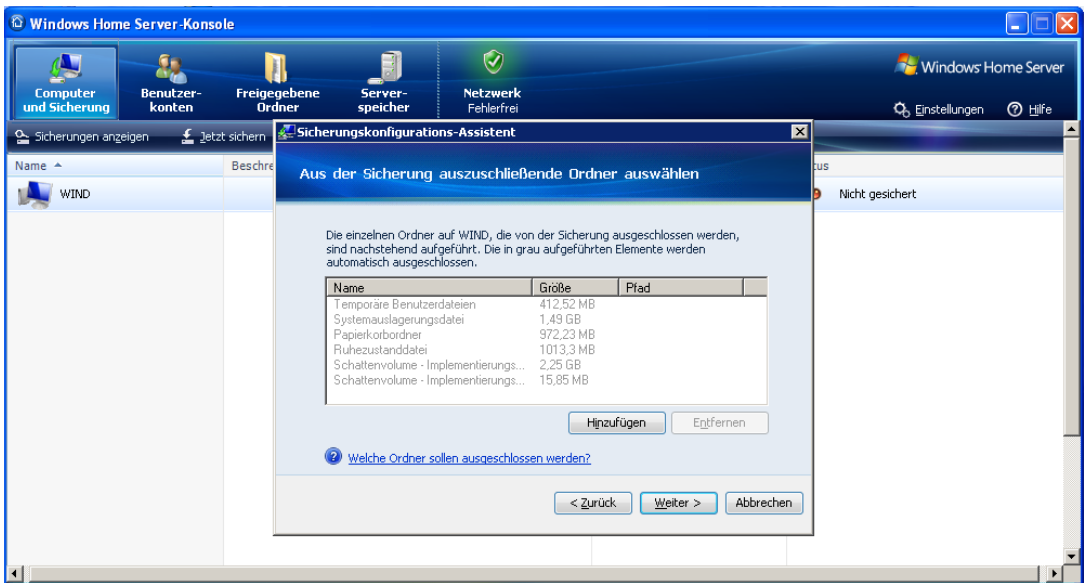

Abbildung 4.7: Vorauswahl von zu ignorierenden Verzeichnissen

durchsucht werden. Über simples Kopieren können die vermissten Daten dann wiederhergestellt werden.

Auch der Server selbst kann mit allen seinen freigegebenen Verzeichnissen gesichert werden. Wird eine externe Festplatte an den Server angeschlossen, bietet ein Konfigurationsdialog die entsprechende Option an. Das Backup des Servers erstellt allerdings keine erneute Sicherheitskopie der Client-Backups[21]. Diese verbleiben auf den internen Festplatten des Servers.

Es ist festzuhalten, dass der WHS eine sehr gute Backup-Lösung für kleine Arbeitsgruppen darstellt, die mit einer homogenen Windows-Infrastruktur arbeiten. In dieser Umgebung kann er seine Stärken voll ausspielen und, abhängig von den Ansprüchen der Anwender, das komplette Management des Netzwerks übernehmen. Leider ist es aufgrund der besonderen Art der Datenspeicherung nicht ohne weiteres möglich, eine Offsite-Backup- oder Verschlüsselungs-Strategie zu ergänzen. Die Skalierung auf mehr als zehn Client-PCs ist nur möglich, indem ein zweiter WHS aufgesetzt und konfiguriert wird. Dieser Mehraufwand für die Administration wäre durch eine flexiblere Lizenzierung des WHS deutlich eleganter zu lösen.

4.2.2 Duplicity

Ein anderes Szenario für eine Arbeitsgruppe ist eine Bürogemeinschaft von Freiberuflern wie Ingenieuren oder Programmierern. Diese könnten aus Gründen der Kostenersparnis

[21] Aufbausatz[Endres 2009a]

eine gemeinsame IT-Infrastruktur betreiben, ansonsten aber ihren jeweils eigenen Geschäften nachgehen. Damit die jeweiligen Backups auf einem gemeinsamen Server sicher vor Manipulation und neugierigen Blicken sind, ist der Einsatz von Verschlüsselungstechnik zu empfehlen.

In diesem Abschnitt wird daher als Beispiel für ein solches Szenario das Backup-Werkzeug Duplicity besprochen.

Duplicity ist ein Kommandozeilenwerkzeug für unixoide Betriebssysteme. Es erstellt mit *GPG*[22] verschlüsselte Backups und transferiert sie über eine Vielzahl von Protokollen zu einem entfernten Server. Dabei ist nur das erste Backup ein volles Level 0 Backup, die nachfolgenden Sicherungen erfolgen inkrementell. Das Problem, gleichzeitig eine inkrementelle Sicherung zu gewährleisten und dabei mit vollständig verschlüsselten Backups zu arbeiten, löst Duplicity sehr elegant.

Das erste Vollbackup besteht aus vielen einzelnen, etwa 5 Megabyte großen, mit GPG-verschlüsselten Archiven, die in der Duplicity-Nomenklatur *Difftars* heissen. Zusätzlich werden Prüfsummen der gesicherten Dateien gebildet und verschlüsselt in sogenannten *Sigtars* abgelegt. Zuletzt erstellt Duplicity eine verschlüsselte Liste aller gesicherten Dateien und Verzeichnisse in einer *Manifest* genannten Datei.

Um das erste und alle folgenden inkrementellen Backups anzulegen, vergleicht Duplicity die in den *Sigtars* aller bisherigen Backups vorhandenen Prüfsummen. So ist feststellbar, welche Dateien sich geändert haben und beim aktuellen Backuplauf gesichert werden müssen.

Diese Dateien werden dann inklusive *Sigtar* und *Manifest* auf die gleiche Weise gespeichert wie das erste Vollbackup.

Aufgrund der zahlreichen Möglichkeiten der Übertragung und der zugrundeliegenden asymetrischen Verschlüsselung auf GPG-Basis ist Duplicity relativ komplex in der Konfiguration[23]. Die Zeitschrift c't hat daher ein Wrapper-Script[24] entwickelt, welches mittlerweile unter dem Namen `duply` Einzug in die offiziellen Quellen von Duplicity gehalten hat und weiter verbessert wurde. So wurde die Beschränkung auf die Übertragung per *FTP*[25] aufgehoben, Stapelverarbeitung hinzugefügt und die Konfigurationsdatei vereinfacht.

Die Konfiguration für das Beispielszenario beschränkt sich mit `duply` auf die Erstellung eines Profils für jeden zu sichernden Computer sowie eines Cronjobs zur automatisierten Ausführung des Backups.

[22] GNU Privacy Guard
[23] [DEBIAN-ADMINISTRATION.ORG 2005]
[24] [RÜTTEN 2006]
[25] File Transfer Protocol

Ein Profil wird mit `duply profilname create` angelegt und erstellt eine Verzeichnisstruktur mit Konfigurationsdatei, fertigen Scripts für die Ausführung vor und nach dem eigentlichen Backup, sowie einer Ausschlussliste für zu ignorierende Dateien und Verzeichnisse. Die Möglichkeit vor und nach dem Backup Scripte auszuführen ermöglicht es beispielsweise, eine Datenbankanwendung vor dem eigentlichen Backup zu stoppen und so in einem garantiert konsistenten Zustand zu sichern[26].

Ein generisches `duply`-Profil ist in Listing 3 auf Seite x im Anhang zu finden.

Ein `duply` Backupvorgang mit den einzelnen Schritten der Durchführung ist in Listing 4.3 zu sehen.

Listing 4.3: Duply Backuplauf

```
1 dennis@ubuntu:~$ duply dennis backup
2 Start duply v1.5.1.2, time is 12/17/09 13:33:50.
3 Using profile '/home/dennis/.duply/dennis'.
4 Using installed duplicity version 0.4.10, gpg 1.4.6 (Home:
    ~/.gnupg)
5 Test - Encryption with passphrase (OK)
6 Test - Decryption with passphrase (OK)
7 Test - Compare Original w/ Decryption (OK)
8 Cleanup - Delete '/tmp/duply.6987.1261053230_*'(OK)
9
10 --- Start running command PRE 13:33:50.449 ---
11 Skipping n/a script '/home/dennis/.duply/dennis/pre'.
12 --- Finished 13:33:50.487 - Runtime 00:00:00.037 ---
13
14 --- Start running command BKP 13:33:50.522 ---
15 Running duplicity - OK
16 Output: PASSPHRASE=test TMPDIR=/tmp /usr/bin/duplicity --
    verbosity 5 --gpg-options= --exclude-globbing-filelist /
    home/dennis/.duply/dennis/exclude /home/dennis/ ssh://
    dennis:@backup_server/opt/duply/backup_dennis
17 --- Finished 13:33:50.567 - Runtime 00:00:00.045 ---
18
19 --- Start running command POST 13:33:50.600 ---
20 Skipping n/a script '/home/dennis/.duply/dennis/post'.
21 --- Finished 13:33:50.635 - Runtime 00:00:00.035 ---
```

Wird dieses Kommando mit einem Cronjob jede Stunde automatisiert ausgeführt, ist das komplette Home-Verzeichnis des Anwenders stets aktuell auf dem Server gesichert, ohne das der Anwender sich um die Integrität des Server-Administrators kümmern müsste.

[26] Vgl. Abschnitt 3.2.1 „Cold Backup“ auf Seite 25

Ein Restore wird ebenfalls über die Kommandozeile ausgeführt. Die drei Tage alte Version einer einzelnen Datei kann beispielsweise per `duply fetch fom/diplomarbeit.tex /home/dennis/restore/diplomarbeit.tex 3D` in einem anderen Verzeichnis wiederhergestellt werden. Das gleiche Vorgehen funktioniert auch mit ganzen Verzeichnissen und anderen Zeitangaben.

Für die Pflege des Backups bietet `duply` die Optionen `status` und `verify` an. Es bietet sich an, diese Kommandos, genauso wie etwa die Erstellung eines weiteren Vollbackups pro Monat, per Cronjob ausführen zu lassen und somit über den aktuellen Status des Backups auf dem laufenden zu bleiben.

Es ist bei dem beschriebenen Vorgehen essentiell wichtig, ein Backup des `duply`-Profils vorzuhalten. Ohne die darin gespeicherten Informationen und kryptographischen Schlüssel ist das Backup nicht wiederherstellbar[27].

Zusammenfassend ist Duplicity, auch in Verbindung mit `duply`, ein eher technisches Werkzeug für erfahrene Unix- oder Linux-Anwender. Dafür ist es äußerst mächtig in seinen Fähigkeiten und implementiert eine sichere Übertragung und Speicherung aller Backups.

Neben den beiden exemplarisch vorgestellten Szenarien gibt es, gerade im Bereich der kleinen und mittleren Arbeitsgruppen, eine unüberschaubare Vielzahl an möglichen Lösungen für den jeweils individuellen Backup- und Restore-Bedarf. Die beiden vorgestellten Lösungen sind dabei an entgegengesetzten Enden des Spektrums der Möglichkeiten angesiedelt, um eben diese große Spanne zu verdeutlichen. Für ihren jeweiligen Einsatzzweck eignen sie sich ausgezeichnet. Es wird immer Einbußen im Komfort der Handhabung eines Backup-Systems geben, wenn Verschlüsselung ins Spiel kommt. Ohne deren Einsatz sind auch netzweite, bequem zu administrierende Lösungen wie der Windows Home Server realisierbar.

4.3 Enterprise

Der Enterprise-Bereich steht dem oben vorgestellten Bereich der Arbeitsgruppen in Sachen Diversität in nichts nach. Hier sind es nur üblicherweise noch mehr Maschinen, die dazu auch noch häufig als Server dienen. Dementsprechend sind die Datenmengen höher und die zur Verfügung stehenden Zeitfenster für Backup und Restore kleiner. Viele Unternehmen setzen außerdem verschiedene Betriebssysteme ein und benötigen dennoch eine zentrale Backup-Lösung.

Gerade im Enterprise-Backup-Bereich gibt es eine Vielzahl von kommerziellen Anbietern, die Lösungen für so gut wie jede Anforderung anbieten. In dieser Arbeit liegt der

[27] Vgl. Abschnitt 3.3.4 „Verschlüsselte Backups“ ab Seite 31

Fokus für den Enterprise-Bereich jedoch auf Lösungen aus der Open Source Welt. Diese sind in vielen Fällen ebenso mächtig wie die kommerziellen Ableger. Um mangelnden Support muss sich heute ebenfalls kein Unternehmen mehr sorgen. Auf Open Source Produkte spezialisierte Firmen bieten, gegen eine entsprechende Bezahlung, Installation, Konfiguration und Business-Support von Open Source Software an.

In diesem Abschnitt wird zunächst `rsnapshot` als Beispiel für das zentralisierte Backup und Restore einer dezentralen Linux-Server-Infrastruktur vorgestellt. Dieses Programm erweitert ein reguläres `rsync`-Backup um konfigurierbare Rotationszyklen und die Fähigkeit über Scripte Datenbankformate zu sichern.

Im Anschluss wird *Amanda*[28] beschrieben. Ein weit fortgeschrittenes Backup und Restore Programm, das betriebssystemunabhängig eine Vielzahl von Clients und Servern zentral sichern kann und dabei einige herausragende Eigenschaften an den Tag legt.

Diese beiden Programme stehen exemplarisch für ein weites Feld von Werkzeugen aus der Open Source Welt, die für eine Backup- und Restore-Strategie verwendet werden können. Die Entscheidung für ein Produkt, egal ob kommerziell oder Open Source, sollte in großen Umgebungen mit viel Sorgfalt gefällt werden. Eine Umstellung auf ein anderes Produkt ist, gerade dann wenn proprietäre Formate verwendet werden, oftmals mit großen Schwierigkeiten verbunden.

4.3.1 rsnapshot

In dem hier beschriebenen Beispielszenario wird `rsnapshot` eingesetzt, um ca. 100 Linux-Server an unterschiedlichen, miteinander vernetzten Standorten zu sichern. Bei den Servern handelt es sich jeweils um den zentralen File-Server einer Schule, auf dem sowohl Schüler als auch Lehrer ihre Dateien speichern[29]. Da sich die Daten auf den zu sichernden Maschinen fast ausschließlich während des Tages ändern, ist es sinnvoll den Zeitpunkt der Datensicherung in die Nacht zu legen[30]. Um die Daten dieser Umgebung zentral zu sichern, bietet sich ein weiterer Linux-Server, der über eine entsprechende Storage-Anbindung verfügt, als zentrale Maschine an.

Die Entwicklung von `rsnapshot` basiert auf der Idee, eine reine Spiegelung von Daten per `rsync` um Snapshots zu ergänzen[31]. Diese werden nach einem, durch den Anwender konfigurierbarem Schema rotiert. Um den Platzbedarf der Sicherungen möglichst gering zu halten, arbeitet `rsnapshot` mit Hard Links. Dadurch erscheint jedes inkrementelle

[28] Advanced Maryland Automatic Network Disk Archiver
[29] [Rauprich 2009]
[30] Vgl. Abschnitt 2.2.2 „Sicherungszeitpunkte und Zyklen“ ab Seite 9
[31] [Ruebel 2004]

Backup als vollständiger Datensatz im Dateisystem, ohne das für unveränderte Dateien Speicherplatz verschwendet wird[32].

Zusätzlich bietet `rsnapshot` die Möglichkeit, durch entsprechende Scripts weitere Aktionen durchzuführen, etwa um Datenbanken zu sichern[33].

Diese Fähigkeiten, ergänzt durch eine recht einfache Konfiguration[34], machen `rsnapshot` zu einer sehr guten Lösung für das beschriebene Szenario.

Nach erfolgter Installation auf dem Backup-Server werden die einzelnen Backup-Punkte in der `rsnapshot.conf` Datei konfiguriert. Hier wird jedes zu sichernde Verzeichnis, inklusive der Adresse des Servers auf dem das Verzeichnis liegt, in einer Zeile notiert. Am Ende jeder Zeile wird das lokale Sicherungsverzeichnis angegeben. Soll statt der einfachen Sicherung eines Verzeichnisses im Dateisystem ein Script ausgeführt werden, startet die Zeile anstelle von `backup` mit `backup_script` und es wird der lokale Ort des auszuführenden Scripts angegeben.

Beide Verfahren sind beispielhaft in Listing 4.4 zu sehen.

Listing 4.4: rsnapshot Backup-Konfiguration

```
1 backup            1.2.3.4:/etc/ server_i/
2 backup_script     /script_dir/custom_backup_script.sh i_mysql/
```

Eine vollständige Konfigurationsdatei findet sich in Listing 4 auf Seite xiii im Anhang.

Anwendungsfälle für den Einsatz eines Scripts sind üblicherweise die Sicherung von Datenbanken. Es können aber auch beliebig komplexe Scripts ausgeführt werden, etwa wenn ein Dienst von einer zu stoppenden Datenbank abhängig ist. Ein Beispiel für ein solches Vorgehen ist in Listing 4.5 zu sehen. Das Script verbindet sich mit dem Server, stoppt den *LDAP*[35]-Server und sichert erst dann die Daten. Nach dem Backup wird der LDAP-Server erneut gestartet. Außerdem muss hier der *DHCP*[36]-Server neu gestartet werden, da dessen Konfiguration in der gerade gesicherten Datenbank gespeichert wird.

Listing 4.5: Beispiel-Script für die Sicherung einer LDAP-Datenbank

```
1 #!/bin/sh
2
3 serverip="1.2.3.4"
4
5 ssh $serverip /etc/init.d/slapd stop
```

[32] Vgl. Abschnitt 4.1.2 „Time Machine“ ab Seite 39
[33] Vgl. Abschnitt 3.2 „Backup von Datenbanken“ auf Seite 24
[34] [Rosenquist 2005]
[35] Lightweight Directory Access Protocol
[36] Dynamic Host Configuration Protocol

```
6 ssh $serverip tar -czvf - /var/lib/ldap > ldap_backup.tar.gz
7 ssh $serverip /etc/init.d/slapd start
8 ssh $serverip /etc/init.d/dhcp3-server restart
```

Ein anderes Beispiel für den sinnvollen Einsatz von Scripts ist beispielsweise die Sicherung von mehreren Subversion-Repositories, ohne dass jedes einzeln angegeben werden müsste. Das passende Script befindet sich in Listing 4.6.

Listing 4.6: Beispiel-Script für die Sicherung von Subversion-Repositories

```
1 #!/bin/sh
2
3 SVN_DIR="/var/lib/subversion"
4 BACKUP_DIR="."
5 SVN_REPOS=‘ls $SVN_DIR‘
6
7 for repos in $SVN_REPOS; do
8     svnadmin dump --incremental --deltas --quiet $SVN_DIR/
        $repos \
9         > $BACKUP_DIR/$repos.svndump
10 done
```

So können mit `rsnapshot` alle Sicherungsaufgaben durchgeführt werden, die sich per Script realisieren lassen. Alles, was als Datei oder Verzeichnis ausgegeben werden kann, wird nach Abschluss des Scripts in den konfigurierten lokalen Verzeichnissen gesichert. Sollte der Output des Scripts identisch mit bereits vorhandenen Dateien sein, so werden Hard Links erstellt um Speicherplatz einzusparen. Diese Flexibilität ist ein großer Vorteil im Vergleich zu vielen kommerziellen Backup-Produkten.

Eine weitere erwähnenswerte Eigenschaft von `rsnapshot` ist seine breite Auswahl an Übertragungsprotokollen. So können die gesicherten Daten nicht nur per `rsync`-Protokoll übertragen werden sondern beispielsweise auch per *SSH*[37]-Verbindung. Über eine solche verschlüsselte Verbindung kann die Sicherheit der Daten auch dann gewährleistet werden, wenn das Netzwerk, durch das die Backups übertragen werden, als nicht vertrauenswürdig angesehen wird.

Das Rotationsschema wird bei `rsnapshot` durch eine Kombination aus Einträgen in der `rsnapshot.conf` Datei und den damit korrespondierenden Cronjobs konfiguriert. Der entsprechende Abschnitt[38] aus dem Listing 4 von Seite xiii des Anhangs ist zur Verdeutlichung in Listing 4.7 auf Seite 51 noch einmal wiederholt.

[37] Secure Shell
[38] Zeile 60 ff.

Listing 4.7: Backup Intervalle aus `rsnapshot.conf`

```
1 interval          daily   7
2 interval          weekly  4
3 interval          monthly 6
```

Diese drei Zeilen spezifizieren, dass sieben Versionen der täglichen Backups aufgehoben werden, vier der wöchentlichen und sechs der monatlichen. Die Zeitpunkte zu denen die jeweiligen Backups durchgeführt werden stehen in einer Crontab-Datei wie sie in Listing 4.8 zu sehen ist:

Listing 4.8: rsnapshot Cronjob Definitionen

```
1 50 23 * * * /usr/bin/rsnapshot daily
2 00 22 * * 6 /usr/bin/rsnapshot weekly
3 00 21 1 * * /usr/bin/rsnapshot monthly
```

Die täglichen Backups werden hier jeden Tag um 23:50 Uhr durchgeführt, die wöchentlichen jeden Samstag um 22:00 Uhr und die monatlichen an jedem ersten Tag eines Monats um 21:00 Uhr.

Mit diesem Schema hat das tägliche Backup genügend Zeit, um die original Dateien von ihrem jeweiligen Server zu holen. Die anderen beiden Zyklen arbeiten bereits vorher ihre Rotationen ab. Der monatliche Intervall funktioniert dabei wie folgt:

1. Der älteste, hier also sechste, monatliche Snapshot wird entsorgt.
2. Die übrigen Snapshots werden nach hinten rotiert. Also wird der fünfte Snapshot zum sechsten, der vierte zum fünften und so weiter.
3. Der älteste wöchentliche Snapshot wird zum neuesten monatlichen Snapshot. Dadurch muss der nächste wöchentliche Backuplauf seinen letzten Snapshot nicht mehr selbst löschen und läuft daher schneller.

Analog dazu wird der älteste tägliche Snapshot einmal in der Woche zum jüngsten Snapshot der wöchentlichen Rotation. Während aller dieser Vorgänge bleibt das Konzept der Hard Links bestehen um Speicherplatz einzusparen. So entsteht nach und nach eine Verzeichnisstruktur wie sie auszugweise in Listing 4.9 zu sehen ist.

Listing 4.9: rsnapshot Verzeichnisstruktur

```
1 |-- daily.0
2 |   |-- server_i
3 |   |-- server_i_ldap
4 |   |-- server_i_mysql
5 |   |-- server_ii
```

```
6 |    |-- server_ii_ldap
7 |    `-- server_ii_mysql
8 |-- daily.1
9 |    |-- server_i
10 (...)
11 |-- weekly.0
12 |    |-- server_i
13 |    |-- server_i_ldap
14 |    |-- server_i_mysql
15 |    |-- server_ii
16 |    |-- server_ii_ldap
17 |    `-- server_ii_mysql
18 (...)
```

Der Restore-Vorgang soll in dem vorliegenden Szenario möglichst durch den lokalen Administrator des Servers durchgeführt werden können, um Verluste von Daten zeitnah zu beheben. Dabei soll jeder Administrator nur auf die Backups des eigenen Servers zugreifen können[39].

Um dies zu ermöglichen, wird zunächst das Script[40] aus Listing 5 auf Seite xix im Anhang eingesetzt. Es erstellt aus der oben stehenden original `rsnaphot`-Verzeichnisstruktur aus Listing 4.9, die nach Servern sortierte Struktur aus Listing 4.10.

Listing 4.10: Browse Backup Verzeichnisstruktur

```
1 |-- server_i
2 |    |-- 12-15-2009 -> /opt/rsnapshot/daily.6/server_i
3 |    |-- 12-16-2009 -> /opt/rsnapshot/daily.5/server_i
4 |    |-- 12-17-2009 -> /opt/rsnapshot/daily.4/server_i
5 |    |-- 12-18-2009 -> /opt/rsnapshot/daily.3/server_i
6 |    |-- 12-19-2009 -> /opt/rsnapshot/daily.2/server_i
7 (...)
8 |-- server_ii
9 |    |-- 12-15-2009 -> /opt/rsnapshot/daily.6/server_ii
10 |    |-- 12-16-2009 -> /opt/rsnapshot/daily.5/server_ii
11 (...)
```

So entsteht eine saubere Trennung der Daten nach Quelle, die außerdem noch jeweils nach dem Datum ihrer Erstellung sortiert sind. Dabei wird wiederum nur sehr wenig Speicherplatz verbraucht, da die neue Verzeichnisstruktur lediglich mit symbolischen Links auf die regulären Backup-Daten verweist.

[39] Vgl. Abschnitt 3.3.2 „Zugriffsberechtigungen“ ab Seite 28
[40] [Shackelford 2006]

Diese Verzeichnisse können dann mit passenden Zugriffsrechten versehen werden und dem jeweiligen lokalen Administrator über das Netzwerk zugänglich gemacht werden. Dieser kann dann durch simples Kopieren der Daten den Restore-Vorgang durchführen. Es besteht so keine Notwendigkeit den Backup-Administrator zu kontaktieren, der gesamte Workflow kann lokal geschehen. Um eine nachträgliche Manipulation der Backups auszuschließen, sind diese Netzwerkfreigaben schreibgeschützt.

Ein umfassendes Backupkonzept mit `rsnapshot` erfordert einigen Aufwand bei der Implementierung und kommt nicht ohne eine sorgfältige individuelle Konfiguration und Anpassungen von diversen Scripts aus. Die sich bietenden Möglichkeiten sind aber enorm und rechtfertigen den anfänglichen Aufwand durch die leichte Erweiterbarkeit des Systems.

Besonders die Möglichkeit des individuellen Restore über das Netzwerk erspart viel Zeit und ermöglicht sehr zeitnahe Reaktionen auf einen eventuellen Datenverlust. Durch diese Möglichkeit findet, neben den regulären Logdateien, außerdem ein kontinuierliches Testen der erfolgten Backupsätze statt. Die Übertragung der Backups findet außerdem verschlüsselt statt und ist somit auch für den Transfer durch nicht vertrauenswürdige Netzwerke geeignet. Die Speicherung erfolgt allerdings im Klartext, daher ist es notwendig, dass die lokalen Administratoren der Server dem Administrator des Backupservers ihre jeweiligen Daten anvertrauen.

4.3.2 Amanda

Amanda ist eine komplette Open Source Backup- und Restore-Lösung. Sie wird seit 1991 stetig weiter entwickelt und erfreut sich einer großen Zahl an Anwendern. Amanda enthält alle in dieser Arbeit bisher angesprochenen Fähigkeiten und bietet ein gelungenes Gesamtkonzept mit einigen Alleinstellungsmerkmalen[41].

So ist es möglich, unterschiedliche Betriebssysteme auf einem einzigen Amanda-Server zu sichern. Zu diesem Zweck werden jeweils die nativen Instrumente des Betriebssystems benutzt. Diese Vorgehensweise führt dazu, dass die Sicherungskopien im Ernstfall auch ohne Amanda für einen Restore benutzt werden können. Diese Offenheit der zugrunde liegenden Architektur ist ein sehr großer Vorteil gegenüber dem Einsatz von proprietären Datenformaten für die Sicherheitskopien. Amanda ist in der Lage, das Konzept des D2D2T umzusetzen und sowohl auf Festplatten als auch auf Magnetbänder zu sichern[42]. Beides kann auch gleichzeitig eingesetzt werden, um die Möglichkeiten der beiden Medien jeweils ideal auszunutzen. Für die Verschlüsselung der Sicherheitskopien, sowie für die Verschlüsselung ihrer Übertragung, sorgt Amanda mit offenen Verschlüsselungsstan-

[41] [Preston 2007]
[42] Vgl. Abschnitt 3.3.3 „Offsite Backup“ ab Seite 29

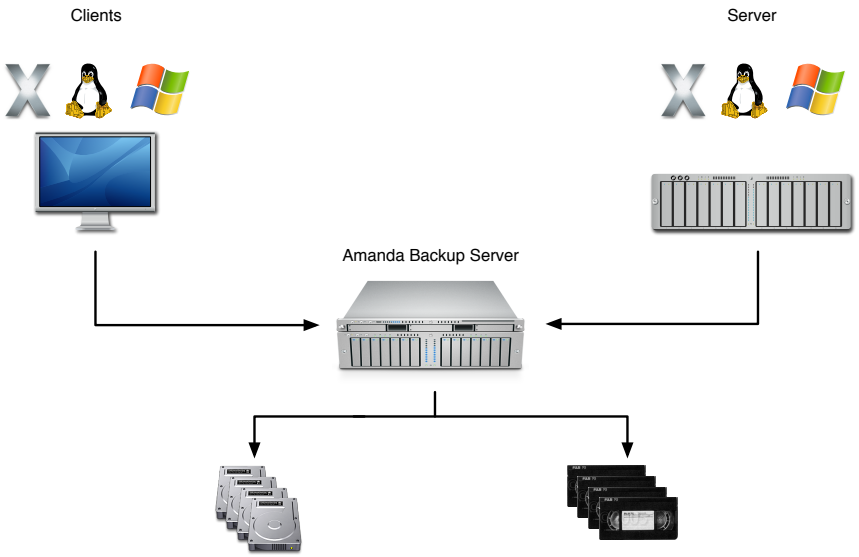

Abbildung 4.8: Amanda Cross Platform Backup

dards wie GPG oder *SSL*[43]. Die Durchführung der Verschlüsselung kann hierbei entweder auf dem Client, oder auf dem Server erfolgen. So kann die Performance der zur Verfügung stehenden Infrastruktur ideal ausgelastet werden.

Außerdem unterstützt Amanda die Sicherung von Daten in der *S3*[44]-Infrastruktur von Amazon. Dieser neue Ansatz für Offsite-Backups kann die Zeit für einen Restore-Vorgang, entsprechende Bandbreite vorausgesetzt, erheblich beschleunigen[45].

Es existieren verschiedene Firmen, die Installation, Betrieb, Support und Schulungen rund um Amanda anbieten. Auch hier zahlt sich der Open Source Gedanke aus, da es nicht nur eine einzige Firma gibt, die das komplette Produkt kontrolliert. Diese Tatsache ermöglicht Planungssicherheit, da das eigene Backup nicht nur von dem Schicksal einer einzigen Firma abhängig ist.

Trotz dieser großen Anzahl an interessanten und bemerkenswerten Eigenschaften werden in diesem Abschnitt nur die folgenden, herausragenden Punkte näher betrachtet:

- Parallele Durchführung von Sicherungsläufen durch das Konzept der Holding Disk.
- Integrierte Kalkulation für Zeitfenster, Zeitpunkte und Level aller Backups.

[43] Secure Socket Layer
[44] Amazon Simple Secure Storage Service
[45] Vgl. Abschnitt 5.1.2 „Cloud-Backup“ ab Seite 65

Holding Disk

Amanda benutzt eine so genannte Holding Disk, auf der Backups zwischengespeichert werden, bevor sie auf das tatsächliche Backupmedium geschrieben werden. Besonders im Falle von Magnetbändern ist das ein äußerst sinnvolles Vorgehen. Bandlaufwerke arbeiten aufgrund ihrer Bauweise sequentiell und können ihre volle Geschwindigkeit nur erreichen, wenn sie kontinuierlich betrieben werden. Ohne einen kontinuierlichen Datenstrom wird es außerdem nötig, das Band zu stoppen, auf die Daten zu warten, das Band ein Stück zurück zu spulen und dann wieder anlaufen zu lassen, wenn genug Daten da sind. Dieser Effekt wird in Anlehnung an die typische Schuhputz-Bewegung *Shoe Shining* genannt. Dieses Stoppen und Starten des Bandes ist der Lebensdauer der Laufwerkmechanik und der Bänder eher abträglich.

Zu eben diesem Shoe Shining kommt es bei einem direkten Backup auf Band aber unweigerlich, da selbst über Gigabit angebundene Clients die Daten nicht schnell genug liefern können, um den Effekt zu vermeiden. Wird dagegen ein Zwischenspeicher, etwa ein über Glasfaser angebundenes schnelles RAID-Laufwerk benutzt, kann Shoe Shining vermieden werden. Die Daten mehrerer Clients können parallel auf die Holding Disk geschrieben werden. Dabei kann jeder Client so schnell schreiben wie er an den Server angebunden ist. Eventuellen Engpässen an der Netzwerkseite des Servers kann, im Gegensatz zu den meisten Clients, mit weiteren Netzwerk-Schnittstellen begegnet werden.

Ist das Backup dann auf der Holding Disk angekommen, werden die Daten in der maximal möglichen Geschwindigkeit auf das Band geschrieben. Dies kann entweder erst dann erfolgen, wenn der komplette Backuplauf aller Clients durchgeführt wurde, oder aber sobald der erste Client sein Backup auf der Holding Disk abgeliefert hat.

Auf diese Art ergänzen sich die Medien Magnetband und Festplatte hervorragend. Beide können ihre Stärken voll ausspielen und sorgen für eine Entlastung des jeweils anderen Mediums.

Amanda Backup-Kalkulation

Übliche Backup-Software ist meist auf die gleiche Weise konfigurierbar. So wird etwa jeden Sonntag ein volles Backup erstellt und an den anderen Tagen erfolgen inkrementelle Sicherungen. Dieser Zeitplan ist natürlich für andere Zeiträume anpassbar und kann auch mit mehreren Stufen des inkrementellen Backups durchgeführt werden, das Prinzip bleibt dabei aber immer gleich[46].

[46] Vgl. Abschnitt 3.1.2 „Inkrementelles Backup“ ab Seite 18

Amanda geht hier einen deutlich anderen Weg[47]. Anstelle einer starren Konfiguration mit exakter Angabe von Zeitpunkten und Stufen des Backups, erhält Amanda gewisse Regeln für das Backup und berechnet dann die notwendigen Stufen und Zeitpunkte selbst. Dabei ist das Ziel der Berechnungen, über alle zu sichernden Clients, eine konstante Datenmenge und damit ein konstantes Zeitfenster für die Sicherung zu erreichen.

Eine solche Regel könnte etwa lauten: „Es muss von jedem Client innerhalb von sieben Tagen ein volles Backup angefertigt werden. An allen anderen sechs Tagen sollen inkrementelle Backups durchgeführt werden, wobei die maximale Zeit zwischen den vollen Backups sieben Tage beträgt.“ Dieser Zeitraum zwischen den vollen Backups wird *Dump Cycle* genannt.

Ausgestattet mit diesem Regelwerk beginnt jeder Backuplauf damit, dass jeder zu sichernde Client die Dateien auflistet, die sich seit dem letzten Backup verändert haben und wie groß der daraus resultierende Bedarf an Speicherplatz ist. Nachdem diese Daten von allen Clients an den Server gesendet wurden, berechnet dieser die ideale Kombination von inkrementellen und vollen Backups für alle Clients.

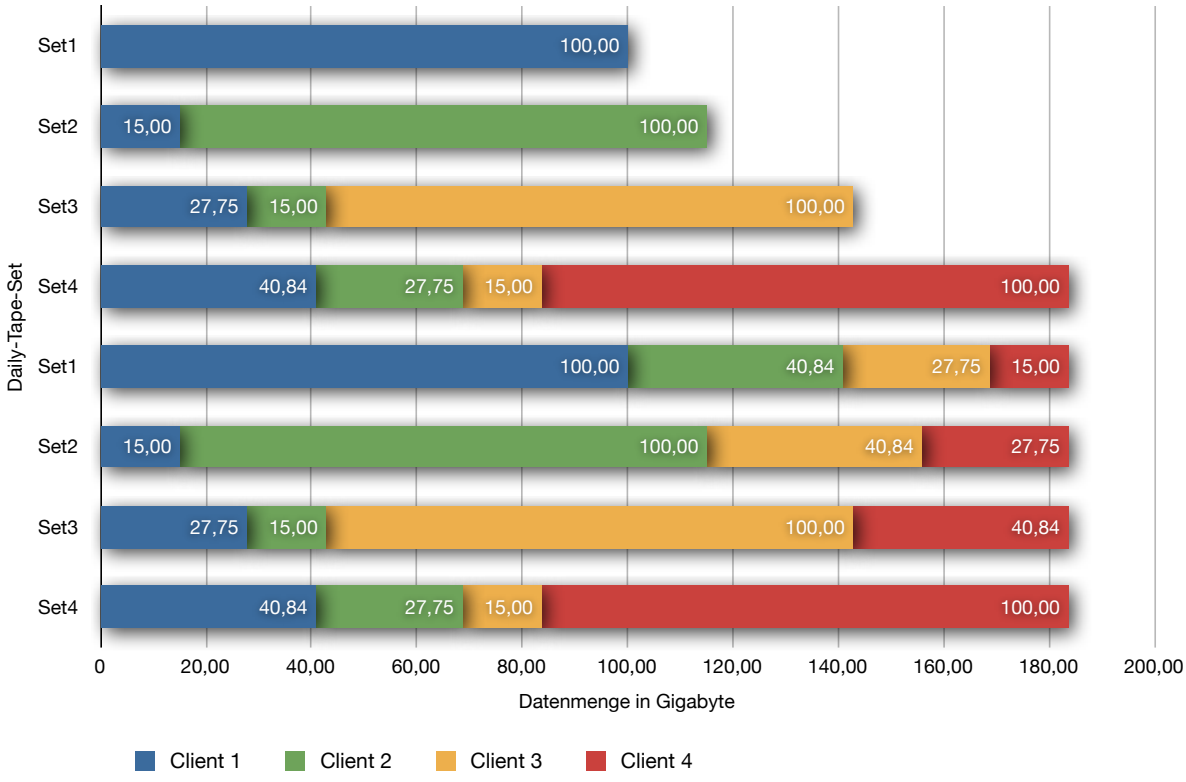

Abbildung 4.9: Beispiel einer Amanda Backup-Planung

[47] [Preston 2007]

Angenommen es existierten vier Clients, die, wie in Abbildung 4.8 auf Seite 55 schematisch dargestell, durch einen Amanda-Server gesichert werden und jeder Client hat 100 GB zu sichernde Daten die sich jeden Tag zu 15 % ändern. Der Dump Cycle beträgt in diesem Beispiel vier Tage[48].

Es ergibt sich für die Sicherung auf die Bänder DailySet1 bis DailySet4 eine Reihenfolge von Backupläufen wie sie auf Abbildung 4.9 auf Seite 57 zu sehen ist. Da der Dump Cycle vier Tage beträgt und es vier zu sichernde Clients gibt, muss jeden Tag ein Viertel der Clients ein volles Backup durchführen. Dies geschieht in den ersten vier Tagen nacheinander für jeden Client. Nach den ersten vier Tagen erfolgt dann jeweils ein volles Backup eines Clients und inkrementelle Backups der übrigen.

So gelangt das Backup-System, bei gleich bleibender Änderungsrate und Clientanzahl, nach vier Tagen zu einer kontinuierlich zu sichernden Datenmenge in Höhe von 183,59 Gigabyte. Dabei bleibt das ursprüngliche Rotationsprinzip bestehen und die vorgegebenen Regeln werden beachtet. Die Tabelle 4.1 zeigt die einzelnen Läufe über zwei Dump Cycles, also acht Tage.

Tabelle 4.1: Beispiel für Backupdaten in Gigabyte

	Set1	Set2	Set3	Set4	Set1	Set2	Set3	Set4
Client 1	100,00	15,00	27,75	40,84	100,00	15,00	27,75	40,84
Client 2		100,00	15,00	27,75	40,84	100,00	15,00	27,75
Client 3			100,00	15,00	27,75	40,84	100,00	15,0
Client 4				100,00	15,00	27,75	40,84	100,00
Total	100,00	115,00	142,75	183,59	183,59	183,59	183,59	183,59

Bei dem vorgestellten Beispiel auf der Abbildung 4.9 und der Tabelle 4.1 auf Seite 58 ist zu beachten, dass im dritten inkrementellen Backuplauf von Client 1 nicht erneut die kompletten 15 Gigabyte gesichert werden müssen. Es werden vielmehr, wie auf Abbildung 4.10 auf Seite 59 schematisch dargestellt, 15 % der Daten geändert, die bereits am zweiten Tag in DailySet2 gesichert wurden.

Da Amanda diese Überlappung berücksichtigt, und weil $30\ \text{GB} - (15\ \text{GB} \cdot 15\%) = 27,75\ \text{GB}$, nimmt der erste Client am dritten Tag nicht 30 Gigabyte Platz in Anspruch, sondern nur 27,75 Gigabyte.

Aus dem gleichen Grund und weil $45\ \text{GB} - (27,75\ \text{GB} \cdot 15\%) = 40,84\ \text{GB}$, beträgt der Platzbedarf für den ersten Client im vierten Backuplauf 40,84 Gigabyte anstelle von glatten 45 Gigabyte.

Diese beispielhaften Zahlen sind natürlich nur unzureichend geeignet die Realität wiederzugeben, bieten aber dennoch einen guten Einblick in die generelle Funktionsweise

[48] [Frisch 2002]

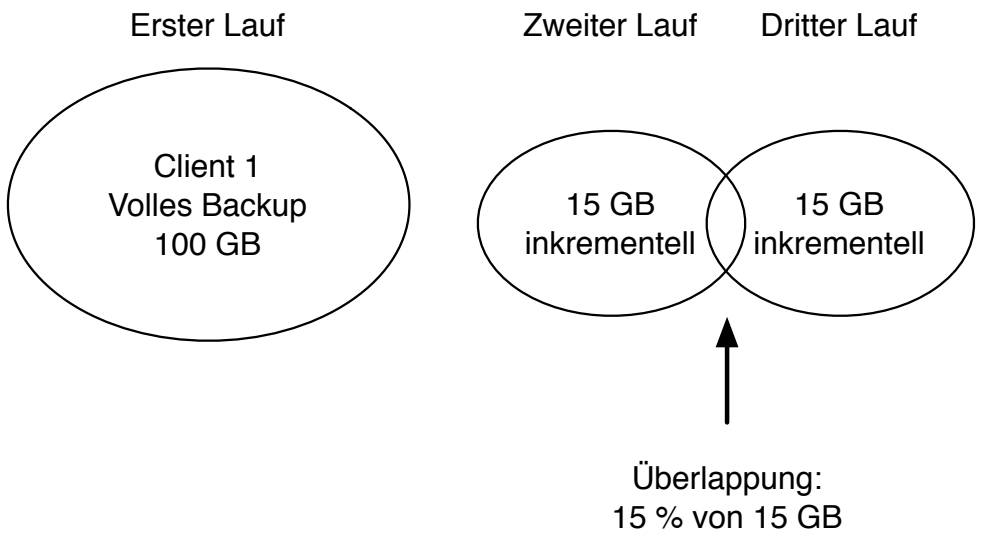

Abbildung 4.10: Überlappung von Daten zwischen zwei Backupläufen

der Backup-Planung durch Amanda. In der Realität sind feste Änderungsraten von Daten äußerst selten und in manchen Umgebungen werden täglich Clients hinzugefügt und entfernt. Außerdem benutzt Amanda nicht nur, wie in dem obigen Beispiel angenommen, inkrementelle Backups des ersten Levels, sondern vielmehr alle neun Level, um den Platzbedarf der Backups höchstmöglich zu optimieren.

Durch diese Eigenschaft vermeidet Amanda, dass es einen vollen Backuplauf pro Woche gibt, der alle Ressourcen bis an ihre Leistungsgrenzen belastet, diese während der restlichen Woche, in der ausschließlich inkrementelle Backups erstellt werden, aber kaum ausgenutzt werden. Mit dem von Amanda geplanten, immer gleichen Backupfenster sind strategische Entscheidungen leichter zu treffen, da die Ansprüche an die Ressourcen nie überproportional und plötzlich ansteigen, sondern langsam und stetig. Eine Anpassung des Dump Cycles oder der zugrundeliegenden Hardware wird so erst dann nötig, wenn Amanda alle zur Verfügung stehenden Ressourcen bestmöglich ausgelastet hat.

Wie bereits am Anfang dieses Abschnittes erläutert, verfügt Amanda, über die beiden hier beschriebenen Eigenschaften hinaus, über einen beeindruckenden Fähigkeitenkatalog. Durch die fast 20 Jahre Entwicklungszeit und den Open Source Prozess der Entwicklung ist die Qualität der Software außerordentlich hoch und die zukünftige Erweiterung der Software, wie etwa die Offsite-Sicherung auf Internet-Services, gesichert. Dazu kommt die Möglichkeit, externe Unterstützung hinzuziehen zu können, wenn das eigene Wissen nicht mehr ausreicht um eine Problemstellung zu lösen.

Amanda ist daher bestens geeignet, um große Client- und Server-Umgebungen im Enterprise-Bereich zuverlässig zu sichern und wiederherzustellen. Es steht kommerziellen Pendants weder in seinen Fähigkeiten, noch in seinen Support-Möglichkeiten nach.

5 Ausblick und Fazit

In diesem letzten Kapitel der Arbeit wird ein Ausblick auf die zukünftigen Entwicklungen im Bereich Backup & Restore gewagt. Dabei wird der Fokus einerseits auf die grundlegende und sehr systemnah angesiedelte Unterstützung von Backups auf Dateisystemebene mit *ZFS*[1] gelegt. Andererseits wird auch die Entwicklung der Backup-Services im Internet betrachtet. Diese so genannten Cloud-Backups erfreuen sich, gerade in der jüngsten Vergangenheit, zunehmender Beliebtheit.

Abgeschlossen wird die Arbeit dann mit einem Fazit der gesammelten Erkenntnisse.

5.1 Zukünftige Entwicklung

5.1.1 ZFS - Das Backupdateisystem?

ZFS wird seit 2006 von Sun als Standard-Dateisystem für die Betriebssysteme der Solaris-Familie eingesetzt. Es wird unter einer Lizenz entwickelt, die es erlaubt ZFS auch in Betriebssystemen zu implementieren, die nicht von Sun hergestellt werden und hat daher auch den Weg in einige andere Betriebssysteme gefunden[2].

Fähigkeiten von ZFS

ZFS ist ein 128-Bit Dateisystem und bietet daher eine praktisch unbegrenzte Dateisystemgröße von 256 Zettabyte[3]. Einzelne Dateien können bis zu 16 Exabyte[4] groß sein. Es integriert RAID-Funktionalität und damit Ausfallsicherheit bei Hardwaredefekten direkt auf Dateisystemebene. Dies geschieht durch einen völlig neuen Ansatz in der Verwaltung von Partitionen oder Volumes auf Festplatten. Traditionelle Dateisysteme erstellen auf der physikalisch vorhandenen Festplatte jeweils eine oder mehrere Partitionen, die dann jeweils mit einem Dateisystem formatiert werden, damit sie durch ein Betriebssystem

[1] Zettabyte File System
[2] [Dawidek 2007]
[3] Das entspricht einem Dateisystem von 274.877.906.944 Terabyte Größe
[4] Dieser Wert entspricht 16.777.216 Terabyte

genutzt werden können. Wenn mehrere Volumes oder Festplatten zu einem großen Dateisystem zusammengefasst werden sollen, ist der Einsatz von weiterer Soft- oder Hardware für das Management unvermeidlich.

In ZFS wird dagegen aus beliebig vielen physikalischen Festplatten ein großer Storage Pool gebildet. Aus diesem Storage Pool werden dann beliebig viele einzelne ZFS-Dateisysteme gebildet, die untereinander in beliebigen RAID-Konstellationen stehen können. Dabei achtet ZFS selbst darauf, dass die physikalische Ausfallsicherheit für alle Dateisysteme, die abhängig voneinander sind, gegeben ist[5]. Es wird also nie eine gespiegelte Konstellation auf der gleichen physikalischen Festplatte abgebildet. Eine schematische Darstellung des Unterschieds zwischen traditionellen Dateisystemen und ZFS findet sich auf der Abbildung 5.1.

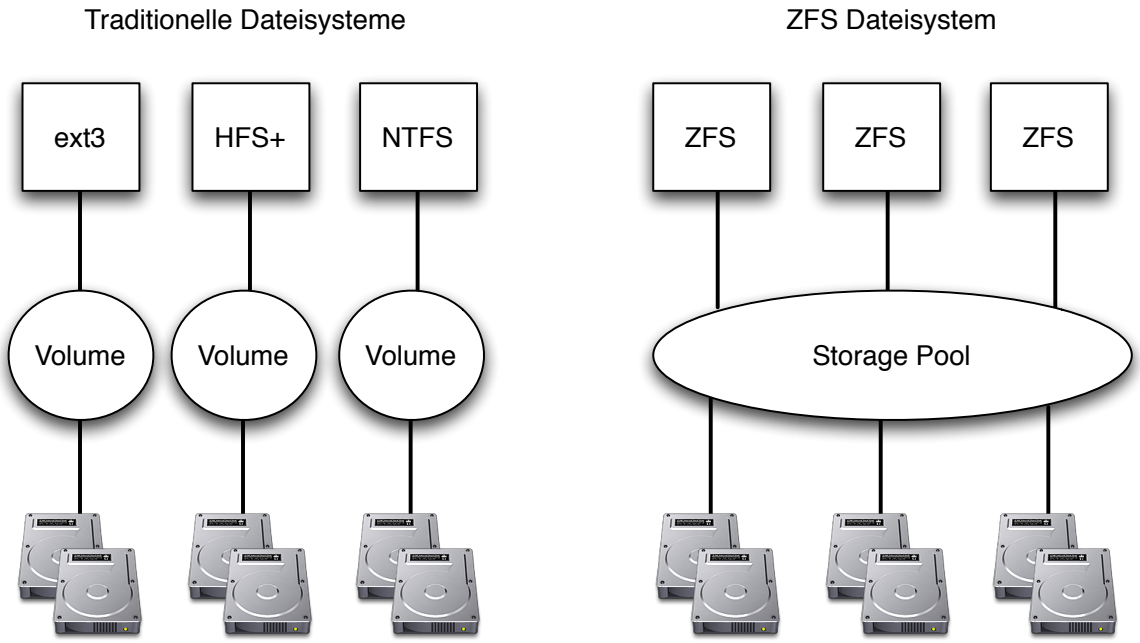

Abbildung 5.1: ZFS Volumemanagement vs. traditionelle Dateisysteme

Diese Eigenschaften machen es für die Aufgabe, Backups vorzuhalten, bereits sehr interessant. Einmalig ist aber die simple Erstellung und Verwaltung von Snapshots auf Dateisystemebene. Mit sehr einfachen administrativen Mitteln ist es möglich, Snapshots eines kompletten Dateisystems anzulegen, zu verwalten und bei Bedarf auch wiederherzustellen. Solange sich die Daten nicht verändern, nimmt so ein Snapshot, vergleichbar mit einem Hard Link, so gut wie keinen Speicherplatz ein[6]. Da alle Operationen auf Dateisystemebene geschehen, finden sie praktisch im Moment ihrer Ausführung statt.

[5] [Zisler 2007]
[6] Vgl. Abschnitt 4.1.2 „Time Machine“ ab Seite 39

Ein ZFS-Dateisystem namens `filesystem` lässt sich in dem Pool `rpool` mit dem Befehl `zfs create rpool/filesystem` anlegen[7]. Snapshots lassen sich sofort danach mit `zfs snapshot rpool/filesystem@name` anlegen und können beliebig benannt werden. Die Anzeige von Snapshots geschieht mit `zfs list`, eine gekürzte Ausgabe des Kommandos findet sich in Listing 5.1.

Listing 5.1: Anzeige von ZFS-Snaphots

```
1 # zfs list -t snapshot | grep "rpool/filesystem"
2 rpool/filesystem                  143M  3.92G   143M
3 rpool/filesystem@name
4 rpool/filesystem@dienstag
5 rpool/filesystem@zfs-auto-snap:frequent-2009-12-29-12:00
6 rpool/filesystem@zfs-auto-snap:hourly-2009-12-29-12:00
```

Auch die Wiederherstellung ist denkbar einfach mit `zfs rollback rpool/filesystem@name` machbar. Ein solcher Restore erfolgt, wie der Snapshot, praktisch auf der Stelle.

Alle erstellten Snapshots können, wie in Listing 5.2 zu sehen, außerdem über das Netzwerk in andere Storage-Pools gesendet werden. So ist über die Redundanz des lokalen Storage-Pools hinaus eine erweiterte Ausfallsicherheit möglich.

Listing 5.2: Übertragung eines ZFS-Snaphots über das Netzwerk

```
1 # zfs send rpool/filesystem@name | ssh 192.168.5.183 "/usr/
    sbin/zfs receive rpool/restore"
```

Da Snapshots so auch auf andere Hosts übertragen werden können, ist ein Restore selbst nach einem kompletten Verlust der Hardware einfach möglich.

Für Anwender steht neben den Kommandozeilen-Werkzeugen auch eine grafische Oberfläche namens `Time Slider` bereit[8].

Dieses Werkzeug muss nur einmal eingeschaltet werden und ist standardmäßig bereits sehr sinnvoll vorkonfiguriert. Auf der Abbildung 5.2 auf der Seite 63 ist die Standardkonfiguration für den weiter oben erstellten Pool `rpool` zu sehen. Es ist hier auch möglich, bestimmte Dateisystempfade von der Erstellung von Snapshots auszuschließen. Dies kann etwa bei rein temporär genutzten Daten, die sich schnell ändern, sinnvoll sein, um Speicherplatz zu sparen.

Die Standardkonfiguration veranlasst auch die Aktivierung der schon in Listing 5.1 zu sehenden Auto-Snapshots und beinhaltet die folgenden Rotationszyklen solange noch mehr als 80 % des Dateisystems unbelegt sind[9]:

[7] [Kay 2009]
[8] [Chénedé 2008]
[9] [Foster 2008]

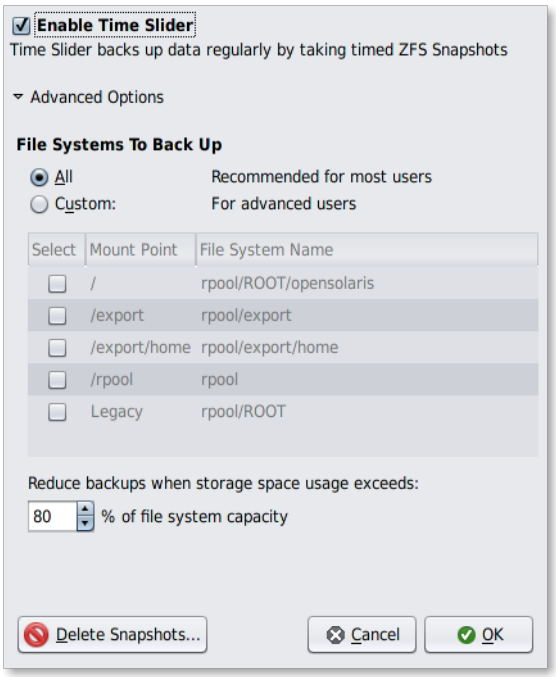

Abbildung 5.2: Time Slider Manager

- `frequent`: Alle 15 Minuten ein Snapshot, es werden die letzten vier Snapshots aufgehoben.
- `hourly`: Jede Stunde ein Snapshot, es werden die letzten 24 Snapshots aufgehoben.
- `daily`: Pro Tag ein Snapshot, es werden 31 Snapshots aufgehoben.
- `weekly`: Pro Woche ein Snapshot, die letzten vier Snapshots werden aufgehoben.
- `monthly`: Pro Monat ein Snapshot, es werden 12 Snapshots aufgehoben.

Dieses Rotationsschema erinnert sehr an das bereits von `rsnapshot`[10] bekannte, ist aber im Gegensatz zu diesem auf Dateisystemebene implementiert und deutlich leichter zu konfigurieren.

Die Wiederherstellung von Dateien und Ordnern aus Snapshots funktioniert über die auf Abbildung 5.3 auf Seite 64 zu sehende Oberfläche. Mit dem großen Schieberegler können die Snapshots nach Erstellungsdatum durchsucht werden. Der Restore selber geschieht dann durch simples kopieren der benötigten Dateien oder Verzeichnisse. Außerdem ist es möglich, manuell einen Snapshot anzulegen oder nicht mehr benötigte Snapshots zu löschen.

[10] Vgl. Abschnitt 4.3.1 „rsnapshot“ ab Seite 49

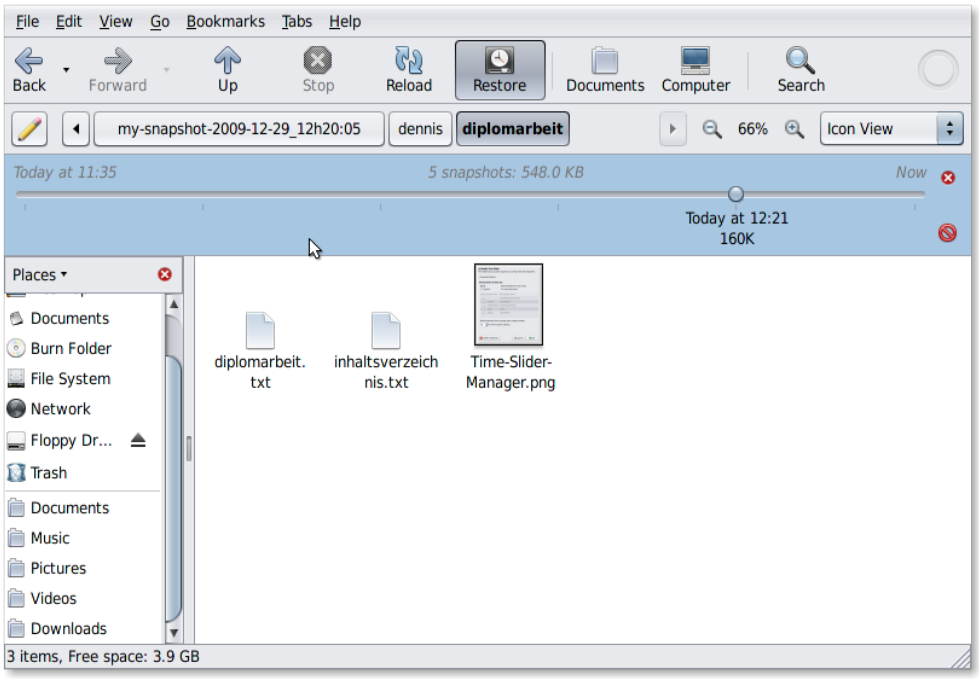

Abbildung 5.3: Time Slider in Aktion

Wird ein ZFS-Dateisystem auf einem Server eingesetzt, der Windows-Clients Dateien- und Verzeichnis-Freigaben über das Netzwerk bietet, zeigt sich die Verwandtschaft der ZFS Snapshots mit den Schattenkopien[11]. Windows Clients sind dann ohne weitere Software oder Konfigurationsbemühungen in der Lage, die von ZFS erstellten Snapshots über die nativen Windows-Werkzeuge zu durchsuchen und Restore-Vorgänge durchzuführen[12]. Zwar fehlen auf der Windows Seite die Namen der Snapshots, die jeweilige Erstellungszeit wird aber angezeigt und auch das Durchsuchen von Versionen einer einzelnen Datei ist mögich.

Diese Interoperabilität bewirkt, dass die Vorteile von ZFS auch auf Betriebssystemen zur Verfügung stehen, die zwar weit verbreitet sind, jedoch keine native Unterstützung für ZFS anbieten.

Probleme von ZFS

Bei allen technischen Vorzügen hat ZFS auch mit Problemen zu kämpfen, die seiner Zukunft als Dateisystem, nicht nur für Backups, sehr abträglich sein könnten. So ist es bisher aus lizenzrechtlichen Gründen nicht möglich, ZFS in den Linux-Kernel zu imple-

[11] Vgl. Abschnitt 4.1.1 „Backup und Restore Center“ auf Seite 35
[12] [Wright 2009]

mentieren[13]. Ohne direkt mit dem Kernel arbeiten zu können, ist die Benutzung von ZFS zwar auch möglich, jedoch mit teilweise empfindlichen Performanceeinbußen verbunden. Dass die technische Möglichkeit besteht, zeigt die Implementierung von ZFS in die Betriebssysteme der BSD-Familie, die über eine liberalere Lizenzierungspolitik verfügen.

Die Hauptentwicklungsfirma Sun wurde außerdem von Oracle aufgekauft[14]. Oracle arbeitet an einem Dateisystem namens *btrfs*[15], welches über einen sehr ähnlichen Funktionsumfang wie ZFS verfügt. Zwar ist die Entwicklung von btrfs noch lange nicht so weit wie die von ZFS, die Ziele sind aber durchaus vergleichbar[16]. Ob Oracle beide Dateisysteme parallel weiter entwickelt, ist heute noch nicht zu sagen.

Auch die ursprünglichen Pläne von Apple, ZFS in Mac OS X zu integrieren, scheiterten offensichtlich an Lizenzproblemen. Nachdem es in der OS X Version 10.5 bereits experimentelle Lesefähigkeiten für ZFS gab, sind diese in Version 10.6 wieder verschwunden. Auch die Entwicklungsseite unter `zfs.macosforge.org` wurde nach einer kurzen Meldung über die Einstellung der Entwicklung im Juni 2009 vom Netz genommen und ist heute nur noch über Webarchive zugänglich[17].

So ist es in der Tat fraglich, ob eine breitere Anwenderschaft in Zukunft die beeindruckenden Fähigkeiten von ZFS nutzen können wird. Die technischen Voraussetzungen sind sehr gut, das Dateisystem ist klar für die Zukunft positioniert. Allein die lizenzrechtlichen Bedenken auf Seiten anderer Betriebssystementwickler und die unklare zukünftige Geschäftspolitik von Sun und Oracle trüben die Zukunftsaussichten von ZFS.

5.1.2 Cloud-Backup

Cloud-Backup bedeutet nichts anderes als Offsite-Backups auf Servern im Internet zu speichern. Durch das aktuelle Hype-Thema Cloud-Computing hat sich der neue Name anstelle der Bezeichnung Online-Backup eingebürgert.

Amazon S3

Einer der größten Anbieter von Speicherplatz ist der bekannte Online-Händler Amazon mit seinem S3 Angebot[18]. Dieser vermarktet den Speicherplatz seiner riesigen Infrastruktur zu sehr attraktiven Preisen. So kostet die Speicherung eines Gigabytes 15 US-Cent

[13] [Andrews 2007]
[14] [Tillman und Lobo 2009]
[15] B-tree File System
[16] [Aurora 2009]
[17] [Dilger 2009]
[18] [Amazon Corporation 2009]

pro Monat. Der Transfer von Daten kostet ca. 10 US-Cent pro Gigabyte. Die Preise ändern sich leicht nach dem Volumen und dem Speicherort[19] der Daten. Für den Transfer von sehr großen Datenmengen bietet Amazon seinen Anwendern außerdem an, Festplatten einzusenden und deren Inhalt gegen Gebühr auf die S3-Infrastruktur zu kopieren.

Der dort angebotene Speicherplatz kann von bereits etablierten Backup-Lösungen wie Duplicity[20] oder Amanda[21] als zusätzliches Ziel genutzt werden, um eine Offsite-Lösung ohne eigene Infrastruktur zu etablieren. Ebenso ist es mit verschiedenen Softwarelösungen möglich, den Speicherplatz auf dem lokalen System einzubinden und ein ganz eigenes Backup-System für das Offsite-Backup darauf aufzusetzen.

Dieses Prinzip funktioniert sowohl für Anwender von Einzelplatzsystemen, die S3 als alleiniges Medium für ihre Backups nutzen, als auch für kleine und große Firmen die eine zusätzliche Sicherheitsschicht über ihre bestehenden Sicherheitsstrategien ziehen möchten.

Andere Cloud-Backup-Anbieter

Abbildung 5.4: Beispiel für eine Online-Backup-Software

Neben Amazon gibt es mittlerweile eine ganze Reihe von Anbietern, die sich darauf spezialisiert haben, Speicherplatz im Internet anzubieten. Das Vorgehen ist dabei immer

[19] Aktuell bietet Amazon zwei US-Standorte und einen in der EU an
[20] Vgl. Abschnitt 4.2.2 „Duplicity“ ab Seite 45 und die Zeilen 22 bis 24 des Listings 3 auf Seite x des Anhangs
[21] Vgl. Abschnitt 4.3.2 „Amanda“ ab Seite 54

gleich: Eine Software wird auf dem lokalen System installiert und für die Konfiguration der zu sichernden Verzeichnisse benutzt[22].

Ein Beispiel für die grafische Oberfläche einer solchen Software findet sich auf Abbildung 5.4 auf der Seite 66. Im Anschluss an die Installation startet der Transfer auf die Server des Anbieters. Dabei wurde vor nicht sehr langer Zeit meistens nur eine einfache Spiegelung der lokalen Daten veranlasst. Ein gutes Backup sollte dagegen immer einige Versionen vorhalten, um nicht nur vor Hardwareausfall, sondern auch vor versehentlichen Löschungen und anderen Gefahren zu schützen[23]. Die aktuellen Angebote bieten aber alle auch inkrementelle Sicherungen an, bei denen vorhergehende Versionen von veränderten Daten eine Zeit lang aufgehoben werden. Auf den Zeitraum hat der Anwender oftmals gar keinen Einfluss. Teilweise ist der Zeitraum über aufpreispflichtige Zusatzangebote erweiterbar.

Die meisten Anbieter bieten auch die Verschlüsselung der Daten, sowohl während der Übertragung, als auch für die Speicherung selber, an. Auf die Verschlüsselungsart und die Sorgfalt bei der Implementierung hat der Anwender hier genau so wenig Einfluss, wie auf die Zeitspanne, über welche die Daten aufgehoben werden. Gerade hier ist es in der Vergangenheit zu eklatanten Sicherheitsmängeln gekommen[24]. Außerdem ist unklar, inwieweit der Anbieter des Cloud-Backups selbst Einblick in die Daten seiner Anwender nehmen kann.Vorsichtige Naturen sollten hier bereits auf dem eigenen Computer Verschlüsselungstechniken einsetzen. Dazu kommen bei internationalen Anbietern auch noch unterschiedliche Datenschutzbestimmungen und sonstige gesetzliche Vorschriften, die teilweise erheblich von den deutschen und europäischen Rechtsnormen abweichen können.

Problematisch ist es, dass Anwender von Cloud-Backups nie genau wissen können, wo genau ihre Daten lagern und wer Zugriff darauf haben könnte. Backup ist Vertrauenssache, daher sollten Anwender und Firmen, die mit dem Gedanken spielen, ihre Backups offsite und online zu lagern, viel Zeit bei der Wahl des Anbieters investieren. Sollte dieser einmal sein Geschäft aufgeben und die Originaldateien genau zu diesem Zeitpunkt verlustig gehen, ist ein Restore kaum noch realisierbar.

Hier ist eine Abwägung zwischen dem Wert der zu sichernden Daten, den Kosten für ein Cloud-Backup und der Vorhaltung einer eigenen Offsite-Infrastruktur zu treffen.

Problemfeld Bandbreite

Ist einmal eine Entscheidung für eine Cloud-Backup-Lösung getroffen, ist das nächste Problem die zur Verfügung stehende Bandbreite für Backup & Restore. Sowohl private

[22] [Ochs 2009]
[23] Vgl. Abschnitt 2.1.2 „Gefahrenarten für Daten“ ab Seite 6
[24] [Bleich und Schmidt 2008]

Tabelle 5.1: Übertragung von 1 Terabyte Daten bei 80 % Nutzung der Bandbreite

Upload-Bandbreite	Benötigte Tage
T1 (1,544 Mbit/s)	82 Tage
10 Mbit/s	13 Tage
T3 (44,736 MBit/s)	3 Tage
100 Mbit/s	1-2 Tage
1000 Mbit/s	weniger als 1 Tag

Anwender als auch Firmen verfügen hierzulande in der Regel über mehr Download- als Upload-Geschwindigkeit. Dieses Missverhältnis macht das initiale Backup außerordentlich langwierig. Die gängigen Anbieter können zwar jederzeit pausieren und einen einmal begonnenen Upload wieder aufnehmen, ohne von vorne beginnen zu müssen, die gesamte Zeit für den ersten Upload kann trotzdem leicht Tage oder Wochen betragen. So dauert der Transfer von einem Terabyte Daten über eine 10 MBit/s Upload-Anbindung bei 80 % Auslastung der Bandbreite fast zwei Wochen. Beispiele für andere Bandbreiten können der Tabelle 5.1 entnommen werden[25]. Die Alternative, Datenträger über den Postweg zum Anbieter zu schicken, ist zwar effizienter, jedoch auch oftmals um ein Vielfaches teurer.

Durch die hohen Download-Bandbreiten ist der Restore von einzelnen Dateien oder Ordnern aus einem Cloud-Backup in der Regel deutlich schneller, als eine Sicherung von einem, eventuell sogar offsite gelagerten, Magnetband einzuspielen. Das gilt aber nur, solange die Bandbreite hoch genug oder die Dateien klein genug sind. Sobald es in den Bereich von einigen Gigabyte wiederherzustellender Daten geht, kehrt sich das Verhältnis um. Für alle Anwender ist es außerdem schwierig, eine Wiederherstellung des kompletten Systems nach einem Hardwaredefekt durchzuführen. Sofern die Möglichkeit überhaupt gegeben ist, dauert es doch ungleich länger als ein Betriebsystem von einem lokalen Datenträger aufzuspielen.

Derzeit ist das Cloud-Backup als alleinige Sicherungsmaßnahme nicht zu empfehlen.

Einen vertrauenswürdigen und auch sonst passenden Anbieter vorausgesetzt, ist Cloud-Backup aber unter Umständen sehr gut geeignet, um als Offsite-Backup zu dienen oder eine vorhandene Offsite-Infrastruktur zu ergänzen und zu entlasten[26].

Sowohl private als auch geschäftliche Anwender können in Zukunft von den günstigen Preisen und der Skalierbarkeit einer Lösung wie Amazon's S3 profitieren. Die Anbindung in vielen traditionellen Backup-Lösungen ist bereits gegeben und der Anbieter genießt das Vertrauen vieler Kunden.

[25] [Amazon Corporation 2009]
[26] [Whitehouse 2009]

Mit zunehmender Verbreitung von Breitbandzugängen zum Internet im Bereich von 100 Mbit/s und mehr, werden auch die kleineren Anbieter von Cloud-Backup-Lösungen ihr Geschäftsmodell besser umsetzen können[27]. Die Breitbandstrategie der Bundesregierung sieht immerhin vor, dass bis zum Jahr 2014 75 % der deutschen Haushalte mit 50 Mbit/s Download-Bandbreite versorgt werden sollen[28]. Zahlen zu geplanten Upload-Bandbreiten nennt die Strategie jedoch nicht.

5.2 Fazit

Niemand verliert gerne seine Daten. Ganz egal ob es sich um Familienfotos oder die Kundendatenbank eines globalen Unternehmens handelt.

Diese Arbeit hat gezeigt, wie essentiell wichtig es für jeden Computeranwender ist, durch gute Backups ein ausreichendes Maß an Datensicherheit zu gewährleisten. Für verschiedene, beispielhafte Fälle wurden Lösungen vorgestellt und ihre Fähigkeiten und Grenzen aufgezeigt. Wie relevant das Thema tatsächlich ist, wird anhand einiger Zahlen deutlicher:

2008 haben alleine die amerikanischen Konsumenten 3,6 Zettabyte[29] an Informationen konsumiert. Das entspricht im Durchschnitt 34 Gigabyte pro Tag und Person[30].

Bereits 2002 wurden weltweit etwa 5 Exabyte[31] an *neuen* Informationen produziert. Aktuellere Zahlen liegen hierzu bedauerlicherweise nicht vor. Führt man sich aber die Menge der konsumierten Informationen vor Augen, so ist davon auszugehen, dass diese Zahl eher gestiegen als gesunken ist. Von diesen neuen Informationen wurden 92 % auf Festplatten gespeichert[32]. Festplatten erleiden Defekte und fallen aus[33]. Die Frage ist nicht, ob es die eigene Festplatte treffen wird, die Frage ist nur wann es passieren wird. Und für genau diesen unerfreulichen Tag werden alle Backups der Welt angefertigt, überprüft und schließlich wiederhergestellt.

Leider sind Backup & Restore in der Praxis trotz allem immer noch ein Randthema. Oft wird erst zum Ende eines Systemdesigns bemerkt, dass es noch keine Lösung für die Sicherung der Daten und des Systems gibt. Das Ergebnis ist dann oftmals ein Flickenteppich aus einzelnen Backup-Lösungen die niemand administriert und überwacht. Ein erfolgreicher Restore-Vorgang wird so zum Glücksspiel.

---

[27] [Vinh 2009]
[28] [BMWI 2009]
[29] Das entspricht unvorstellbaren $3{,}865 \cdot 10^9$ Terabyte
[30] [Bohn und Short 2009]
[31] Das entspricht 5.242.880 Terabyte
[32] [Lyman und Varian 2003]
[33] [Pinheiro et al. 2007]

In der Literatur wird das Thema ebenfalls häufig stiefmütterlich behandelt. Informationen und Anweisungen über die Sicherung der Daten des behandelten Software-Produktes finden sich nur in eingeschobenen Kapiteln oder sogar nur im Anhang des jeweiligen Buches. Oftmals wird auch schlicht auf die Produkte von Drittanbietern verwiesen. Eine große Ausnahme ist das, auch in dieser Arbeit oft zitierte, Werk *Backup & Recovery* von Preston W. Curtis. Curtis beschreibt umfassende Lösungen und führt den geneigten Leser ausgesprochen unterhaltend an die Materie heran. Sollten Leser dieser Arbeit an weitergehender Lektüre interessiert sein, so sei ihnen dieses Werk besonders ans Herz gelegt.

Backup & Restore sind wichtige Bestandteile einer jeden umfassenden Sicherheitsstrategie, die IT-Technologie betrifft. Dies wird sich auch in der absehbaren Zukunft nicht ändern. Ganz im Gegenteil werden mit der zunehmenden Produktion von Informationen auch immer mehr Backup-Lösungen benötigt. Diese müssen in Zukunft nicht nur skalierbar sein, um die Massen an Daten zu bewältigen, sondern auch auf allen Ebenen gut bedienbar bleiben. Es muss sowohl Lösungen für den einzelnen Computer zuhause, als auch für global dezentralisierte Rechenzentren geben. Dazu sollte es einfach sein, von einem Produkt auf ein anderes zu wechseln, wenn sich die Ansprüche an die eingesetzte Backuplösung ändern. Um eine solche Migration zu ermöglichen, sollten Lösungen mit offenen, gut dokumentierten Formaten verwendet werden. Vom rohen Dateisystem, über die Komprimierung und die Verschlüsselung der Daten bis hin zu den eingesetzten Hardwareschnittstellen wird jeder Bestandteil benötigt, um im Falle eines Falles eine erfolgreiche Wiederherstellung zu ermöglichen. Fehlt nur ein Glied in der Kette, sind die Daten verloren.

Um diese Katastrophe zu vermeiden, sollte sich wirklich jeder Anwender Gedanken um seine Sicherungsstrategie machen. Vorhandene Lösungen müssen von Zeit zu Zeit überprüft werden, ob sie den eigenen Ansprüchen noch gerecht werden. Administratoren und sonstige Verantwortungsträger für die Daten Anderer sind hierbei in der Pflicht, besonders viel Sorgfalt walten zu lassen. Wenn diese Voraussetzungen erfüllt sind, können wir alle ein wenig beruhigter die nächsten Schritte auf dem Weg in das Informationszeitalter unternehmen.

Übrigens, wann war *Ihr* letztes Backup?

Literaturverzeichnis

[Amazon Corporation 2009] Amazon Corporation (2009). *Amazon Simple Storage Service Homepage*. Amazon Web Services, `http://aws.amazon.com/s3/`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 65, 68

[Anderson 2008] Anderson, Ross J. (2008). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, Indianapolis, 2. Aufl. Zitiert auf S. 27

[Andrews 2007] Andrews, Jeremy (2007). *Linux: ZFS, Licenses and Patents*. kerneltrap.org, `http://kerneltrap.org/node/8066`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 65

[Aurora 2009] Aurora, Valerie (2009). *A short History of btrfs*. lwn.net, `http://lwn.net/Articles/342892/`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 65

[Ballman 2001] Ballman, Janette (2001). *Merrill Lynch Resumes Critical Business Functions Within Minutes of Attack.*. In: Disaster Recovery Journal, Fall 2001. Zitiert auf S. 8

[Bleich und Schmidt 2008] Bleich, Holger und J. Schmidt (2008). *Web Tresore. Wackelige Sicherheit bei Online-Backup-Anbietern*. In: c't, (08/2008). Zitiert auf S. 67

[BMWI 2009] BMWI (2009). *Bundesministerium für Wirtschaft und Technologie - Breitbandstrategie der Bundesregierung*. Berlin. Zitiert auf S. 69

[Bögeholz 1999] Bögeholz, Harald (1999). *Leben am Abgrund. Nur regelmäßige Datensicherung schützt vor Verlusten*. In: c't, (11/1999). Zitiert auf S. 9

[Bohn und Short 2009] Bohn, Roger E. und J. E. Short (2009). *How much Information? 2009 Report on American Consumers*. Global Information Industry Center, University of California, San Diego. Zitiert auf S. 69

[Bott et al. 2009] Bott, Ed, C. Siechert und C. Stinson (2009). *Windows 7 Inside Out*. Microsoft Press, Redmond, 1. Aufl. Zitiert auf S. 36

[BSI 2005] BSI (2005). *Bundesamt für Sicherheit in der Informationstechnik - IT-Grundschutz-Katalog*. Bundesanzeiger-Verl., Köln, 10. Aufl. Zitiert auf S. 6, 27

[BSI 2008] BSI (2008). *Bundesamt für Sicherheit in der Informationstechnik - Informationssicherheit und IT-Grundschutz: BSI-Standards 100-1, 100-2 und 100-3*. Praxiswissen. Bundesanzeiger-Verl., Köln, 2., überarb. Aufl. Zitiert auf S. 27

[Burleson 1999] Burleson, Donald K. (1999). *Oracle SAP Administration*. O'Reilly, Cambridge, 1. Aufl. Zitiert auf S. 24

[Chénedé 2008] Chénedé, Erwann (2008). *ZFS Snapshot Visualization in GNOME*. blogs.sun.com, `http://blogs.sun.com/erwann/entry/zfs_on_the_desktop_zfs`(Abgerufen am: 28. Dezember 2009). Zitiert auf S. 62

[Cohasset Associates, Inc. 2005] Cohasset Associates, Inc. (2005). *LTO Ultrium WORM Magnetic Tape Technology*. LTO Whitepaper. Zitiert auf S. 5

[Cooperstein und Richter 1999] Cooperstein, Jeffrey und J. Richter (1999). *Keeping an Eye on Your NTFS Drives: the Windows 2000 Change Journal Explained*. In: Microsoft Systems Journal, `http://www.microsoft.com/msj/0999/journal/journal.aspx`(Abgerufen am: 12. November 2009). Zitiert auf S. 20

[Dawidek 2007] Dawidek, Pawel Jakub (2007). *Porting the ZFS file system to the FreeBSD operating system*. FreeBSD Whitepaper. Zitiert auf S. 60

[debian-administration.org 2005] debian-administration.org (2005). *Unattended, Encrypted, Incremental Network Backups: Part 1*. debian-administration.org, `http://www.debian-administration.org/articles/209`(Abgerufen am: 13. November 2009). Zitiert auf S. 46

[Dilger 2009] Dilger, Daniel Eran (2009). *Apple shuts down ZFS open source project*. AppleInsider, `http://www.appleinsider.com/articles/09/10/23/apple_shuts_down_zfs_open_source_project.html`(Abgerufen am: 29. Dezember 2009). Zitiert auf S. 65

[Dorion 2008] Dorion, Pierre (2008). *Tape Encryption FAQ*. searchdatabackup.com, `http://searchdatabackup.techtarget.com/generic/0,295582,sid187_gci1335639,00.html`(Abgerufen am: 24. November 2009). Zitiert auf S. 33

[Endres 2009a] Endres, Johannes (2009a). *Aufbausatz. Windows Home Server aufsetzen und ausbauen*. In: c't, (15/2009). Zitiert auf S. 45

[Endres 2009b] Endres, Johannes (2009b). *Innenarchitektur. Der passende Server für kleine Arbeitsgruppen*. In: c't, (15/2009). Zitiert auf S. 43

[Exabyte 2005] Exabyte (2005). *Disk-to-Disk-to-Tape: Where Disk Fits Into Backup*. Exabyte Whitepaper. Zitiert auf S. 30

[Fok 2007] Fok, Christine (2007). *A Guide to Windows Vista Backup Technolo-*

gies. In: Microsoft Tech Net Magazine, `http://technet.microsoft.com/en-us/magazine/2007.09.backup.aspx`(Abgerufen am: 20. November 2009). Zitiert auf S. 36

[Foster 2008] Foster, Tim (2008). *ZFS Auto Snapshot Readme*. blogs.sun.com, `http://blogs.sun.com/timf/resource/README.zfs-auto-snapshot.txt`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 62

[Frisch 2002] Frisch, Æleen (2002). *Essential System Administration*. O'Reilly, Cambridge, 3. erweiterte Aufl. Zitiert auf S. 58

[Garfinkel et al. 2003] Garfinkel, Simson, G. Spafford und A. Schwartz (2003). *Practical Unix and Internet Security*. O'Reilly, Cambridge, 3. erweiterte Aufl. Zitiert auf S. 6

[Gieselmann 2004] Gieselmann, Hartmut (2004). *Blaues Gedächtnis. Professionelle Datensicherung der nächsten Generation*. In: c't, (6/2004). Zitiert auf S. 5

[Kampffmeyer 2003] Kampffmeyer, Ulrich (2003). *Revisionssichere Archivierung und Dokumentenmanagement im Licht neuer rechtlicher Anforderungen*. In: *audicon und ErnstYoung Roadshow Mai 2003 "Die Intelligenz der digitalen Steuerprüfung nach den GDPdU"*. Zitiert auf S. 5

[Kay 2009] Kay, Dominic (2009). *Working with Solaris ZFS Snapshots*. Solaris 10 How To Guides. Zitiert auf S. 62

[Leber 1998] Leber, Jody (1998). *Windows NT Backup & Restore*. O'Reilly, Cambridge, 1. Aufl. Zitiert auf S. 19

[Luther 2004] Luther, Jörg (2004). *Backup mit Konzept*. TecChannel.de, `http://www.tecchannel.de/storage/backup/402355/backup_mit_konzept`(Abgerufen am: 02. November 2009). Zitiert auf S. 20, 21

[Lyman und Varian 2003] Lyman, Peter und H. R. Varian (2003). *How much Information? 2003*. International Computer Science Institute, University of California, Berkeley. Zitiert auf S. 69

[Microsoft Corporation 2003] Microsoft Corporation (2003). *How Volume Shadow Copy Service Works*. Microsoft Tech Net, `http://technet.microsoft.com/en-us/library/cc785914(WS.10,printer).aspx`(Abgerufen am: 20. November 2009). Zitiert auf S. 38

[Microsoft Corporation 2006] Microsoft Corporation (2006). *Single Instance Storage in Microsoft Windows Storage Server 2003 R2: A Solution for Managing Duplicate Files*. `http://download.microsoft.com/download/`

8/a/e/8ae7f07d-b888-4b17-84c3-e5a1976f406c/SingleInstanceStorage.doc(Abgerufen am: 17. Dezember 2009). Zitiert auf S. 44

[Microsoft Corporation 2007] Microsoft Corporation (2007). *Windows Home Server: Getting Started*. Zitiert auf S. 44

[Microsoft Corporation 2009] Microsoft Corporation (2009). *Using Windows Home Server to Back Up Windows Small Business Client Computers*. Microsoft Tech Net, http://technet.microsoft.com/en-us/library/ee378513(WS.10,printer).aspx(Abgerufen am: 16. Dezember 2009). Zitiert auf S. 44

[Ochs 2009] Ochs, Susie (2009). *Online Storage Battle: Which Cloud Back-Up Service Reigns Supreme?*. Maclife.com, http://www.maclife.com/article/reviews/online_storage_battle_which_cloud_backup_service_reigns_supreme(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 67

[Pinheiro et al. 2007] Pinheiro, Eduardo, W.-D. Weber und L. A. Barroso (2007). *Failure Trends in a Large Disk Drive Population*. In: *Proceedings of the 5th USENIX Conference on File and Storage Technologies*. Zitiert auf S. 5, 7, 69

[Preston 1999] Preston, W. Curtis (1999). *Unix Backup & Recovery*. O'Reilly, Cambridge, 1. Aufl. Zitiert auf S. 8, 12

[Preston 2007] Preston, W. Curtis (2007). *Backup & Recovery*. O'Reilly, Cambridge, 1. Aufl. Zitiert auf S. 12, 22, 54, 57

[Rauprich 2009] Rauprich, Thomas (2009). *Erstellung und Umsetzung eines IT-Konzeptes für den Bildungsbereich am Beispiel der städtischen Schulen in Essen*. Diplomarbeit, Fachhochschule für Oekonomie und Management (FOM), Essen. Zitiert auf S. 49

[Rosenquist 2005] Rosenquist, Nathan (2005). *rsnapshot HowTo*. http://rsnapshot.org/howto/1.2/rsnapshot-HOWTO.en.pdf(Abgerufen am: 17. Dezember 2009). Zitiert auf S. 50

[Ruebel 2004] Ruebel, Mike (2004). *Easy Automated Snapshot-Style Backups with Linux and Rsync*. http://www.mikerubel.org/computers/rsync_snapshots/(Abgerufen am: 17. Dezember 2009). Zitiert auf S. 49

[Rütten 2006] Rütten, Christiane (2006). *Hinter Schloss und Siegel. Backups auf nicht vertrauenswürdige FTP-Server*. In: c't, (13/2006). Zitiert auf S. 46

[Schmidt 2006] Schmidt, Jürgen (2006). *Beruhigungsmittel. Backups für kleine Linux-Server*. In: c't, (07/2006). Zitiert auf S. 41

[Schneier 2005] Schneier, Bruce (2005). *Public Disclosure of personal Data Loss*.

Schneier on Security, `http://www.schneier.com/blog/archives/2005/06/public_disclosu.html`(Abgerufen am: 02. November 2009). Zitiert auf S. 30

[Shackelford 2006] Shackelford, Daniel (2006). *Ruby script for creating symlinks*. rsnaphsot discuss mailinglist, `https://lists.sourceforge.net/lists/listinfo/rsnapshot-discuss`(Abgerufen am: 21. Dezember 2009). Zitiert auf S. 53

[Siracusa 2007] Siracusa, John (2007). *Mac OS X 10.5 Leopard: The Ars Technica Review*. arstechnica.com, `http://arstechnica.com/apple/reviews/2007/10/mac-os-x-10-5.ars/`(Abgerufen am: 14. Dezember 2009). Zitiert auf S. 39

[Tillman und Lobo 2009] Tillman, Karen und R. Lobo (2009). *Oracle buys Sun*. Oracle Press Release, `http://www.oracle.com/us/corporate/press/018363`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 65

[Vinh 2009] Vinh, Khoi (2009). *Backing Up Over Broadband*. subtraction.com, `http://www.subtraction.com/2009/12/10/backing-up-over-broadband`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 69

[Whitehouse 2009] Whitehouse, Lauren (2009). *Cloud backup - the pros and cons*. searchstorage.techtarget.com.au, `http://searchstorage.techtarget.com.au/articles/30320-Cloud-backup-the-pros-and-cons`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 68

[Wright 2009] Wright, Alan (2009). *Using Windows Previous Versions to access ZFS Snapshots*. blogs.sun.com, `http://blogs.sun.com/amw/entry/using_the_previous_versions_tab`(Abgerufen am: 30. Dezember 2009). Zitiert auf S. 64

[Yurin 2007] Yurin, Maxim (2007). *The History of Backup*. backuphistory.com, `http://www.backuphistory.com`(Abgerufen am: 23. Juli 2009). Zitiert auf S. 13

[Zisler 2007] Zisler, Harald (2007). *Solaris 10 & OpenSolaris*. Franzis, Poing, 1. Aufl. Zitiert auf S. 61

[Zivadinovic und Beier 2009] Zivadinovic, Dusan und A. Beier (2009). *In Lauerstellung. Dokumentenbackup mit c't-TriggerBack für Mac OS X*. In: c't, (22/2009). Zitiert auf S. 10

[Zwicky 1991] Zwicky, Elizabeth D. (1991). *Torture-testing Backup and Archive Programs: Things You Ought to Know But Probably Would Rather Not*. Zitiert auf S. 12

Abkürzungsverzeichnis

Amanda ..... Advanced Maryland Automatic Network Disk Archiver. Zuerst benutzt auf S. 49

btrfs ..... B-tree File System. Zuerst benutzt auf S. 65

CD ..... Compact Disc. Zuerst benutzt auf S. 5

D2D2T ..... Disk to Disk to Tape. Zuerst benutzt auf S. 30

DB2 ..... DataBase 2. Zuerst benutzt auf S. 24

DHCP ..... Dynamic Host Configuration Protocol. Zuerst benutzt auf S. 50

DVD ..... Digital Versatile Disc. Zuerst benutzt auf S. 5

FTP ..... File Transfer Protocol. Zuerst benutzt auf S. 46

GPG ..... GNU Privacy Guard. Zuerst benutzt auf S. 46

LDAP ..... Lightweight Directory Access Protocol. Zuerst benutzt auf S. 50

MySQL ..... My Structured Query Language. Zuerst benutzt auf S. 24

NAS ..... Network Attached Storage. Zuerst benutzt auf S. 15

NTFS ..... New Technology Filesystem. Zuerst benutzt auf S. 38

RAID ..... Redundant Array of Independent Disks. Zuerst benutzt auf S. 7

RPO ..... Recovery Point Objective. Zuerst benutzt auf S. 13

RTO ..... Recovery Time Objective. Zuerst benutzt auf S. 12

S3 ..... Amazon Simple Secure Storage Service. Zuerst benutzt auf S. 55

SAN ..... Storage Area Network. Zuerst benutzt auf S. 15

SBS ..... Small Business Server. Zuerst benutzt auf S. 44

SLA ..... Service Level Agreement. Zuerst benutzt auf S. 24

SPOF ..... Single Point of Failure. Zuerst benutzt auf S. 33

SSH ..... Secure Shell. Zuerst benutzt auf S. 51

SSL ..... Secure Socket Layer. Zuerst benutzt auf S. 55

UDO ..... Ultra Density Optical. Zuerst benutzt auf S. 5

UNIVAC I ..... UNIVersal Automatic Computer I. Zuerst benutzt auf S. 13

USN Journal ... Update Sequence Number Journal. Zuerst benutzt auf S. 20

VSS ..... Volume Shadow Copy Service. Zuerst benutzt auf S. 38

WHS ..... Windows Home Server. Zuerst benutzt auf S. 43

WoL ..... Wake on LAN. Zuerst benutzt auf S. 44

WORM ..... Write Once Read Multiple (Times). Zuerst benutzt auf S. 5

ZFS ..... Zettabyte File System. Zuerst benutzt auf S. 60

Schlagwortverzeichnis

Anhang

Listing 3: Generisches Duply Profil

```
# gpg key data (for symmetric encryption comment out GPG_KEY)
GPG_KEY='_KEY_ID_'
GPG_PW='_GPG_PASSWORD_'
# gpg options passed from duplicity to gpg process (default='')
# e.g. "--trust-model pgp|classic|direct|always"
#   or "--compress-algo=bzip2 --bzip2-compress-level=9"
#GPG_OPTS=''

# credentials & server address of the backup target (URL-Format)
# syntax is
#   scheme://[user:password@]host[:port]/[/]path
# probably one out of
#   file:///some_dir
#   ftp://user[:password]@other.host[:port]/some_dir
#   hsi://user[:password]@other.host/some_dir
#   cf+http://container_name
#   imap://user[:password]@host.com[/from_address_prefix]
#   imaps://user[:password]@host.com[/from_address_prefix]
#   rsync://user[:password]@other.host[:port]::/module/some_dir
#   rsync://user[:password]@other.host[:port]/relative_path
#   rsync://user[:password]@other.host[:port]//absolute_path
#   # for the s3 user/password are AWS_ACCESS_KEY_ID/AWS_SECRET_ACCESS_KEY
#   s3://[user:password]@host/bucket_name[/prefix]
#   s3+http://[user:password]@bucket_name[/prefix]
#   scp://user[:password]@other.host[:port]/some_dir
#   ssh://user[:password]@other.host[:port]/some_dir
#   tahoe://alias/directory
#   webdav://user[:password]@other.host/some_dir
#   webdavs://user[:password]@other.host/some_dir
```

```
30 ###
31 TARGET='scheme://user[:password]@host[:port]/[/]path'
32 # optionally the username/password can be defined as extra
     variables
33 # setting them here _and_ in TARGET results in an error
34 #TARGET_USER='_backend_username_'
35 #TARGET_PASS='_backend_password_'
36
37 # base directory to backup
38 SOURCE='/path/of/source'
39
40 # Time frame for old backups to keep, Used for the "purge"
     command.
41 # see duplicity man page, chapter TIME_FORMATS)
42 # defaults to 1M, if not set
43 #MAX_AGE=1M
44
45 # Number of full backups to keep. Used for the "purge-full"
     command.
46 # See duplicity man page, action "remove-all-but-n-full".
47 # defaults to 1, if not set
48 #MAX_FULL_BACKUPS=1
49
50 # verbosity of output (5 for gpg errors, 9 for bug fixing)
51 # default is 4, if not set
52 #VERBOSITY=5
53
54 # temporary file space. at least the size of the biggest
     file in backup
55 # for a successful restoration process. (default is '/tmp',
     if not set)
56 #TEMP_DIR=/tmp
57
58 # sets duplicity --time-separator option (since v0.4.4.RC2)
     to allow users
59 # to change the time separator from ':' to another character
      that will work
60 # on their system.  HINT: For Windows SMB shares, use --time
     -separator='_'.
61 # NOTE: '-' is not valid as it conflicts with date separator
     .
62 # ATTENTION: only use this with duplicity < 0.5.10, since
     then default file
```

```
63 #            naming is compatible and this option is pending
     depreciation
64 #DUPL_PARAMS="$DUPL_PARAMS --time-separator _ "
65
66 # activates duplicity --short-filenames option, when
     uploading to a file
67 # system that can't have filenames longer than 30 characters
     (e.g. Mac OS 8)
68 # or have problems with ':' as part of the filename (e.g.
     Microsoft Windows)
69 # ATTENTION: only use this with duplicity < 0.5.10, later
     versions default file
70 #            naming is compatible and this option is pending
     depreciation
71 #DUPL_PARAMS="$DUPL_PARAMS --short-filenames "
72
73 # activates duplicity --full-if-older-than option (since
     duplicity v0.4.4.RC3)
74 # forces a full backup if last full backup reaches a
     specified age, for the
75 # format of MAX_FULLBKP_AGE see duplicity man page, chapter
     TIME_FORMATS
76 #MAX_FULLBKP_AGE=1M
77 #DUPL_PARAMS="$DUPL_PARAMS --full-if-older-than
     $MAX_FULLBKP_AGE "
78
79 # sets duplicity --volsize option (available since v0.4.3.
     RC7)
80 # set the size of backup chunks to VOLSIZE MB instead of the
     default 25MB.
81 # VOLSIZE must be number of MB's to set the volume size to.
82 #VOLSIZE=50
83 #DUPL_PARAMS="$DUPL_PARAMS --volsize $VOLSIZE "
84
85 # more duplicity command line options can be added in the
     following way
86 # don't forget to leave a separating space char at the end
87 #DUPL_PARAMS="$DUPL_PARAMS --put_your_options_here "
```

Listing 4: rsnapshot Konfigurationsdatei

```
1  #####################################################
2  # rsnapshot.conf - rsnapshot configuration file #
3  #####################################################
4  #                                                   #
5  # PLEASE BE AWARE OF THE FOLLOWING RULES:           #
6  #                                                   #
7  # This file requires tabs between elements          #
8  #                                                   #
9  # Directories require a trailing slash:             #
10 #   right: /home/                                   #
11 #   wrong: /home                                    #
12 #                                                   #
13 #####################################################
14
15 ##########################
16 # CONFIG FILE VERSION #
17 ##########################
18
19 config_version  1.2
20
21 ############################
22 # SNAPSHOT ROOT DIRECTORY #
23 ############################
24
25 # All snapshots will be stored under this root directory.
26 snapshot_root   /opt/rsnapshot/
27
28 # If no_create_root is enabled, rsnapshot will not
     automatically create the
29 # snapshot_root directory. This is particularly useful if
     you are backing
30 # up to removable media, such as a FireWire drive.
31 #
32 #no_create_root 1
33
34 #################################
35 # EXTERNAL PROGRAM DEPENDENCIES #
36 #################################
37
38 # LINUX USERS:   Be sure to uncomment "cmd_cp". This gives
     you extra features.
```

```
39 # EVERYONE ELSE: Leave "cmd_cp" commented out for
       compatibility.
40 #
41 # See the README file or the man page for more details.
42 #
43 cmd_cp          /bin/cp
44
45 # uncomment this to use the rm program instead of the built-
       in perl routine.
46 cmd_rm          /bin/rm
47
48 # rsync must be enabled for anything to work.
49 cmd_rsync       /usr/bin/rsync
50
51 # Uncomment this to enable remote ssh backups over rsync.
52 cmd_ssh         /usr/bin/ssh
53
54 # Comment this out to disable syslog support.
55 cmd_logger      /usr/bin/logger
56
57 # Uncomment this to specify a path to "du" for disk usage
       checks.
58 cmd_du          /usr/bin/du
59
60 ##########################################
61 #           BACKUP INTERVALS             #
62 # Must be unique and in ascending order #
63 # i.e. hourly, daily, weekly, etc.      #
64 ##########################################
65
66 # The interval names (hourly, daily, ...) are just names and
        have no influence
67 # on the length of the interval. The numbers set the number
       of snapshots to
68 # keep for each interval (hourly.0, hourly.1, ...).
69 # The length of the interval is set by the time between two
       executions of
70 # rsnapshot <interval name>, this is normally done via cron.
71 # Feel free to adapt the names, and the sample cron file
       under /etc/cron.d/rsnapshot
72 # to your needs. The only requirement is that the intervals
       must be listed
73 # in ascending order. To activate just uncomment the entries
       .
```

```
74
75 #interval        hourly  6
76 interval         daily   7
77 interval         weekly  4
78 interval         monthly 6
79
80 ###############################################
81 #               GLOBAL OPTIONS                #
82 # All are optional, with sensible defaults    #
83 ###############################################
84
85 # If your version of rsync supports --link-dest, consider
       enable this.
86 # This is the best way to support special files (FIFOs, etc)
        cross-platform.
87 # The default is 0 (off).
88 # In Debian GNU cp is available which is superior to
       link_dest, so it should be
89 # commented out (disabled).
90 #
91 #link_dest      0
92
93 # Verbose level, 1 through 5.
94 # 1    Quiet           Print fatal errors only
95 # 2    Default         Print errors and warnings only
96 # 3    Verbose         Show equivalent shell commands being
       executed
97 # 4    Extra Verbose   Show extra verbose information
98 # 5    Debug mode      More than you care to know
99 #
100 verbose         3
101
102 # Same as "verbose" above, but controls the amount of data
       sent to the
103 # logfile, if one is being used. The default is 3.
104
105 loglevel        3
106
107 # If you enable this, data will be written to the file you
       specify. The
108 # amount of data written is controlled by the "loglevel"
       parameter.
109 logfile /var/log/rsnapshot.log
110
```

```
111 # The include and exclude parameters, if enabled, simply get
      passed directly
112 # to rsync. If you have multiple include/exclude patterns,
     put each one on a
113 # seperate line. Please look up the --include and --exclude
     options in the
114 # rsync man page for more details.
115 #
116 #include         ???
117 #include         ???
118 #exclude         ???
119 #exclude         ???
120
121 # The include_file and exclude_file parameters, if enabled,
     simply get
122 # passed directly to rsync. Please look up the --include-
     from and
123 # --exclude-from options in the rsync man page for more
     details.
124 #
125 #include_file    /path/to/include/file
126 #exclude_file    /path/to/exclude/file
127
128 # Default rsync args. All rsync commands have at least these
      options set.
129 #
130 #rsync_short_args        -a
131 #rsync_long_args         --delete --numeric-ids --relative --
     delete-excluded
132
133 # Enable reporting for rsync
134 #
135 rsync_long_args --delete --numeric-ids --relative --delete-
     excluded --stats
136
137
138 # ssh has no args passed by default, but you can specify
     some here.
139 #
140 #ssh_args        -p 22
141
142 # Default arguments for the "du" program (for disk space
     reporting).
```

```
143 # The GNU version of "du" is preferred. See the man page for
      more details.
144 #
145 #du_args        -csh
146
147 # If this is enabled, rsync won't span filesystem partitions
      within a
148 # backup point. This essentially passes the -x option to
     rsync.
149 # The default is 0 (off).
150 #
151 #one_fs         0
152
153 # If enabled, rsnapshot will write a lockfile to prevent two
      instances
154 # from running simultaneously (and messing up the
     snapshot_root).
155 # If you enable this, make sure the lockfile directory is
     not world
156 # writable. Otherwise anyone can prevent the program from
     running.
157 #
158 lockfile        /var/run/rsnapshot.pid
159
160 ###################################
161 ### BACKUP POINTS / SCRIPTS ###
162 ###################################
163
164
165 ############################################
166
167 # Beispielserver I
168 #
169 # LDAP-Backup
170 #
171 backup_script   /opt/scripts/server_i_ldap_backup.sh
            server_i_ldap/
172 #
173 # MySQL-Backup
174 #
175 backup_script   /opt/scripts/server_i_mysql_backup.sh
            server_i_mysql/
176 #
177 # Dateisystem-Backup
```

```
178 #
179 backup  192.168.1.1:/home/schueler_ordner/
      server_i/
180 backup  192.168.1.1:/home/lehrer_ordner/
      server_i/
181 backup  192.168.1.1:/home/klassen_ordner/
      server_i/
182 backup  192.168.1.1:/home/pool/
              server_i/
183
184 ##############################################
185
186
187 # Beispielserver ii
188 #
189 # LDAP-Backup
190 #
191 backup_script   /opt/scripts/server_ii_ldap_backup.sh
                      server_ii_ldap/
192 #
193 # MySQL-Backup
194 #
195 backup_script   /opt/scripts/server_ii_mysql_backup.sh
                      server_ii_mysql/
196 #
197 # Dateisystem Backup
198 #
199 backup  192.168.1.2:/home/schueler_ordner/
      server_ii/
200 backup  192.168.1.2:/home/lehrer_ordner/
      server_ii/
201 backup  192.168.1.2:/home/klassen_ordner/
      server_ii/
202 backup  192.168.1.2:/home/pool/
              server_ii/
203
204 ##############################################
```

Listing 5: Browse Backup Script

```
#!/usr/bin/ruby

base_dir = "/opt/rsnapshot/"
link_dir = "/opt/browse_backup/"
rs_conf = "/etc/rsnapshot.conf"
conf = File.open(rs_conf)
folders = []
backups = conf.select { |line| /^backup/ =~ line}
backups.each { |entry|
   name = entry.split[2].chop!
   folders << name
}

folders.each {|folder|
   Dir.open(base_dir).each {|entry|
       unless entry.include? 'lost+found' or entry == '.' or
           entry == '..'
           mtime = File.open(base_dir+entry).mtime
           link_name = mtime.mon.to_s+"-"+mtime.day.to_s+"-"
              +mtime.year.to_s
           begin
               if File.symlink?(link_dir+folder+"/"+
                  link_name)
                   p "Removing old links for #{folder}"
                   File.delete(link_dir+folder+"/"+link_name
                      )
               end
               p "Creating new links for #{folder}"
               File.symlink(base_dir+entry+"/"+folder+"/"+
                  folder, link_dir+folder+"/"+link_name)
           rescue SyntaxError, NameError, StandardError
               p "links creation failed! Error: #{$!}"
           end
       end
   }
}
```

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Ich erkläre mich damit einverstanden, dass die Arbeit der Öffentlichkeit zugänglich gemacht wird. Ich erkläre mich damit einverstanden, dass die Digitalversion dieser Arbeit zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

12. Januar 2010, Dennis Wegner