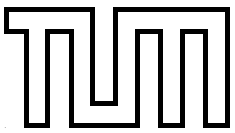


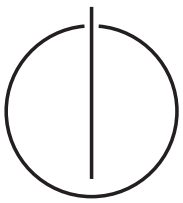
Technische Universität München

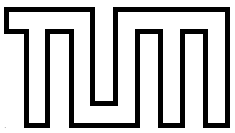
Fakultät für Informatik

Diplomarbeit in Informatik

Analyse und Erstellung eines Datenmodells und Konzeption einer geeigneten Werkzeug-Architektur für das Configuration Management am LRZ

Markus Gillmeister





Technische Universität München

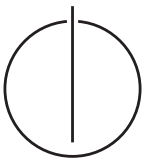
Fakultät für Informatik

Diplomarbeit in Informatik

Analyse und Erstellung eines Datenmodells und Konzeption einer geeigneten Werkzeug-Architektur für das Configuration Management am LRZ

Analysis and creation of a data model and conception of a appropriate toolset for the configuration management at the LRZ

Bearbeiter:	Markus Gillmeister
Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Dr. Michael Brenner (LRZ) Silvia Knittl (TU München) Christian Richter (LRZ)
Abgabedatum:	15. September 2009



Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. September 2009

. . . . . .
(Unterschrift des Kandidaten)

Abstract

Die stetig wachsenden Anforderungen an IT-Dienstleister nach immer höherer Qualität (Ausfallsicherheit, schnelle Bearbeitung von Störungen) führen zur Einführung einer prozessorientierten IT nach Best-Practices wie ITIL, CobiT oder der Norm ISO/IEC20000. Durch gewachsene Strukturen besteht für die Dienstleister jedoch nicht die Möglichkeit, bei der Planung von einer „grünen Wiese“ auszugehen (green field approach). In der Regel existieren bereits eine Vielzahl an Tools, Datenbanken und Dokumenten, die Informationen über die IT-Systeme und deren Abhängigkeiten enthalten. Diese Informationen sind meist nicht oder nur unzureichend miteinander verknüpft. Man spricht in diesem Zusammenhang von Daten-Silos oder Management Data Repositories (MDR).
Die Orientierung an Standards wie ISO/IEC20000 erfordert eine Bündelung aller Informationen über Configuration Items, SLAs, Verträgen und Dokumentationen in einer zentralen Configuration Management Database (CMDB). Dabei werden einzelne Daten-Silos zugunsten der CMDB-Einführung aufgelöst. Andere hingegen, die wichtige Funktionen für das Unternehmen erfüllen, die über die Funktionalität einer CMDB hinausgehen (Monitoring, Software-Deployment), bleiben weiterhin aktiv. Diese Tools müssen nun auf geeignete Weise mit der CMDB verknüpft werden, um Informationen zu synchronisieren. Dabei gehen die Hersteller von CMDB-Tools unterschiedliche Wege, ein standardisiertes Vorgehen gibt es nicht. Abhilfe hierzu soll ein Standard für föderierte Datenquellen (CMDBf) schaffen.

Diese Arbeit gibt einen Überblick über das Configuration Management und vermittelt, was für einen Aufbau und Betrieb einer CMDB notwendig ist. Da man in der Praxis immer von mehreren Datenquellen ausgehen kann, wird der derzeit im Entwurf befindliche Standard „CMDBf“ für verteilte Datenquellen untersucht.
Die Entwicklung eines Konzeptes zum Aufbau des Configuration Managements am Leibniz-Rechenzentrum (LRZ) dient als praktische Umsetzung. Das Rechenzentrum bereitet derzeit die Einführung des IT-Servicemanagement-Tools „iET ITSM“ der Firma iETSolutions vor. Hierzu wird primär das standardisierte und abteilungsübergreifende Verfahren zur Provisionierung von virtuellen Maschinen analysiert. Außerdem werden die an den Prozessen beteiligten MDRs bewertet. Auf Basis der Ergebnisse wird ein Informationsmodell erstellt. Ein Integrations- und Migrationskonzept für die Realisierung dieses Modells wird exemplarisch in „iET ITSM“ realisiert. Abschließend werden die MDRs „LRZ Netzdoku“ und „VMware Infrastructure“ an die Software angebunden.

Inhaltsverzeichnis

1. Einführung

Die Beherrschbarkeit komplexer IT-Infrastrukturen ist ohne die Einführung von Prozessen und Tools heutzutage nicht mehr möglich. Unternehmen können sich dabei an - speziell für den IT-Dienstleistungsbereich - definierten Leitlinien, die IT-Unternehmen bei der Einführung, Umsetzung und ständigen Verbesserung unterstützen, orientieren: Best-Practice-Ansätze wie IT Infrastructure Library (ITIL) [OGC05] [OGC07], Control Objectives for Information and Related Technology (CobiT) [Inf07], Microsoft Operations Framework (MOF) [Mic08] sowie die international gültige Norm ISO/IEC20000 [ISO05a] geben ihnen Ideen und Werkzeuge an die Hand, um ein prozessorientiertes IT-Servicemanagement aufzubauen.

1.1. Motivation

IT-Dienstleister versuchen ihre Kunden langfristig an Ihr Unternehmen zu binden, indem sie die Kundenzufriedenheit in den Fokus ihres Handels stellen. Dabei ist es notwendig, nicht nur durch die Wirtschaftlichkeit, sondern insbesondere durch die Qualität der Angebote zu überzeugen, um sich einen Wettbewerbsvorteil gegenüber der Konkurrenz zu verschaffen. Ein hochwertiges IT-Servicemanagement ist dabei der Schlüssel zur Betriebs- und Prozessicherheit einer IT-Infrastruktur [Sur08]. Der ungebrochene Trend in Richtung Service- und Qualitätsorientierung zeigt, dass immer mehr Unternehmen diese Möglichkeiten nutzen und so eine kontinuierliche Prozessverbesserung mit Hilfe von geschultem Personal und dem Einsatz eines IT-Servicemanagement-Tools erreichen. Eine erfolgreiche unternehmensweite Zertifizierung besitzt - sofern sie kommuniziert und beworben wird - sowohl eine positive Innen- wie auch Außenwirkung für das Unternehmen. Intern wird das Mitarbeiterbewusstsein für Qualität und effiziente Prozesse gestärkt. Nach Außen hin sorgt eine Zertifizierung für Vertrauen in das Unternehmen und führt zu einer besseren Wettbewerbsposition bei Auftragsvergaben.

Die Ausrichtung auf IT-Servicemanagement führt früher oder später zur Einführung eines Tools, welches alle Unternehmensprozesse miteinander verzahnt und somit eine gemeinsame Datenbasis schafft. Man spricht dabei von der Configuration Management Database (CMDB).

Während des Trainings und der Tool-Einführung wird oftmals von einer „grünen Wiese“ ausgegangen. Diese Annahme einer „grünen Wiese“ beschreibt den Idealfall, in dem es keine zu berücksichtigenden, bestehenden Prozesse eines Unternehmens gibt. Daher kann alles von Grund auf geplant und spezifiziert werden. Dies ist für die Tool-Einführung optimal, da jeder Hersteller eigene Konventionen verwendet und der Prozess an das Tool angepasst werden kann.

Dieser Fall ist in der Praxis jedoch nicht anzutreffen. Je nach Reifegrad des Unternehmens wurden bereits Prozessdefinitionen und Ablaufpläne erstellt. Oftmals existieren auch nicht-dokumentierte Abläufe (=Konventionen), die von Mitarbeiter zu Mitarbeiter weitergegeben und nicht formal festgeschrieben wurden. Zudem werden benötigte Informationen zur

Durchführung der Prozesse in mehreren voneinander unabhängigen Systemen und Programmen festgehalten (sogenannte Daten-Silos). Diese Strukturen bestehen gewöhnlich bereits mehrere Jahre und haben sich fest im Unternehmen etabliert.

Die Einführung eines zentralen IT-Servicemanagement-Tools führt dazu, dass sich einerseits definierte Abläufe und Konventionen zum Teil enorm verändern und es andererseits notwendig wird, vorhandene Software mit Datenbeständen zu evaluieren und auf Ihren Fortbestand oder die Integration in einem gemeinsamen Datenverbund (CMDB) zu untersuchen. Da dieser Schritt meist sehr stark in die Arbeitsweise der Mitarbeiter eingreift, ist es erforderlich, die Einführung frühzeitig zu planen und zu kommunizieren. Nur so können Verständnis und Akzeptanz durch die Mitarbeiter erreicht werden. Bei fehlendem Verständnis durch mangelnde Kommunikation mit den Mitarbeitern kann die Einführung von ITSM auf schwerwiegende Widerstände stoßen und letztendlich sogar scheitern.

Ziel dieser Arbeit ist es, eine bestehende Unternehmensstruktur mit zahlreichen isolierten Datenbeständen (Daten-Silos) zu analysieren, Wege aufzuzeigen, diese Datenbestände miteinander zu vernetzen und so für den IT-Servicemanagement Prozess „Configuration Management“ einen gemeinsamen und aktuellen Datenbestand aufzubauen. Dabei können sich die Konfigurationsinformationen auch über mehrere Tools hinweg erstrecken. Man spricht dabei von einer verteilten bzw. föderierten Configuration Management Database (CMDBf). Derzeit ist ein Kommunikationsstandard für CMDBf in Arbeit, um Informationen zu synchronisieren. Der praktische Teil der Arbeit wird am Leibniz-Rechenzentrum (LRZ) durchgeführt. Das LRZ baut in den kommenden Jahren einen vollständig prozessorientierten IT-Betrieb auf, mit dem mittelfristigen Ziel sich nach ISO/IEC20000 zertifizieren zu lassen.

1.2. Leibniz-Rechenzentrum (LRZ)

Das Leibniz-Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften (BAdW) mit Sitz am Forschungszentrum Garching bei München ist IT-Dienstleister in der Wissenschafts- und Hochschullandschaft Münchens. Eine Kernaufgabe des Rechenzentrums ist der Betrieb und die Wartung des Münchner Wissenschaftsnetzes (MWN). Das MWN ist die zentrale Kommunikations-Infrastruktur aller Münchner Hochschulen und anderer wissenschaftlicher Einrichtungen. Es umfasst derzeit 60 Standorte an denen insgesamt mehr als 65.000 Geräte (Server, PCs, Drucker, aktive Netzkomponenten) angeschlossen sind [Lei09]. Für das MWN werden zahlreiche zentrale Dienste wie z.B. Web-, Mail- und Directory-Server durch das LRZ bereitgestellt. Diese Dienste können wiederum von den einzelnen Instituten oder Lehrstühlen (Kunden) genutzt werden. Des Weiteren gibt es Hoch- und Höchstleistungsrechner, deren Rechenzeit zum Teil bundesweit angefragt werden kann. Im Gegensatz zu anderen Unternehmen obliegt dem LRZ ausschließlich der Betrieb und die Wartung der Infrastruktur (Verkabelung, Anschlüsse, Netzkomponenten, Accesspoints, Telefonie) sowie die Bereitstellung der zentralen Dienste (Server, Speicher, Höchstleistungsrechner). Die Endgeräte (PCs, Drucker) werden von den Instituten selbst gekauft, angeschlossen und gewartet und obliegen nicht dem IT-Management des LRZ.

Intern ist das LRZ in vier Abteilungen untergliedert:

- Benutzernahe Dienste und Systeme (BDS)
  Zuständig für MWN-weite Dienste wie Directory's, E-Mail, Internetdienste und Datenbanken sowie am LRZ befindliche Grafik-, Visualisierungs- und Multimedia-Einrichtungen

- Hochleistungssysteme (HLS)
  Zuständig für die Serverumgebung, Speichersysteme und die Höchstleistungsrechner
- Kommunikationsnetze (KOM)
  Zuständig für Aufbau, Betrieb und Wartung der Kommunikationsinfrastruktur
- Zentrale Dienste (ZD)
  Verwaltungs-Aufgaben, Gebäudetechnik

1.3. Herangehensweise

Aufgabe dieser Arbeit ist es, mit Hilfe von Analysen von Unternehmensprozessen ein Informationsmodell für die Configuration Management Database (CMDB) zu erstellen. Auf Basis dieses Modells soll anschließend eine geeignete Werkzeug-Architektur erarbeitet und ein Weg zur Datensynchronisierung aufgezeigt werden. Dazu soll der Entwurf der föderierten CMDB (CMDBf) auf seine Tauglichkeit untersucht werden.

Diese Untersuchung geschieht im Rahmen von Kapitel 2, in dem zunächst der grundlegende Prozess „Configuration Management“ näher betrachtet wird. Es werden die Aufgaben dieses Prozesses beschrieben und aufgezeigt, was für den Aufbau einer CMDB notwendig ist. Da es in der Regel in Unternehmen nicht eine einzige, zentrale CMDB geben kann, wird auf den Ansatz verteilter (föderierter) CMDBs eingegangen. Diese Idee wurde zuerst von Gartner [Col06] dokumentiert. In ITIL v3 [OGC07] wird dies als Configuration Management System (CMS) bezeichnet. Die Schwierigkeit dabei ist die gegenseitige Synchronisation und die Definition des Orts der Wahrheit (Reconciliation). Dazu existiert ein Normvorschlag der CMDBf-Workgroup, der analyisiert wird. In Kapitel 3 werden einige ausgesuchte Use-Cases am LRZ näher betrachtet und deren Tool-Abhängigkeiten analysiert. Auf Basis dieser Daten geschieht in Kapitel 4 die Bewertung der Tools auf ihre zukünftige Eignung als Datenlieferant für die CMDB. In Kapitel 5 wird zunächst auf Basis der in den vorangegangen Kapiteln gewonnenen Daten ein Informationsmodell erzeugt. Bevor das Modell im zukünftigen ITSM-Tool „iET ITSM“ von iETSolutions umgesetzt werden kann, wird zunächst die Software kurz vorgestellt und das vom Hersteller vorgesehene CMDB-Datenmodell analysiert. Anschließend kann das erstellte Informationsmodell in „iET ITSM“ umgesetzt werden. In Kapitel 6 werden exemplarisch zwei Datenquellen (LRZ Netzdoku und VMware Infrastructure) im Rahmen eines Proof-of-concept angebunden. Kapitel 7 stellt eine Zusammenfassung, ein Fazit und einen Ausblick auf zukünftige Möglichkeiten und Denkansätze vor. Abbildung 1.1 zeigt das Vorgehen als Schaubild.

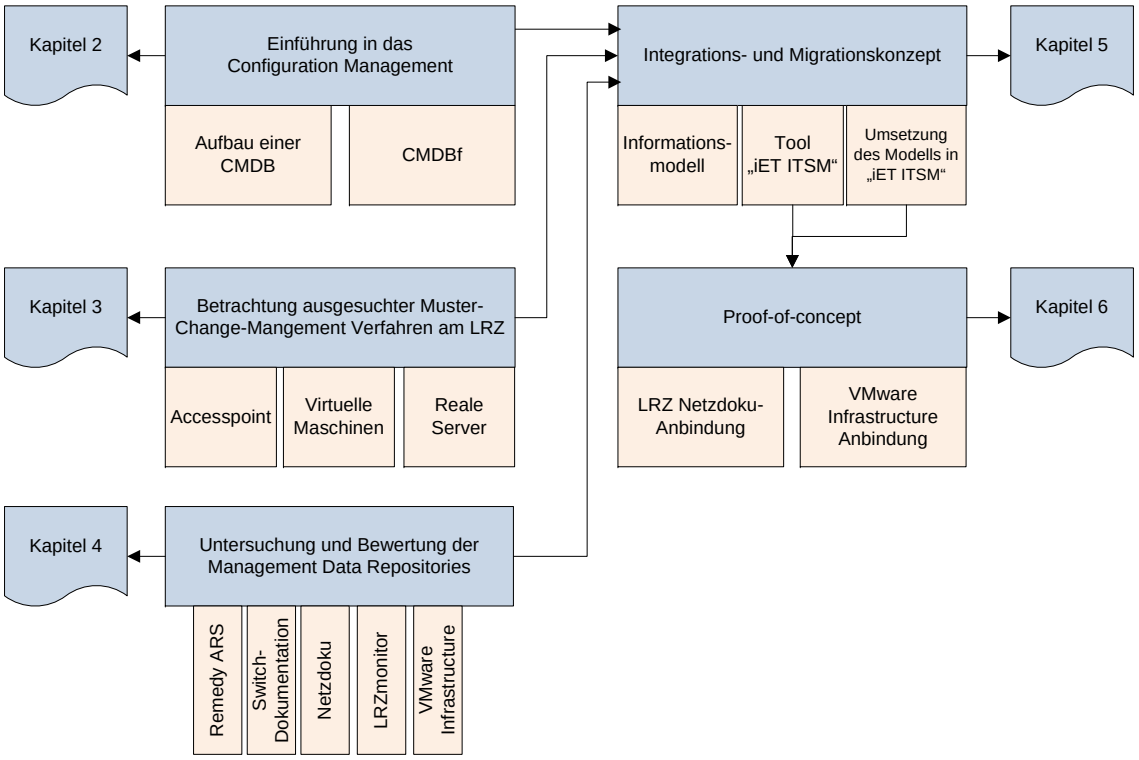


Abbildung 1.1.: Visualisierung der Herangehensweise

2. Configuration Management - State of the Art

Das Configuration Management kann nicht als isolierter Prozess innerhalb des IT-Servicemanagements betrachtet werden. Vielmehr bildet es mit Hilfe der Configuration Management Database (CMDB) die Basis für den Erfolg der anderen Prozesse.

In diesem Kapitel werden die Funktionen und Fähigkeiten einer CMDB dargestellt. Dazu definiert Kapitel 2.1 die Eigenschaften einer CMDB. Für den Aufbau und den Betrieb ist ein strukturiertes Vorgehen gemäß dem Deming-Kreislauf empfehlenswert, um somit eine kontinuierliche Verbesserung der Datenqualität zu erreichen (Kapitel 2.2). Wie die Qualität einer CMDB bestimmt werden kann, erläutert Kapitel 2.3. In der Praxis ist überwiegend die Situation anzutreffen, dass mehrere isolierte Datenquellen (Daten-Silos) im Unternehmen eingesetzt werden. Eine Vernetzung der Informationsquellen ist bislang nur sehr individuell möglich. Abhilfe soll hier ein neuer Standard für föderierte Datenquellen (CMDBf) [CMD08] schaffen. Die Möglichkeiten des neuen Standards werden in Kapitel 2.4 kritisch bewertet.

2.1. CMDB-Grundlagen

Die Configuration Management Database (CMDB) ist laut ITIL [OGC05] eine Datenbank, die ein logisches Modell der gesamten IT-Infrastruktur und der Services allen anderen Prozesse bereitstellt. In der CMDB werden alle Configuration Items (CIs), die für das IT-Servicemanagement wichtig sind, abgespeichert. Unter CI versteht man Infrastrukturkomponenten (PCs, Switches) ebenso wie Softwarekomponenten, Lizenzen und Verträge. Jedes CI enthält Attribute (technische, wirtschaftliche und verwalterische Daten), die das CI näher beschreiben. Ein CI kann aus Komponenten bestehen, die nicht als separates CI gepflegt werden sollen um den Überblick über die CMDB zu gewährleisten. Beispiele für Komponenten können Netzwerkkarten oder Festplatten sein, die man detailliert erfassen möchte. Die Art der Implementierung solcher Komponenten ist herstellerabhängig.

Die CIs stehen dabei im Allgemeinen in Beziehung zueinander, können beispielsweise miteinander verbunden oder eingebaut sein. Diese Relationen werden ebenfalls innerhalb der CMDB abgebildet. Beispiele gängig verwendeter Relationen sind:

Teil von: eine Komponente ist in einer anderen Komponente verbaut, beispielsweise ein Einschubmodul in einem Switch. Die verbaute Komponente wird separat erfasst, da sie jederzeit ausgetauscht werden kann.

installiert auf: in der Regel bei wichtiger Software oder Serverdiensten genutzt. Ermöglicht beispielsweise während einer Impact-Analyse im Rahmen des Change Managements Aussagen über die Auswirkung auf Dienste zu treffen.

verbunden mit: kennzeichnet eine direkte Verbindung eines CIs mit einem anderen. Wird bei Netzverbindungen und Stromzuleitungen verwendet. Je nach Unternehmensszenario

gibt es hierbei mehrere Fallunterscheidungen (z.B. nach Art der Netzverbindung).

Die Kontrolle über die CMDB obliegt ausschließlich dem Configuration Management. Der Prozess muss sicherstellen, dass vor dem Ausführen von Änderungen Baselines der betroffenen CIs angelegt werden [ISO05a]. Baselines stellen eine Art Snapshot dar und ermöglichen es, einen SOLL-IST Vergleich zu ziehen.

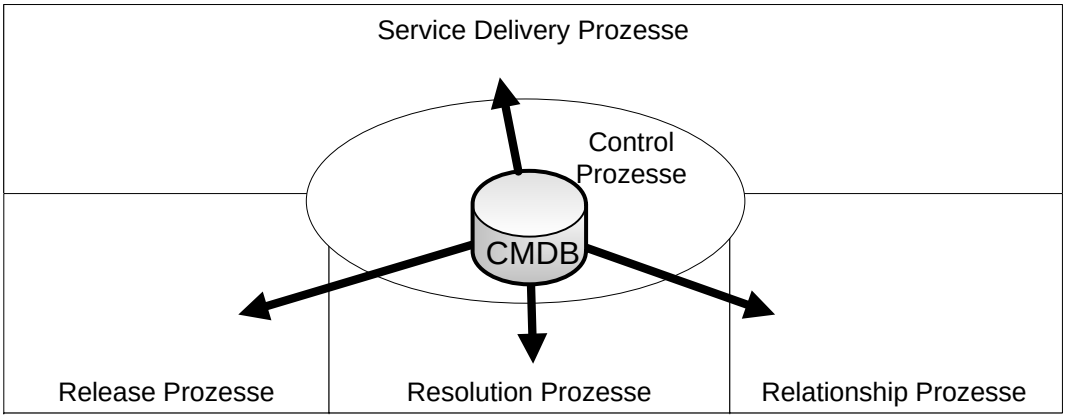


Abbildung 2.1.: CMDB in Beziehung zu den anderen ISO/IEC20000-Prozessen

Eine umfassende und aktuell gehaltene CMDB unterstützt darüber hinaus die Einhaltung gesetzlicher Anforderungen an das Unternehmen. So sind Unternehmen abhängig von Unternehmensform, -standort und -branche an Gesetze und Vorgaben gebunden, die beispielsweise das Vorgehen zum Risiko-Management, Datenhaltung und Buchführung festlegen. Beispiele hierfür sind das BDSG (Bundesdatenschutzgesetz), GDPdU (Grundsätze zum Datenzugriff und Prüfbarkeit digitaler Unterlagen), KonTaG (Gesetz zur Kontrolle und Transparenz im Unternehmensbereich), SOX (Sarbanes-Oxley-Act (USA)) und die 8. EU Richtlinie (die 2006 durch die Richtlinie 2006/43/EG ersetzt wurde). Ziel dieser Compliances-Regelungen ist es, das rechtlich und ethisch korrekte Verhalten von Firmen, Organen und Mitarbeitern zu gewährleisten [MB08]. Da die Geschäftsprozesse immer mehr mit den IT-Systemen verzahnt sind, ziehen diese Anforderungen auch IT-Compliances nach sich. Das IT-Governance Institute definiert diese Anforderungen allgemein als „Einhaltung und Umsetzung regulatorischer Anforderungen im weitesten Sinne mit dem Ziel eines verantwortungsvollen Umgangs mit allen Aspekten der Informationstechnik“ [MB08]. IT-Servicemanagement nach ISO/IEC20000, ITIL, CobiT oder MOF unterstützt die Sicherstellung der IT-Compliances [And09].

Somit bildet die CMDB das Rückgrat für alle Unternehmens- und IT-Prozesse [Det08], da sie in Verbindung zu allen Prozessen steht (siehe Abbildung 2.1). Wie einzelne Prozesse nach ISO/IEC20000 durch die CMDB unterstützt werden, wird in Tabelle 2.2 dargestellt.

Prozess	**CMDB-Unterstützung**
Budgeting & Accounting for IT services	CI-Daten werden mit Finanzdaten verknüpft und ermöglichen so eine Kosten- und Leistungsverrechnung
Business Relationship Management	Sämtliche Kundenkontakte werden vorgehalten. Bei Bedarf ist es möglich die CIs des Kunden anzuzeigen.
Capacity Management	Mit Hilfe von hinterlegten Incidents können Kapazitätsengpässe schnell lokalisiert werden
Change Management	Unterstützung bei der Impact-Analyse um Auswirkungen bei Changes festzustellen
Incident Management	Incidents werden mit den vorliegenden CIs verknüpft. Ein Zugriff auf eine Known-Error-Datenbank ermöglicht eine schnelle Problemlösung
Information Security Management	Eine CMDB erlaubt ein vielschichtiges Sicherheitskonzept um ausschließlich autorisierten Personen Zugriff auf definierte Teilbereiche oder Informations-Artefakte zu gewährleisten
Release Management	Informationen für die Release-Planung werden bereitgestellt
Problem Management	CIs, die mit Incidents verknüpft wurden, können für schnelle Analysen herangezogen werden und so langfristig Probleme behoben werden
Service Continuity & Availability Management	Mit Hilfe von Baselines kann im K-Fall auf einen konsistenten Datenbestand zurückgegriffen werden. Des Weiteren können durch eine Überwachung von CIs Schwachstellen lokalisiert werden
Service Level Management	Informationen über abgeschlossene SLAs werden vorgehalten und mit entsprechenden CIs verknüpft, so dass eine frühzeitige Warnung erfolgen kann, falls ein ServiceLevel nicht eingehalten werden kann.
Supplier Management	ein Zugriff auf alle relevanten Daten der Lieferanten ermöglicht einen raschen Überblick
Service Reporting	Da die CMDB alle für das Unternehmen wichtigen Konfigurationsinformationen bereithält, können so schnell aussagekräftige Berichte erzeugt werden

Abbildung 2.2.: Prozessunterstützung durch die CMDB in Anlehnung an [Det08]

2.2. Aufbau und Betrieb einer CMDB

Die Struktur der CMDB stellt die Weichen für den Erfolg der CMDB. Begonnen wird mit einer Anforderungsanalyse, in der Use-Cases für die CMDB erfasst werden. Auf Basis der Use-Cases wird zunächst ein Informationsmodell erstellt, dass alle zu erfassenden Informationen und deren Abhängigkeiten zueinander enthält. Schließlich wird das Informationsmodell in ein Datenmodell überführt und im Anschluss implementiert. Nach der Implementierungsphase erfolgen diverse Tests, die ggf. zur Veränderung des Datenmodells führen, bis schließlich die CMDB in den Live-Betrieb geht.

Bei dieser Vorgehensweise gibt es klare Parallelen zum Software Engineering. Mit Hilfe des V-Modells (vgl. Abbildung 2.3), einer Projektmanagement-Methode für Software-Entwicklungsprojekte, werden dort komplexe Entwicklungen gemanaged. Da auch die initiale Entwicklung der CMDB vergleichbar mit der Entwicklung einer Software ist, lässt sich das Vorgehen aus dem V-Modell auf die Domäne IT-Servicemanagement übertragen [And09].

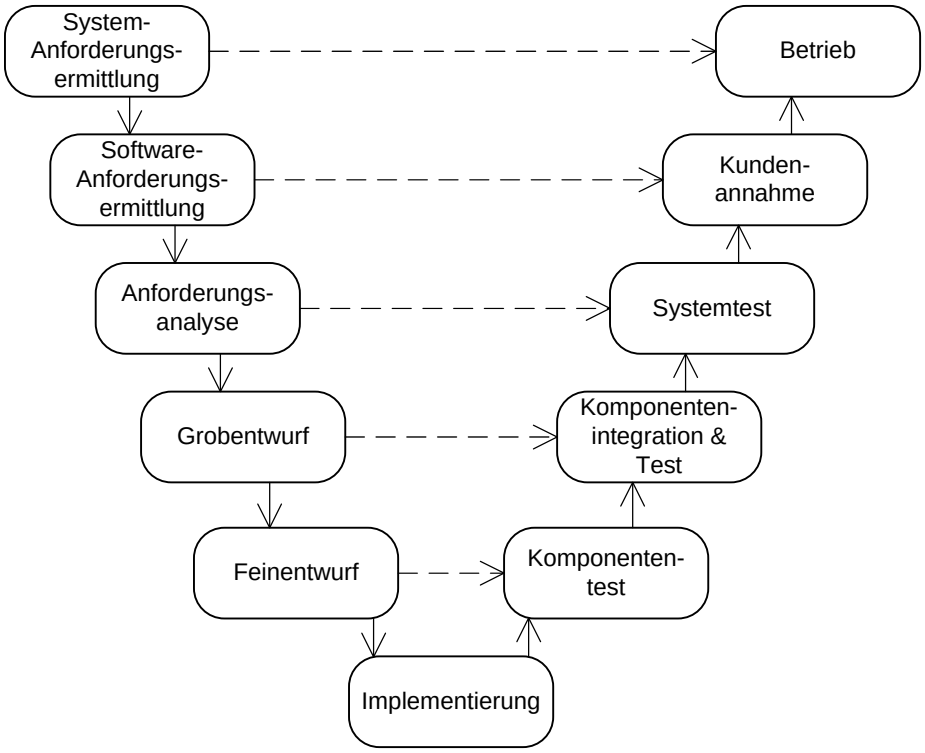


Abbildung 2.3.: V-Modell der Softwareentwicklung nach [BD07]

Nach der Inbetriebnahme der CMDB muss durch das Configuration Management eine stetige Erfolgskontrolle und kontinuierliche Verbesserung durchgeführt werden. Dieser Kreislauf ist besser bekannt als PDCA-Zyklus (Plan-Do-Check-Act). Dieser Zyklus geht auf W.E. Deming zurück [Tag05] und beschreibt ein Verfahren zur kontinuierlichen Verbesserung, das auch im Configuration Management zum Einsatz kommt.

2.2.1. Plan

In der ersten Phase ist es zunächst notwendig sich Gedanken über Umfang und Ziele der CMDB zu machen (Scoping). Dabei ist es nötig, sämtliche Unternehmensprozesse und Verfahren zu betrachten, um daraus die benötigten Informationen abzuleiten. Ein standardisiertes Vorgehen gibt es dafür nicht. Üblicherweise findet diese Projektphase im Rahmen von Workshops statt, an denen alle beteiligten Personen und Abteilungen teilnehmen. Hierbei wird auf Basis von Anwendungsfällen herausgearbeitet, welche Informationen in der CMDB enthalten sein müssen. Grundsätzlich sollten nur Daten in die CMDB, die auch benötigt werden. Sobald Daten prozess- oder abteilungsübergreifend genutzt werden, sollten diese in der Datenbank auftauchen. Neben den Daten müssen auch sämtliche physischen und logischen Abhängigkeiten in dieser Phase erfasst werden [Ros08a]. Grundsätzlich sollten nur Daten in der CMDB liegen, die auch gepflegt werden können. Unterteilt man seine Komponenten in zu viele kleine Bausteine können sehr schnell mehrere tausend CIs entstehen, die man in der Praxis nicht benötigt und den Überblick verschlechtern. Auch ein zu hoher Detailgrad, d.h. es werden zu viele Informationen erfasst, kann den Administrationsaufwand unnötig erhöhen.

Welche Information tatsächlich erfasst werden sollten, ist unternehmensspezifisch und abhängig von den dortigen IT-Prozessen. ITIL sagt diesbezüglich nur „everything you wish to keep under control“ [Det08].

Um den Prozess „Configuration Management“ formal zu definieren sollte gemäß ISO/IEC20000-2 [ISO05b] zu Beginn ein Configuration Management Plan erstellt werden. Dieses Dokument sollte folgende Informationen enthalten:

1. Umfang, Ziele, Richtlinien, Rollen und Verantwortlichkeiten
2. Etablierung der Configuration Management Prozesse
   Anlage, Kontrolle, Aktualisierung und Berichterstattung über CIs
3. Anforderungen an Verantwortlichkeit, Nachvollziehbarkeit und Auditierbarkeit
   Dabei sind sowohl gesetzliche, firmeneigene und Sicherheits-Anforderungen zu berücksichtigen
4. Kontrolle der Konfiguration
   Zugriff, Sicherheit, Versionierung, Release-Kontrollen
5. Sicherheits- und Kontrollmechanismen für die Benutzeroberfläche und System-Schnittstellen
   Dieser Punkt ist besonders dann wichtig, falls mehrere getrennte Organisationen verwaltet werden müssen
6. Ressourcen-Entwicklung, um CIs zu erfassen und aktuell zu halten
7. Management von Zulieferern und Subunternehmen im Hinblick auf Configuration Management

Hilfe bei der Erstellung von Configuration Management Plänen bieten unter anderem das Web-Portal ConfigurationKit [cmt], das einen beispielhaften Configuration Management Plan nach ITIL und CobiT anbietet. In Cobit entspricht das Configuration Management dem dortigen Prozess DS9: „Manage the Configuration“ der Prozessgruppe „Deliver and Support“. Als Inputs für diesen Prozess werden

- Anwender-, Support-, Techniker- und Administrator-Handbücher
- Installierte Configuration Items
- Relevanz der IT Configuration Items

genannt.

Zu Beginn des Configuration Managements sollte das Verfahren, wie Configuration Items implementiert und die Aktualisierung von Informationen in der CMDB erfolgt, festgelegt werden. Gemäß ISO/IEC20000-2 sollte ein angemessener Grad an Automatisierung implementiert werden. Das bedeutet, dass man einen guten Mittelweg zwischen vollständiger Automatisierung (was in der Praxis selten erfolgreich sein wird) und hoher Flexibilität und damit steigender Komplexität (schnelle Anpassung an neue/geänderte Anforderungen) finden sollte.

Configuration Management läßt wie viele der ITL-basierten Prozesse eine Beliebigkeit der Details der Prozessgestaltung zu [Bre07].

2.2.2. Do

Innerhalb dieses Zyklus-Abschnitts erfolgt die Umsetzung der vorher durchgeführten Planungen. Der Aufbau der CMDB kann mittels zwei Methoden erfolgen: „top-down“ oder „bottom-up“. Der „top-down“-Ansatz startet mit der Betrachtung großer IT-Services, die für das Unternehmen von Bedeutung sind und verfeinert kontinuierlich die Betrachtungsweise. Das Modell gewinnt somit mehr Details. Beim „bottom-up“-Ansatz beginnt man in der Regel mit der Betrachtung lokaler Daten-Silos, wie z.B. isolierten Access-Datenbanken und Excel-Listen, und arbeitet sich langsam an die IT-Services heran. Die Gefahr hierbei ist, dass man sich in Details verliert und damit keine aussagekräftige Struktur erzeugt. iETSolutions, ein Hersteller eines ITSM-Tools, empfiehlt daher den „top-down“-Ansatz [Det08]. Auch die ISO/IEC20000-Norm gibt Hinweise zum Umfang und Detaillierungsgrad der CMDB: „...es muss sichergestellt sein, dass der Grad der Kontrolle den Geschäftsanforderungen, Fehlerrisiken und Kritikalität des Service entspricht“. Diese Aussage führt wiederrum zum „top-down“-Ansatz.

Idealerweise beginnt man mit zwei bis drei wichtigen Services des Unternehmens und erweitert die CMDB nach und nach. So erzeugt man eine aussagekräftige CMDB. Dazu werden zunächst relevante CI identifiziert und im zweiten Schritt genauer definiert.

Mit Hilfe des Configuration Management Plans sollte es sehr einfach sein, die relevanten Configuration Items zu identifizieren. Zu den relevanten Elementen zählen alle Arten von Software und Informationssystemen (sowohl Eigenentwicklungen als auch gekaufte Systeme), Lizenzen, physische Elemente (Arbeitsgeräte), Dokumentationen und Verträge.

Im zweiten Schritt werden die CI's näher spezifiziert. Es werden die entsprechenden physischen und funktionalen Eigenschaften definiert, die als Attribute dem CI mitgegeben werden. Außerdem werden physische und logische Abhängigkeiten und Beziehungen zwischen CIs definiert. Sie sollten auf angemessener Ebene erfolgen, d.h. es sollten nur Abhängigkeiten erfasst werden, die auch später wieder benötigt und gepflegt werden können. Relationen ermöglichen später die Beurteilung von Änderungen (Impact Analyse). Zuletzt sollte die Nachvollziehbarkeit für ein späteren Audit und Reporting sichergestellt werden.

2.2.3. Check

Die Überprüfung des Control-Prozesses Configuration Management dient zum formellen Abgleich der realen Infrastruktur und Services mit dem dokumentierten logischen Abbild. Außerdem wird die reale Funktionalität und Performance mit dem dokumentierten logischen Abbild verglichen [ISO05b]. Ein Audit sollte in regelmäßigen Abständen (z.B. jährlich), vor und nach wesentlichen Änderungen, nach einem Katastrophenfall sowie auch stichprobenartig durchgeführt werden. Damit soll sichergestellt werden, dass Service-Management-Anforderungen konform mit dem Service-Management-Plan sowie ISO/IEC20000 sind und die Anforderungen effektiv umgesetzt und gepflegt werden. Die Planung des Audit-Programms ist zwingende Voraussetzung von ISO/IEC20000-1 [ISO05a]. Dabei müssen vorab Ziele festgelegt werden, deren Ergebnisse und Lösungsmöglichkeiten aufzuzeichnen sind. Es gibt verschiedene Arten von Assessments und Audits:

Selbsteinschätzung Abteilung überprüft eigene Verfahren und Leistungen. Häufig ist diese Art der Bewertung nicht objektiv.

Internes Audit Die Auditierung findet innerhalb der Organisation statt. Der Auditor gehört zur gleichen Organisation, ist aber nicht direkt in den überprüfenden Bereich involviert.

Lieferanten-Audit Auditierung des Zulieferers. Ermöglicht Abläufe besser abzustimmen.

Externes Audit Auditierung durch unabhängige, externe Organisation. Im Rahmen von Standard-Audits ist dies meist ein Registered Certified Body (RCB), in Deutschland in der Regel TÜV Süd Akademie GmbH.

2.2.4. Act

Nachdem in der Check-Phase alle Defizite festgestellt wurden, erfolgt im letzten Schritt die Planung, Umsetzung und Bewertung von Verbesserungsmaßnahmen. Gemäß ISO/IEC20000 muss es eine veröffentlichte Policy zur Service-Verbesserung geben (Service Improvement Policy). Alle vorgeschlagenen Service-Verbesserungen sind zu bewerten, dokumentieren, priorisieren und autorisieren. Die Kontrolle der Aktivitäten muss nach einem Plan erfolgen. Typische Probleme, die im Configuration Management auftreten können, sind

- Ungenügende Verknüpfung zwischen CIs und Dokumentation
  Zur Verbesserung ist eine Schulung der Mitarbeiter notwendig
- Aktualität der CMDB nicht gesichert
  Verbesserungspotenzial kann eine automatische IST-Erfassung durch Tools bieten
- Dokumentation der CIs kann nicht als Analyse für Changes genutzt werden
  Tritt ein solches Problem auf, ist es dringend erforderlich, die Struktur der CMDB zu überprüfen und ggf. das Scoping zu verändern um den Nutzen des Configuration Managements wiederherzustellen

2.3. Qualität der Prozesse und der CMDB

Die Einführung eines durchgängigen IT-Servicemanagements erfordert einen Wandel hin zur prozessorientierten IT und die Erstellung und Überarbeitung von Prozessen. Um die

Prozessqualität eines Unternehmens zu bewerten, eignet sich das CMMI-Reifegradmodell (Capability Maturity Model Integration) am Besten. Es gibt derzeit drei CMMI-Modelle, angepasst an spezielle Aufgaben (Development, Acquisition und Services). Im Falle der IT-Dienstleister sollte das sogenannte „CMMI for Services“-Modell betrachtet werden [cmm09a]. Durchgeführte Untersuchungen des Software Engineering Institute ergeben, dass die meisten Unternehmen Prozessdefinitionen erstellt haben und diese auch aktiv leben (Reifegradstufe 3). Durch kontinuierliche Überwachung und Verbesserung ist es somit möglich die Prozessreifegrade auf Stufe 4 (Quantitativly Managed) und Stufe 5 (Optimizing) zu heben. Wie die Studie zeigt, gibt es noch eine sehr große Anzahl an Unternehmen, die lediglich Stufe 2 (Managed) erreicht haben [cmm09b]. Abbildung 2.4 visualisiert das Ergebnis der Studie.

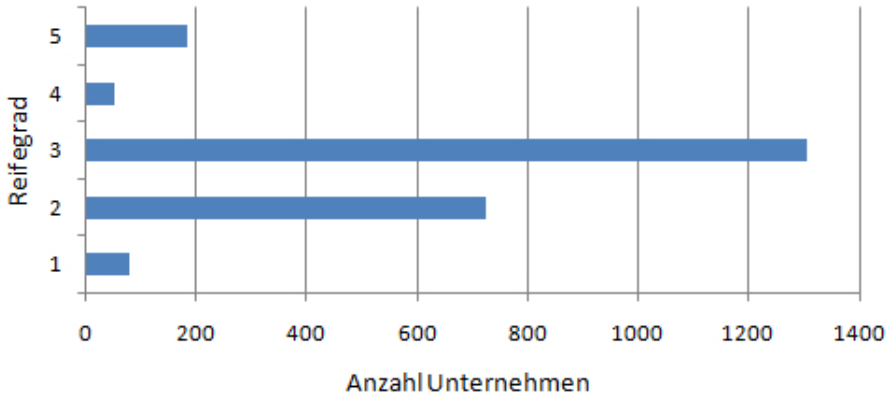


Abbildung 2.4.: Ergebnisse der Untersuchungen bzgl. der Reifegrade

Der Prozess-Reifegrad steht in einem direkten Zusammenhang mit der Qualität der CMDB. Gerade Unternehmen mit niedrigem Reifegrad verwenden oft zahlreiche Daten-Silos in denen Daten - teilweise redundant - vorgehalten werden. Diese Daten-Silos sind gemäß der Definition bereits eigenständige CMDBs. Somit lässt die Qualität der CMDB(s) Rückschlüsse auf den Reifegrad zu.

Aber auch die CMDB lässt sich direkt bewerten. ComConsult [Jak08] teilt die Unternehmens-CMDB in ein Sterne-System ein, dass ähnlich zu den Reifegradstufen ist.

1* Werte-Listen
Es werden getrennte Datenquellen mit überlappenden Informationen betrieben, die in den meisten Fällen manuell gepflegt werden. Änderungen werden nicht an die anderen Datenquellen propagiert. Auskunft, Datenquellenpflege und Erweiterung ist oft von einzelnen Personen abhängig

2* Item-Ketten
Höhere Informationsmenge in den Datenquellen. Es werden Objekte mit Attributen verwaltet und zum Teil miteinander verknüpft. Hoher Pflegeaufwand zur Aufrechterhaltung der Datengüte. Erhöhter Abstimmungsaufwand mit anderen Abteilungen bei der Pflege gemeinsamer Informationen. Notwendigkeit zur Definition erster Standardprozesse.

3* Item-Netze

Im Wesentlichen identisch zur 2*-CMDB, allerdings deutlich erhöhte Objekt-Anzahl, Detailgrad und thematische Vielfalt sowie dichtere Vernetzung der CIs.

4* Regeln
Mechanismen zur automatischen Pflege von Relationen. Plausibilitätsregeln unterstützen den Nutzer bei der Pflege der Daten.

5* Prozesse
Einbeziehung aller IT-Prozesse. Kontinuierlicher PDCA-Zyklus

Auch die Unternehmensberatung santix AG verwendet ein Reifegradmodell für das Configuration Management, um eine Bewertung ihrer Kunden vorzunehmen [Ros08b]. Dabei wird ein Stufenplan, wie er in Abbildung 2.5 zu sehen ist, verwendet. Die einzelnen Punkte dienen als Milestones und können bei erfolgreicher Implementierung abgehakt werden. Um ein bestimmtes Level zu erreichen, ist es erforderlich, alle der genannten Punkte einer Ebene zu implementieren. Stufe 1 erreicht ein Unternehmen implizit.

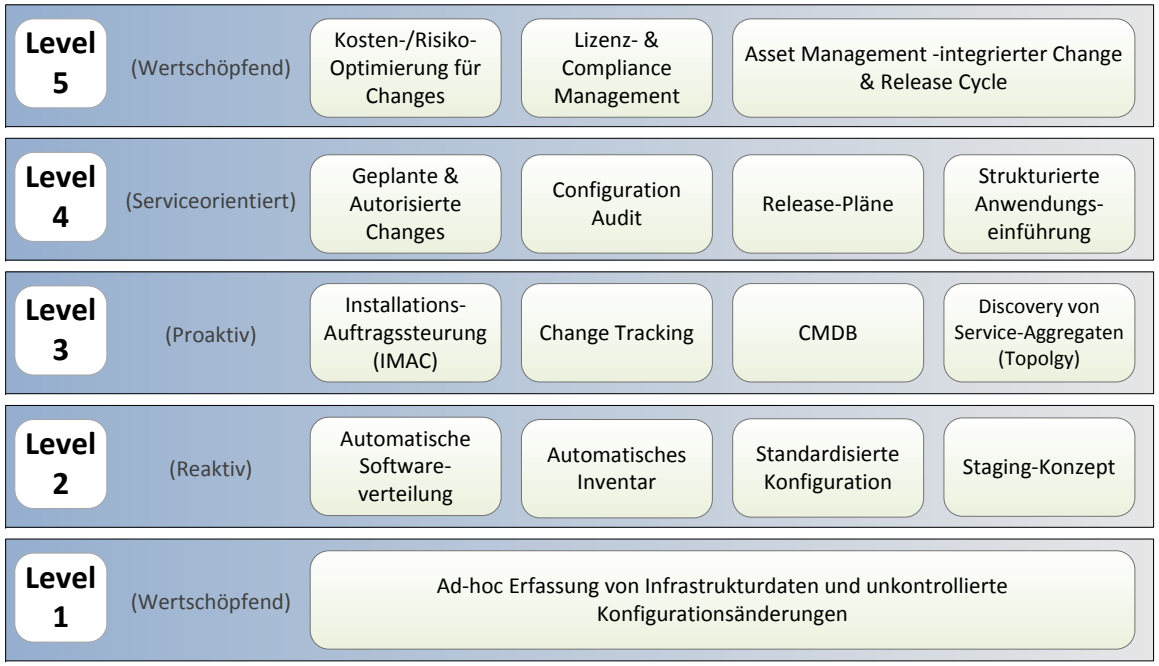


Abbildung 2.5.: Reifegradstufen im Configuration Management nach Santix [Ros08b]

2.4. Mehrere Datenquellen - verteilte CMDBs

Zunächst wird in Abschnitt 2.4.1 ein Überblick über das grundsätzliche Konzept gegeben. Der Normvorschlag der CMDBf Workgroup wird in Abschnitt 2.4.2 näher untersucht.

2.4.1. Grundidee

Bei der Einführung von IT-Servicemanagement und dem Ziel einer Zertifizierung nach ISO/IEC20000 in einem Unternehmen wird man selten bei Null anfangen. In der Regel gibt

es eine Vielzahl bestehender Management-Systeme, die über die Jahre hinweg eingesetzt und erweitert wurden. Diese Systeme wurden aber meistens eigenständig betrieben, d.h. ohne Verbindung zu anderen Tools. Man spricht hier von Daten-Silos, isolierten Systemen oder Management Data Repositories (MDR) [CMD08]. Zusammengenommen enthalten diese MDRs in der Regel das benötigte Wissen um ein IT-Servicemanagement zu realisieren und bilden damit die Basis für eine CMDB.

Die Idee ist es, diese MDRs miteinander zu einer großen Datenbank zu vernetzen um mit Hilfe eines ITSM-Tools auf alle aktuellen Daten aller Subsysteme (Assetmanagement-, ERP- oder Monitoring-Software) zuzugreifen. Damit beschleunigen sich die Unternehmensprozesse, da man in Zukunft nur noch ein Tool benötigt, das alle Informationen bereitstellen kann. ITIL V3 [OGC07] verwendet den Begriff des Configuration Management System (CMS), das aus einer oder mehrerer CMDBs bestehen kann. Diese Weiterentwicklung zeigt bereits die Annäherung an die Realität.

Das zentrale Tool benötigt somit Schnittstellen zu diversen Systemen. Bisher gibt es noch keine genormte Schnittstellendefinition, die die Kommunikation und den Informationsaustausch zwischen der CMDB und den MDRs regelt. Ein Schritt in diese Richtung geht der im Entwurf befindliche Ansatz der CMDBf, der im folgenden Abschnitt näher beschrieben wird.

Unterdessen stellen die Hersteller von ITSM-Tools eigene Schnittstellen für den Import von Daten bereit. In der Regel können die Tools Datenbankverbindungen zu anderen Systemen aufbauen (ODBC), Informationen aus Verzeichnisdiensten wie LDAP und Active Directory auslesen sowie CSV-Dateien (im Tool-Hersteller spezifischen Format) aus einem Verzeichnis importieren, Abbildung 2.6 verdeutlicht dies. Der automatische Export aus den ITSM-Tools in andere MDRs ist in der Regel nicht out-of-the-box möglich. Hierfür müssen in der Regel eigene Lösungen entwickelt werden.

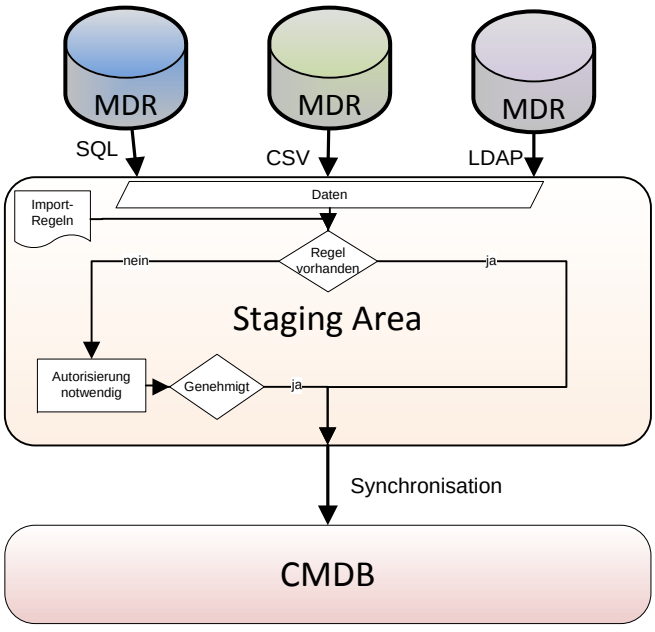


Abbildung 2.6.: Import-Mechanismus gängiger ITSM-Tools

Die importierten Daten aus den MDRs dürfen keinesfalls direkt in die CMDB übernommen werden und dort die bereits vorhandenen CI-Informationen überschreiben. Das würde eine Verletzung der Anforderungen an eine CMDB nach ISO/IEC20000-1 [ISO05a] bedeuten, da die Nachvollziehbarkeit nicht gegeben ist. Aus diesem Grund wird in den Tools in der Regel eine „Staging Area“ zur Zwischenspeicherung genutzt. In diesem Puffer befinden sich alle neu gewonnenen Informationen (IST-Daten) und werden mit den SOLL-Daten aus der CMDB verglichen. Bei Abweichungen der beiden Werte ist ein Eingriff des Configuration Managers notwendig. Das Verfahren ermöglicht die Entdeckung unautorisiert durchgeführter Changes sowie das Aufspüren von Fehlern in der Prozesskette.

Um das Prinzip zu verdeutlichen, wird dies beispielhaft für ein Arbeitsupgrade an einem CI durchgespielt: So soll einem Server der Arbeitsspeicher von 4 auf 8 GB erhöht werden. Dazu wird der entsprechende Change angelegt und gegebenenfalls durch den Change Manager autorisiert (falls es sich nicht um einen Standard-Change handelt). Dabei werden die neuen SOLL-Daten (8 GB) für das CI eingetragen. Sobald das Monitoring-Tool 8 GB Arbeitsspeicher vorfindet und dies an die „Staging Area“ weitergibt, kann der Change erfolgreich abgeschlossen werden.

2.4.2. CMDBf

Für den Import und die gemeinsame Verwendung von Daten aus MDRs gibt es bisher keinen Standard. Aus diesem Grund wurde im Jahr 2007 die CMDB Federation Workgroup ins Leben gerufen. Das Konsortium besteht aus sechs großen Firmen (BMC Software, CA, Fujitsu, HP, IBM und Microsoft). BMC, CA, HP und IBM entwickeln eigene ITSM-Suiten für den Markt und gehören laut Gartner-Marktanalyse (Magic Quadrant) zu den „Big-Playern“ im Bereich IT Service Desk [CB08].

Die Aufgabe der Arbeitsgruppe besteht darin, einen Standard für den Zusammenschluss heterogener MDRs zu einer CMDB zu finden und die Kommunikation und den Informationsaustausch festzulegen.

Die erste Version des Entwurfs wurde am 22.Oktober 2007 veröffentlicht. Am 4.Januar 2008 gab es ein Update des Entwurfs auf Version 1.0b [CMD08]. Seit diesem Zeitpunkt befindet sich der Normvorschlag im Entwurfsstatus. Abbildung 2.7 zeigt den prinzipiellen Aufbau, der nachfolgend näher beschrieben ist.

Architektur der CMDBf

Client Ein anfragender Client kann via Schnittstelle („Query Service“) direkt mit der CMDBf kommunizieren, um so eine aggregierte Ansicht der Daten zu bekommen. Der Normvorschlag erlaubt auch die direkte Kommunikation mit den einzelnen MDRs. Grundsätzlich erfolgt der Zugriff nur lesend, der Entwurf enthält keine Interface-Definition für einen schreiben Zugriff.

Management Data Repository (MDR) Die MDRs halten die Daten über gemanagte Ressourcen bereit und können auch sogenannte Prozess-Artefakte wie z.B. Incident-Records enthalten. Der Überbegriff in der Norm lautet dafür „Item“. MDRs geben über einen Query Service ihre Informationen auf Anfrage an die CMDBf weiter (PULL-Verfahren) können aber auch aktiv Informationen an die CMDBf übermitteln (PUSH-Verfahren).

Federating CMDB (CMDBf) Die Federating CMDB enthält Daten aus allen MDRs (federated data), kann aber Informationen enthalten (non-federated data). Es ist möglich, dass eine CMDBf Daten ihrerseits von einer anderen CMDBf bekommt, allerdings verwendet man hier wieder den Begriff des MDR. Die CMDBf stellt neben einem „Query Service“ für Anfragen von Clients zusätzlich einen „Registration Service“ bereit, der es MDRs ermöglicht, im PUSH-Verfahren Daten zu übermitteln.

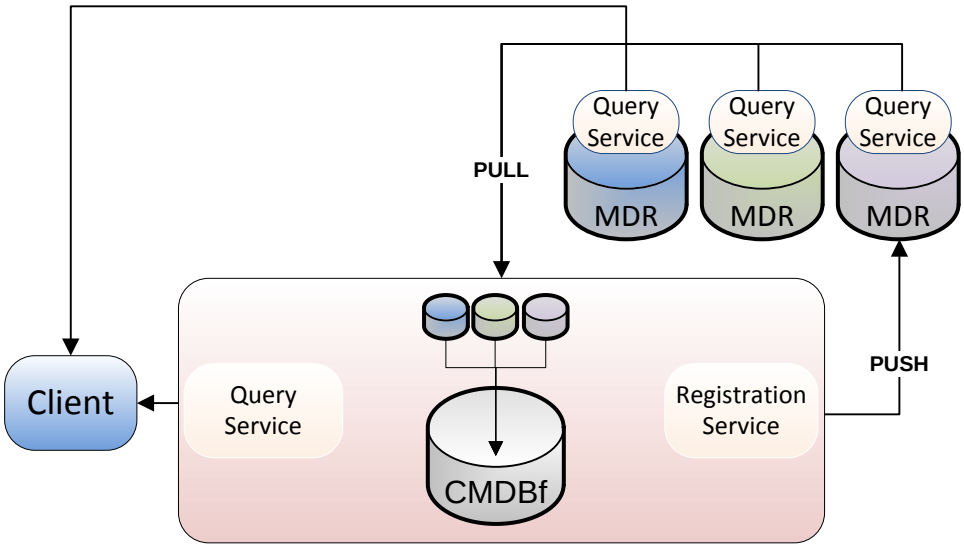


Abbildung 2.7.: Struktur der CMDBf

Kommunikation und Datenmodell

Die Kommunikation läuft plattformunabhängig mit Hilfe von Web Services ab. Als Schnittstelle zwischen den verschiedenen Systemen dient SOAP 1.1[BEK+00]. Die Beschreibung des Web Service erfolgt mit WSDL 1.1[CCMW01].

Ein Aspekt der gegenseitigen Synchronisierung ist der Identitätsabgleich der einzelnen Items („Identity Reconciliation“). So kann eine einzelne Ressource (Configuration Item, Relation) in mehreren MDRs auf unterschiedliche Weise identifiziert werden. Eine Voraussetzung ist aber, dass die Ressource innerhalb der MDR eindeutig identifizierbar ist und dieses Identifizierungsmerkmal (ID) sich im Laufe der Zeit nicht ändert. Aufgabe der CMDBf ist es, die IDs aus den jeweiligen MDRs zusammenzuführen und daraus eine „Ressource Identity“ zu genieren. Dies kann mit Hilfe einer automatischen Analyse oder durch manuelle Eingaben geschehen.

Die Definition des Datenmodells führt zunächst den Begriff der „managed data“ ein. Dabei handelt es sich um Informationen, die von MDRs erfasst und mit der CMDBf ausgetauscht werden. Es werden drei Element-Typen unterschieden.

Item Ein Item kann entweder eine Ressource (wie z.B. ein Computer-System oder eine Anwendung) oder ein Prozess-Artefakt (Incident record, RfC-Formular) sein. Es ist somit eine Übermenge des klassischen Configuration Items nach ITIL oder ISO/IEC20000.

Jedes Item muss mindestens eine eindeutige ID innerhalb der MDR besitzen (Item-ID), die über die komplette MDR-Lebenszeit konstant bleibt. Jede MDR besitzt innerhalb des CMDBf-Verbunds eine eindeutige MDR-ID. Zusätzlich besitzt jedes Item eine „Instance ID“, die sich aus der MDR-ID und der Item-ID zusammensetzt. Ein Client kann somit aus der Instance-ID die MDR-ID extrahieren und via „Query Service“ weitere Informationen zu diesem Item von der entsprechenden MDR anfordern. Sollte nun die CMDBf im Rahmen der Synchronisierung Daten eines Items verändern, so muss die Instance-ID ebenfalls verändert werden. Die CMDBf überschreibt dabei die MDR-ID mit ihrer eigenen MDR-ID. Dieses Konzept erlaubt Caching. Das bedeutet, dass die CMDBf auch direkt auf Anfragen der Clients an MDRs beantworten kann. Die Informationen aus der CMDBf dürfen dabei von den Informationen aus der MDR abweichen.

Relationship Ein Relationship repräsentiert eine Verbindung zwischen zwei Items. Man unterscheidet zwischen Quell- und Ziel-Item. Beispiele für Relationen zwischen Items wurden bereits in Kapitel 2.1 beschrieben. Jede Relation muss - wie auch schon ein Item - eine eindeutige ID innerhalb einer MDR besitzen. Nach der Reconciliation-Phase kann eine Relation auch mehrere IDs besitzen.

Record Ein Record enthält Informationen über ein Item oder ein Relationship. Es können sowohl informative als auch deskriptive Informationen enthalten sein. Records können von unterschiedlichen Typen sein, um unterschiedliche Informationen zu speichern (beispielsweise Asset- und Konfigurationsinformationen). Im Entwurfsdokument der Workgroup wurden Records mit Zeilen einer SQL-Datenbank verglichen. Jeder Record ist somit eine Zeile, die bestimmte Informationen vorhält. Dabei ist es auch möglich, dass identische Informationen in mehreren zu einem Item gehörige Records gespeichert werden. Jeder Record enthält neben den Nutzdaten auch Metadaten. Dazu zählt primär die eindeutige ID zur Identifizierung. Optional kann ein Record auch einen Zeitstempel der letzten Änderung, eine Baseline-ID, die die Zugehörigkeit zu einer Baseline angibt und eine Snapshot-ID, die die Verbindung zu einem Snapshot definiert, enthalten.

Im Anhang A wird der Aufbau einer beispielhaften Anfrage kurz erläutert.

Kritik

Der Nutzen der CMDBf-Standardisierung wird bereits vor der Erklärung zum Standard kritisch hinterfragt. Kritiker bemängeln, dass die CMDB Federation Workgroup entscheidende Punkte in ihrem Paper nicht definiert[Eng09b]: So wird zwar definiert, wie Tools Daten austauschen können, aber nicht, was sie explizit austauschen. Es gibt kein Datenmodell, Schema oder eine Semantik für den Datenaustausch. Somit kann nicht garantiert werden, dass unterschiedliche Toolhersteller kompatibel zueinander sind. Diese Tatsache wird von den Mitgliedern der CMDB Federating Workgroup relativiert. In einem Whitepaper von CA werden die Vorteile von CMDBf, wie z.B. die Kompatibilität unterschiedlicher Hersteller, hervorgehoben [DLM+09]. Die dort gemachten Äußerungen werden aber ebenfalls als zu überzogen von Kritikern bemängelt [Eng09a].

Auch Gartner untersuchte, ob es mit Hilfe der CMDBf in Zukunft einen Standard im Configuration Management geben wird[AC07]. Durch die ungenaue Spezifikation einiger Aspekte

(Datenformat und Schema) werden die Hersteller nach wie vor ihre eigenen Lösungen implementieren. Zudem möchten Hersteller aus wirtschaftlicher Sicht gerne ihre eigene Produkte und Komplettlösungen verkaufen, die eine CMDBf im eigentlichen Sinne unnötig machen. Sollte die Distributed Management Task Force (DTMF) CMDBf zum Standard erklären, werden die Toolhersteller gezwungen sein, diesen Standard in Ihre Produkte umzusetzen. Bis dahin sollte man laut Gartner sein Augenmerk auf die bereits vorhandenen Integrationsmöglichkeiten der Tools lenken.

2.5. Zusammenfassung

In diesem Kapitel wurden grundsätzliche Anforderungen an das Configuration Management herausgearbeitet. Zur Hauptaufgabe des Configuration Managements gehört der Aufbau und Betrieb einer Configuration Management Database (CMDB). Diese Datenbank enthält alle unternehmensrelevanten Informationen über Hardware, Software, Verträge sowie deren physische und logische Verbindungen und Abhängigkeiten zueinander. Der Aufbau einer CMDB unterscheidet sich für jedes Unternehmen, da jedes Unternehmen unterschiedliche Business-Prozesse mit unterschiedlichen Anforderungen implementiert. Für den Aufbau einer CMDB kann man sich an standardisierten Verfahren zum Projektmanagement und Software Engineering orientieren (V-Modell). Mit Hilfe des PDCA-Zyklus nach Deming erreicht man eine kontinuierliche Verbesserung der CMDB, die für die Effizienz in einem Unternehmen maßgeblich ist. Um die Qualität der CMDB herauszufinden kann man sich an Reifegradmodellen ähnlich dem CMMI-Modell orientieren. In der Praxis wird man selten auf eine isolierte CMDB treffen. Vielmehr gibt es zahlreiche - zunächst isolierte - Daten-Silos, sogenannte Management Data Repository's (MDRs), die ihre Daten mit der zentralen CMDB synchronisieren müssen. Für den Integrations- und Synchronisationsprozess gibt es das Konzept der föderierten CMDB (CMDBf). Dieser Normvorschlag befindet sich allerdings noch im Entwurfsstadium. Daher ist es fraglich, ob das CMDBf-Datenaustauschformat sich nach der Standardisierung durchsetzen wird.

Mit Hilfe des in diesem Kapitel erarbeiteten Wissens sollen nun im Folgenden einige Standard-Verfahren am LRZ untersucht werden, um daraus ein Konzept für ein Informationsmodell zu erstellen.

3. Analyse von Standard-Verfahren für die Einführung von IT Servicemanagement am LRZ

Der Wunsch nach Verbesserung der internen Abläufe, der Servicequalität sowie einer exakten Prozessdefinition ist am LRZ die Motivation für den Aufbau eines durchgängigen IT-Servicemanagements. Zu diesem Zweck wurde im Oktober 2007 ein Arbeitskreis gegründet (AK ITSM), mit der Aufgabe die Einführung des Servicemanagements zu planen und zu realisieren. Ziel des AK ITSM ist die mittelfristige Zertifizierung des LRZ als Unternehmen nach ISO/IEC20000.

Für die Umsetzung von IT-Servicemanagement wurde ein ITSM-Tool gekauft, das allen Anforderungen des LRZ genügt, um die Basis für eine prozessorientierte IT zu schaffen. Als Tool wurde im Rahmen einer ausführlichen Tool-Evaluation „iET ITSM 5.0“ von iET Solutions ausgewählt.

Vor dem Einsatz eines ITSM-Werkzeugs ist es erforderlich, die notwendigen Voraussetzungen zu schaffen. Es muss entschieden werden, welche MDRs in Zukunft durch den Einsatz des Tools obsolet werden. Für diese Tools muss ein Migrationsplan erstellt werden. Des Weiteren gibt es in der Regel Datenquellen, die weiterhin bestehen bleiben müssen. Für diese Datenquellen muss eine Integration vorbereitet werden. Zuletzt erfolgt eine Anpassung des ITSM-Tools, da jedes Unternehmen andere Bedürfnisse besitzt.

3.1. Vorgehensweise bei der Analyse

Zunächst werden Verfahren ausgewählt (Abschnitt 3.2), die als Erstes für den Einsatz des ITSM-Tools vorbereitet werden, um wichtige Use-Cases abzubilden. Hierzu werden auf Basis von Prozessdefinitionen und Verfahrensanweisungen - sofern vorhanden - die Abläufe analysiert. Dabei wird weniger Wert auf eine exakte Beschreibung des Ablaufs gelegt, sondern darauf, das Verfahren aus Sicht des Configuration Managements zu analysieren.

Neben einem groben Verfahrensablauf ist es vor allem wichtig zu wissen, welche Rollen an dem Prozess beteiligt sind und welche Tools eingesetzt werden, um Konfigurationen vorzunehmen und unterstützende Informationen zu erhalten. Bei den eingesetzten Tools werden nur diejenigen betrachtet, die Informationen über ein zukünftiges Configuration Item speichern, die von mehreren Personen genutzt werden. Dabei handelt es sich gemäß der Definition [CMD08] um Management Data Repositorys (MDRs), die später integriert oder konsolidiert werden. Die Bewertung der Tools geschieht in Kapitel 4.

3.2. Verfahren

Die Einbindung aller Unternehmensabläufe in das ITSM-Tool geschieht üblicherweise in mehreren Stufen. Zunächst sollten gemäß [Det08] nur zwei bis drei wichtige Services in das

Tool integriert und die übrigen Services nach und nach hinzugezogen werden. Auf diese Weise wird eine aussagekräftige CMDB erzeugt und man verliert sich nicht zu Beginn in zu vielen Details.

Am LRZ wurde als Musterprozess der Betrieb von virtuellen Maschinen ausgewählt. Der Grund liegt darin, das es sich in Zukunft um einen der wichtigsten Prozesse handeln wird, da diese Dienstleistung kostenpflichtig nach außen hin angeboten wird. Somit ist es auch wichtig, dass es für diesen Prozess ein standardisiertes Verfahren gibt. Zudem bietet diese Dienstleistung einen sehr guten Querschnitt durch viele Bereiche des LRZ und führt dazu, dass im ITSM-Tool viele wichtige Bereiche zu Beginn berücksichtigt werden.

Im Rahmen dieser Arbeit werden zudem noch zwei weitere Verfahren (Accesspoint-Installation und physische Maschinen) in die Konzeption mit eingebunden, da diese Verfahren in naher Zukunft ebenfalls standardisiert ablaufen werden und die Sichtweise auf die Organisationsstruktur abrunden. Das Accesspoint-Verfahren erweitert das Modell der Netzinfrastruktur. Physische Server erweitern das Modell dagegen in erster Linie um Anbindungsinformationen (unter anderem Power Distribution Unit (PDU)) und Standortinformationen, die dem Modell der virtuellen Infrastruktur fehlen.

3.2.1. Installation eines Accesspoints

Im Rahmen einer Diplomarbeit wurde der Prozess der Installation von Accesspoints bereits ausführlich untersucht, bewertet und optimiert [Sch08]. Für diese Analyse wurden der optimierte Prozess betrachtet und um die Sichtweise aus dem Configuration Management erweitert.

Das LRZ stellt im gesamten MWN die Infrastruktur für den drahtlosen Zugang ins Netz bereit. Das WLAN-Netz wird dabei kontinuierlich erweitert und verbessert. Institute (Kunde) können jederzeit eine Erweiterung der WLAN-Infrastruktur beantragen. Handelt es sich um öffentlich zugängliche Räume werden die Kosten durch eine Kostenstelle des LRZ abgedeckt. Handelt es sich aber um private Institutsräume muss das Institut bzw. der Lehrstuhl für die Kosten aufkommen. In beiden Fällen kümmert sich das LRZ um die Beschaffung, Konfiguration, Installation und Wartung des Accesspoints. Abbildung 3.1 zeigt die in diesem Kapitel betrachteten Use-Cases.

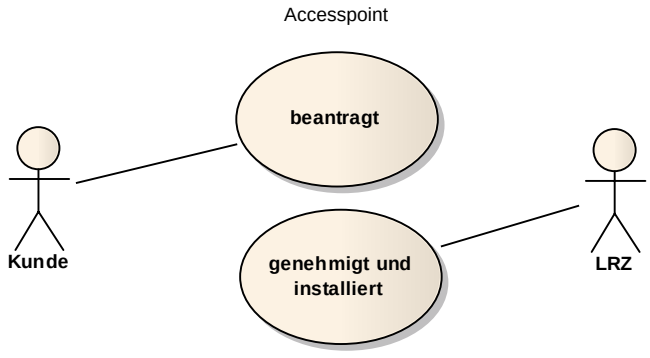


Abbildung 3.1.: Use-Case des Accesspoint-Verfahrens

Verfahrensüberblick

Ein Kunde des LRZ (Change Requester) stellt einen informellen Antrag (Request for Change), der via Brief, Fax oder E-Mail eingereicht wird. Die Gesamtleitung der Abteilung KOM (Change Authority) überprüft den Antrag auf Korrektheit und Vollständigkeit und legt das Accounting fest (Kostenübernahme durch das LRZ oder Kunden). Der Antragsteller erhält daraufhin einen formlosen Bescheid mit dem Ergebnis der Prüfung. Im Falle einer Genehmigung wird an dieser Stelle ein sogenannter Laufzettel angelegt. In diesem einseitigen Word-Dokument werden alle relevanten Informationen für die Installation Schritt für Schritt vervollständigt. Als nächstes legt der WLAN-Beauftragte den Standort des Accesspoints auf Basis von Gebäude- und Bauplänen vorläufig fest. Hierzu dienen in Microsoft Visio oder Microsoft Paint angelegte Pläne. In der Regel liegen die Gebäudepläne bereits im LRZ vor, in Ausnahmefällen müssen diese separat angefordert werden. Die Netzwartung des zu diesem Standort gehörenden Unterbezirks prüft nun vor Ort die Gegebenheiten. So wird ein geeigneter Netz- und Stromanschluss lokalisiert. Die Accesspoints werden in der Regel durch Power-over-Ethernet mit Strom versorgt. Es muss sich somit ein geeigneter Switch in der Nähe befinden. Sollte an diesem Standort kein entsprechender Netzanschluss vorhanden sein, müssen zuerst die nötigen Komponenten installiert werden. Sobald alle Voraussetzungen erfüllt sind, kann der WLAN-Beauftragte die Konfiguration des Accesspoints vornehmen und das Gerät für den Einsatz vorbereiten. Dazu nutzt er neben Tools zur Accesspoint-Installation die Software „Remedy ARS“, um sich eine IP-Adresse reservieren zu lassen. Parallel dazu bereiten die Switch-Administratoren die Konfiguration aller Switches vor, die sich auf dem Weg von zentralen Netzkomponenten zu dem neuen Accesspoint befinden. In der Regel müssen dazu die entsprechenden VLANs geschaltet werden. Die Switch-Administratoren nutzen dabei Microsoft Visio, die die Anbindung zu den Instituten visualisiert, sowie ggf. die LRZ Netzdoku. Anschließend wird der Accesspoint von der Netzwartung vor Ort installiert und zusammen mit dem WLAN-Verantwortlichen ein Funktionstest durchgeführt. Hierzu wird Software eingesetzt, die die Feldstärke, Reichweite und Abdeckung des Accesspoints misst. Da es sich um ein reines Diagnose-Tool handelt wird es im Weiteren nicht betrachtet. Nach erfolgreicher Installation kann der Change Agent den RfC abschließen und den Kunden über den neuen Accesspoint informieren.

Beteiligte Rollen

An diesem Prozess ist neben dem externen Kunden ausschließlich die Abteilung KOM beteiligt, wie zusammenfassend aus der Tabelle 3.2 entnommen werden kann. Die Kommunikation zwischen Kunde und LRZ erfolgt formlos per eMail, Fax oder Brief. Zunächst informiert der Abteilungsleiter KOM den Kunden über die Genehmigung oder Ablehnung des Antrags. Im weiteren Verlauf übernimmt der Gruppenleiter der KOM-Betriebsgruppe die Kommunikation mit dem Kunden. Intern erfolgt die Kommunikation zwischen den beteiligten Rollen bevorzugt über eMail. Die Installationstermine (für etwaige Netzkomponenten bzw. den eigentlich Accesspoint) werden in „Remedy ARS“ als KOM Change Record (KCR), einer Vorstufe zu den Change Records im IT-Servicemanagement-Sprachgebrauch, dokumentiert und in regelmäßigen Meetings kommuniziert.

Prozessrolle	Zugeordnete Stellen(n)
Change Requester	Kunde
Change Authority	Abteilungsleiter KOM
Change Agent	Gruppenleiter KOM - Betrieb
WLAN-Beauftragter	Abteilung KOM - Betrieb
Field Support	Abteilung KOM - Netzwartung
Switch-Administrator	Abteilung KOM - Betrieb

Abbildung 3.2.: Rollen und involvierte Stellen

Verwendete Datenquellen

Jeder Accesspoint wird zunächst buchhalterisch durch das Asset-Management-Modul von „Remedy ARS“ erfasst. Sämtliche Baupläne liegen als Microsoft Visio-Grundriss oder Paint-Zeichnung vor und dienen der Standortwahl und Frequenzplanung. Anschließend werden alle bisher bekannten Daten in einen Laufzettel eingetragen. Die verantwortliche Person der Netzwartung kann mittels LRZNetzdoku ermittelt werden. An sie wird der Laufzettel zunächst abgegeben und dort weiter vervollständigt. Die Switch-Administratoren nutzen Teilinformationen der LRZNetzdoku, ihre eigenen Visio-Zeichnungen für die Zusammenhänge zwischen den Switches auf dem Weg zum Accesspoint sowie Excel-Listen für Zentralswitches. Zur Konfiguration des Accesspoints wird zunächst die IP-Adresse im entsprechenden Modul in „Remedy ARS“ reserviert und der DNS-Name mit Hilfe der „Nixu NameSurfer Suite (NSS)“ in den DNS-Servern eingetragen. Parallel dazu pflegt der WLAN-Beauftragte eine Excel-Liste für die Dokumentation aller zu einem Standort gehörenden Accesspoints.

Management Data Repository	Verwendungszweck
LRZ Netzdoku	Dokumentation der Switche/VLANs/Standorte
Microsoft Word	Laufzettel
Microsoft Excel	Datentabelle Accesspoints, Switch-Konfiguration
Microsoft Visio/Paint	Bau-, AP- und Frequenzpläne
Nixu NameSurfer Suite (NSS)	DNS-Konfiguration
Remedy ARS	Asset-Management

Abbildung 3.3.: Eingesetzte MDRs für das Accesspoint-Verfahren

Schritt	Aktivität	Change Requester	Change Authority	Change Agent	WLAN-Beauftragter	Field-Support	Switch-Administrator	LRZ Netzdoku	Microsoft Excel	Microsoft Word	Microsoft Visio/Paint	Nixu NameSurfer Suite (NSS)	Remedy ARS
1	RfC einreichen	R											
2	Antrag prüfen und genehmigen	I	RA		I								
3	Vorbereitung												
3.1	Accesspoint beschaffen			A	R							x	
3.2	Standort vorläufig festlegen				R					x			
3.3	Laufzettel anlegen				R				x				
3.4	Installationsvoraussetzung prüfen					R							
3.5	Installationsvoraussetzung schaffen					R							
3.6	Accesspoint vorbereiten				R			x			x	x	
3.7	Switch-Konfiguration						R	x	x				
4	Accesspoint-Installation					R							
5	Installation überprüfen				R	C							
6	RfC abschließen	I		R									

Tabelle 3.1.: Gesamtübersicht über das Verfahren für Accesspoints

3.2.2. Einrichtung einer virtuellen Maschine

Das LRZ ist momentan dabei eine große VM-Infrastruktur aufzubauen, um dem steigenden Bedarf an Hardware für Server zu reduzieren. Ein Teil der Infrastruktur soll schließlich gegen Entgelt an Endkunden vermietet werden. Da auch viele LRZ-Dienste intern mittlerweile virtualisiert betrieben werden, entwickelt sich der Komplex „Virtuelle Infrastruktur“ immer mehr zu einer bedeutenden Dienstleistung. Aus diesem Grund ist es notwendig, die Abläufe weitestgehend zu standardisieren und Prozessdefinitionen festzulegen.

Mit Hilfe der neuen Infrastruktur können ca. 3200 virtuelle Maschinen mit vernünftiger Ausstattung gleichzeitig betrieben werden. Gerade für das Anlegen und die Ersteinrichtung von virtuellen Maschinen benötigt man einen effizienten Workflow. Das bedeutet, dass die Maschinen idealerweise nach der Genehmigung und Freigabe automatisch eingerichtet werden sollten.

Die Einrichtung von Maschinen mit Windows- und Linux-Betriebssystemen übernehmen jeweils zwei voneinander unabhängige Abteilungen. Für die Einrichtung mit Linux-Betriebssystemen gibt es bereits einen ersten Entwurf zur Verfahrensdokumentation [BR09].

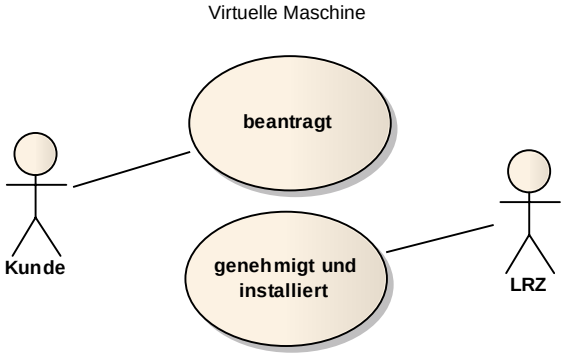


Abbildung 3.4.: Use-Case des Verfahrens für virtuelle Maschinen

Verfahrensüberblick

Ein Kunde (LRZ intern oder extern) meldet schriftlich Bedarf an. Hierzu gibt es bereits ein RfC-Formular für Linux-Betriebssysteme, dass alle relevanten Daten aufnimmt. Dazu zählen auch VM-spezifische Daten wie unter Anderem die Größe des Arbeitsspeichers und die Anzahl der CPUs. Abhängig vom zu installierenden Betriebssystem genehmigt der Abteilungsleiter der Linux oder Windows-Gruppe (Change Authority) die Einrichtung der neuen virtuellen Maschine. Anschließend erfolgt die Netzkonfiguration durch den VM-Beauftragten. In der Regel sind bereits alle benötigten VLANs auf dem VMware ESX-Cluster eingerichtet. Bei Bedarf müssen diese angelegt werden. Pro gewünschtes Subnetz wird eine virtuelle Netzwerkkarte für die virtuelle Maschine eingerichtet. Die IP-Adressen und DNS-Namen werden in „Remedy ARS“ beantragt und reserviert. Als nächstes erfolgt die Erstellung und Platzierung der neuen virtuellen Maschine im VMware-Cluster. Für die automatischen Betriebssysteminstallationen (jeweils mehrere Linux- und Windowsversionen) gibt es bereits vorgefertigte Templates. Von diesen Templates wird das passende ausgewählt und entsprechend in dem ESX-Cluster eingerichtet. Die Platzierung erfolgt nach den Angaben des RfC-Formulars in dem jeweiligen Ressource-Pool (Test oder Produktion) und zugehörigem

Datastore (Test- oder Produktions-Datastore). Außerdem erfolgt nun die Anpassung der virtuellen Maschine. Dazu werden die erste IP-Adresse und der DNS-Name eingetragen, gegebenenfalls die Hardware-Daten der virtuellen Maschine verändert (wie z.B. RAM-Größe, zusätzliche Festplatte). Während die Installation der virtuellen Maschine angestoßen wurde, kann - je nach Wunsch des Kunden - ein Backup-Bereich (Node) zur Datensicherung eingerichtet werden. Das LRZ nutzt zur Datensicherung „IBM Tivoli Storage Manager“ (TSM). Die Client-Software ist sowohl für Linux als auch Windows verfügbar. Die Einrichtung einer TSM-Node erfolgt durch die Gruppe DAT. Diese benötigt den Node-Namen (in der Regel identisch zum Namen der virtuellen Maschine), eine grobe Schätzung der zu sichernden Datenmenge, der Anzahl der Prozessoren und die Festlegung des Backup-Schedulings. Nach erfolgreicher Einrichtung durch die Gruppe DAT bekommt der VM-Verantwortliche die entsprechenden Einstellungen (TSM-Server, Portnummer, Initial-Passwort) zurück, die während der nun folgenden Post-Install-Prozeduren in der virtuellen Maschine konfiguriert werden müssen. Die Post-Install-Prozeduren sind abhängig vom gewählten Betriebssystem. Beide Systeme werden zunächst per Hand angepasst. So wird beispielsweise der TSM-Client konfiguriert. Bei Linux-Systemen wird die Maschine anschließend in das Tool „LRZmonitor“ eingetragen, dass der Einrichtung von Software, der Aktualisierung des Systems und der Überwachung dient. Bei Windows-Systemen übernimmt dies der „Microsoft System Center Configuration Manager“, dessen Client bereits in dem Betriebssystem-Template enthalten ist. Sollte der Kunde direkten Zugang auf die VMware-Konsole wünschen, muss dies der VM-Beauftragte konfigurieren. Dazu muss bereits eine Benutzerkennung im MWN-weiten Active Directory bestehen. Der VM-Beauftragte kann nun dem Benutzer die entsprechenden Rechte an der VM einrichten. Als nächstes erfolgt die Dokumentation der neuen Maschine in „Remedy ARS“. Dazu wird für die virtuelle Maschine ein Geräteticket angelegt, für jede virtuelle Netzwerkkarte ein Einzelticket vom Typ „IP“. Abschließend erfolgt ein kurzer Funktionstest, in dem alle Parameter überprüft werden, sowie die Erreichbarkeit der Maschine sichergestellt wird. Sollte dieser Test erfolgreich verlaufen sein, erfolgt eine Meldung über Fertigstellung an den Kunden und die Change Authority.

Beteiligte Rollen

Da das Verfahren abhängig vom gewählten Betriebssystem ist, teilen sich mehrere Abteilungen die Einrichtung der virtuellen Maschine.

Prozessrolle	**Zugeordnete Stellen(n)**
Change Requester	Kunde
Change Authority (Linux)	Gruppenleiter HLS - COS
Change Authority (Windows)	Gruppenleiter BDS - PC
VM-Beauftragter	Abteilung HLS - COS
TSM-Beauftragter	Abteilung HLS - DAT
Linux-Administrator	Abteilung HLS - COS
Windows-Administrator	Abteilung BDS - PC

Abbildung 3.5.: Rollen und involvierte Stellen

Verwendete Datenquellen

Die Provisionierung einer neuen virtuellen Maschine wird mit Hilfe des VMware Infrastructure Clients durchgeführt. Dort wird auf Basis eines hinterlegten Templates der Setup-Prozess der Maschine angestoßen. Somit handelt es sich um das wichtigste Tool in diesem Prozess. Des Weiteren findet eine organisatorische Erfassung mit Hilfe von „Remedy ARS“ statt. Dort werden sowohl die Maschine als auch die Netzwerkkarten, die benutzten IP-Adressen und DNS-Namen erfasst. Für das Backup-Management steht das Tool „DATweb“, eine webbasierte Schnittstelle zum IBM „Tivoli Storage Manager“, bereit. Abhängig vom installierten Betriebssystem verwalten am Ende des Prozesses die jeweiligen Tools (MS SCCM oder LRZmonitor) die Software-Konfigurationen. Diese Tools nutzen dazu Daten aus ihren jeweiligen Konfigurations-Datenbanken, um entsprechende Software und Updates einzuspielen.

Management Data Repository	**Verwendungszweck**
DATweb	Anlegen von TSM-Nodes
LRZmonitor	Überwachung und Software-Konfigurationsmanagement für Linux
Microsoft System Center Configuration Manager (SCCM)	Software-Konfigurationsmanagement für Windows
Remedy ARS	IP-Adressen & Inventarisierung der VM
VMware Infrastructure	Konfiguration der virtuellen Maschinen
Microsoft Excel	Switch-Dokumentation
LRZ Netzdoku	Dokumentation der Switche/VLANs/Standorte

Abbildung 3.6.: Eingesetzte MDRs für virtuelle Maschinen

3.2.3. Einrichtung eines physischen Servers

Für die Beschaffung und Installation eines neuen Servers gibt es am LRZ bisher noch keine Verfahrensbeschreibung und kann nicht im selben Umfang wie die beiden vorangegangenen Verfahren analysiert werden. Trotzdem wird dieser Use-Case im Rahmen dieser Arbeit berücksichtigt, da er in erster Linie das virtuelle Infrastrukturkonzept des LRZ vervollständigt. So muss zunächst Server-Hardware beschafft, installiert und im Serverraum platziert werden. Dabei kann es sich sowohl um einen Clusterknoten für die virtuelle Infrastruktur als auch um einen echten, unabhängigen Server handeln.

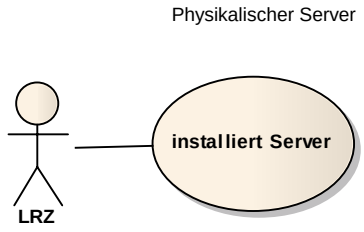


Abbildung 3.7.: Use-Case des Verfahrens für physische Server

Schritt	Aktivität	Change Requester	Change Authority	VM-Verantwortlicher	TSM-Verantwortlicher	Linux-Administrator	Windows-Administrator	DATweb	LRZmonitor	MS SCCM	Remedy ARS	VM-Infrastructure	Switch-Dokumentation	LRZ Netzdoku
1	RfC einreichen	R												
2	Antrag überprüfen			R										
3	Antrag genehmigen	I	R	C										
4	Netzkonfiguration			R							x		x	x
5	Installation und Placement der VM			R		C	C					x		
6	TSM-Node Einrichtung			I	R			x						
7	Post-Install-Prozeduren					R	R		x	x				
8	Einrichtung des Kundenzugangs			R								x		
9	Dokumentation		A	R							x			
10	Funktionstest			R								x		
11	Meldung über Fertigstellung	I	I	R										

Tabelle 3.2.: Gesamtübersicht über das Verfahren für virtuelle Maschinen

Verfahrensüberblick

Der Bedarf an neuer Hardware wird innerhalb der betroffenen Gruppe bzw. Abteilung des LRZ kommuniziert. Nach Genehmigung durch den Vorgesetzten wird die Hardware bestellt. Parallel beginnt die Dokumentation der Hardware in „Remedy ARS“. Hierzu werden die entsprechenden Einzelteil- und Gerätetickets angelegt.

Als nächstes wird ein Standort im Serverraum festgelegt, an dem die Hardware später aufgestellt wird. Nun beginnt die Vorbereitung des Standorts. In Absprache mit den Switch-Administratoren werden die entsprechenden Switch-Ports vorbereitet und mit den benötigten VLANs beschaltet. Dazu werden auf die Excel-Listen zur Switch-Dokumentation und die „LRZ Netzdoku“ zurückgegriffen. Als nächstes wird die Stromversorgung vorbereitet. Je nach Wichtigkeit des Servers erfolgt der Anschluss an die autonome oder normale Stromversorgung. Da das Rechenzentrum als Darkcenter betrieben wird, d.h. die Mitarbeiter arbeiten außerhalb des eigentlichen Rechnergebäudes, sind alle Server an aus der Ferne schaltbare Steckdosen (PDU) angeschlossen. Die Dokumentation, an welcher Steckdose welcher Server angeschlossen ist, erfolgt dabei ausschließlich innerhalb der Konfigurationsdatei des Tools zum Schalten der Steckdosen.

Nach der Lieferung erfolgt die Installation durch die jeweilige Untergruppe der Abteilung. Der Installationsmechanismus ist wie bereits bei der Installation der virtuellen Maschine vom Betriebssystem abhängig. Bei normalen Servern kommen auch hier Tools wie der LRZ-monitor (Linux) oder MS SCCM (Windows) zum Einsatz. Im Falle der Erweiterung der VM-Infrastruktur wird hier ausschließlich der VMware ESXi-Hypervisor installiert. Das System muss abschließend einem ESX-Cluster zugewiesen werden und ist dann einsatzbereit.

Verwendete Datenquellen

Management Data Repository	Verwendungszweck
Konsolenskript	Steuert und dokumentiert die schaltbaren PDUs
LRZmonitor	Überwachung und Software-Konfigurationsmanagement
LRZ Netzdoku	Dokumentation der Switche/VLANs/Standorte
Microsoft Excel	Switch-Dokumentation
Microsoft System Center Configuration Manager (SCCM)	Software-Konfigurationsmanagement für Windows
Remedy ARS	IP-Adressen & Inventarisierung
VMware Infrastructure	Einrichten des neuen Cluster-Knotens

Abbildung 3.8.: Eingesetzte Datenquellen für physische Server

3.3. Zusammenfassung

In diesem Kapitel wurden Verfahren analysiert, die bei der Einführung des ITSM-Tools am LRZ eine wichtige Rolle spielen. Der Prozess „Einrichtung einer virtuellen Maschine“ (Kapitel 3.2.2) wurde durch den AK ITSM als Musterprozess definiert und wird als erstes vollständig in das Tool integriert werden. Dieser Prozess bildet einen guten Querschnitt durch viele Abteilungen und Tools. Gerade für das Configuration Management ist es wich-

tig, zu wissen, welche Tools am Prozess beteiligt werden. Bevor ein Prozess vollständig in das ITSM-Tool integriert werden kann, ist es notwendig, zu analysieren, an welchen Stellen Informationen über ein zukünftiges Configuration Item gespeichert werden. Tools, die relevante Informationen vorhalten, sind damit automatisch MDRs. Mit obiger Analyse wurden die entsprechenden MDRs der jeweiligen Verfahren herausgearbeitet. Im nächsten Schritt ist es nun notwendig, zu entscheiden welche MDRs konsolidiert bzw. integriert werden können. Dies ist Ziel des nächsten Kapitels.

4. Untersuchung und Bewertung potenzieller Management Data Repositories

Auf Basis der in Kapitel 3 ausgewählten Prozesse wurden die beteiligten Tools und Datenquellen herausgearbeitet, die bei der Implementierung der oben genannten Prozesse in das ITSM-Tool eine Rolle spielen. Im diesem Kapitel werden die Aufgaben, Fähigkeiten und Funktionen der einzelnen Tools dargestellt. Da es sich zum Teil um konkurrierende bzw. unterstützende Systeme zur ITSM-Tool-Einführung handelt, wird im Anschluss eine Bewertung auf ihre Tauglichkeit als MDR durchgeführt. Hierzu wird ein Migrations- und Integrations-Wert berechnet, der die Fähigkeit zur Koexistenz mit einer CMDB bewertet.

Zunächst werden im Abschnitt 4.1 die Bewertungskriterien im Detail vorgestellt. In den darauf folgenden Abschnitten werden die Datenquellen untersucht. Da aufgrund der Menge nicht alle analysiert werden können, beschränkt sich die Bewertung auf die Tools, die im Rahmen des CMDB-Aufbaus am LRZ als erstes migriert oder integriert werden sollen.

4.1. Bewertungsrichtlinien

Ziel des Kapitels ist es, am Ende eine Aussage darüber zu treffen, ob die beteiligten Datenquellen mit Produktivschaltung der Configuration Management Database noch benötigt werden. Möglich ist, dass die Funktionalitäten vollständig mit Hilfe der ITSM-Software realisiert werden. Ein paralleler Betrieb beider Datenbasen würde den Administrationsaufwand nur unnötig erhöhen und langfristig zu Inkonsistenzen führen. Eine Migration und - damit verbunden - eine Abschaltung der bisherigen Datenquelle wäre die logische Konsequenz. Andererseits kann es aber sein, dass die Datenquelle weitere Aufgaben wahrnimmt und deren Konfigurations-Informationen an anderer Stelle genutzt werden. Die Abschaltung dieses Tools wäre nicht möglich. Vielmehr muss ein Weg gefunden werden, den Datenbestand mit der CMDB zu synchronisieren. Dies gelingt nur, wenn geeignete Schnittstellen vorhanden sind.

Auf Basis von internen Dokumenten, Interviews und Tool-Tests werden im Folgenden die potenziellen MDRs bewertet. Hierfür wurde eine Bewertungskatalog entwickelt. Dabei werden zwei Bewertungszahlen, der Integrations-Score und Migrations-Score, errechnet. Für jedes Kriterium kann somit eine Aussage getroffen werden, ob eine Integration oder Migration begünstigt wird. In der Regel sind die Kriterien binär, d.h. die Bedingung ist erfüllt (1 Punkt) oder nicht erfüllt (0 Punkte). Mehrstufige Kriterien werden an der jeweiligen Stelle erläutert. Jedes Kriterium wird zudem auf einer Skala von 0 (ohne Bedeutung) bis 4 (sehr wichtig) gewichtet. Die Tabelle 4.1 listet alle Gewichtungen mit ihrer Bedeutung auf.

Auf Basis der gewichteten Kriterien wird am Ende ein Prozentwert für die Integration und die Migration errechnet. Diese Werte sollen eine Einschätzung geben, ob die Datenquelle weiterbestehen kann oder ob eine Migration sinnvoller ist. Der höhere Wert von beiden gibt die Tendenz an.

Die Bewertung gliedert sich in drei Teilbereiche „Allgemeines“, „Funktionen“ und „Technik&Schnittstellen“ die im folgenden vorgestellt werden.

Wertung	**Gewicht**
ohne Bedeutung	0
weniger wichtig	1
wichtig	2
sehr wichtig	4

Abbildung 4.1.: Gewichtung

4.1.1. Allgemeines

Die Kategorie Allgemeines geht mit 25% in die Gesamtwertung ein.

Es wird bewertet, ob das Tool lediglich abteilungsintern oder übergreifend eingesetzt wird. Dies ist ausschließlich für den Integrations-Wert von Bedeutung, da man bei unternehmensübergreifender Nutzung von einer Vielzahl von unterschiedlichen Nutzerkreisen ausgehen muss und somit eine Integration vorteilhafter sein kann.

Ist ein Hersteller-Support vorhanden bzw. ein Wartungsvertrag abgeschlossen, so kann das Tool weiterhin betrieben werden, was für eine Integration spricht. Umgekehrt ist es sinnvoller, das Tool bei fehlender Unterstützung zu migrieren.

Die nächsten beiden Kriterien sind nicht ausschließlich objektiv einschätzbar. Zum Einen wird die Bedeutung des Tools für das Unternehmen bewertet. Hierbei handelt es sich um eine dreistufige Ausprägung (hoch, mittel, niedrig). Als „hoch“ wurde klassifiziert, was bei einem Ausfall des Tools die Service-Erbringung des Unternehmens maßgeblich eingeschränkt ist. „Mittel“ bedeutet, dass der Betrieb gestört ist, aber keine Einschränkung kritischer (evtl. mit SLA-versehener) Systeme besteht. Mit „niedrig“ ist gemeint, dass es zu keinen Einschränkungen kommt. Eine „hohe“ Bedeutung unterstreicht die Wichtigkeit des Tools und spricht auf der einen Seite für eine Integration. Auf der anderen Seite kann man aber darauf schließen, dass das Tool wichtige und kritische Konfigurations-Informationen vorhält, die somit auch im ITSM-Tool stehen sollten. Dies würde zur Migration führen.

Als letztes wird noch der Aufwand für die Wartung berücksichtigt. Unterliegt das Tool einem hohen Wartungsaufwand, spricht dies eher für eine Migration. Ein niedriger Wartungsaufwand ist eher ein Argument für die Integration.

Entscheidend für die Berechnung dieses Abschnitts sind die Gewichtungen der einzelnen Punkte, die in Tabelle 4.2 aufgeführt sind.

	Migrations-Wert	**Integrations-Wert**
Nutzungsart	ohne Bedeutung	wichtig
Hersteller-Support	sehr wichtig	wichtig
Bedeutung für das Unternehmen	sehr wichtig	sehr wichtig
Aufwand für Wartung	weniger wichtig	sehr wichtig

Abbildung 4.2.: Bewertungskriterien für Kategorie Allgemeines

4.1.2. Funktionen

Die Kategorie Funktionen geht mit 25% in die Gesamtwertung ein und enthält nur einen einzigen Punkt. Sollte das Tool noch für Funktionen eingesetzt werden, die das zukünftige ITSM-Tool nicht beherrscht (z.B. Software-Deployment, Monitoring), so begünstigt das die Integration und verhindert eine Migration. Im umgekehrten Fall, wenn das Tool ausschließlich nur zum Verwalten von Informationen dient, die auch in der Configuration Management Database gespeichert sind, ist eine Migration sinnvoller als eine Integration. In Tabelle 4.3 findet sich die Gewichtung für die Migrations- und Integrations-Werte.

	Migrations-Wert	**Integrations-Wert**
Zusätzliche Funktionen	sehr wichtig	sehr wichtig

Abbildung 4.3.: Bewertungskriterien für Abschnitt Funktionen

4.1.3. Technik & Schnittstellen

Die Kategorie Technik & Schnittstellen geht mit 50% in die Gesamtwertung ein und ist somit die wichtigste Kategorie bei der Beurteilung. Denn nur mit geeigneten Schnittstellen und einem guten Datenmodell des Tools ist es möglich, überhaupt eine permanente Integration durchzuführen. Somit müssen alle Punkte erfüllt sein, um einen hohen Integrationswert in dieser Kategorie zu erreichen.

Zunächst wird dokumentiert, ob eine Datenbank zur Speicherung von Informationen vorhanden ist. Ist dies der Fall, wird das - in der Regel relationale - Datenbankschema auf die Qualität untersucht. Nach Kemper [KE09] kommt es zu Anomalien bei der Verwendung von Relationen-Schemata die ungenügender Qualität unterliegen:

- Updateanomalie
  Es existieren Informationen mehrfach im Schema. Bei einem Update werden evtl. nicht alle Informationen aktualisiert. Zudem erhöhen redundant gespeicherte Informationen den Speicherbedarf und führen zu Leistungseinbußen, da mehrere Einträge aktualisiert werden müssen.
- Einfügeanomalie
  Das Problem entsteht, wenn man zwei Entitytypen aus der realen Anwendungswelt miteinander vermischt hat. Fügt man nur eine von beiden Entity's ein, müssen die Werte der anderen Entity auf NULL gesetzt werden.
- Löschanomalie
  Das Problem tritt auf, wenn man eine Information bei vermischten Entitytypen löschen will. Dadurch kann es passieren, dass man unter Umständen auch die Information der anderen Entity verliert.

Um diese Probleme zu vermeiden, ist es notwendig das Relationen-Schema zu normalisieren.

- 1.Normalform
  Alle Attribute müssen atomare Wertebereiche (Domänen) haben. Zusammengesetzte mengen- oder relationenwertige Attributdomänen sind somit unzulässig.
- 2.Normalform
  Voraussetzung ist die 1.Normalform sowie das jedes Nichtschlüsselattribut von jedem Schlüsselkandidaten voll funktional abhängig ist.

- 3.Normalform
  Um die 3.Normalform zu erreichen, muss zunächst die 2.Normalform erreicht werden. Außerdem muss sichergestellt werden, dass jedes Nichtschlüsselattribut von keinem Schlüsselkandidaten transitiv abhängt.

Wenn eine Datenbank und eine hinreichende Normalisierung des Schemas vorliegt, wird dies für eine Migration positiv bewertet. Für eine Integration hingegen, muss zwingend eine Datenbank vorhanden sein oder eine geeignete Im- und Exportschnittstelle (siehe nächste Kriterien)

Nach der Betrachtung der Datenbank wird das Tool auf vom Hersteller vorgesehene Import- und Exportschnittstellen untersucht. Eine direkte Verbindung zur Datenbank zählt dabei nicht als Schnittstelle. Gängige Schnittstellen sind in der Regel CSV-Dateien, SOAP-Schnittstellen und sonstige API-Schnittstellen. Für die Migration ist es positiv, wenn eine Export-Schnittstelle existiert. Eine Import-Schnittstelle ist ohne Bedeutung.

Als vorletztes wird bewertet, ob das MDR in der Lage ist, ein gespeichertes Configuration Item eindeutig zu identifizieren. Sollte das nicht der Fall sein, ist eine Integration mit automatischer Synchronisierung nicht möglich, da es Probleme bei der Zuordnung der CI-Informationen gibt. Eine Migration ist somit die bessere Möglichkeit.

Zuletzt wird geprüft ob es aktive Schnittstellen zu anderen Systemen gibt. Sollte das MDR Informationen aus anderen Tools beziehen, wird eine Migration dadurch erschwert. Für eine Integration ist dies jedoch ein Vorteil, da die Verbindungen weiterhin ohne Änderung bestehen bleiben können.

Tabelle 4.4 listet die Gewichtung der einzelnen Kriterium für den Migrations- und Integrations-Wert auf.

	Migrations-Wert	**Integrations-Wert**
Datenbank-Normalisierung (falls Speicherung in Datenbank)	weniger wichtig	sehr wichtig
Auto. Datenexport-Schnittstelle	sehr wichtig	sehr wichtig
Auto. Datenimport-Schnittstelle	ohne Bedeutung	sehr wichtig
Eindeutige Identifizierung von CIs	sehr wichtig	sehr wichtig
Verbindung zu anderen Systemen	wichtig	sehr wichtig

Abbildung 4.4.: Bewertungskriterien für Abschnitt Technik und Schnittstellen

4.2. Remedy Action Request System

Das Action Request System (ARS) ist ein Workflow-Managenment-Tool für mittlere und große Unternehmen, dass sich gut an Unternehmensbedürfnisse anpassen lässt. Seit 1994 wird ARS am LRZ eingesetzt. Ursprünglich wurde es nur für die Unterstützung des Incident Managements benutzt. Im Laufe der Jahre wurde das Tool von LRZ-Mitarbeitern kontinuierlich erweitert. So gibt es unter anderem eine Inventar-Datenbank, die alle relevanten CIs verwaltet und somit als sehr einfach konzipierte CMDB verstanden werden kann.

4.2.1. Allgemeines

Die Software wird von allen Abteilungen im LRZ genutzt. Seit vielen Jahren wird Version 6.3 betrieben, da wegen der Anpassungen ein Upgrade nicht möglich ist. Derzeit ist Version

7.5 aktuell. Für die veraltete Version gibt es keinen Herstellersupport mehr. Die Wartung und kontinuierliche Anpassung an neue Gegebenheiten ist mit „hoch“ einzustufen.

4.2.2. Funktionen

Die Software enthält einige Programmpunkte, die im Folgenden beschrieben werden. Zur besseren Übersicht wurde ein Informationsmodell auf Basis der in den Funktionen erhobenen Daten angelegt (Abbildung 4.5).

Problemmanagement / Trouble Ticket

Die Sektion erfüllt im Wesentlichen die Anforderungen, die ITIL oder ISO/IEC20000 an ein Incident- und Problemmanagement stellen. Es werden dabei drei verschiedene Arten von Tickets unterschieden:

Quick-Ticket Ein Quick-Ticket enthält im Gegensatz zu einem Trouble-Ticket ein reduziertes Eingabeformular. Es ist für Incidents gedacht, die direkt im First-Level Support bearbeitet und geschlossen werden. Aus diesem Grund werden weniger Informationen abgefragt. Mit Quick-Tickets sollen somit auch minor Incidents erfasst werden, um eine spätere Auswertung vornehmen zu können.

Trouble-Ticket Trouble-Tickets sind für Probleme konzipiert worden, die nicht direkt vom First-Level-Support gelöst werden können und somit an Spezialisten aus dem Second- und Third-Level weitergereicht werden müssen. Auf zwei Bildschirmmasken (Tab-Pages) werden detaillierte Informationen zum Kunden (Kontaktdaten) und zur Problembeschreibung abgefragt. Hierbei hilft ein Fragenkatalog, die benötigten Informationen abzufragen. Ein Trouble-Ticket kann auch durch das System automatisch angelegt und befüllt werden. Hierfür gibt es eine Web- sowie eMail-Schnittstelle. Auf der dritten Tab-Page können die Mitarbeiter interne Bearbeitungsschritte erfassen. Die vierte Maske enthält ein detailliertes Log des Incidents, so dass eine Nachverfolgung aller Änderungen jederzeit gegeben ist. Die Kommunikation zwischen Benutzer und LRZ verläuft vollständig über das Trouble-Ticket-System.

Master-Ticket Ein Trouble-Ticket, dass inhaltlich interessant ist, kann durch Setzen einer Checkbox zu einem Master-Ticket werden. Im Wesentlichen soll damit eine Art Knowledge-Database aufgebaut werden, die vom First-Level-Support heranziehbar ist.

Beschaffungswesen

Unter dem Punkt Beschaffungswesen befindet sich ein Asset Management System das im Laufe der Zeit auch Aufgaben des Configuration Managements übernommen hat. So kann mit Hilfe eines elektronischen Beschaffungsformulars (Stabeg) der Einkauf neuer Hardware dokumentiert werden. Für jede beschaffte Hardware wird zunächst ein sogenanntes Einzelteil-Ticket angelegt, dass schließlich mit einem Geräte-Ticket verbunden wird. So können leicht zusammengebaute Komponenten (wie sie beispielweise bei einem Switch auftreten) dargestellt werden. Anders ausgedrückt ist ein Einzelteil in der Regel ein physisches Gerät, dass mit einem Inventaraufkleber versehen werden kann. Somit wird beispielsweise für einen Laptop zunächst ein Einzelteil-Ticket und im Anschluss daran ein Geräte-Ticket erzeugt, dass

aus genau einem Einzelteil besteht. Mit dem Modul „Beschaffungswesen“ werden auch virtuelle Maschinen verwaltet. Hierfür ist das Verfahren identisch mit dem Anlegen einer physischen Maschine. Es wird ein Einzelteil- und Geräteticket erzeugt. Die Dokumentation der Hardwarebeschaffung aus Einzelteil- und Geräteticket wird auch als „Dokuticket“ [AH07] bezeichnet.

Stabeg Stabeg steht für „Statusblatt für bestellte bzw. beschaffte Geräte“. Es handelt sich dabei um die elektronische Version eines kaufmännischen Bestellformulars für Hardware. Im System besteht das Eingabeformular aus zwei Bildschirmmasken. Zum einem werden allgemeine Bestellinformationen eingegeben (Maske „Stabeginfo“). Dazu zählen unter anderem Besteller, eine grobe Klassifizierung der bestellten Ware (z.B. RH für Rechnerhardware) und ein Feld für Hersteller mit optionaler Produktangabe (damit auch unterschiedliche Modelle vom gleichen Hersteller bestellt werden können). Die Angabe einer Stückzahl der bestellten Ware ermöglicht, dass mit einem Klick die entsprechende Anzahl an Einzeltickets generiert und automatisch mit dem Stabeg-Eintrag verbunden werden.

Einzelteil-Ticket Das Einzelteil-Ticket wird für jede bestellte Komponente angelegt. Es stellt die „kleinste“ dokumentierbare Einheit dar, der in der Regel eine Inventarnummer zugeordnet wird [AH07]. Eine Stabeg-Bestellung führt somit automatisch zu Einzelteiltickets. Innerhalb des Tickets können neben den Produktmerkmalen auch netzwerkspezifische Einstellungen (MAC-Adresse, IP-Adresse, PXE-Gruppe) eingegeben werden. Hier zeigt sich die Schwäche der Einzelteil-Tickets: Die Netzinformationen stehen auch bei nicht-netzbezogenen Geräten (z.B. Serverschränke) zur Verfügung und werden dort nicht genutzt. Auf der anderen Seite reicht die Eingabemaske in der Regel bei Serversystemen mit mehreren Netzwerkkarten nicht aus.

Geräte-Ticket Mit dem Geräte-Ticket wird ein logisches Gerät verwaltet. Ein logisches Gerät besteht aus mindestens einem Einzelteil. Auch hier können wieder netzwerkspezifische Parameter eingetragen werden: Uplink/Downlink-Port, DNS-Name, Betriebssystem und Betriebssystem-Version.

Dienst- und Systemdokumentation

Die Idee dieser Funktion war es, Abhängigkeiten von Diensten untereinander abzubilden, um dadurch eine Impact-Analysemöglichkeit für das Change Management sowie eine Hilfe für das Incident-Management zu erreichen. Allerdings wird diese Funktion seit einigen Jahren nicht mehr genutzt und gepflegt.

Change Management

Mit Hilfe von „KOM-Change-Record“ (KCR) wurde ein einfaches Mittel geschaffen, die anstehenden Wartungsarbeiten der Abteilung KOM zu verwalten und zu dokumentieren. In groben Zügen entspricht das Ganze dem Prozess Change Management nach ITIL oder ISO/IEC20000. Auf einer einseitigen Bildschirmmaske wird der Kunde der Change-Maßnahme, eine detaillierte Beschreibung des Changes sowie ein Bearbeiter erfasst. Zusätzlich wird ein Realisierungstermin aufgenommen. Mit Hilfe dieser Daten wird für die wöchentliche Abteilungssitzung ein Forward Schedule of Change (FSC) angelegt.

IP-Adressverwaltung

Mit Hilfe der IP-Adressverwaltung können IP-Adressen reserviert werden. Die Reservierung geschieht dabei ausschließlich auf organisatorischer Ebene. Die in Remedy hinterlegten IP-Adressen werden an kein anderes System weitergegeben. Zusätzlich kann der DNS-Name und ein optionales Ablaufdatum der Reservierung angegeben werden. Außerdem kann eine Verbindung zu einem Einzelteil-Ticket hergestellt werden.

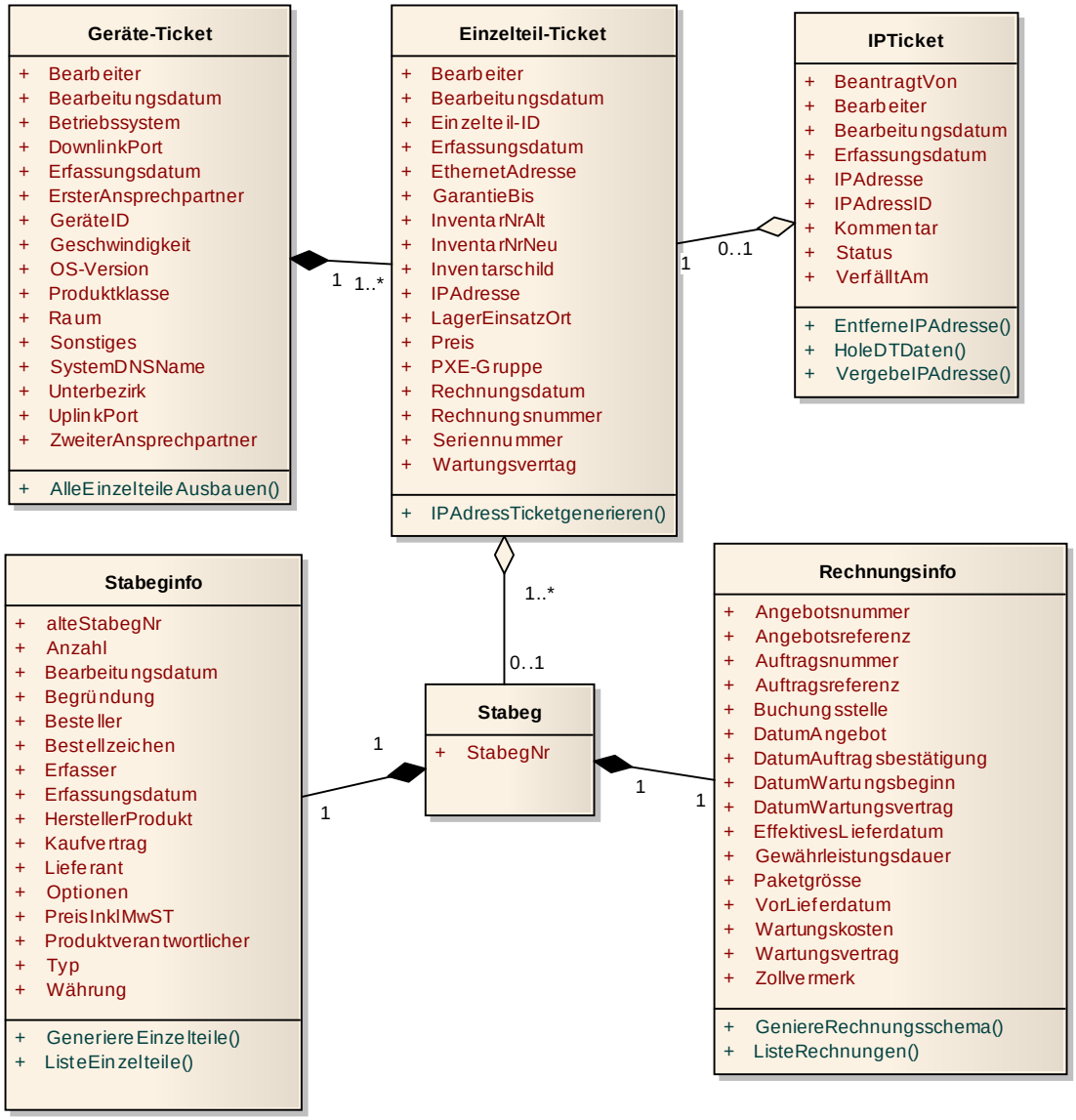


Abbildung 4.5.: Informationsmodell Remedy ARS

4.2.3. Technik & Schnittstellen

Die Software speichert die Daten in einer Oracle-Datenbank ab. Jede Konfigurations-Information trägt eine eindeutige ID. Somit ist eine Synchronisierung mit der CMDB prinzipiell

möglich. Die LRZ Netzdoku (Abschnitt 4.4) baut die Datenbank des Tools auf, um Asset-Informationen abzufragen. Es gibt sonst keine vom Hersteller vorgesehenen Schnittstellen.

4.2.4. Bewertung

Remedy ARS ist das mitunter am häufigsten eingesetzte, abteilungsübergreifende Tool. Durch die zahlreichen Anpassungen ist es zu einem Allround-Tool geworden, dass einige tägliche Arbeitsabläufe des LRZ unterstützt. Wie häufig das Tool eingesetzt wird und welche Funktionen dabei genutzt werden, zeigt eine Umfrage von Christian Richter Anfang des Jahres 2009 [Ric09]. Wie man dem Ergebnis in Abbildung 4.6 entnehmen kann, sind die am häufigsten nachgefragten Funktionen das Trouble-Ticket (Incident-Management) sowie die Geräte- und Einzelteil-Tickets (Configuration Management).

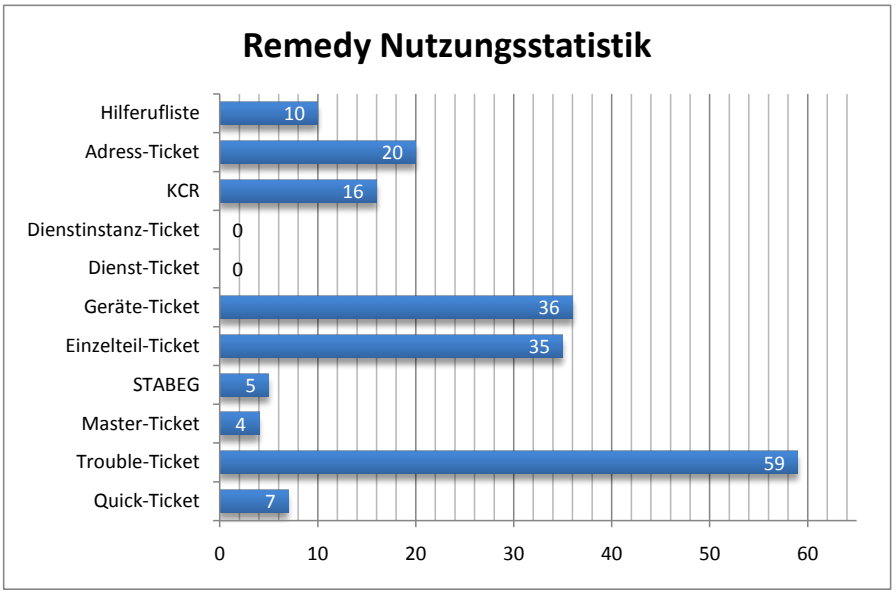


Abbildung 4.6.: Remedy Nutzungsstatistik

Durch die vielen Anpassungen ist aber mittlerweile die Grenze der Leistungsfähigkeit des Tools erreicht. Trotzdem müssten erhebliche Änderungen vorgenommen werden, um das Tool an den aktuellen Bedarf des LRZ anzupassen. Aus diesem Grund wurde durch den Arbeitskreis ITSM geplant, „Remedy ARS“ mit Einsatz von „iET ITSM“ nicht mehr weiter zu pflegen und alle Daten zu migrieren. Zu demselben Ergebnis kommen auch die Untersuchungen dieser Arbeit. So ergibt sich für „Remedy ARS“ ein Migrationswert von 68%, aber nur ein Integrationswert von 47% (siehe Bewertungstabelle 4.16). Die Empfehlung liegt somit bei einer Migration.

4.3. Switch-Dokumentation

Mit Hilfe dieser Dokumentation werden die Konfigurationsinformationen der einzelnen Switchports aufgezeichnet. Die Dokumentation der Switches erfolgt mit Hilfe von Excel-Listen.

4.3.1. Allgemeines

Obwohl Switches ausschließlich von der Abteilung KOM betreut werden, geschieht die Nutzung der Listen unternehmensweit. So kann jeder auf die Listen zugreifen und Patchaufträge direkt in die Listen schreiben. Da die Dokumentation hausintern aufgebaut und gepflegt wird, ist jederzeit eine Änderung schnell und unkompliziert möglich. Die Bedeutung der Listen ist mit „mittel“ anzusetzen, da es ohne die Listen zwar zu erheblichen Einschränkungen kommt, die Informationen aber zum Teil direkt aus den Switches ausgelesen werden können.

4.3.2. Funktionen

Grundsätzlich lassen sich zwei verschiedene Arten von Switch-Dokumentationen feststellen: Zum einen Switches, die sich im Rechnergebäude des LRZ befinden (im Folgenden bezeichnet als Zentralswitches) und zum anderen Switches, die sich an den jeweiligen Standorten und Unterbezirken befinden (bezeichnet als Standortswitches). Bei den Standortswitches werden zudem noch Visio-Zeichnungen angefertigt um die Pfad-Traversierung zum Netzmittelpunkt grafisch darzustellen.

Zentralswitches

An diesen Geräten werden in der Regel Server angeschlossen. Für die Dokumentation ist die Abteilung „Betrieb Kommunikationsnetze“ verantwortlich. Auf dem globalen Netzlaufwerk gibt es hierzu ein Verzeichnis[1] in welchem die aktuelle Version der Dokumentation vorliegt. Zur leichteren Handhabbarkeit wurde die Dokumentation auf mehrere Excel-Dateien aufgeteilt. Die Bezeichnung der Excel-Dateien gibt Aufschluss darüber, welche Serverschränke in welchem Serverraum (NSR, DAR) gemeint sind.

Die Dateinamen folgen diesem Format:

[Gruppe]-[Raum]-[Reihe][Schrank—Schränke]-[version].xls

Gruppe Bezeichnet die Zugehörigkeit. Mögliche Bezeichnungen sind BVB (Bibliothek), COS (Linux/Cluster), DAT (Datensicherung), INT und PC

Raum Gibt den Serverraum an. Mögliche Bezeichnungen: NSR (Netz- und Serverraum) und DAR (Datei- und Speichersysteme)

Schrank Schränke werden von A-Z bezeichnet, wobei A und Z jeweils Kopfstationen einer Reihe bilden. Werden mehrere Schränke in einer Datei abgebildet so werden die Buchstaben in aufsteigender alphabetischer Reihenfolge angehängt. Jeder Schrank liegt innerhalb der Excel-Datei in einem separaten Tabellenblatt.

Version Jede Person die Änderungen an der Datei vornimmt muss die Version um eins erhöhen (Kopie der Datei). Alte Versionen werden nach einer gewissen Zeit von den Switch-Administratoren in einen Archiv-Ordner verschoben.

Neue oder geänderte Einträge folgen einem festen Prozessablauf. Zunächst werden diese mit grüner Hintergrund-Farbe in die entsprechende Excel-Liste eingetragen. Anschließend werden die Switch-Administratoren über die Änderungen informiert. Diese nehmen die Änderungen

[1] I:\Share\KOM\Serverinfos

vor (via Console auf den entsprechenden Switch) und entfernen die Farbmarkierung. Bei zu löschenden Einträgen ist das Verfahren identisch, hier wird stattdessen mit roter Farbe gearbeitet. Abbildung 4.7 zeigt exemplarisch einen Ausschnitt aus einem Excel-Sheet.

Registrierte Daten pro Server:

- Servername
- IP-Adresse (*)
- Mac-Adresse (*)
- Dienst
  Die Dienste, die der Server bereitstellt (z.B. Mail, AFS, DNS, WWW, NASfiler)
- Dienstressourcen
  Eine weiterführende Angabe, wer den Dienst benötigt (z.B. Campus LMU). Die Bezeichnung wurde nicht generell vorgegeben
- Abhängigkeiten von anderen Diensten
  Die Angabe erfolgt im selben Format wie in der Spalte Dienst
- Rack
  Hier wird Raum und Schrankbezeichnung eingetragen
- Verantwortlicher
  Ansprechpartner für dieses Gerät. Kann eine einzelne Person oder eine Gruppe sein
- Switchname
  Bezeichnung des Switches
- VLAN-ID
  Aufgeschaltete VLAN-ID
- Switchport
  Switchport (Angabe einer einfachen Zahl)
- Netzanschlüsse
  Anzahl der Anschlüsse des Gerätes an das Netz
- Tagging
  Aktiviertes VLAN-Tagging (ja/nein)
- Letzte Änderung
  Datum und Kürzel der Person, die die letzte Änderung an dieser Zeile durchgeführt hat

(*) bei mehreren IP-Adressen mehr Zeilen in dem Excel-Datenblatt

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Serveranschlussdaten**	**SWM2-1WR**												
2	Servername	IP-Adresse	MAC-Adresse	Dienst	Dienstressource	Abhängigkeiten von anderen Diensten	Rack	Verant-wortlicher	Switchname	VLAN-ID	Switch ports	Netzan-schlüsse	Tagging	letzte Änderung
3														
4	datsrv31.lrz-muenchen.de	[redacted]	[redacted] (eth0)	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	8	41	1		
5	datsrv31.lrz-muenchen.de	[redacted]	[redacted] (eth1)	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	8	42	1		
6	datsrv31.lrz-muenchen.de	[redacted]	[redacted] (eth2)	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	8	43	1		
7	datsrv31.lrz-muenchen.de	[redacted]	[redacted] (eth3)	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	8	44	1		
8	datsrv31.lrz-muenchen.de	[redacted]	[redacted]	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	24	1	1		
9	datsrv32.lrz-muenchen.de	[redacted]	[redacted] (eth0)	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	8	37	1		
10	datsrv32.lrz-muenchen.de	[redacted]	[redacted] (eth1)	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	8	38	1		
11	datsrv32.lrz-muenchen.de	[redacted]	[redacted] (eth2)	ABS	TSM-Server HABS		DAR-5L	[redacted]	SWM2-1WR	8	39	1		

Abbildung 4.7.: Beispiel-Liste für Zentralswitche

Standortswitches

Diese Switches stehen verteilt innerhalb des gesamten MWN und dienen zum Anschluss der Endgeräte an das Netz. In der Regel teilen sich mehrere Lehrstühle eines Instituts einen

Switch. Zur logischen Trennung der einzelnen Lehrstühle werden portbasierte VLANs verwendet. Für die Switch-Administratoren ist es nicht wichtig zu wissen, welche Geräte angeschlossen sind, sondern an welchem Port welches VLAN geschaltet und welche Datendose mit welchem Switchport verbunden ist. Von daher unterscheidet sich die Dokumentation dieser Switches erheblich von der der Zentralswitches. Die einzelnen Excel-Dateien befinden sich geordnet nach Bezirk und Unterbezirk auf dem globalen Netzlaufwerk.[2] Für die Durchführung der Dokumentation ist die Abteilung „Wartung Kommunikationsnetze“ zuständig. Für jeden Unterbezirk existiert somit eine eigene Excel-Datei, sofern es für diesen Standort eine Patchliste gibt. Bisher scheint es noch kein Template für eine Patchliste zu geben, da die im Rahmen dieser Arbeit betrachteten Patchlisten allesamt optisch und z.T. funktional unterschiedlich aufgebaut waren. So gibt es beispielsweise einige Standorte in denen die Excel-Datei Makros enthält, die ein Sortieren nach Rack oder Raum ermöglicht. Grundsätzlich enthält jede Patchliste folgende Daten:

Allgemeines Zu den allgemeinen Daten zählen Name des Standorts, Unterbezirks-Kürzel, Gebäude-Nr. und Raumnummer.

Rack/Panel An einem Standort können sich mehrere Racks befinden. Diese werden in der Regel mit Buchstaben (A-Z) gekennzeichnet. Des Weiteren folgt die Panelport-Bezeichnung.

Dosenbeschriftung Mit Dosenbeschriftung ist die Netzdose auf Endanwender-Seite gemeint. In jedem Raum der Gebäude, in dem sich die strukturierte Verkabelung befindet, befinden sich Netzdosen, an denen Mitarbeiter ihre netzfähigen Geräte einstecken können. Jede Netzdose ist mit einem Panelport verbunden. Die Beziehung zwischen Panelport und Dose wird somit fix bei der Installation der Netzwerkleitungen definiert. Die Dosenbeschriftung folgt üblicherweise der Konvention „Raumnummer + aufsteigende Zahl“.

Switch Gibt an, mit welchem Switch dieses Panelport verbunden ist.

Port Spezifiziert mit welchem Port am oben genannten Switch der Panelport verbunden wurde.

Bemerkung Das Bemerkungsfeld ist ein optionales, freies Kommentarfeld. Es wird häufig verwendet, um auf besondere Geräte hinzuweisen, die an dem Port verbunden sind (beispielsweise ein PoE-Switch) oder welches portbasierte VLAN auf diesem Port eingerichtet wurde.

Status In diesem Feld wird dokumentiert wann an dieser Zeile im Excel-Sheet eine Änderung vorgenommen wurde.

Zusätzlich zu der Excel-Liste für Standortswitches gibt es Visio-Zeichnungen. Sie dienen dazu, einen schnellen grafischen Überblick über die Zusammenhänge der Switches von dem jeweiligen Standort bis zum Netzmittelpunkt (Zentralswitch) zu erlangen. Ein Beispiel ist in Abbildung 4.9 zu sehen. Diese Zeichnung wird benötigt, sobald ein VLAN auf dem Standortswitch eingerichtet werden muss. So ist es notwendig, das gewünschte VLAN auch auf dem gesamten Pfad bei allen Switches einzurichten. Die grafische Dokumentation erleichtert diese Arbeit. Abbildung 4.10 zeigt das erhobene Informationsmodell.

[2] I:\Share\KOM\DOKU

	A	B	C	D	E	F
1	5405	Chemie Garching				
2	AH	47 222	Lichtenbergstr. 4			
3						
4	Rack / Panel	Dosenbes.	Switch	Port	Bemerkung	Status
5	TP					
6	A-1-01	57 101 - 1	SWV3-4AH	D 24	ap01-5ah POE-8 Port	
7	A-1-02	57 101 - 2	SWV3-4AH	C 08	Bauchemie Vlan 772	
8	A-1-03	57 101 - 3	SWV3-4AH	C 02	Bauchemie Vlan 772	
9	A-1-04	57 101 - 4				
10	A-1-05	57 103 - 1				
11	A-1-06	57 103 - 2	SWV3-4AH	C 10	Bauchemie Vlan 772	
12	A-1-07	57 103 - 3				
13	A-1-08	57 103 - 4	SWV3-4AH	C 09	Bauchemie Vlan 772	
14	A-1-09	57 104 - 1	SWV3-4AH	C 03	Bauchemie Vlan 772	
15	A-1-10	57 104 - 2	SWV3-4AH	C 01	Bauchemie Vlan 772	
16	A-1-11	57 104 - 3				
17	A-1-12	57 104 - 4				
18	A-1-13	57 105 - 1				

Abbildung 4.8.: Beispiel-Liste für Standortswitches

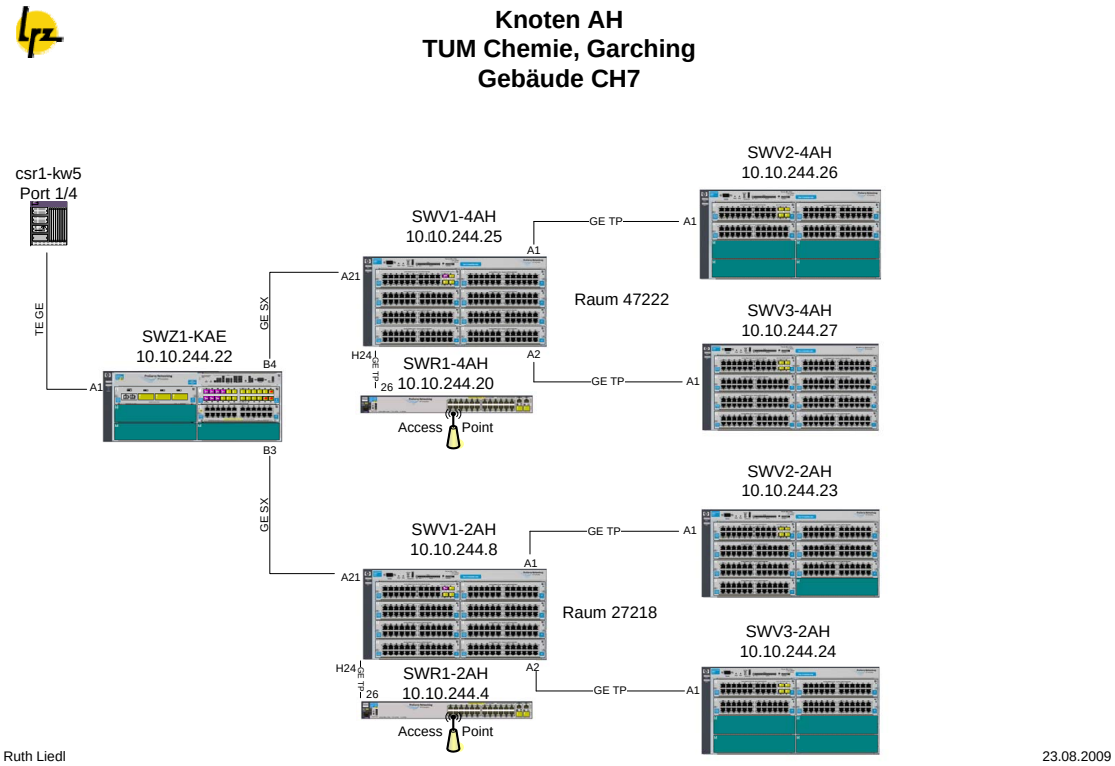


Abbildung 4.9.: Beispiel einer Switch-Pfad-Dokumentation

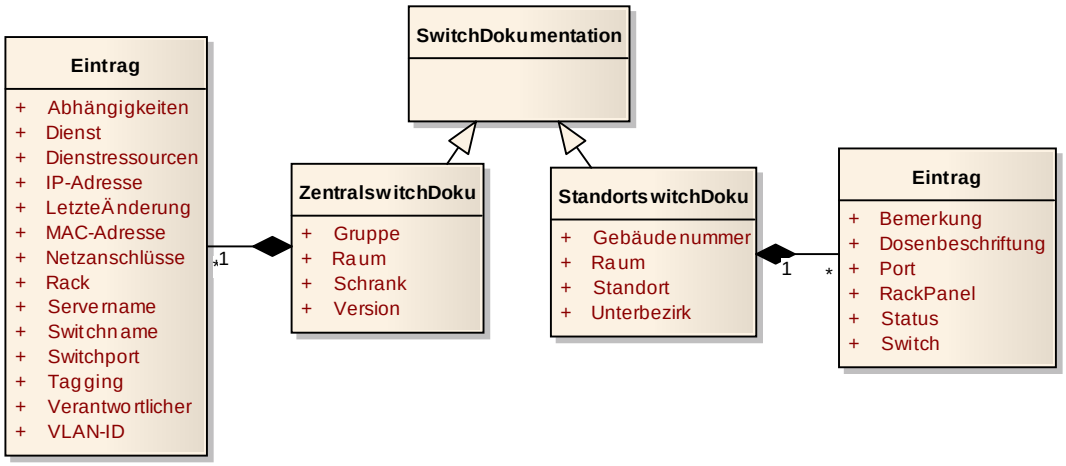


Abbildung 4.10.: Informationsmodell der Switch-Dokumentation

4.3.3. Technik & Schnittstellen

Zu diesem Punkt gibt es bei Excel-Listen wenig zu sagen. Da es sich bei einer Excel-Liste um keine Datenbank handelt, keinerlei Schnittstellen existieren und die Excel-Dateien alle ähnlich, aber nie exakt gleich aufgebaut sind, ist eine Integration nahezu unmöglich. Ein automatischer CSV-Import würde aufgrund der variablen Struktur der Einträge scheitern.

4.3.4. Bewertung

Mit Hilfe der Excel-Listen wird der komplette Soll-Zustand der Switch-Infrastruktur dokumentiert. Die Listen der Zentralswitches und deren Eintragungsverfahren sind weitestgehend standardisiert, weisen aber trotzdem Mängel auf. So fehlen oftmals die Angaben zu Diensten und Dienstabhängigkeiten, wodurch eine Impact-Analyse nicht möglich ist. Zudem ist nur eine begrenzte Auditierbarkeit der Listen gegeben. Die Dokumentation der Standortswitches besitzt noch kein standardisiertes Verfahren. Außerdem weist die Dokumentation große Lücken auf, da die Listen früher nicht eingesetzt worden sind. Lediglich neuere Standorte wurden wie oben beschrieben dokumentiert. Eine Analyse ergibt, dass es schwer möglich ist, diese Art der Dokumentation mit einem ITSM-Tool zu integrieren/synchronisieren. Diese Datenquelle muss vollständig migriert werden, die Daten sollten in Zukunft direkt innerhalb des ITSM-Tools gepflegt werden.

4.4. LRZ Netzdoku

Das Tool „Netzdoku“ ist das zentrale Tool der Abteilung KOM. Es handelt sich dabei um eine Eigenentwicklung des LRZ. Innerhalb des Tools werden organisatorische und technische Informationen zur Dokumentation der MWN-weiten Kommunikationsinfrastruktur erfasst. Dazu gehören

- Komponenten
- Anschlüsse
- Subnetze (organisatorische und IST-Daten)
- Routen (IST-Daten)

- VLANs (organisatorische und IST-Daten)
- Personen (Verantwortliche, Betreuer)
- Bezirke
- Institute
- Räume / Racks

4.4.1. Allgemeines

Das User Interface des Tools ist webbasiert und wird sowohl von Mitarbeitern aus allen Abteilungen des LRZ als auch von Netzverantwortlichen der angeschlossenen Institute genutzt. Netzverantwortliche erhalten nur lesenden Zugriff auf Teilinformationen wie z.B. Kontakt- und Subnetzinformationen.

Der Aufwand zum Betrieb der Anwendung wird mit „niedrig“ eingestuft, da kaum Änderungen vorgenommen werden müssen. Falls doch geschieht dies schnell und unkompliziert, da es sich um ein hausinternes Tool handelt.

4.4.2. Funktionen

Neben den organisatorischen Informationen (SOLL-Daten) gibt es bei einigen Punkten (VLAN, Routen, Subnetze) auch IST-Daten. Bei diesen Informationen handelt es sich um Live-Daten, die aus den Netzkomponenten regelmäßig ausgelesen werden.

Das Schema der „LRZ Netzdoku“ ist wie folgt aufgebaut:

Komponente Unter dem Überbegriff „Komponente“ werden sämtliche Netzkomponenten (Switches, Router, Repeater u.ä.) zusammengefasst. Neben einer Bezeichnung (Alias) werden auch Inventar-Daten erfasst, die aus Remedy ARS synchronisiert werden (Inventarnummer und Ticketnummer). Eine Komponente wird einem Raum zugeordnet und optional einer Liste von Personen (PersonMap) zugeordnet.

PhyPort Jede Komponente besitzt eine beliebige Anzahl an physischen Netzwerkports. Jeder Port trägt eine Bezeichnung und gehört einer PortKlasse an. Die Bezeichnung bildet das Tool logisch aus den in der Entität Produkt hinterlegten Daten der Komponente.

PhyPortKlasse Mit Hilfe dieser Entität werden die Ports nach Schnittstelle und Geschwindigkeit klassifiziert.

LogPort Jeder physische Netzwerkport stellt logische Ports zur Verfügung. Jeder Port besitzt eine IP-Adresse und befindet sich in einem Subnetz.

Subnet Ein Subnetz wird durch seine Adresse und Netzmaske charakterisiert. Teile der Subnetze werden an Institute und Abteilungen vergeben welche in der Entität Subnetpart definiert sind.

SubnetPart Es werden die Subnetz-Teilbereiche definiert, die von einzelnen Instituten oder Personen reserviert wurden. Da die Bereiche nicht dauerhaft vergeben werden müssen können Reservierungsbeginn und -ende eingetragen werden. Neben den organisatorischen Informationen werden auch Konfiguration bzgl. Adressumsetzung und Sicherheit festgehalten.

VLAN Jedes Subnetz liegt in einem oder mehreren VLANs. Ein VLAN wird von einer Menge von Personen betreut (PersonMap) und besitzt neben der VLAN-ID noch eine Klassifizierung der Sicherheitseinschränkungen (Nutzer, Management, Tunnel u.ä.)

Raum Innerhalb dieser Entität werden Raum und Racks erfasst. Jede Komponente befindet sich zunächst in einem Raum. Optional können hier die einzelnen Racks erfasst werden. Ein Raum befindet sich in einem Unterbezirk.

Unterbezirk In dieser Entität werden Bezirke und Unterbezirke erfasst. Ein Bezirk ist ein Standort, an dem sich ein oder mehrere vom LRZ betreute Organisationen, Institute und Gebäude befinden. Er wird durch einen einzigen Buchstaben gekennzeichnet (Beispielsweise „A“ für das Hochschulgelände Garching). Ein Unterbezirk ist eine logische Unterteilung des Bezirks. Dabei handelt es sich in der Regel um eine Aufteilung nach Gebäuden. Ein Unterbezirk wird durch den Buchstaben des Bezirks plus einem weiteren Buchstaben gekennzeichnet (z.B. „AH“ für das Chemiegebäude, Bauteil CH7 im Bezirk des Hochschulgeländes Garching).

URL Mit Hilfe der Entität „URL“ können zu den Bezirken, Unterbezirken und Räumen Dateien und Links hinterlegt werden. Für (Unter)bezirke wird im Allgemeinen ein Lageplan (unter anderem via Google Maps) hinterlegt. Für Räume und Racks wird oftmals auf die Zusammenhangs-Zeichnungen in Visio verwiesen, die auf dem zentralen, internen Netzlaufwerk des LRZ liegen.

Institut Ein Institut ist eine Untereinheit einer übergeordneten Organisation (z.B. „Technische Universität München“). Da jedes Institut Dienstleistungen des LRZ in Anspruch nehmen kann, kann man den Begriff des Instituts mit Kunde gleichsetzen. Jedes Institut besitzt ein eindeutiges Kennzeichnen, dass aus 4-5 Zeichen besteht.

Person Eine Person ist Mitglied eines Instituts. In der Regel werden in der Netzdoku ausschließlich Ansprechpartner für den Netzbetrieb (Administratoren und Netzverantwortliche) hinterlegt. Die Personen-Liste eines Instituts entspricht somit keiner Mitarbeiterliste.

PersonMap Mit dieser Hilfsentität ist es möglich mehrere Personen als Verantwortliche bzw. Ansprechpartner einzutragen. Die Reihenfolge der Liste kann individuell festgelegt werden.

Produkt Eine Komponente ist eine Instanz eines Produkts. Ein Produkt in der Netzdoku gibt die Bezeichnung für Ports oder Slots (vorwiegend bei Switches) vor, die bei der Entität „PhyPort“ berücksichtigt werden. Das Mapping zwischen den Namenskonventionen und der Bezeichnung bei der Entität „PhyPort“ findet ausschließlich auf logischer Ebene innerhalb der Applikation statt.

Produktklasse Mit Hilfe der Entität „Produktklasse“ werden die Produkte klassifiziert. Dies ermöglicht eine komfortablere Suche innerhalb des Tools.

Hersteller Jedes Produkt stammt von einem Hersteller.

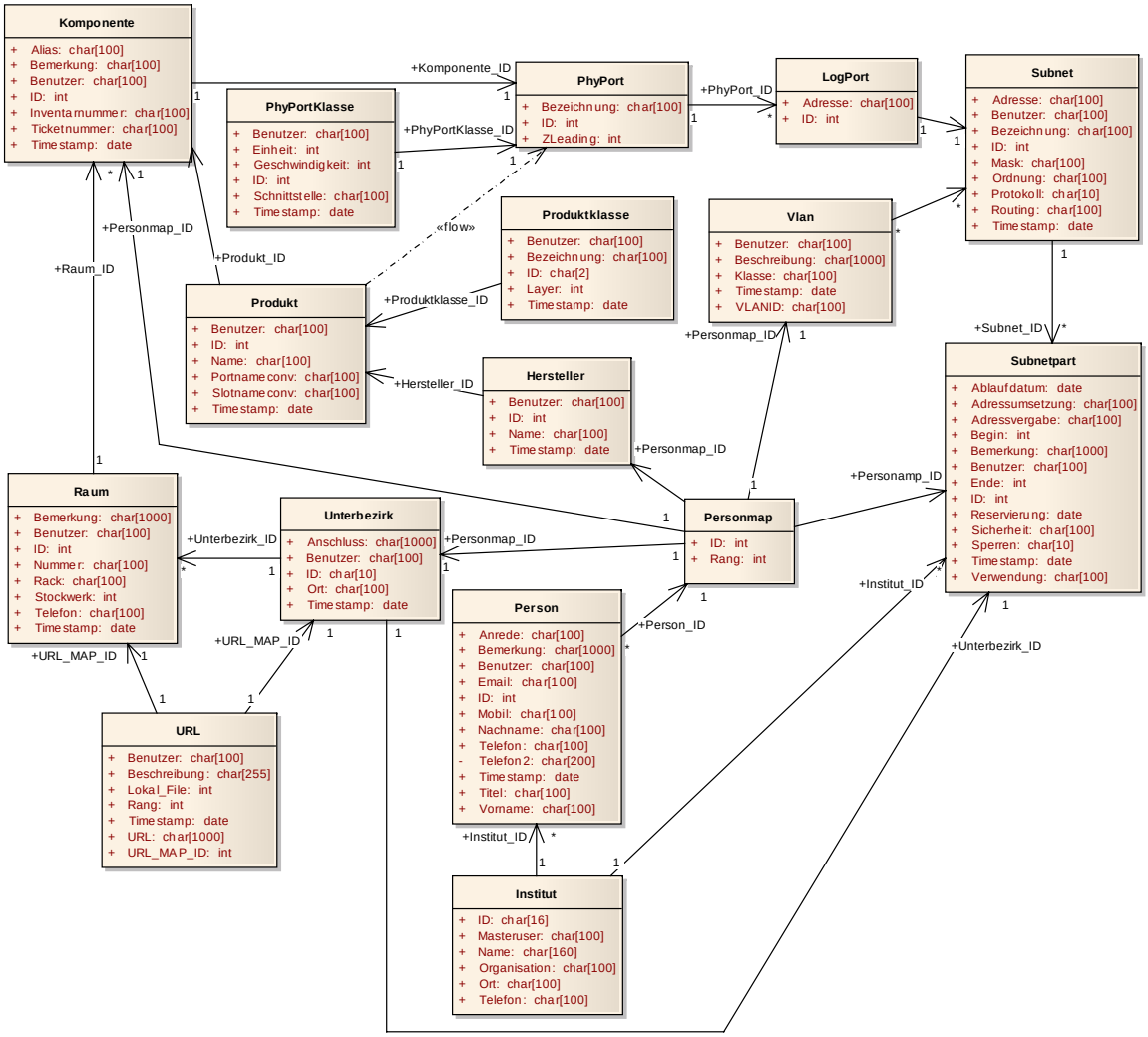


Abbildung 4.11.: Datenmodell Netzdoku

4.4.3. Technik & Schnittstellen

Als technische Plattform dienen zwei relationale Datenbanksysteme (Oracle und PostgreSQL). Die Daten werden vom Web- bzw. Applikationsserver Zope 2 für das User Interface aufbereitet. IST-Daten werden mittels Perl-Skript automatisch in die PostgreSQL-Datenbank geschrieben. Die weiteren Informationen befinden sich in der Oracle-Datenbank und werden manuell gepflegt. Das Datenbankschema weist allerdings einige Schwachstellen auf. So sind einige Datenbankfelder nicht normalisiert und Informatinen werden zum Teil mehrfach abgespeichert. Die Sicherstellung der referentiellen Integrität erfolgt ausschießlich auf Anwendungsebene.

Die „LRZ Netzdoku“ besitzt eine Schnittstelle zu „Remedy ARS“ (s. Seite 34), die den Benutzer bei der Neuanlage von Netzkomponenten unterstützt und - wenn möglich - die Daten aus den Einzelteil - und Geräte-Tickets übernimmt. Zudem verfügt das Tool über eine Anbindung an das zentrale Identity-Portal des LRZ (LRZ SIM).

4.4.4. Bewertung

Die „LRZ Netzdoku“ ist ein mächtiges Tool, dass primär von der Abteilung KOM für ihre tägliche Arbeit genutzt wird. Zusätzlich können die Netzverantwortlichen der Kunden auf eine reduzierte Sicht zugreifen. Dies erschwert eine Migration der Datenbasis zunächst, ist allerdings langfristig unumgänglich. Da das Datenmodell einige Schwachstellen aufweist, ist eine automatische Synchronisation mit dem ITSM-Tool nur schwer möglich. Der Migrationswert des Tools beträgt 54%, der Integrationswet nur 45% (siehe Tabelle 4.16). Obwohl die Differenz beider Werte nur sehr klein ist, geht die Tendenz Richtung Migration.

4.5. LRZmonitor

Mit zunehmender Zahl an Linux-Systemen im LRZ bestand die Notwendigkeit diese Systeme zentral über ein Interface zu managen. Existierten im Jahr 2000 nur ein Dutzend Systeme, sind es 2009 bereits 1550. Aus diesem Grund wurde im Jahr 2000 ein eigenes Tool, der LRZmonitor, entwickelt. Kernaufgabe des Tools ist die Sammlung von Konfigurationsparametern (Reporting) sowie das Software-Deployment. Die momentan eingesetzte und aktuelle Version ist 0.39.

4.5.1. Allgemeines

Die Administration des Tools erfolgt innerhalb der Abteilung HLS. Der Aufwand für die laufende Wartung wurde mit „niedrig“ angesetzt, da selten Änderungen am Tool direkt vorgenommen werden müssen. Primär müssen neue Linux-Rechner im Tool angelegt werden.

4.5.2. Funktionen

Ziel der Entwicklung war es von Anfang an, die Linux-Clients nicht durch zusätzliche Hintergrund-Agents zu belasten. Somit wurde das System agentenlos konzipiert. Ein Hauptskript („chk-all.sh“) wird einmal täglich per Cronjob auf dem Client aufgerufen und führt folgende Skripte und Funktionen aus:

1. Zunächst werden die Daten über den Client gesammelt und in einzelnen Dateien, getrennt nach Information, lokal abgelegt.

2. Anschließend werden eventuell vorhandene Zusatzskripts gestartet, die gewünschte Änderungen auf dem Client durchführen.

3. Als nächstes erfolgt ein RPM-Paket-Update. Clients werden damit mit neuer und aktualisierter Software versorgt.

4. Einzelne Konfigurationsdateien werden im nächsten Schritt mit einem zentral abgelegten Muster für jeden Client verglichen und gegebenfalls ersetzt.

5. Anschließen werden Dateisystemberechtigungen überprüft und korrigiert.

6. Das Skript prüft, ob ein Backup erfolgreich erstellt wurde und startet ggf. den Backup-Scheduling-Prozess.

7. Als nächstes wird die Auslastung des Dateisystem überprüft.

8. Im Falle einer virtuellen Linux-Maschine wird ein Zusatzskript angestoßen, dass einige VM-spezifische Überprüfungen (z.B. korrekt installierte VMware Tools) und Korrekturen vornimmt.

9. Für jeden oben genannten Schritt wurden Dateien erzeugt, die ein Resultat der Überprüfung bzw. Konfigurationsanpassung enthalten. Am Ende werden nun alle Dateien in ein Verzeichnis auf dem Server übertragen. Um Bandbreite zu schonen, wird vor dem Senden geprüft, ob sich die Datei geändert hat. Sollte dies nicht der Fall sein, wird die Datei nicht übertragen. Bei kritischen Fehlern (RPM-Installation fehlgeschlagen, Dateiberechtigungen falsch) werden zusätzlich die entsprechenden Administratoren per Mail benachrichtigt.

Mit der Überprüfung der Dateisystemberechtigungen und Konfigurationsdateien übernimmt der LRZmonitor zugleich die Funktion als Intrusion Detection System. Da im Falle eines Einbruchs zwangsläufig Konfigurationsdateien und Dateisystemberechtigungen geändert werden, kann dies mit Hilfe der täglich laufenden Check-Skripten entdeckt und gemeldet werden.

Abbildung 4.12.: Screen aus LRZmonitor

Erhobene Daten Erfasst werden sogenannte Harddata und Softdata. Unter Harddata versteht man Hostdaten, die zum großen Teil nicht durch das Betriebssystem ausgelesen werden können. Darunter fallen Serien- und Inventarnummer, Aufstellungsort, Modellbezeichnung, Kaufdatum, Dienstbeschreibung- und abhängigkeiten, Namen der System- und Dienstadministratoren mit jeweiliger Mailadresse, Bedienungsanleitung für Operateure und Wartungsarbeiten an Hard- und Software. Softdata sind Systemdaten, die in regelmäßigen Abständen von den Clients abgefragt werden. Die Abfrage geschieht derzeit alle 12 Stunden. Unter Softdata fallen unter Anderem CPU,RAM, Festplattenbelegung, Uptime, RPM-Paketauswahl. Abbildung 4.13 zeigt das Informationsmodell des Tools.

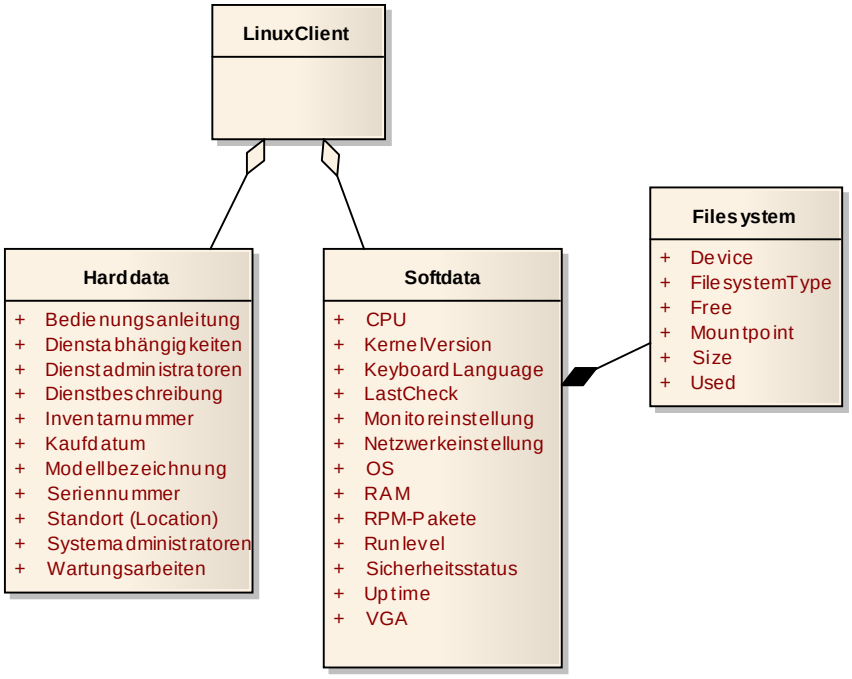


Abbildung 4.13.: Datenmodell des LRZmonitor 0.39

4.5.3. Technik & Schnittstellen

Die vielen Konfigurationsdateien der einzelnen Clients werden für einen Überblick grafisch aufgearbeitet. So verfügt der LRZmonitor über ein Webinterface - basierend auf HTML, PHP und CSS - um die entsprechenden Informationen anzuzeigen. Farbige Symbole kennzeichnen den Status der einzelnen Überprüfungen und verschaffen den Administratoren schnell einen Überblick, bei welchem Client Probleme aufgetreten sind bzw. welcher Client mit seinen regelmäßigen Statusmeldungen im Verzug ist.

Mit der kommenden Version 0.4 sollen die dateibasierten Reports durch eine MySQL-Datenbank abgelöst werden, was erheblich zur Performance-Steigerung des Web-Frontends beitragen wird. Hierfür gibt es bereits ein erstes Datenbankkonzept. Das Upgrade wurde allerdings im Hinblick auf die Einführung des ITSM-Tools verschoben, da der LRZmonitor in Zukunft Daten mit dem ITSM-Tool austauschen soll. Hierfür werden die Datenbankstruktur und die Schnittstellen des LRZmonitors an das ITSM-Tool angeglichen.

4.5.4. Bewertung

Nach Abschluss aller Betrachtungen erlangt der LRZmonitor eine Integrations-Wert von 57% und lediglich einen Migrations-Wert von 32% (siehe Bewertungstabelle 4.16). Somit liegt die Empfehlung klar bei der Integration in das ITSM-Tool. Dieses Ergebnis verwundert nicht, denn gerade die zusätzlichen Features des LRZmonitors, wie Intrusion Detection System und Softwareverteilungssystem für Linux, machen das Tool unersetzlich. Bisher gibt es noch keine geeignete Verbindung zwischen dem LRZmonitor und dem iET ITSM-Tool. Dies ist allerdings primär auf das noch auszuarbeitende Datenmodell zurückzuführen. Da das Tool am LRZ aktiv gepflegt und weiterentwickelt wird, sollte es kein Problem sein, auf Basis eines Datenmodells dieses Tool mit einer geeigneten Schnittstelle zu versehen, um Informationen zwischen CMDB und LRZmonitor zu synchronisieren.

4.6. VMware Infrastructure

Die „VMware Infrastructure“ ist eine Programmsuite bestehend aus dem ESX-Server und VMware „vCenter“. Derzeit wird Virtual Infrastructure 3.5 Enterprise eingesetzt. Im Augenblick laufen am LRZ auf zwei Cluster insgesamt ca. 330 virtuelle Maschinen. Demnächst wird die virtuelle Infrastruktur erweitert und ermöglicht mit der kommenden Ausbaustufe den Parallelbetrieb von ca. 3000 virtuelle Maschinen. Die Nutzung der Infrastruktur soll dann für Kunden des LRZ zur Verfügung stehen.

4.6.1. Allgemeines

Der gesamten virtuellen Infrastruktur kommt eine hohe Bedeutung zu, da mittlerweile viele Produktiv-Server virtualisiert wurden. Der Einsatz erfolgt dabei sowohl im gesamten LRZ, als auch bereits jetzt schon für einige Kunden. Der Aufwand für Wartung wird mit „hoch“ bewertet, da selbst kleine Updates vor dem Rollout sehr genau geprüft werden müssen. Zudem erfolgen Routineaufgaben bisher ausschließlich manuell. Für einige Komponenten (wie z.B. dem „Lifecycle Manager“) wurde kaum Dokumentationsmaterial zur Verfügung gestellt, was zu einer hohen Einarbeitungsphase führte.

4.6.2. Funktionen

Die Kernaufgabe des Tools ist es, den Betrieb virtueller Maschinen sicherzustellen. Innerhalb der virtuellen Infrastruktur gibt es folgende Komponenten:

Datacenter Ein Datacenter repräsentiert ein Rechenzentrum [RSW+09]. Es dient dabei primär der organisatorischen Abgrenzung. Innerhalb eines Datacenters existieren ein oder mehrere Cluster.

Cluster Ein Cluster enhält eine Anzahl von Hosts, die ihre Ressourcen dem Cluster zur Verfügung stellen. Innerhalb von Clustern ist es möglich, VMware HA (Hochverfügbarkeit) und VMware DRS (dynamische Ressourcenverteilung) zu nutzen. Jeder Cluster enthält mindestens einen „Resource Pool“.

Host Ein Host ist ein Computer, auf dem das Virtualisierungsbetriebssystem „VMware ESX“ läuft. Er stellt dem Cluster CPU- und Arbeitsspeicherressourcen zur Verfügung.

Resource Pool Ein „Resource Pool“ dient dazu, die Ressourcen des Clusters in mehrere Gruppen aufzusplitten. So kann einem „Resource Pool“ eine bestimmte maximale Arbeitsspeicher- und CPU-Nutzung vorgegeben werden. „Resource Pools“ können auch eine hierarchische Struktur aufweisen. Innerhalb dieser Struktur befinden sich die virtuellen Maschinen.

Virtuelle Maschine Genau wie physische Computer besteht eine virtuelle Maschine aus (virtuellen) Hardwarekomponenten (CPU, RAM u.ä.). Jede virtuelle Maschine besitzt mindesten eine virtuelle Netzwerkkarte und eine Festplatte.

Virtuelle Netzwerkkarte Eine virtuelle Netzwerkkarte verhält sich im Wesentlichen genauso wie eine physische Netzwerkkarte. Sie besitzt eine MAC-Adresse und in der Regel genau eine IP-Adresse. Jede virtuelle Netzwerkkarte ist mit einem virtuellen Switch (vSwitch) verbunden.

vSwitch An einen vSwitch werden sowohl virtuelle als auch physische Netzwerkkarten gebunden. Ein vSwitch besitzt eine begrenzte Menge an Ports. Auf einem Host können maximal 127 vSwitches eingerichtet werden, die jeweils bis zu 1016 Ports zur Verfügung stellen können. Zur logischen Trennung innerhalb eines vSwitches können Portgruppen (Portgroups) definiert werden.

Portgroup Eine Portgroup fasst einzelne Ports zu Gruppen zusammen. Alle Anschlüsse einer Portgroup besitzen dieselben Eigenschaften (z.B. die gleiche VLAN-ID). Portgroups sind mit portbasierten VLANs in physischen Switches vergleichbar.

Virtuelle Festplatte Eine virtuelle Festplatte besteht aus einer großen Datei, die zusammen mit der virtuellen Maschine im Datastore abgelegt wird [RSW+09].

Datastore Ein Datastore ist ein Festplattenspeicher, auf dem die virtuellen Maschinen gespeichert werden. Bei Nutzung eines Clusters mit Live-Migration-Funktionalität ist ein gemeinsamer Zugriff von mehreren Hosts auf den Datenspeicher notwendig, was die Verwendung eines SANs o.ä. erfordert.

Im Rahmen eines Arbeitskreises wurde hierzu von der Abteilung HLS/COS ein Informationsmodell zum Betrieb virtueller Maschinen angefertigt (Abbildung 4.14). In diesem Modell wird die Komponente „Datacenter“ nicht erwähnt, da es am LRZ derzeit nur ein Datacenter gibt.

4.6.3. Technik & Schnittstellen

Die Konfigurationsinformationen werden in einer SQL-Datenbank gespeichert und vom VMware vCenter Server genutzt. Um die virtuelle Infrastruktur zu managen wird der „VMware Infrastructure Client“ verwendet. Zur Authentifizierung der Benutzer greift das vCenter auf das „Active Directory“ einer Windows-Domäne zurück. Zur Entwicklung eigener Applikationen und Tools stellt VMware ein Software Development Kit (SDK) zur Verfügung. Mit Hilfe von Webservices (SOAP/WSDL) können Befehle an das vCenter abgesetzt und Informationen ausgelesen werden.

Um Kunden die Bestellung neuer und Verwaltung bestehender virtueller Maschinen zu ermöglichen, wurde der „VMware Lifecycle Manager“ gekauft. Dabei handelt es sich um ein

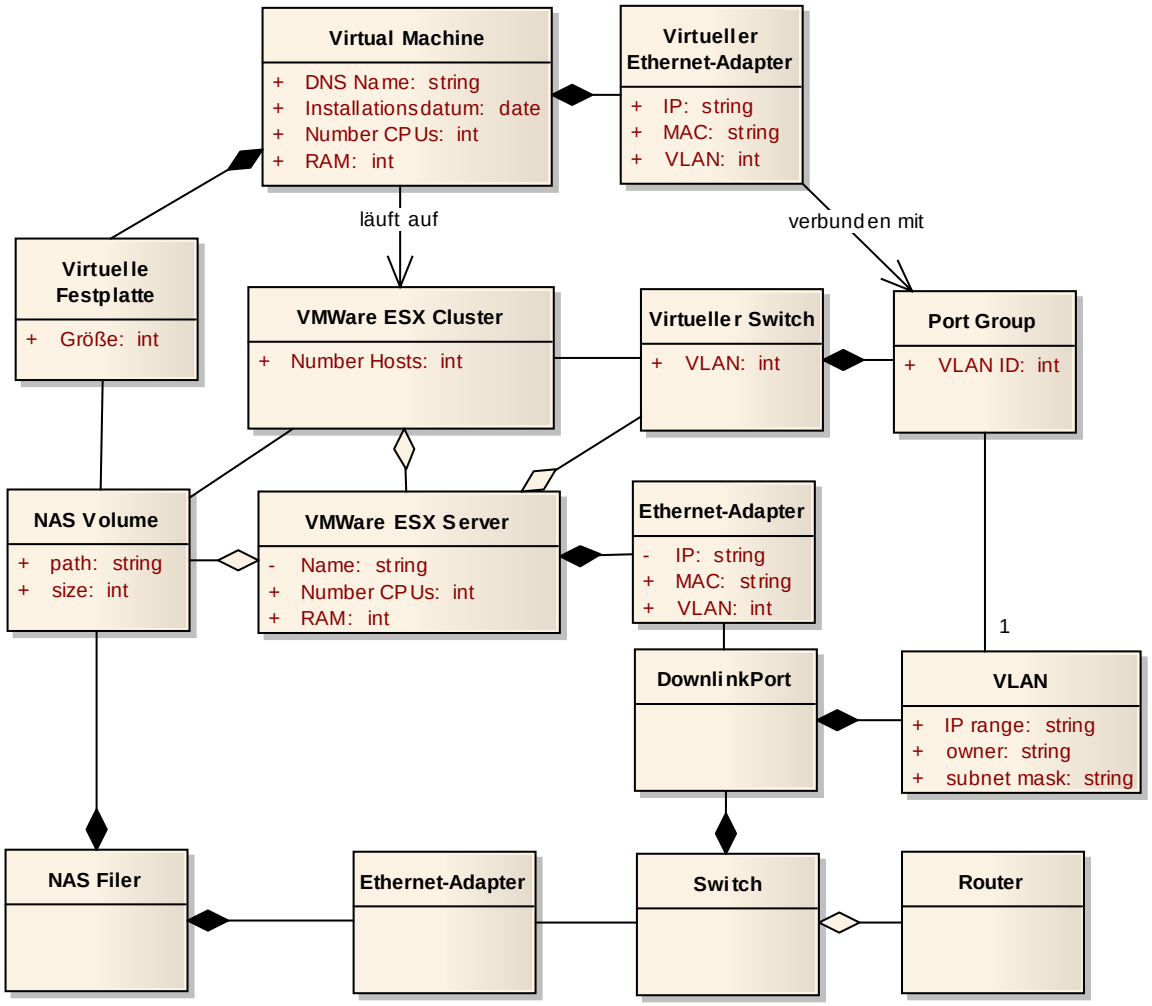


Abbildung 4.14.: Informationsmodell der virtuellen Infrastruktur erstellt von Berner [Ber09]

anpassbares Webinterface, dass auf das Workflow-System („VMware Orchestrator“) zum automatisierten Lebenszyklus-Management zurückgreift.

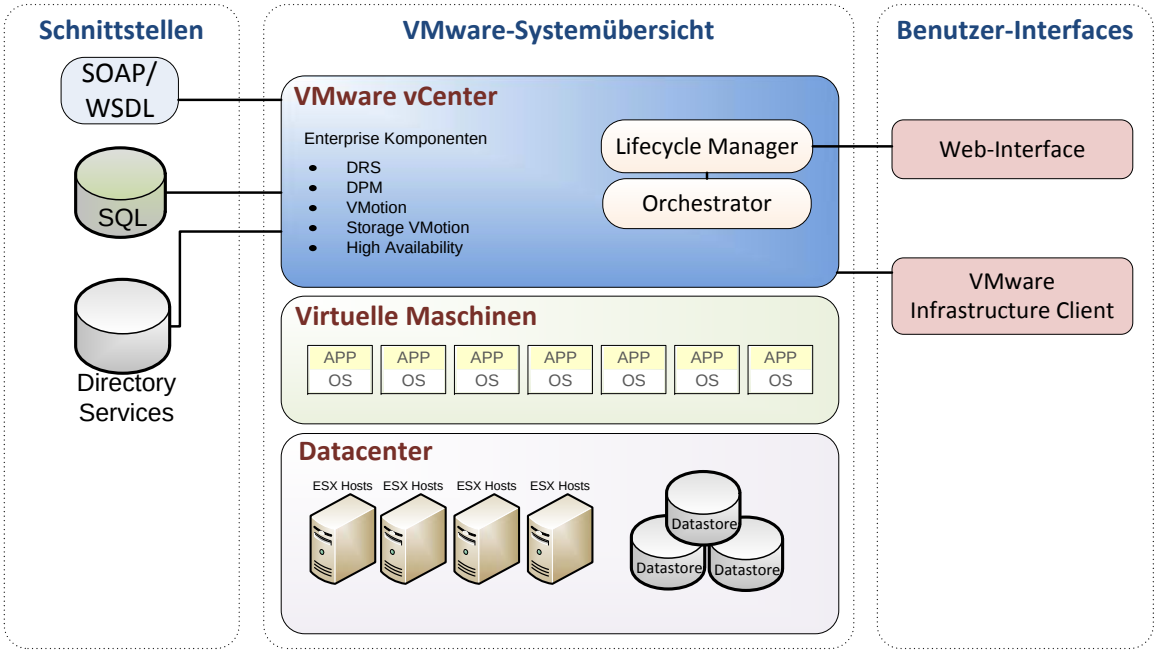


Abbildung 4.15.: Überblick über das VMware System mit Schnittstellen

4.6.4. Bewertung

Wie auch die Bewertungstabelle (Abbildung 4.16) zeigt, ist der einzig sinnvolle Weg die Integration der Informationen zwischen CMDB und vCenter-Datenbank. Durch die demnächst rasch steigende Anzahl an zu verwaltenden virtuellen Maschinen ist ein sauber definierter Prozessablauf notwendig, der mit dem zukünftigen ITSM-Tool umgesetzt werden kann. Aus diesem Grund wurde der Betrieb und das Management der virtuellen Infrastruktur auch vom Arbeitskreis ITSM des LRZ als erstes Szenario zur Implementierung ausgewählt.

4.7. Zusammenfassung

Auf Basis der in Abschnitt 4.1 erstellten Bewertungsrichtlinien wurden Datenquellen analysiert und bewertet, die bei als erstes zum Aufbau der CMDB herangezogen werden. Abbildung 4.16 enthält die Bewertung in tabellarischer Form. Dabei zeigt sich, das einige Tools und Datenquellen, wie z.B. die Switch-Dokumentation oder Remedy ARS, keinen wirklichen Mehrwert zu einer CMDB bieten und zukünftig nicht mehr weiter eingesetzt werden sollten. Bei Tools, deren Funktionalität erheblich größer ist und somit weiter bestehen sollen, muss ein Integrationskonzept erstellt werden, dass die Synchronisation von relevanten Konfigurationsinformationen mit der CMDB sicherstellt. Dazu muss zunächst ein übergreifendes Informations- und Datenmodell erzeugt werden, mit dessen Hilfe eine Toolintegration geplant werden kann. Dies geschieht im Folgenden Kapitel.

Integrations- / Migrations- Scoreboard	LRZ Netzdoku	LRZ LRZmonitor 0.39	Remedy Action Request System 6.3	LRZ Switch-Dokumentation	VMware Infrastructure 3.5 Enterprise
ALLGEMEINES (25%)					
Nutzungsart	unternehmensweit	abteilungsintern	unternehmensweit	unternehmensweit	unternehmensweit
Hersteller-Support	vorhanden	vorhanden	nicht vorhanden	vorhanden	vorhanden
Bedeutung f.d. Unternehmen	hoch	mittel	mittel	mittel	hoch
Aufwand für Wartung	niedrig	niedrig	hoch	niedrig	hoch
FUNKTIONEN (25%)					
Zusätzliche Funktionen	nein	ja *Monitoring, IDS, Software-Deployment*	nein	nein	ja *VMware steuern*
TECHNIK & SCHNITTSTELLEN (50%)					
Speicherung in Datenbank	ja	nein	ja	nein	ja
Datenbank-Normalisierung	nein	n.a.	ja	n.a.	ja
Auto. Daten-Export-Schnittstelle	nein	nein	nein	nein	ja
falls ja, welche					*SOAP*
Auto. Daten-Import-Schnittstelle	nein	nein	nein	nein	ja
falls ja, welche					*SOAP*
Auto. eindeutige Ident. von CIs möglich	ja	ja	ja	nein	ja
Verbindung zu anderen Systemen	ja	nein	ja	nein	ja
falls ja, welche	*Remedy ARS, IST-Import, LRZ SIM*		*LRZ Netzdoku*		*Active Directory*
ERGEBNIS					
Migrations-Score (%)	54	32	68	44	52
Integrations-Score (%)	45	57	47	20	100
Tendenz zu	**Migration**	**Integration**	**Migration**	**Migration**	**Integration**

Abbildung 4.16.: Scoreboard

5. Integrations- und Migrationskonzept

Im folgenden Kapitel wird das Integrations- und Migrationskonzept für das Szenario am LRZ erarbeitet. Als Basis dienen dazu, die in Kapitel 4 gewonnen Informationen über die MDRs, die am Verfahren für die Provisionierung von VM beteiligt sind. Das erstellte Konzept wird dabei auf das vom LRZ beschaffte Tool „iET ITSM“ ausgerichtet.

5.1. Erstellung des Informationsmodells

Zunächst wird ein - von „iET ITSM“ unabhängiges - Informationsmodell erzeugt. Die notwendigen Daten zur Erstellung des Modells liefert im Wesentlichen Kapitel 4, in dem die einzelnen MDRs mit ihren jeweiligen Datenmodellen analysiert und bewertet wurden.

Schon bei der Erstellung der einzelnen Informationsmodelle der MDRs wurde deutlich, das es nicht möglich ist, alle Modelle in einem einzigen Schritt zu einem großen Gesamtmodell zusammenzufassen. Das liegt daran, dass die einzelnen Informationsmodelle unterschiedliche Sichten auf Prozesse besitzen, aber auch dass einige Modellierungen nicht mehr zeitgemäß sind und überarbeitet werden müssen.

Somit wird mit einem leeren Modell gestartet, dass nun schrittweise erweitert wird.

5.1.1. Phase 1: Stammdaten

Bei der Erstellung des Informationsmodells wird mit den Stammdaten begonnen. Zu den Stammdaten zählen dynamische Bewegungsdaten und statische Grunddaten. Unter dynamischen Bewegungsdaten versteht man Bestellungen und Lieferungen. Zu den statischen Grunddaten zählen Mitarbeiter, Kunden und Lieferanten [Win08]. Die Stammdaten sind somit für die weitere Erfassung und korrekte Verarbeitung unerlässlich.

Unter der Entität „Kunde“ fallen alle Institute und Lehrstühle für die das LRZ Dienstleistungen erbringt und die bereits über eine Institutions-Kennung als eindeutigen Schlüssel verfügen. Die einzelnen Kunden gehören dabei Organisationen (z.B. „Technische Universität München“) an. Jeder Kunde verfügt über mindestens einen Standort, der als eigene Entität modelliert wird und für das LRZ von Bedeutung ist. Dieser Standort befindet sich in einem Gebäude, das wiederum in einem Bezirk liegt. Die Bezirke geben vor, wer primärer Arealbetreuer ist. Diese feingranulare Aufteilung ist gerade für die Abteilung Netzwartung notwendig. So gibt es für jeden Standort und jedes Gebäude detaillierte Netzpläne. Auch das LRZ mit seinen Mitarbeitern wird der Entität „Kunde“ zugeordnet. Grund dafür ist, dass Mitarbeiter des LRZ Dienstleistungen in Anspruch nehmen können (z.B. eine virtuelle Maschine) und sonst doppelt (in einer Entität „Kunde“ und „Personal“) auftauchen würden. Mitarbeiter eines Kunden werden in der Entität „Kontakt“ festgehalten. Kontakte können zusätzlich die Rolle Master-User für einen Kunden einnehmen. Pro Kunde kann es mehrere Ansprechpartner geben, die vom LRZ bei Problemen und Anfragen kontaktiert werden. Allerdings müssen diese nicht zwingend Mitarbeiter des Kunden sein.

Für das Lieferantenmanagement wird eine Entität „Lieferanten“ angelegt, die alle betriebsrelevanten Informationen aufnimmt.

Für die dynamischen Bewegungsdaten (Bestellungen und Lieferungen) wurde bisher „Remedy ARS“ als Datenbank genutzt. Zur Dokumentation des Bestellprozesses wird das Stabeg-Formular (Statusblatt für bestellte bzw. beschaffte Geräte) verwendet, dass auch mit der ITSM-Einführung in leicht veränderter Form erhalten bleibt. In „Remedy ARS“ gibt es die Hierarchie von Einzel- und Gerätetickets. Diese wurde nicht in dieser Form in das Datenmodell übernommen, da sie nicht allgemeingültig ist. So funktioniert sie nicht bei Hardware, die aus einem Teil besteht (z.B. Laptop). Bei Switches hingegen ist die Aufteilung sinnvoll. Da aus Einzelteilen bzw. Komponenten zusammengesetzte Hardware ebenfalls wieder vom Typ „Hardware“ ist, ist an dieser Stelle keine weitere Untergliederung notwendig.

Damit sind alle relevanten Stammdaten modelliert.

5.1.2. Phase 2: Kerndienstleistung Netzinfrastruktur

Kerngeschäft des LRZ ist die Bereitstellung der Netzinfrastruktur. Daher findet hier die erste Erweiterung des Datenmodells statt.

Die Betrachtung beginnt zunächst mit Switches. Jeder Switch besteht in der Regel aus mehreren Modulen, die als Komponenten in das Gerät eingebaut werden. Sowohl Switch als auch Switchmodul erben Informationen aus der abstrakten Klasse „Hardware“ und benötigen somit keine weiteren Parameter. Ein Switch-Modul stellt eine Anzahl an physischen Ports zur Verfügung. Die Ports tragen eine Bezeichnung, die den einzelnen Port eindeutig identifiziert. Auf der anderen Seite kann ein physischer Port auch von einer Netzwerkkarte stammen, die wiederum eine Komponente eines Servers ist. Das eindeutige Kennzeichen einer Netzwerkkarte ist die MAC-Adresse. Sie wird nicht als Spezialisierung der Hardware modelliert, da eine Netzwerkkarte in der Regel keine eigene Inventarnummer besitzt und somit buchhalterisch nicht extra erfasst wird. Bei Standortswitches werden zusätzlich Dosen- und Panelbeschriftung, wie bereits im Kapitel 4.3 beschrieben, festgehalten.

Auf OSI-Schicht 2 können dem Port VLANs zugeordnet werden, die entweder portbasiert oder getaggt sind. Letzteres wird in einem Attribut vermerkt. Da hinter einem VLAN eine organisatorische Gliederung steckt, wurde dies zu einer eigenen „Entität“. So gibt es für jedes VLAN eine kurze Beschreibung, eine Klassifizierung (die der LRZ Netzdoku entnommen ist) und einen Kontakt, der als Ansprechpartner fungiert.

Auf OSI-Schicht 3 werden IP-Adressen und damit auch die DNS-Namensauflösung definiert. Da jeder physische Port eine beliebige Anzahl an logischen Ports zur Verfügung stellen und somit auch beliebig viele IP-Adressen und DNS-Namen nutzen kann, werden IP- und DNS-Konfiguration in die Entität „LogischerPort“ ausgelagert. Gerade größere Server verbinden mehrere Ports zu einem Einzelnen, um mehr Bandbreite zu nutzen (bonding). Es können aber auch mit nur einer Netzwerkkarte und einem virtuellen Interfaces mehrere Subnetze bedient werden. Jedes Subnetz dient einem gewissen Zweck und wird von einem Kunden oder einer LRZ-Abteilung, die im folgenden Datenmodell ebenfalls wieder als Kunde modelliert wird, genutzt. Ein Subnetz definiert sich eindeutig durch seine Subnetzadresse und die Subnetzmaske. Hier wurde das bisherige Schema der LRZ Netzdoku abgeändert, da dieses Subnetzinformationen mehrfach gespeichert hatte. Die weiteren technischen Informationen wie z.B. Adressumsetzung und Sicherheit werden übernommen.

Neben Switches gibt es noch Accesspoints als Netzinfrastrukturkomponenten. Accesspoints erben ihre Stammdaten-Informationen aus der Entität „Hardware“. Als zusätzliche Attribute

werden die Funkkanäle, auf denen sie senden, festgehalten. Außerdem kann jeder Accesspoint ein oder mehrere SSIDs ausstrahlen. SSIDs werden zur Strukturierung in einer eigenen Entität festgehalten.

5.1.3. Phase 3: Physische Server

Physische Server sind ebenfalls vom Typ „Hardware“. Zusätzlich werden Ausstattungs- und Konfigurationsmerkmale erfasst. Dazu zählen CPU, Arbeitsspeicher und Grafikkartentyp. Jeder Server wird von einem Kontakt im LRZ betreut. Ein Server besteht zudem aus mindestens einer Festplatte, die in der Entität „HDD“ modelliert wird, da einige Zusatzinformationen erfasst werden. So wird unter anderem festgehalten, wie groß die Festplatte ist, wo sie am Server angebunden ist (Device), welches Dateisystem darauf läuft und ob ein Backup dieser Festplatte gemacht wird. Für Linuxserver, die mit dem LRZmonitor verwaltet werden, gibt es zum Teil hinterlegte Anleitungen. Diese werden über die Entität „Dokumentation“ an den Server angebunden.

Das Leibniz-Rechenzentrum wird als Darkcenter betrieben, d.h. das Personal sitzt außerhalb des Rechnergebäudes. Daher sind alle Server an schaltbare Steckdosen angeschlossen (Power Distribution Units), die via SNMP gesteuert werden. Diese Konfigurationsinformation wird ebenfalls für jeden Server erfasst.

5.1.4. Phase 4: Virtuelle Infrastruktur

Eine virtuelle Infrastruktur besteht aus einer Anzahl von Servern, die einen Cluster bilden, auf denen die einzelnen virtuellen Maschinen laufen. Da bereits die Entität „Server“ modelliert wurde und diese ausreichend spezifiziert ist, ist es nicht nötig, eine separate Entität für Clusterknoten anzulegen. Die Server können somit Knoten in einem Clusterverbund sein. Ein Cluster wird durch seinen Namen eindeutig gekennzeichnet.

Virtuelle Maschinen werden ebenfalls durch einen Namen eindeutig identifiziert. Sie haben - ähnlich zu realen Servern - Ausstattungsmerkmale wie z.B.die Anzahl der CPUs und Größe des Arbeitsspeichers. Jede Maschine wird durch einen Systemadministrator betreut und von einem Kunden genutzt. Jeder virtuelle Rechner besitzt zudem ein oder mehrere Festplatten bzw. Netzwerkkarten. Festplattenkonfigurationsinformationen werden wie bei realen Servern dokumentiert. Netzwerkkarten werden durch ihre MAC-Adresse eindeutig identifiziert und sind an einen virtuellen Switch angeschlossen. Dieser hat wiederum in der Regel genau ein VLAN geschaltet. Virtuelle Netzwerkkarten stellen logische Ports bereit, die denselben Aufbau wie in Phase 2 beschrieben haben.

5.1.5. Phase 5: Betriebssystem, Dienste und Dienstabhängigkeiten

Als letzten Punkt der Modellierung werden Betriebssystem und Dienste berücksichtigt.

Auf Servern und virtuellen Maschinen läuft ein Betriebssystem. Da zum Betriebssystem einige detaillierte Informationen (Version und Patchlevel) gespeichert werden, wird dies in einer eigenen Entität modelliert.

Dienste sind ein zentraler Punkt für das LRZ. Da ausschließlich Netzinfrastruktur und Basisdienste (wie z.B. Webserver) bereitgestellt werden, werden Service-Level Agreements (SLA) auf Basis von Diensten und nicht einzelner Hardware angelegt. Zudem muss es eine Möglichkeit geben, herauszufinden, welche Komponenten für die Erbringung eines Dienstes

benötigt werden. Die Abhängigkeiten sollen dabei auch in einer grafischen Übersicht dargestellt werden. Dies ermöglicht dem Change- und Incident-Management eine schnelle Auswirkungsanalyse.

Da dies nicht nur am LRZ gängige Praxis ist, wurde in ITIL v3 der Service Catalogue Management-Prozess eingeführt [OGC07]. Dieser baut das in ITIL v2 bekannte Servicekatalog-Konzept weiter aus und führt unter anderem eine klare Trennung zwischen Business Services und Infrastruktur-Services ein. Business Services sind Dienste, die nach außen hin wahrnehmbar sind und die ein Kunde beziehen kann. Diese werden durch SLAs definiert. Auf der anderen Seite gibt es Infrastruktur-Services, die ausschließlich intern sichtbar sind und die nötigen Voraussetzungen zum Betrieb der Business Services schaffen. Sie werden durch UCs oder OLAs definiert [BT09][Kem09].

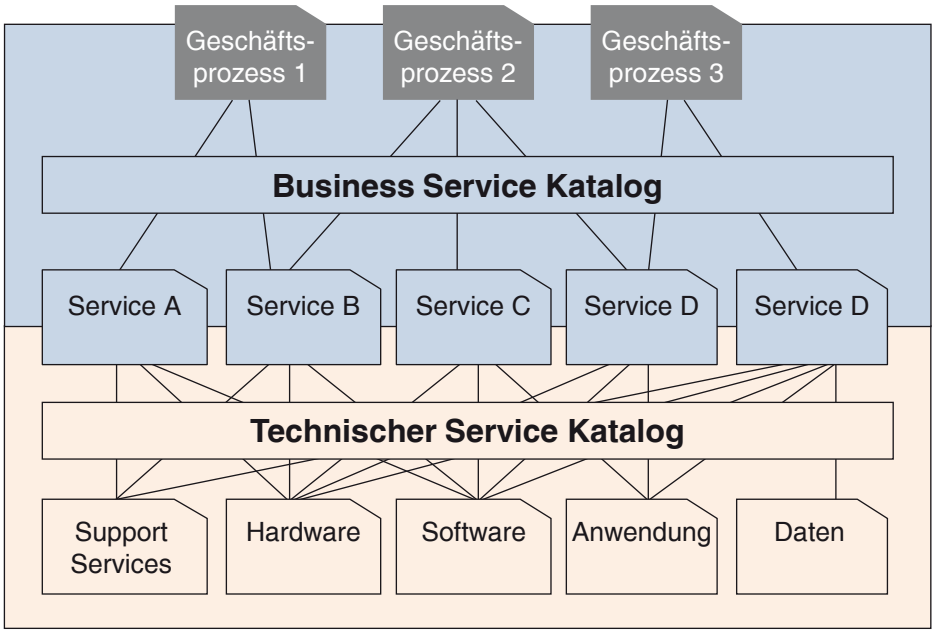


Abbildung 5.1.: Business- und Infrastruktur-Services Modellierung nach [BT09]

Ein eigens dafür eingerichteter Arbeitskreis („AK-Continuity“) beschäftigt sich am LRZ mit dem Thema Abhängigkeiten zwischen Diensten. So wurde ein Plan entwickelt, um nach einem vollständigen Blackout ein kontrolliertes Hochfahren des Rechenzentrums zu ermöglichen. Die Dienste bzw. die entsprechenden Server wurden dazu in Anschaltstufen unterteilt. Zudem wurden Dienstabhängigkeiten aufgezeichnet.

Die Ergebnisse des Arbeitskreises fließen durch die Modellierung der Entität „Dienst“ in das Datenmodell ein.

5.1.6. Finales Informationsmodell

Am Ende der Modellierung ist ein umfangreiches Modell entstanden, dass die wichtigsten Punkte am LRZ darstellt. Wegen der hohen Detailtiefe ist es allerdings nicht mehr möglich, alle Verbindungen überschneidungsfrei einzuzeichnen und übersichtlich darzustellen. Aus diesem Grund wurde darauf verzichtet, bei 1:1 Relationen, wie sie in der Regel bei Kontakten auftreten, eine Verbindung einzuzeichnen. Stattdessen wird lediglich ein Attribut mit dem

Typ der Zielklasse erstellt. Farbige Kennzeichnungen sollen helfen, den Fokus der einzelnen Entitäten besser zu unterscheiden. So werden Stammdaten in grün, Netzkonfigurationen in gelb, virtuelle Infrastruktur in blau und Betriebssystem und Dienste in rot dargestellt. Physische Server und deren zugehörige Komponenten, Dokumentation werden ohne Farbmarkierung dargestellt. Das Informationsmodell findet sich in Abbildung 5.2.

5.2. Toolüberblick: iETSolutions ITSM 5.0

Am LRZ wird in naher Zukunft das ITSM-Tool von iETSolutions eingesetzt. Somit müssen das Informationsmodell und die Anforderungen des LRZ damit umgesetzt werden. Daher wird im folgenden ein kurzer technischer Überblick über das Tool gegeben sowie die CMDB vorgestellt.

5.2.1. Technische Sicht

Als Datenbank dient ein Microsoft SQL Server oder ein Oracle Datenbankserver. Die Basis der iET Software ist die sogenannte „iET Technology“. Dabei handelt es sich um ein Bundle von Komponenten, die zusammen den javabasierten Anwendungsserver bereitstellen. Außerdem wird mit der „iET Technology“ der Client und ein Entwicklungstool mitgeliefert. Bereits mit dieser Grundausrüstung ist es möglich, eigene Applikationen zu entwickeln und für Enduser bereitzustellen. Für IT-Servicemanagement hat iET eigene Applikationen (wie z.B. „Configuration Management“ oder „Service Desk Management“) entwickelt, die als „iET ITSM“ bezeichnet und als Addon zur Technology installiert werden.

iET Technology

Die „iET Technology“ besteht aus mehreren zum Teil voneinander abhängigen oder nebenläufigen Komponenten. Die Komponente „axnet“ stellt die Verbindung zur Datenbank her. Darüber folgt der eigentliche Enterprise Anwendungsserver. Dieser besteht aus dem Hauptprozess „iET Enterprise Service Host“ sowie insgesamt fünf Agenten, die für Benachrichtigungen, Berichte, Eskalationen, Textsuchen und Datenimports (Virtual Incident Processing, VIP) zuständig sind. Clientseitig befindet sich darüber ein Gateway („iET aagtwy“), dass Anfragen entgegennimmt und Ergebnisse cached. Des Weiteren gibt es noch einen „iET Developer Studio Server“, der für die Entwicklung mit dem Programm „iET Developer Studio“ genutzt wird. Die „iET Technology“ enthält zudem noch einige Schnittstellen. So gibt es eine API für die Entwicklung eigener Klassen in Java, .NET oder C++. Mit Hilfe von Crystal Reports ist es möglich, eigene Berichte zu erstellen bzw. vorhandene anzupassen und mit Hilfe des Developer Studios dem Applikationsserver bereitzustellen. Außerdem verfügt die Technology über zahlreiche Kommunikationsschnittstellen zu E-Mail-Servern (Microsoft Exchange), SMS/Pager-Anbindung, Faxsoftware und Computer Telephony Integration (CTI). Abbildung 5.3 gibt einen grafischen Überblick über die Komponenten.

CMDB Intelligence

Die Komponente „CMDB Intelligence“ ermöglicht es, Daten aus Monitoring-Tools oder anderen Quellen in die CMDB zu importieren. Es handelt sich dabei um eine Staging Area, wie sie in Kapitel 2.4 bereits beschrieben wurde. Neuankommende Daten werden zunächst in den

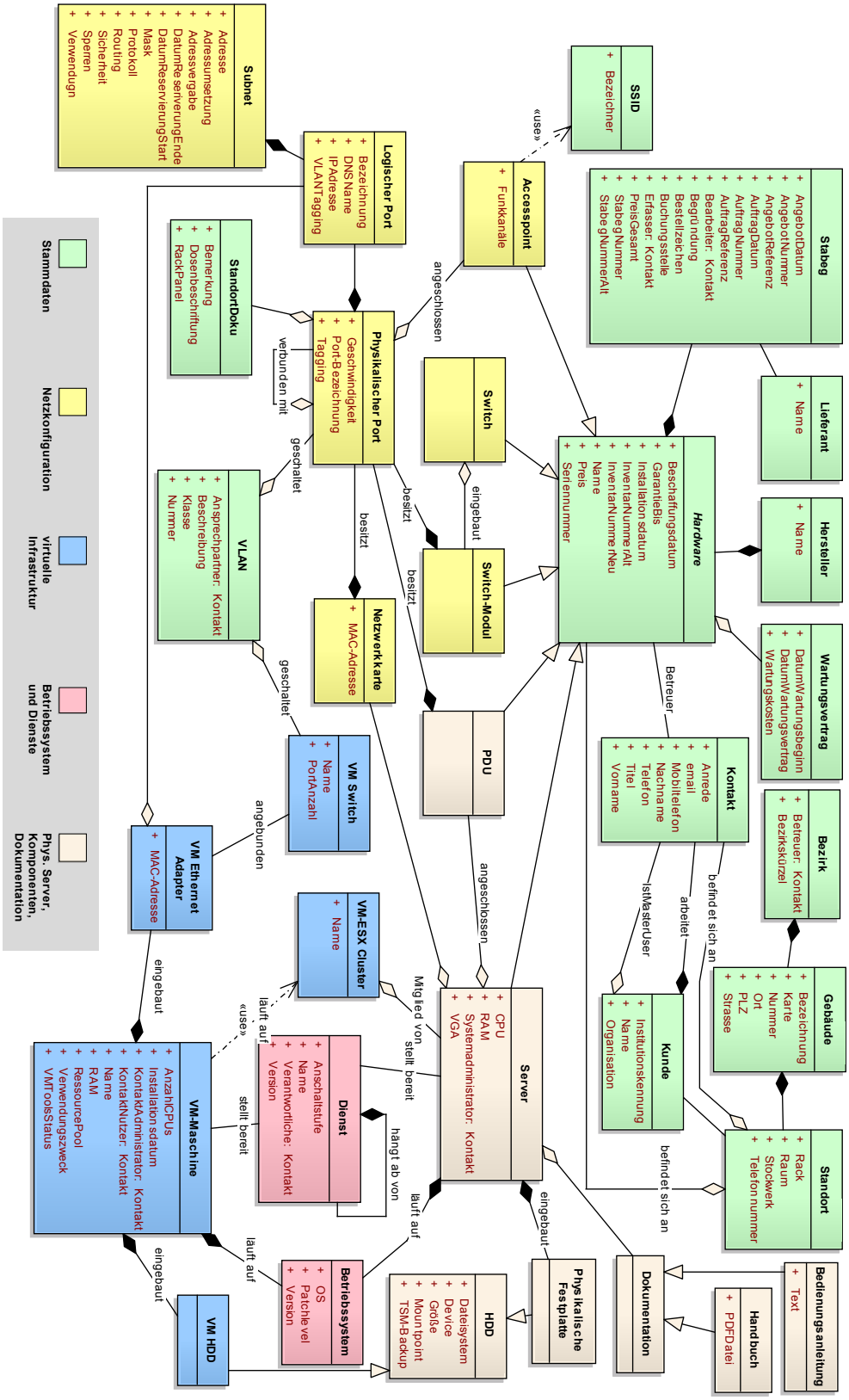


Abbildung 5.2.: Informationsmodell des Szenarios am LRZ

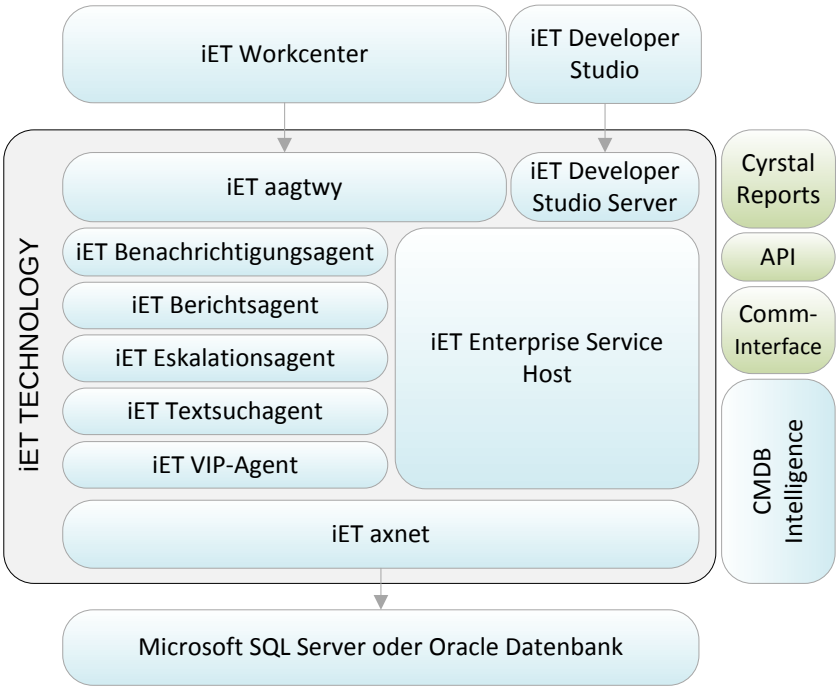


Abbildung 5.3.: Technischer Aufbau des iET ITSM-Tools

sogenannten Inventory-Bereich, einer Datenbank für IST-Daten, geladen und müssen dort einem CI zugeordnet werden. Die Zuordnung kann auch automatisch erfolgen, sofern eine Zuordnungsdefinition erstellt wurde. Als Import-Mechanismus erlaubt „CMDB Intelligence“ XML-Dateien, CSV-Dateien oder ODBC-Verbindungen.

Workcenter

Das Workcenter ist die Benutzeroberfläche für Mitarbeiter. Es handelt sich dabei um einen auf .NET 3.5 basierten Client. Die Software nimmt, wie in Abbildung 5.3 zu sehen ist, ausschließlich eine Verbindung zum „iET aagtwy“ auf, dass sich um die Kommunikation mit dem Applikationsserver und der Datenbank kümmert.

Im Workcenter befindet sich auf der linken oberen Seite die Auswahlmöglichkeit der sogenannten „Applikation“. Darunter befinden sich die Elemente (Formulare, Abfragen, Berichte, Dashboards), die es innerhalb der Applikation gibt. Auf der rechten Seite befindet sich das geöffnete Formular oder ein Abfragedialog. Abbildung 5.4 zeigt einen Screenshot der Benutzeroberfläche. Zu sehen ist die sogenannte „ITSM-Übersicht“ innerhalb der Applikation „Service Desk Management“.

Self Service

Das Self-Service-Portal ist eine webbasierte Applikation (siehe Abbildung 5.5) mit der, Endkunden Incidents, Changes und Service Requests stellen sowie den laufenden Stand einsehen können. Der Benutzer muss sich hierfür mit seiner im System hinterlegten E-Mailadresse und seinem Passwort einloggen.

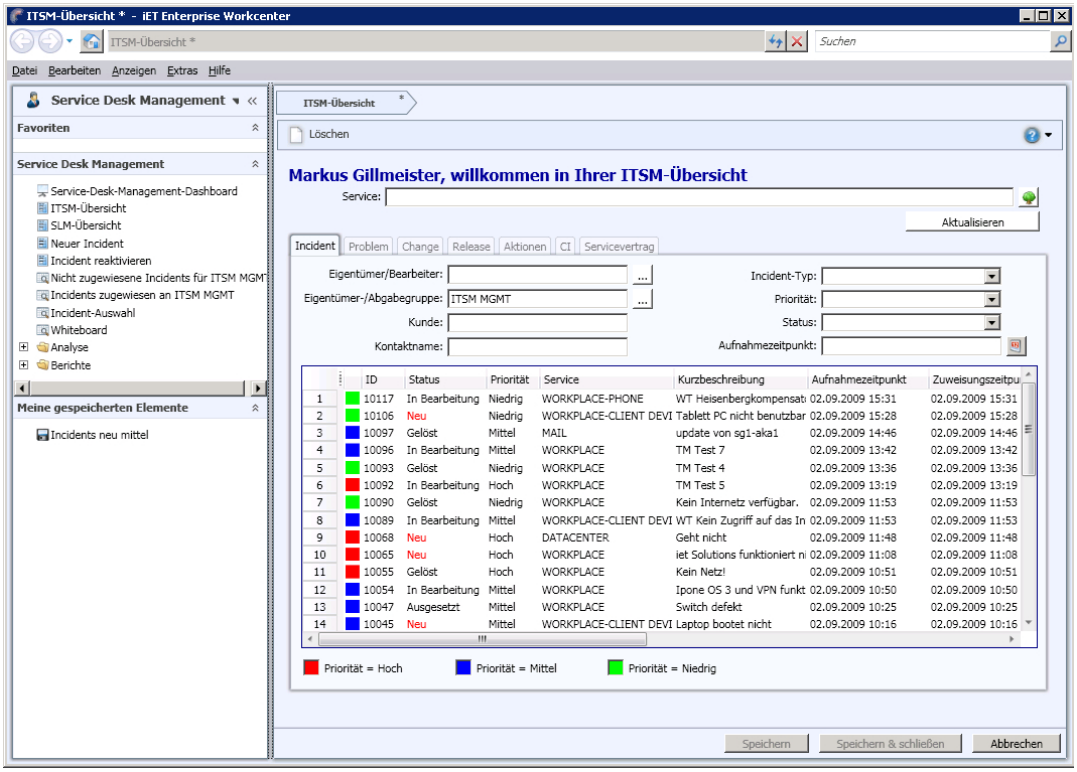


Abbildung 5.4.: iET Workcenter Oberfläche

Das Selfservice-Portal wurde in ASP geschrieben und läuft somit auf einem ASP-fähigen Webserver wie dem Microsoft Internet Information Server (IIS). Die ASP-Skripte des Self-Services laden Formulare und Abfragen, die mit Hilfe des Developer Studios erstellt wurden. Somit ist es auch möglich eigene, neue Formulare für bestimmte Aufgaben zu entwickeln.

Developer Studio

Das Developer Studio ist eine Java-Applikation, die mit Hilfe des Developer Studio Server Hosts eine Verbindung zum iET-Applikationsserver und der Datenbank aufbaut. Das Tool ermöglicht einen kompletten Zugriff auf das Datenmodell der „iET Technology“ und damit auch „iET ITSM“. Mit Hilfe des Tools können alle bestehenden Formulare, Abfragen, Elemente wie Dashboards, grafische Explorer, Datagrids, Datasheets geändert bzw. neue Elemente angelegt werden. Zudem können hier Formulare und Datenfelder mit Java-Klassen verknüpft werden, deren Ausführung vom Workcenter getriggert wird. Auf diese Weise lassen sich beliebige Erweiterungen schreiben. Kapitel 6.3 geht auf die möglichen Java-Klassen-Erweiterungen ein, die man nutzen kann und zeigt anhand des VMware vCenters eine mögliche Umsetzung.

5.2.2. CMDB

Die Configuration Management Database von iET erfüllt im Wesentlichen alle Anforderungen, die an eine CMDB gestellt werden.

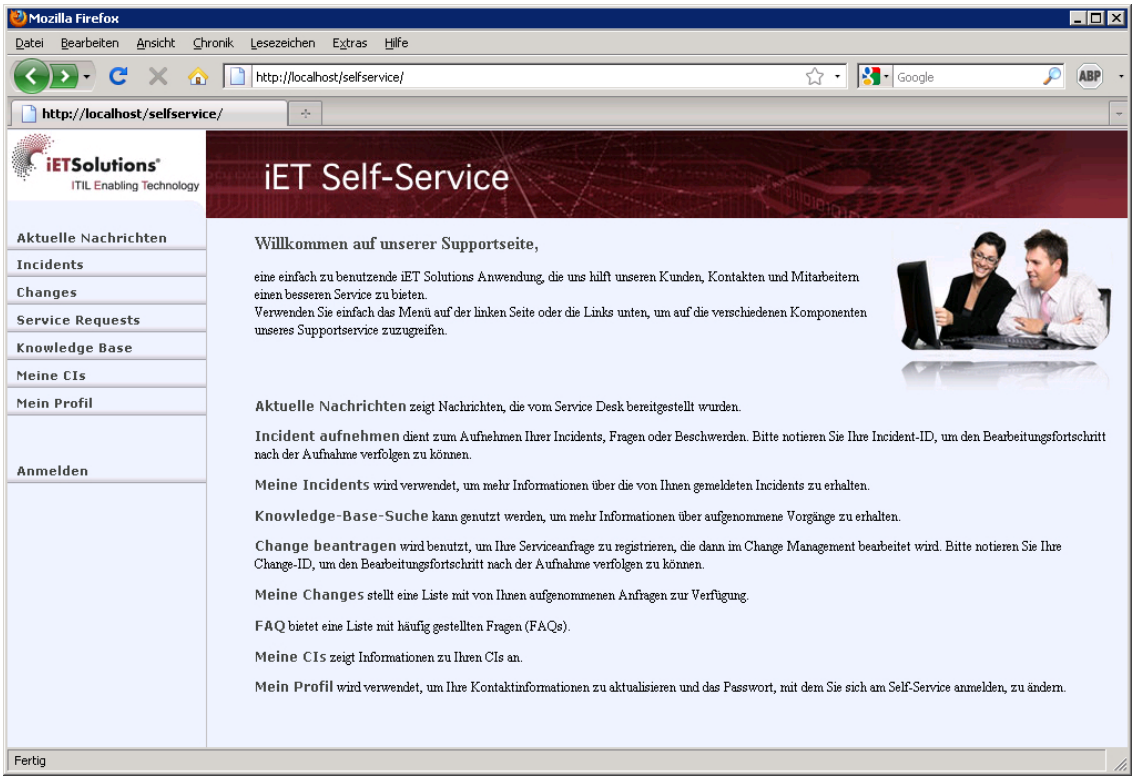


Abbildung 5.5.: iET Self-Service Portal

Ein CI wird mit Hilfe eines Produkts angelegt. Das Produkt dient dabei als Template. Das CI erbt alle Attribute und Informationen des Produkts. Sollte das Produkt später verändert werden vererben sich die Änderungen allerdings nicht automatisch an das jeweilige CI. Das Produkt kann standardmäßig vom Typ Hardware, Software, Dokumentation, Zugriffsrecht oder Service sein (wird als Produkttyp bezeichnet). Zudem gehört jedes Produkt einer Kategorie an, die später das Bildsymbol in der grafischen CMDB definiert.

Jedes CI kann entweder einem Kunden und Kontakt oder einem Mitarbeiter zugeordnet werden. In letzterem Fall handelt es sich um ein System-CI, dass seine Dienste mehreren Kunden bereitstellen kann. Pro CI können beliebig viele Attribute vergeben werden, die vom Datentyp Text, Zahl, Float, Datum oder Zeit sind und verpflichtend oder optional angegeben werden müssen.

In jedes CI können „Komponenten“ eingebaut werden. Die Komponenten müssen vorab als Komponentenprodukt angelegt und einem Produkt als mögliche Komponenten zugeordnet werden. Bei Komponenten handelt es sich um „Massenartikel“, d.h. eine Komponente mit ihren Attributen kann beliebig oft verbaut werden.

Ein CI muss zwingend mindestens einen „Service“ aus dem Service-Baum bereitstellen. Dies entspricht im Normalfall einer Dienstleistung aus Kundensicht. Unter anderen gibt es im Incident Management automatisch ein Mapping zwischen dem gewählten Service und einer Kategorie. Kategorien entsprechen eher einer technischen Sichtweise.

Abschließend lassen sich noch CIs mit anderen CIs in Beziehung setzen. iET hat hierfür gängige Beziehungstypen (z.B. „Teil von“ / „besteht aus“ und „installiert auf“ / „hat installiert“) bereits mitgeliefert. Je nach Beziehungstyp erfolgt eine andere Liniendarstellung

in der grafischen CMDB.

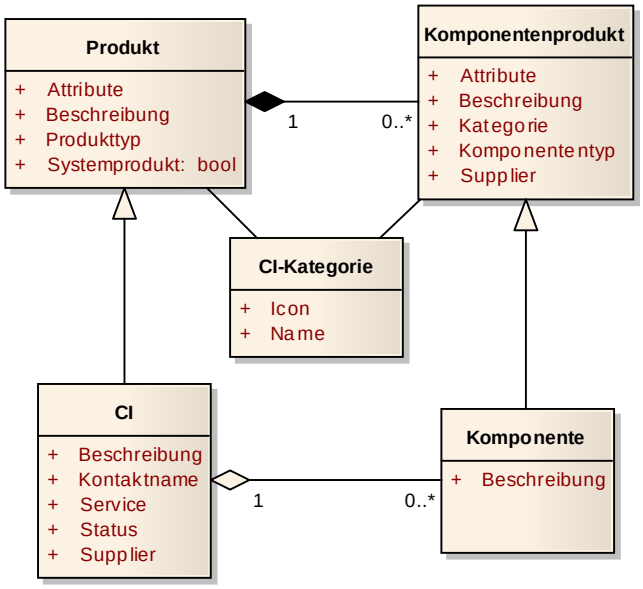


Abbildung 5.6.: iET CMDB

5.3. Konzept zur Einführung von iET ITSM am LRZ

Im folgenden Abschnitt wird ein Konzept erarbeitet, dass das in Kapitel 5.1.6 erarbeitete Informationsmodell in die Software von iETSolutions und die Daten aus den MDRs integriert.

5.3.1. Vorüberlegung zur Umsetzung des Datenmodells in iET

Während der ersten Tests, in denen typische Szenarien des LRZ mit Hilfe der Software nachgestellt wurden, haben sich einige Schwächen der CMDB, wie sie vom Hersteller konzipiert wurde, bei der Umsetzung der Anforderungen und Use-Cases des LRZ gezeigt.

- Es gibt keine Relation zwischen Attributen
- Attributwerte können nur begrenzt auf Plausibilität geprüft werden
- Es gibt keine Verknüpfung zwischen Komponenten
- Bei System-CIs muss pro Service zwingend ein Kunde angegeben werden
- Abhängigkeiten zwischen Services können nicht dargestellt werden

Dies hat Auswirkungen auf die Realisierung des Datenmodells im Hinblick auf Netzkomponenten und Services, da diese mehrschichtig aufgebaut sind, voneinander abhängen und zusätzliche Informationen benötigen. Somit müssen Konfigurationsinformationen, die mit anderen in Beziehung stehen als CI modelliert werden.

5.3.2. MDR-Anbindung

Im folgenden Abschnitt soll kurz beschrieben werden, wie die einzelnen MDRs an die CMDB von iET angeschlossen und welche Schnittstellen dabei verwendet werden können.

Zu migrierende MDRs werden nach dem Export der Daten nicht mehr produktiv genutzt. Der Export kann - aufgrund der Schwächen der MDRs - nicht immer automatisch durchgeführt werden und erfordert einige manuelle Anpassungen, Kontrollen und Nachbearbeitungen. So wird es wegen der Variabilität der Daten in der Switch-Dokumentation nur schwer möglich sein, die Excel-Daten vollständig automatisiert in die CMDB zu übernehmen. Bei „Remedy ARS“ hingegen wird es mit einem eigens dafür entwickelten Export-Skript möglich sein, die Daten nahezu automatisch zu übernehmen. Dabei können einige Daten (hauptsächlich Geräte- und Einzelteiltickets) über die CMDB Intelligence Schnittstelle in Form von XML oder CSV Dateien übernommen werden. Andere Daten wie IP-Adress-Tickets und Stabeg-Daten werden direkt in die eigens dafür angelegten Tabellen geschrieben. Daten der LRZ Netzdoku sollten ebenfalls mit Hilfe von selbstentwickelten Export-Treibern über CMDB Intelligence und direkten Import in die CMDB gelangen.

Grundsätzlich ist beim Import auf die Reihenfolge zu achten. So muss sichergestellt sein, dass bereits alle Kunden, Kontakte und Standorte hinterlegt sind, bevor mit dem Import der weiteren Stammdaten begonnen werden kann, da diese mit Kontakten und Standorten verknüpft sind.

Für die zu integrierenden Tools ist es wichtig eine beständige automatische Im- und Exportschnittstelle zu schaffen, die einen reibungslosen Datenaustausch ermöglicht. So ist es, wie beim CMDBF-Konzept (Kapitel 2.4) notwendig, die entsprechenden CIs eindeutig über mehrere MDRs hinweg zu identifizieren. iET vergibt als eindeutigen Kennzeichner IDs, die für die Identifizierung benutzt werden könnten. Zusätzlich gibt es das Feld „Externe ID“, in dem eine beliebige Zeichenfolge eingetragen werden kann. MDRs wie z.B. „VMware Virtual Infrastructure“ verwenden hingegen den Namen einer virtuellen Maschine als eindeutige Kennzeichnung. Dieser könnte somit in das Feld „externe ID“ eingetragen und bei der Synchronisierung als eindeutiger Identifikator benutzt werden. Allerdings muss sichergestellt sein, dass externe ID - wenn sie einmal festgelegt wurden - nicht mehr durch Benutzer geändert werden.

Der laufende und automatisierte Import in die CMDB sollte bei allen MDRs über die Inventory Staging Area (CMDB Intelligence) erfolgen. Auf diese Weise wird sichergestellt, dass alle eingehenden Daten einer Kontrolle durch das Change Management unterliegen.

Eine Exportschnittstelle, damit MDRs mit aktuellen Daten versorgt werden können, wurde von iET nicht vorgesehen. Die einzige Möglichkeit besteht darin, die entsprechenden Daten direkt aus der SQL-Datenbank zu gewinnen. Damit nicht jede MDR Zugriff auf alle Daten der Datenbank erhält, sollten Entwickler gezielt Views für die einzelnen MDRs erzeugen. Auf diese Weise können sich die Datenstrukturen im Hintergrund ändern, während die MDRs die Daten, die benötigt werden, immer in der gleichen Darstellung erhalten.

Damit es nicht zu Problemen kommt, in welcher Datenbasis sich gerade die aktuellsten Informationen befinden (Ort der Wahrheit), sollten alle beteiligten MDRs geänderte Daten sofort an die Import-Schnittstelle der IET-Software weitergeben.

5.4. Umsetzung des Datenmodells in iET

Das in Kapitel 5.1 erarbeitete Informationsmodell wird nun in ein Datenmodell überführt, dass zur Software von iETSolutions kompatibel ist. Eine direkte Überführung ist aufgrund einiger vom Hersteller nicht vorgesehenen Use-Cases, die in Abschnitt 5.3.1 erwähnt wurden, nicht möglich. Aus diesem Grund werden die einzelnen Punkte des Datenmodells auf ihre Realisierungsmöglichkeit hin untersucht.

5.4.1. Stammdaten

iET verfügt über ein Kundenmanagement-Modul, dass die Entitäten „Kontakt“ und „Kunde“ gut abbildet. Es ist möglich, eine hierarchische Kundenstruktur aufzubauen, so dass dort Universitäten mit ihren jeweiligen Instituten und Lehrstühlen perfekt abgebildet werden können. Das Datenfeld „Institutskennung“ kann einfach nachgerüstet werden. Für die Abbildung der Ansprechpartner sind neue Textfelder notwendig, die eine Verbindung zur Kontakt-Tabelle herstellen, da es sich nicht um Kontakte desselben Kunden handeln muss. Untersuchungen der Datenbasis der Netzdoku haben ergeben, dass derzeit 5973 Ansprechpartner-Einträge vorhanden sind. Davon gibt es 4386 erste Ansprechpartner, 1328 zweite Ansprechpartner, 256 dritte Ansprechpartner und 3 vierte Ansprechpartner. Somit sollten drei Textfelder für Ansprechpartner ausreichend sein.

Das Standortmanagement genügt nicht den Anforderungen des LRZ. Durch die vielschichtigen Untergliederung in Bezirk, Unterbezirk, Gebäude und Standort (Raum und ggf. Rack) muss hierfür eine größere Anpassung vorgenommen werden. So muss das gesamte Formular für Standortmanagement überarbeitet und zusätzlich neue Formulare für Bezirk und Gebäude entwickelt werden.

Da iET ITSM kein Bestellwesen enthält, ist es erforderlich, das Stabeg-Formular als Anpassung hinzuzufügen. Stabeg greift allerdings tief in das Configuration Management und Produktmanagement ein. Jede bestellte Hardware muss nun automatisch als CI angelegt werden. Sollte das Produkt, aus dem das CI erzeugt wird, noch nicht existieren, muss dies ebenfalls erst in Absprache mit dem Configuration Management erzeugt werden. Außerdem ist es sinnvoll, das CI-Formular zu erweitern, um jederzeit auf die Bestelldaten, Wartungsverträge (UC) u.ä. zurückgreifen zu können. Wartungsverträge von Lieferanten und Herstellern werden durch die Applikation „Service Level Management“ vollständig abgedeckt. Die abstrakte Entität „Hardware“ wird separat in iET modelliert. Jedes Produkt vom Typ Hardware wird mit den entsprechenden Attributen angelegt.

5.4.2. Netzinfrastruktur

Da die Kerndienstleistung des LRZ die Bereitstellung der Netzinfrastruktur für viele Organisationen und Institute ist, ist es - wie eine Analyse der entsprechenden MDRs gezeigt hat - wichtig zu wissen, welches VLAN auf welchem Switchport geschaltet ist bzw. welche Ports miteinander verbunden sind. Die Verbindungen sollen dabei automatisch auf beiden Seiten ersichtlich und konsistent sein. Zudem soll es möglich sein, die Zusammenhänge in der grafischen CMDB zu sehen. Außerdem sollen Änderungen der Kontrolle des Change Managements unterliegen. Das CMDB-Modell in iET ITSM lässt hierzu mehrere Möglichkeiten zu, die im folgenden kurz beschrieben werden.

Variante 1

Da Switches in der Regel aus mehreren Modulen mit jeweils vielen Ports bestehen, lassen sich Module als Komponenten realisieren, die wiederum physische Ports in Form von Attributen bereitstellen. In diesen Attributen wird vermerkt, welche Ports verbunden sind bzw. welcher Server angeschlossen ist. Logische Ports werden durch eine zusätzliche Tabelle als Erweiterung zum CMDB-Modell gespeichert.

Komponenten sind Massenartikel, d.h. eine Komponente kann in mehreren CIs gleichzeitig verbaut werden. Damit ist es nicht möglich, Komponentenattribute individuell zu pflegen, da alle Attribute mit ihren Werten vererbt werden. Um dieses Problem zu umgehen müsste für jedes Modul eine neue Komponente angelegt werden.

Ein weiteres Problem ist, dass es keinen Mechanismus gibt, den komplexen Attributinhalt auf Richtigkeit zu überprüfen und die entsprechende Gegenverbindung in der anderen Komponente einzutragen. Der Benutzer müsste somit immer an beiden Ports die entsprechenden Eintragungen vornehmen.

Variante 2

Switch und Switch-Module werden als CIs abgespeichert. Die Eigenschaften der physischen Ports werden als Attribute der Switch-Module gespeichert. Die logischen Ports werden wie auch schon in Variante 1 als zusätzliche Tabelle realisiert.

Auch hier hat man das Problem, dass Attribute keine Beziehungen zu anderen CIs herstellen können. Die Beziehungen müssten manuell auf beiden Seiten gepflegt werden, was den Bedienkomfort der Applikation stark einschränkt. Zudem muss die Ansicht der grafischen CMDB angepasst werden, um alle verbunden Geräte eines Switches anzeigen zu lassen.

Variante 3

Switch, Switch-Module und physische Ports werden als CI angelegt und miteinander in Beziehung gesetzt. Dadurch steigt allerdings die Zahl zu verwaltender CIs schnell in unüberschaubare Bereiche und eine gezielte Suche wird erheblich erschwert.

Variante 4

Switches und Module werden als CIs implementiert. Mit Hilfe einer Anpassung der vorhandenen CMDB werden Verbindungen physischer Ports mit Hilfe von zusätzlichen Parameterfeldern als Teil der Relation zwischen CIs erfasst. Dafür werden - gemäß dem Informationsmodell - drei zusätzliche Parameter benötigt: Quell-Port, Ziel-Port und Tagging. Logische Ports werden wie oben beschrieben in einer separaten Tabelle vorgehalten. Informationen zu VLANs und Einträge für die „StandortDoku“ werden ebenfalls in einer separaten Tabelle abgespeichert und mit der CI-Relation verknüpft.

Diese Variante ist mit etwas Aufwand verbunden, da die entsprechenden Formulare und Tabellen angepasst werden müssen, aber es erlaubt eine konsistente Pflege, welche CIs über welchen Port miteinander verbunden sind und eine grafische Darstellung, die den Anforderungen des LRZ entspricht.

Realisierung

Von den oben dargestellten und in Tabelle 5.7 zusammengefassten Möglichkeiten kristallisiert sich Variante 4 klar als Favorit heraus. Durch eine relativ einfach durchzuführende Anpassung werden alle Anforderungen des LRZ erfüllt, ohne dass die CMDB dabei mit unnötigen Configuration Items überfrachtet wird. Abbildung 5.8 zeigt einen Entwurf des Screen-Designs zur Erfassung der Netzinformationen.

Die Lösung für logische Ports und damit die komplette IP-, DNS- und Subnetzverwaltung wird dabei in ein eigenständiges Formular ausgegliedert, was eine höhere Flexibilität erlaubt.

	Variante 1	**Variante 2**	**Variante 3**	**Variante 4**
Switch	CI	CI	CI	CI
Switch-Modul	Komponente	CI	CI	CI
Physischer Port	Komponenten-attribut	CI-Attribut	CI	Parameter als Teil der Beziehung
Logischer Port	zus. Tabelle	zus. Tabelle	CI-Attribut	zus. Tabelle

Abbildung 5.7.: Möglichkeiten zur Realisierung von Netzkomponenten in iET

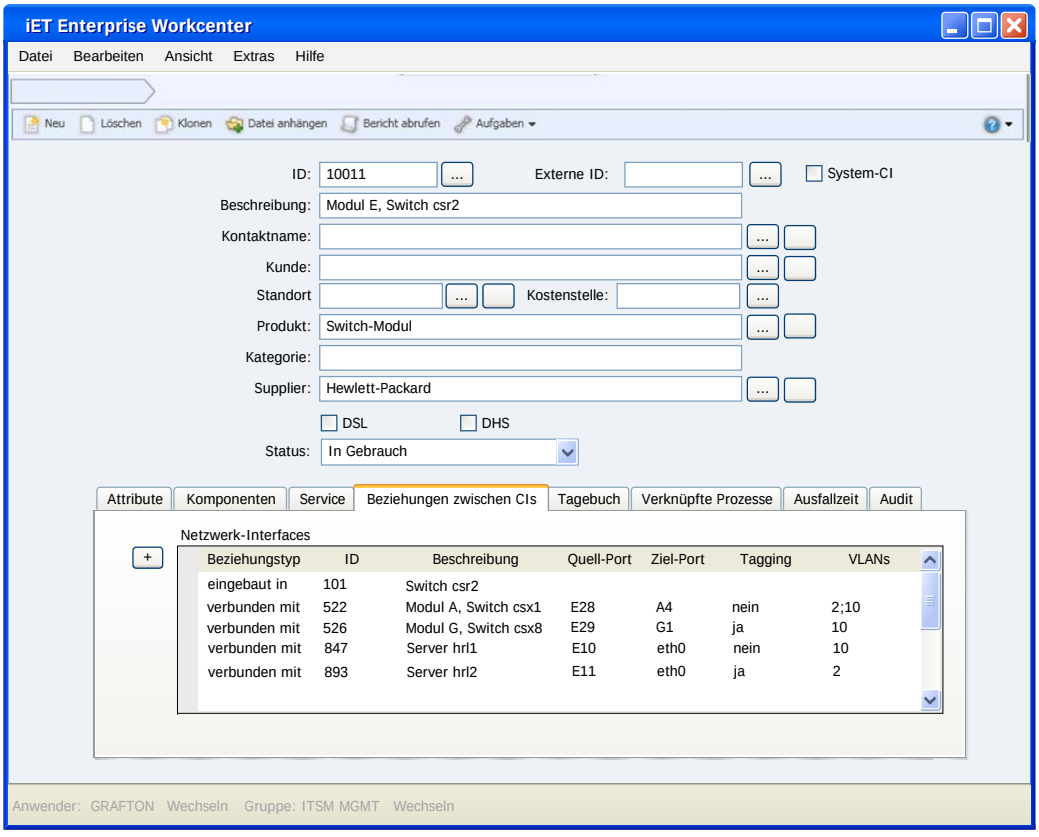


Abbildung 5.8.: Mögliche Gestaltung der CI-Maske für Netzkomponenten

5.4.3. Physische Server

Die Anlage der physischen Server als „System-CI“ erfolgen. Damit müssen diese automatisch einem Mitarbeiter zugewiesen werden, was im Betriebskontext des LRZ am sinnvollsten erscheint. Der Mitarbeiter wird damit als Systemadministrator definiert. Die im Server befindlichen Netzwerkkarten müssen aufgrund der im vorangegangen Abschnitts gewählten Umsetzung als eigene CIs definiert werden. Zwar wäre es auch möglich den Server direkt mit einer anderen Netzkomponente zu verbinden, allerdings verliert man dann die Information der MAC-Adresse, die für den Netzbetrieb von Bedeutung ist.

Da aufgrund von MDRs die angeschlossenen Festplatten wegen ihren benötigten Informationen als eigene Entität modelliert wurden, sollten die Festplatten als Komponenten angelegt werden.

5.4.4. Virtuelle Infrastruktur

Die Bereitstellung virtueller Maschinen wird das LRZ in Zukunft als kostenpflichtige Dienstleistung anbieten. Kunden des LRZ sollen über ein Interface neue Maschinen beantragen, Konfigurationen vorhandener ändern und die Maschine archivieren lassen.

iET ITSM bietet hierfür die Applikation „Request Fullfillment“ an, die mit dem Self-Service-Portal (Kapitel 5.2.1) gekoppelt ist. Auf diese Weise kann der Genehmigungs-Workflow für virtuelle Maschinen auf Seiten des LRZ komfortabel umgesetzt werden. Nach erfolgreicher Genehmigung kann über eine Erweiterung der Software die Provisionierung durch die VM Infrastructure angestoßen werden.

Die Entwicklung des (Web-)Formulars zur Abfrage aller relevanten Informationen über die virtuelle Maschine muss in Rücksprache mit den VM-Administratoren erfolgen. Dasselbe Formular könnte auch intern innerhalb der ITSM-Software zur Anlage neuer Maschinen durch LRZ-Mitarbeiter genutzt werden.

Eine virtuelle Maschine existiert nach erfolgreicher Genehmigung als CI in der CMDB. Die eingebauten Festplatten kann man wie auch bei physischen Servern als Komponenten erfassen. Netzwerkkarten sollten als CIs angelegt werden, um Relationen zum VM-Switch abbilden zu können. Jede Netzwerkkarte stellt beliebig viele logische Ports zur Verfügung, die wiederum in der IP-, DNS- und Subnetzverwaltung administriert werden.

Der virtuelle Switch (VM Switch) sollte als eigenes CI geführt werden. Pro geschaltetes VLAN wird es am LRZ einen virtuellen Switch geben.

Die Entität „ESX Cluster“ wird als Dienst modelliert und somit im folgenden Abschnitt näher behandelt.

5.4.5. Betriebssystem, Dienste und Dienstabhängigkeiten

Betriebssystem-Informationen können als Attribute bei den CI-Klassen „Physischer Server“ und „virtuelle Maschine“ eingetragen werden. Über eingerichtete Abfragen können so schnell Rechner gefunden werden, deren Betriebssystem-Version veraltet ist und ein Update benötigen.

Server oder jede virtuelle Maschine können einen oder auch mehrere Dienste bereitstellen. Dabei handelt es sich um einen Infrastruktur-Service wie er in Kapitel 5.1.5 beschrieben wurde (z.B. „ESX Cluster“). Dienste werden als eigene CIs vom Typ „Service“ modelliert. Zur besseren Unterscheidung sollte jeweils ein Produkt „Business Service“ und „Infrastruktur Service“ angelegt werden.

Services können somit in Beziehung zueinander gesetzt werden, während Server an „Infrastruktur Services“ gebunden sind. Mit Hilfe des zusätzlichen Parameters der Anschaltstufe läßt sich somit ein vollständiger Anschaltplan bzw. Lastabwurfplan generieren. Für die Beziehungen sollte in iET ITSM ein neuer Beziehungstyp („stellt bereit“ / „benötigt“) definiert werden.

5.4.6. Umsetzung als Datenmodell in iET

Nach den detaillierten Überlegungen zu den einzelnen Entitäten ist es nun leicht das erstellte Modell in die CMDB-Struktur von „iET ITSM“ einzuarbeiten. Wie man Abbildung 5.9 entnehmen kann, lassen sich die meisten Entitäten als CIs oder als Attribute zu CIs umsetzen. Lediglich ein kleiner Teil muss als eigenständige Anwendung hinzuentwickelt werden. Dies ist aber durch die offene Datenstruktur von iET kein allzu großes Problem.

5.5. Zusammenfassung

In diesem Kapitel wurde auf Basis der Informationsmodelle der analysierten MDRs ein neues Gesamtmodell erzeugt, dass die meisten Anforderungen der MDRs abdeckt. Nach einer Vorstellung der vom LRZ ausgewählten ITSM-Software iETSolution ITSM 5.0 wurde im Anschluss die Realisierung, d.h. die Integration und Migration vorhandener Daten sowie die Umsetzung innerhalb der ITSM-Software, diskutiert.

Das Datenmodell lässt sich in Teilen gut in die vorhandene CMDB-Struktur einbringen, jedoch ist es an einigen Stellen notwendig Anpassungen der Software durchzuführen. Da iET ein relativ gutes Entwicklungswerkzeug zur Verfügung stellt und das Datenmodell der CMDB offengelegt ist, lassen sich diese Anpassungen gut durchführen.

Um die Implementierung des Datenmodells und die Anbindung von MDRs zu zeigen, wird im folgenden Kapitel dies exemplarisch für die Datenquellen „LRZ Netzdoku“ und „VMware Infrastructure“ durchgeführt.

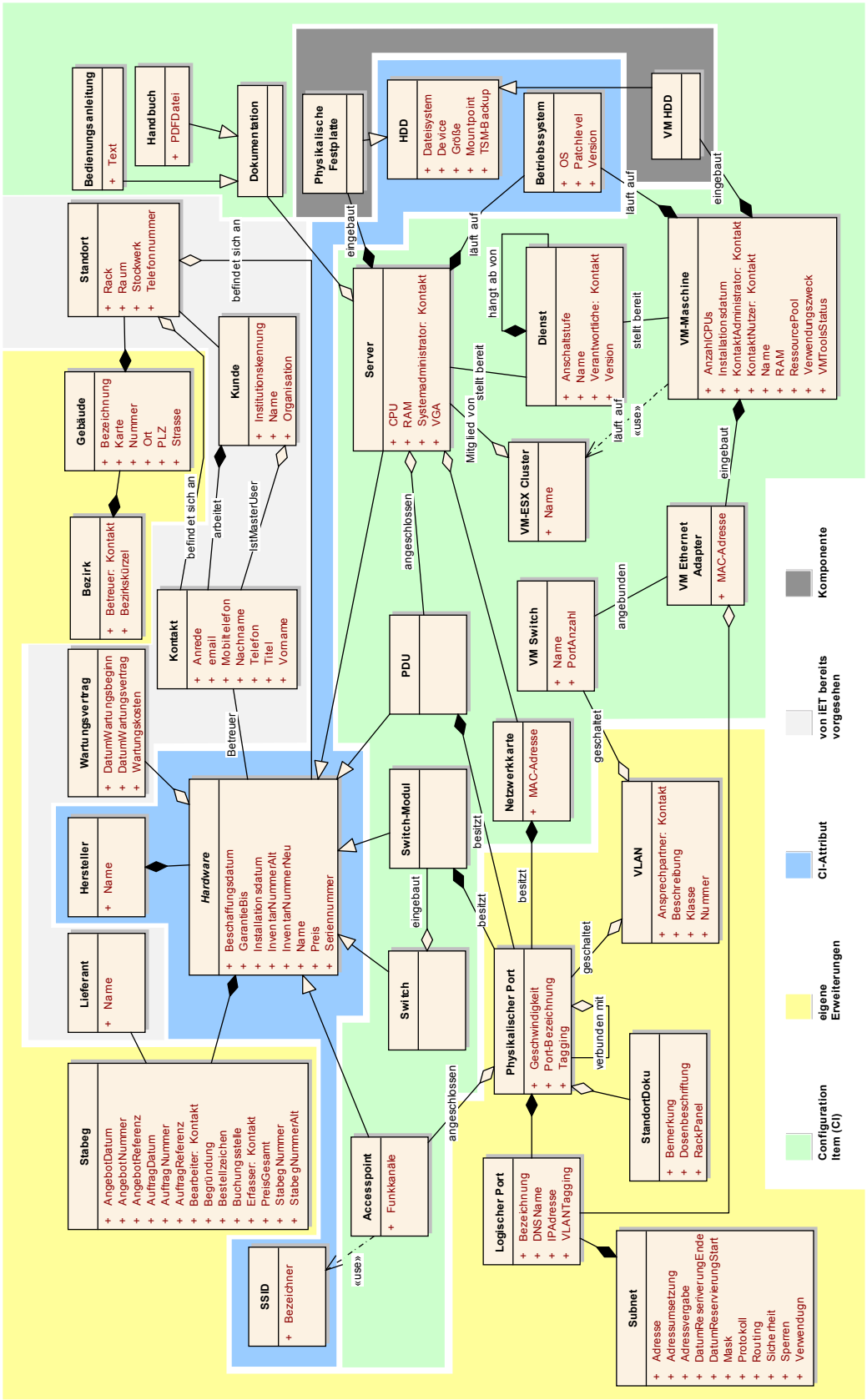


Abbildung 5.9.: Datenmodell mit Realisierungsmöglichkeit in iET

6. Proof-of-concept

Im folgenden Kapitel wird das erarbeitete und für den Einsatz in iET ITSM vorbereitete Datenmodell-Konzept exemplarisch umgesetzt. Dabei werden zunächst Stammdaten aus der LRZ Netzdoku ausgelesen und übernommen. Anschließend wird eine Grundstruktur erzeugt, um den Betrieb der virtuellen Infrastruktur zu simulieren. Über die CMDB Intelligence Importschnittstelle werden die virtuellen Maschinen aus dem vCenter eingelesen.

6.1. Stammdaten aus LRZ Netzdoku

Mit Hilfe der LRZ Netzdoku sollen Stammdaten wie Kunden und deren Kontakte automatisch übernommen werden.

Für die Realisierung ist es notwendig ein Skript zu schreiben, dass die Daten aus der Oracle-Datenbank der LRZ Netzdoku exportiert, aufbereitet und an die MS SQL-Datenbank der ITSM-Software weitergibt. Da eine Datenbankverbindung von zwei Herstellern gleichzeitig aufgebaut werden muss, wurde das Skript in PHP entwickelt. Das vollständige Programm findet sich im Anhang B.

Zunächst mussten mit Hilfe des Developer-Studios kleine Anpassungen vorgenommen werden. So wird das benötigte Feld für die Institutskennung in der Tabelle „Account“ mit den Angaben aus Tabelle 6.1 angelegt. Erste Analysen haben gezeigt, dass einige Datenfelder innerhalb der CMDB zu klein dimensioniert sind und ebenfalls mit Hilfe des Developer Studios bzw. dem MS SQL Server Management Studio vergrößert werden müssen. So muss das Namensfeld in der Tabelle „Account“ sowie die Felder „last_name“ und „phone“ in der Tabelle „account_contact“ vergrößert werden, um die Daten der Netzdoku korrekt zu importieren. Ansonsten werden Einträge abgeschnitten und Informationen gehen beim Import verloren. Abschließend wird im Kundenformular „IT ACCOUNT MANAGEMENT“ ein für die Institutskennung angelegtes Textfeld auf dem Formular platziert (Datenfeldparameter in Abbildung 6.1).

Eigenschaft	**Wert**
Name	institutskennung
Data Type	variable
Display Type	textbox
Case	Mixed Case
Max Length	10
Access Mode	Allow Form Access
Form Label	LRZ-Institutskennung

Abbildung 6.1.: Datenfeld-Parameter im Developer Studio

Nach den Anpassungen kann das eigentlich Skript starten. Das Skript baut zunächst eine Verbindung zu beiden Datenbanken auf. Als erstes werden alle Organisations-Namen (z.B.

„Technische Universität München“) aus der Netzdoku extrahiert und in die ITSM-Datenbank übernommen. Dies dient dazu, um später Hierarchien aufbauen zu können. Anschließend durchläuft das Skript die Instituts-Tabelle und legt das Institut mit seinem Standort und ggf. dem Master-User (als Kontakt) an. Um den Standort korrekt anzulegen, muss dabei das Netzdoku-Feld „ORT“ in seine Bestandteile zerlegt werden. So enthält dieses Feld Postleitzahl und Ort bei Orten in Deutschland. Bei der Anlage der Master-User Kontakts muss ebenfalls aus einem einzigen Feld Anrede, Titel, Vorname und Nachname extrahiert werden. Als letztes können aus der Personen-Tabelle weitere Kontakte des entsprechenden Instituts hinzugefügt werden. Dies verursacht keine weiteren Probleme, da die Daten bereits normalisiert in den entsprechenden Spalten vorliegen.

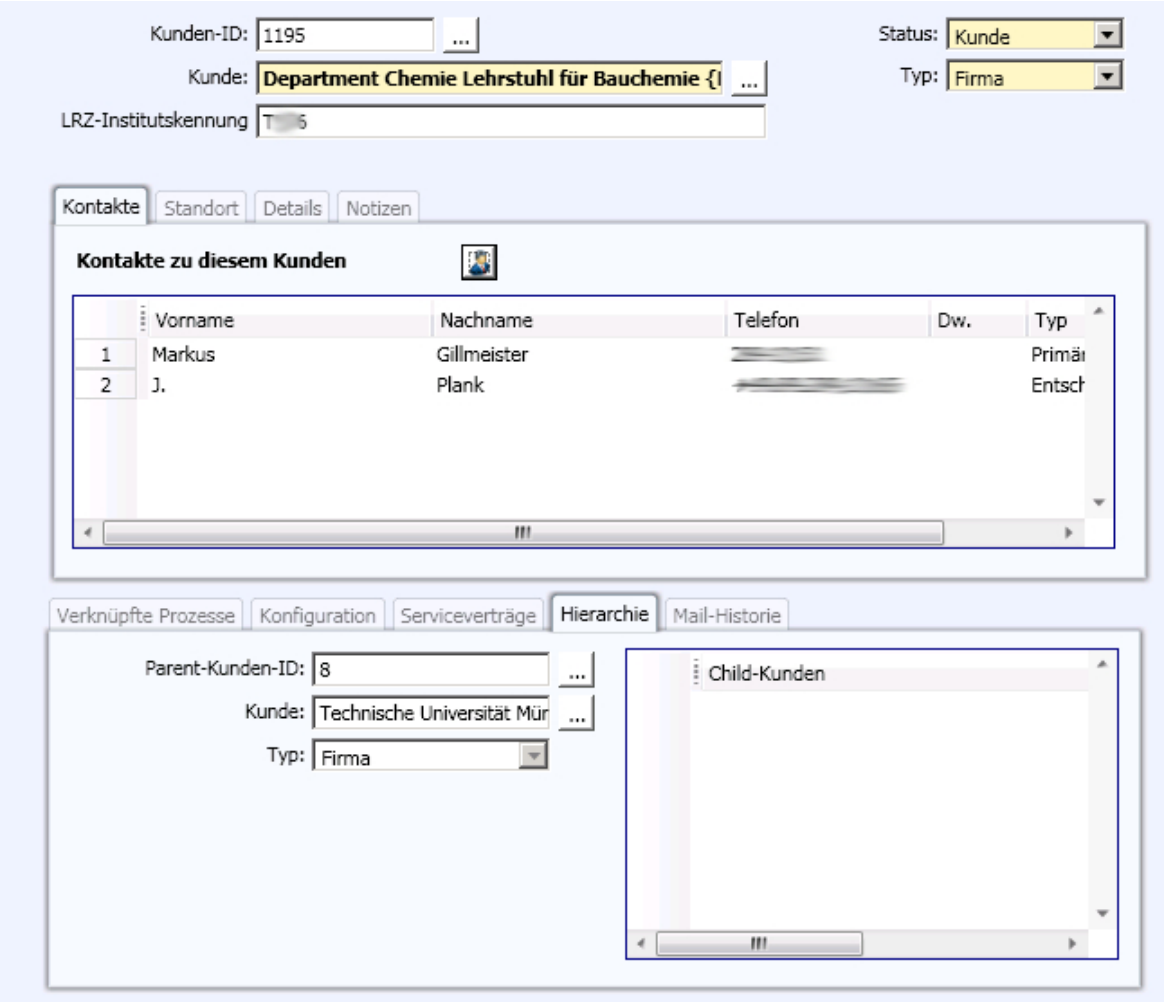


Abbildung 6.2.: Beispielscreen mit importierten Stammdaten

Abbildung 6.2 zeigt ein Beispiel für einen Lehrstuhl an der „Technischen Universität München“. In der Registerkarte sieht man die Hierarchie-Beziehung eingetragen. Das Feld für die Institutskennung wurde wie oben beschrieben hinzugefügt. Als weitere Anpassung müsste man nun entsprechende Such- und Auswahldialoge modifizieren, um gezielt nach der Institutskennung suchen zu können.

6.2. Beispiel-Datenstruktur anlegen

Als nächstes wird ein Teil des Datenmodells 5.1.6 in die CMDB implementiert, um später virtuelle Maschinen zu importieren.

Um die Datenstruktur für die virtuelle Infrastruktur anzulegen müssen vorab einige Dinge eingerichtet werden. Zunächst muss die Anlage eines Lieferanten erfolgen, der die Business- und Infrastruktur-Services erbringt. In vorliegenden Fall wird dabei ein Lieferant mit Namen „Leibniz Rechenzentrum“ erzeugt. Als nächstes müssen die Produktattribute für den Produkttyp „Service“ um „Anschaltstufe“ und „Version“ erweitert werden. Um später für die Services eine bessere Darstellung in der grafischen CMDB zu bekommen werden zwei neue CI-Kategorien angelegt, die ein eigenes Icon erhalten. Aus dem gleichen Grund wird eine Kategorie für „Virtuelle Maschine“ erzeugt. Zuletzt müssen noch CI-Templates (Produkte) für die Business- und Infrastruktur-Services angelegt werden. Ein Business-Service benötigt dabei kein Attribut für die Anschaltstufe, da dieser Dienst funktioniert, sobald alle technischen Services darunter in Ordnung sind. Infrastruktur-Services werden gemäß dem Datenmodell mit den Attributen „Anschaltstufe“ und „Version“ ausgestattet. Diese werden dem Produkttyp „Virtuelles Produkt“ zugewiesen (der neu angelegt wurde), um eine bessere Unterscheidung von physischer Hardware zu erreichen.

Nun können die Produkte „Virtuelle Maschine“, „Virtuelle Netzwerkkarte“, „Server“, „Netzwerkkarte“, „VM Switch“, „Switch“, „Switch-Modul“ sowie die Komponenten „Virtuelle HDD“ und „Festplatte“ gemäß dem Datenmodell mit ihren jeweiligen Attributen angelegt werden.

Gemäß Roll [Rol08] werden nun beispielhaft Komponenten zum Betrieb der virtuellen Infrastruktur manuell erzeugt. Dazu werden Daten aus „Remedy ARS“ und der „LRZ Netzdoku“ herangezogen. Die virtuelle Infrastruktur am LRZ besteht aus zwei ESX-Clustern (Produktion und Test), die jeweils aus 7 bzw. 2 Cluster-Nodes bestehen. Jede Clusternode enthält zwei Netzwerkkarten, die aus Redundanzgründen an zwei unterschiedlichen Switches angeschlossen sind. Dieser Zusammenhang wurde modelliert und kann mit Hilfe der grafischen CMDB dargestellt werden (Abbildung 6.3).

Um die Beziehung der Netzverbindungen über physische Ports zu realisieren, muss dies mit Hilfe des Developer Studio programmiert werden. Dazu wird zunächst die Datentabelle „it_cmdb_ci_ci“ um die Felder Quellport und Zielport erweitert. Anschließend müssen in die entsprechenden Formulare („IT CMDB CI REL ENTRY“ und „IT CMDB CI REL VIEW“) die beiden neu erstellen Datenfelder hinzugefügt werden. Damit die Portbeziehungen auch in der CI-Übersicht angezeigt werden, muss der Query „IT CMDB CI RELATION“ erweitert werden. Abbildung 6.4 zeigt die Erweiterung der Software.

6.3. VMware Infrastructure Anbindung

Zur Anbindung der „VMware Infrastructure“ an „iET ITSM“ werden im Folgenden zwei Anbindungswege genutzt. Zum Einen über einen Live-Zugriff aus dem Programm auf die VMware API (Abschnitt 6.3.1). Zum Anderen die Nutzung der CMDB Intelligence Schnittstelle zum Import von virtuellen Maschinen aus der Infrastruktur 6.3.2.

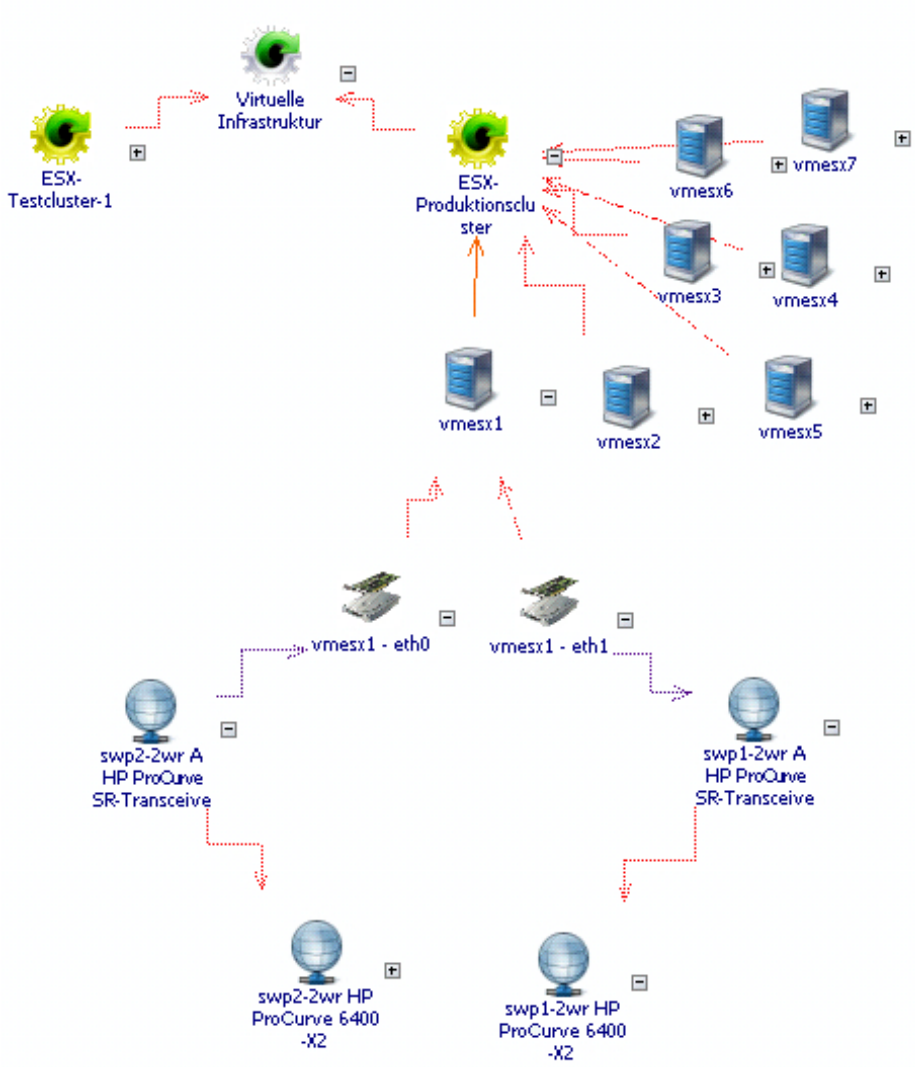


Abbildung 6.3.: Virtuelle Infrastruktur am LRZ in der grafischen CMDB-Darstellung

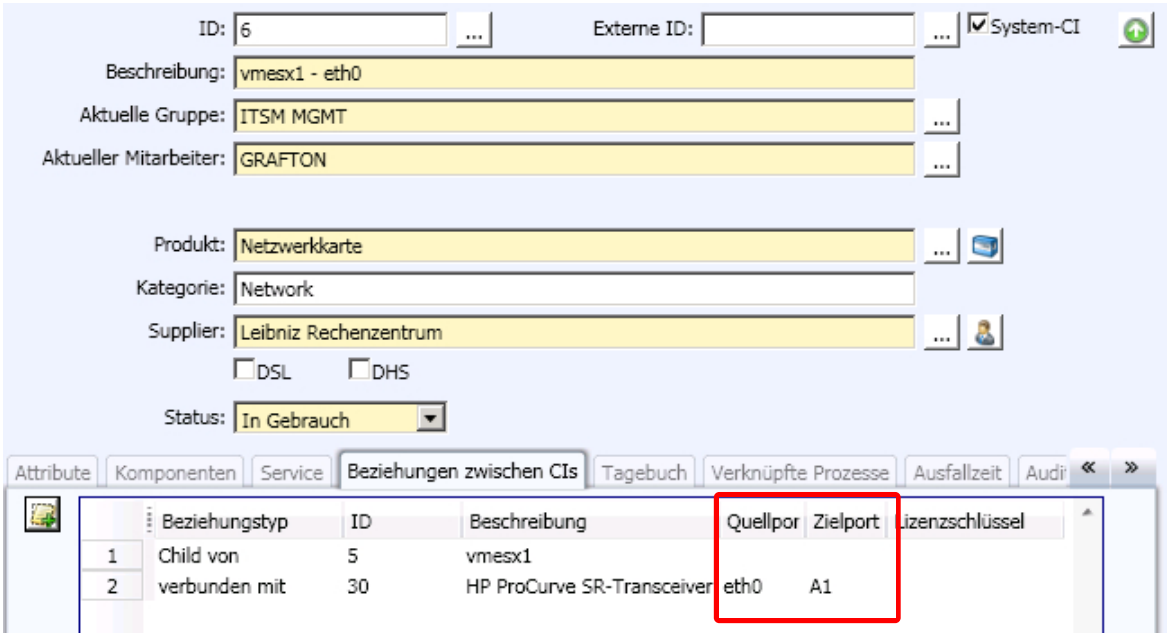


Abbildung 6.4.: Erweiterung der Beziehungen zwischen CIs in iET

6.3.1. Live-Zugriff innerhalb von „iET ITSM“

Um einen Live-Zugriff innerhalb der ITSM-Software auf Daten der virtuellen Infrastruktur zu erlangen, muss mit Hilfe von Java-Klassen eine Erweiterung programmiert werden. Dazu werden im zunächst die grundsätzlichen Java-Schnittstellen des Tools vorgestellt. Im darauf folgenden Abschnitt wird auf die Java-API von VMware eingegangen. Als letztes erfolgt die Anbindung von VMware an iET.

Java-Schnittstelle in iET

Mit Hilfe von Java lässt sich das Klassenmodell des iET Enterprise Hosts erweitern. So ist es möglich, zahlreiche Events (wie z.B. Formular öffnen oder speichern, Textfeld verlassen) abzufangen und mit zusätzlichem Code zu versehen. Abbildung 6.5 zeigt die Funktionen auf, die in eigenen Klassen implementiert werden können.

Das iET Klassenmodell ist hierarchisch aufgebaut. Die Klasse „EntSession“ verwaltet dabei Informationen über den aktuell angemeldeten Benutzer und dessen Gruppenmitgliedschaften. Es ist möglich den Login- bzw. Logoutvorgang sowie einen Benutzerwechsel abzufangen und zu erweitern.

Die Klasse „EntClientObject“ hält Informationen über den aktuellen Benutzeragenten und kann nicht erweitert werden.

Mit Hilfe der Klasse „EntForm“ können Formularfelder ausgelesen und verändert werden. Hierfür gibt es eine Vielzahl an Events, an die man sich binden kann. So ist es möglich beim Laden oder Speichern eines neuen Formulars, Änderungen vorzunehmen bzw. auf die Formulardaten zuzugreifen.

Die Klasse „EntCtrl“ bzw. „EntCtrlBtn“ erlaubt ein Auslesen oder Verändern einzelner Textfelder oder Buttons innerhalb eines Formulars. Dies ist durch die Events „AfterChange“

bzw. „OnChange“, die direkt bei einer Veränderung (z.B. Texteingabe oder Button-Klick) ausgelöst werden, möglich.

Als letztes gibt es die zwei Klassen „ClientAuth“ und „ClientNotify“, die nicht direkt an das Klassenmodell gebunden sind. Mit diesen Klassen ist es möglich ein eigenes Authentifizierungsschema zu realisieren. Der iET Enterprise Host ruft dazu die selbst erstellten Klassen - sofern sie definiert sind - bei jedem Authentifizierungsvorgang auf.

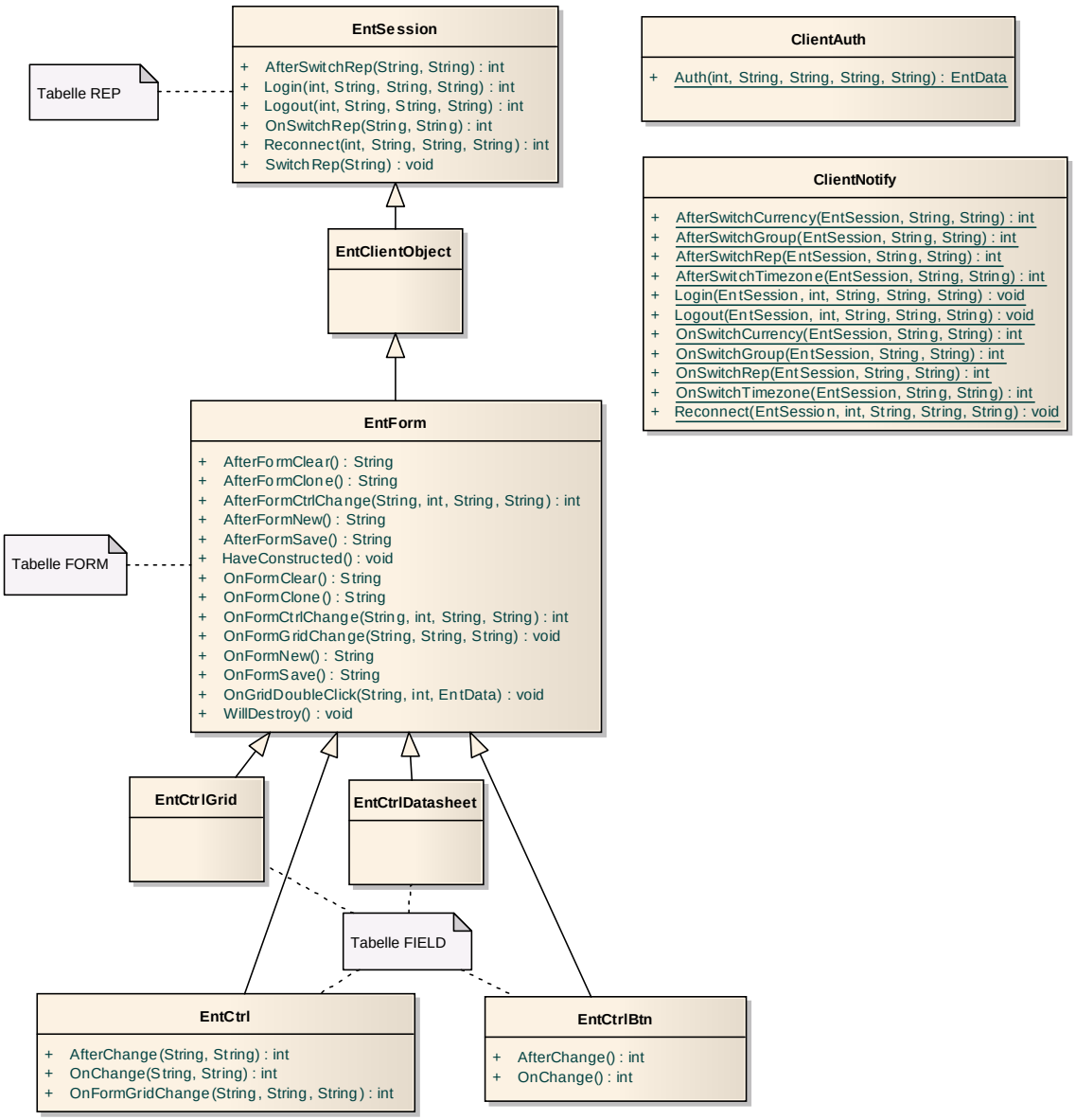


Abbildung 6.5.: iET Klassendiagramm für eigene Erweiterungen erstellt auf Basis von [iET08]

Java-Schnittstelle zu VMware vCenter

Für die „VMware Infrastructure“ gibt es ein offizielles Software Development Kit (VI-SDK) zum Download. Mit Hilfe der damit erstellten API-Libaries kann das vCenter über WebServices (SOAP-Schnittstelle) aus Java heraus angesprochen werden. Allerdings erfordert die VI-SDK ein sehr fundiertes Wissen über das Objekt-Modell von VMware. Außerdem kann nur sehr mühsam eigener Code erzeugt werden. Aus diesem Grund wurde das Open-Source-Projekt „VMware VI (vSphere) Java API“ (VI Java API) [Jin09] ins Leben gerufen. Es kapselt die native VI-API, erleichtert den Zugriff auf das Objekt-Modell erheblich und macht den eigenen Quellcode leserlicher. Daher wurde entschieden, die Anbindung des vCenters mit Hilfe der VI Java API durchzuführen.

Die VI Java API erlaubt mittels einer Klasse „InventoryNavigator“ ein gezieltes Navigieren oder Suchen im VMware Objektbaum. So kann beispielsweise mit der Methode „searchManagedEntities“ gezielt nach allen virtuellen Maschinen gesucht werden. Diese erhält man in Form eines „ManagedEntity“-Arrays zurück. Abbildung 6.6 zeigt die Traversierungsmöglichkeiten auf.

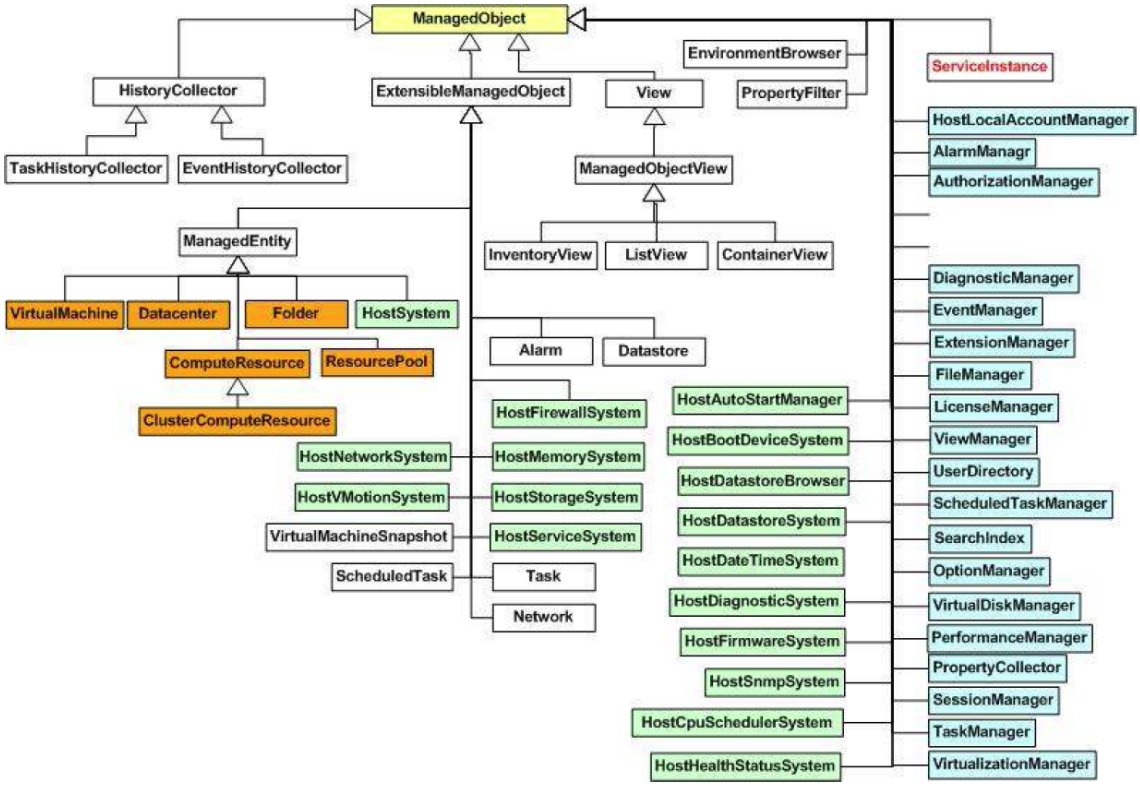


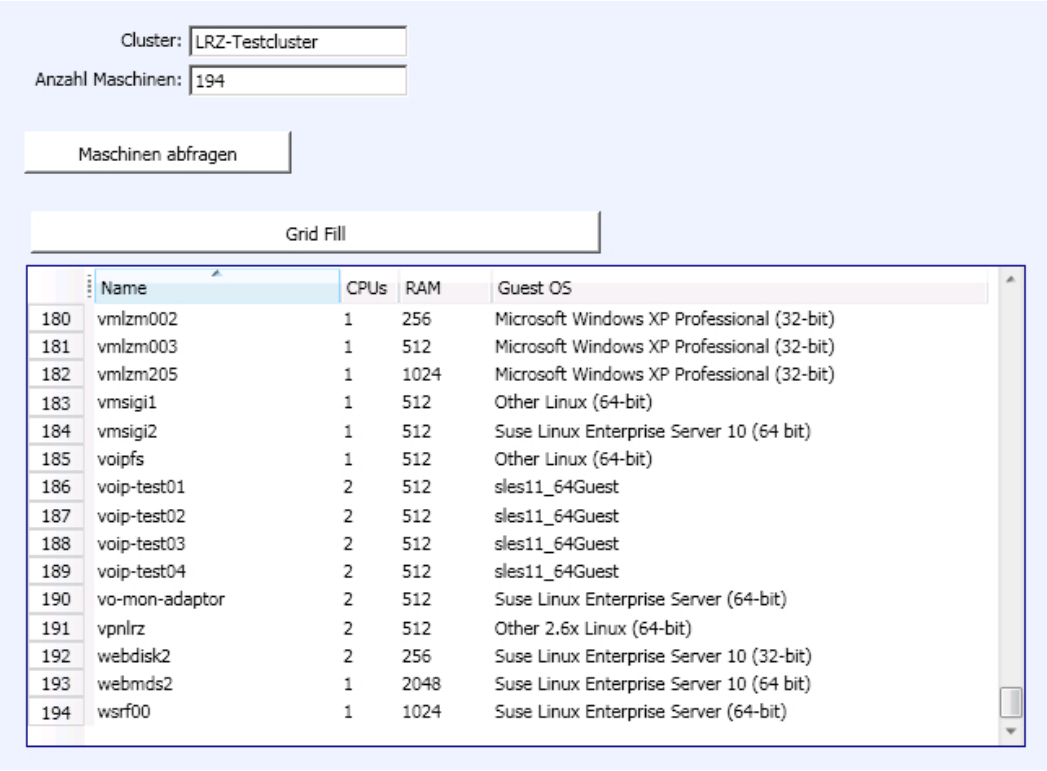
Abbildung 6.6.: VI Java API Objektmodell, Quelle: [Jin08]

Testapplikation

Um nun Daten direkt aus einem iET-Formular heraus von VMware abzufragen, wird zunächst das Formular vorbereitet. Ziel ist es, die Anzahl der virtuellen Maschinen im LRZ-Testcluster nach Klick auf einen Button abzufragen. Als zweiten Schritt (getriggert durch einen zweiten

Button) soll eine Auflistung aller virtuellen Maschinen des Testclusters mit Namen, Anzahl CPUs, Arbeitsspeicher und Betriebssystem erstellt werden. Somit werden für die erste Abfrage zwei Textboxen erstellt. Für die zweite Abfrage wird ein Datagrid mit den vier gewünschten Spalten angelegt. Die zwei Buttons, die die Abfragen auslösen, werden jeweils an eine Klasse vmware1 (Anzahl der Maschinen im Testcluster, Quellcode im Anhang C.1) bzw. vmware2 (Datagrid aller virtueller Maschinen, Quellcode im Anhang D.1) gebunden.

In den Klassen wird die OnChange()-Methode für den Aktionsstart genutzt. In beiden Fällen wird zunächst eine Verbindung zum vCenter hergestellt und beginnend von der Objektwurzel eine Suche nach „ComputeResource“ durchgeführt. Dies ergibt eine Auflistung aller Cluster. Im Falle des Testclusters (Bezeichnung „ESX-Testcluster-1“) werden die untergeordneten ResourcePools abgefragt. Für die Kalkulation der Anzahl der virtuellen Maschinen wird lediglich abgefragt, wie viele Maschinen sich im jeweiligen Ressource-Pool befinden. Für die Erstellung des Datagrids erfolgt die Abfrage der gewünschten Informationen über eine virtuelle Maschine über die Klassen „VirtualMachine“ und „VirtualMachineConfigInfo“. Die Ergebnisse werden anschließend in die entsprechenden Formularfelder bzw. in das Datagrid geschrieben. Abbildung 6.7 zeigt das Ergebnis als Screenshot.



	Name	CPUs	RAM	Guest OS
180	vmlzm002	1	256	Microsoft Windows XP Professional (32-bit)
181	vmlzm003	1	512	Microsoft Windows XP Professional (32-bit)
182	vmlzm205	1	1024	Microsoft Windows XP Professional (32-bit)
183	vmsigi1	1	512	Other Linux (64-bit)
184	vmsigi2	1	512	Suse Linux Enterprise Server 10 (64 bit)
185	voipfs	1	512	Other Linux (64-bit)
186	voip-test01	2	512	sles11_64Guest
187	voip-test02	2	512	sles11_64Guest
188	voip-test03	2	512	sles11_64Guest
189	voip-test04	2	512	sles11_64Guest
190	vo-mon-adaptor	2	512	Suse Linux Enterprise Server (64-bit)
191	vpnlrz	2	512	Other 2.6x Linux (64-bit)
192	webdisk2	2	256	Suse Linux Enterprise Server 10 (32-bit)
193	webmds2	1	2048	Suse Linux Enterprise Server 10 (64 bit)
194	wsrf00	1	1024	Suse Linux Enterprise Server (64-bit)

Abbildung 6.7.: Live-Anbindung zwischen VMware vCenter und iET ITSM

6.3.2. Import virtueller Maschinen über CMDB Intelligence

Die in Abschnitt 6.3.1 abgefragten Daten wurden bisher nicht in die Configuration Management Database von iET übernommen. Einen Speichermechanismus könnte man zwar realisieren, in der Praxis ist dieses Verfahren aber zu umständlich, da es sich um keine au-

tomatische Synchronisation handelt, sondern der Vorgang jedesmal per Hand angestoßen werden muss.

Eine Möglichkeit Konfigurationsinformationen automatisiert zu importieren bietet die Komponente „CMDB Intelligence“ von iET. Dabei werden Daten aus Monitoring-Tools und sonstigen Datenquellen in eine Inventory Staging Area geschrieben. Diese Zwischenebene funktioniert wie in Kapitel 2.4.1 beschrieben. Die Daten der Inventory Staging Area werden dabei mit Regelsätzen auf in der CMDB bekannte CIs gemappt.

CMDB Intelligence bietet den Import über XML-Dateien (nach einem definierten DTD-Schema), CSV-Dateien und einer ODBC-Schnittstelle an. Für die Abfrage der virtuellen Maschinen wird wieder die VI Java API aus Abschnitt 6.3.1 genutzt. Somit eignet sich eine XML-Datei als Import-Schnittstelle am besten, da in Java relativ einfach XML-Dateien erzeugt werden können.

Das Import-Tool von iET stellt eine DTD-Datei zur Verfügung (Listing 6.1), die den Aufbau der XML-Dateien beschreibt, die das Tool verarbeiten kann. So können unter dem Hauptknoten „root“ beliebige viele CI's aufgelistet werden. Von jedem CIs muss in immer Fall ein eindeutiger Schlüssel („key“) des Monitoring-Tools angegeben werden. Außerdem ist die Angabe des CI-Typs und des Produkts verpflichtend. Dabei kann es sich prinzipiell um beliebigen Text handeln, da die dort gemachten Angaben später im Programm unter der Applikation „CMDB Intelligence“ abgeglichen werden. Zudem kann jedes CI Komponeten und Attribute enthalten. Obwohl die DTD die Verwendung der Knoten „PARENT“ und „RELATION“ definiert, erlaubt der Import Server von iET dies in der aktuellen Version leider nicht. Somit lassen sich keine Beziehungen zwischen CIs automatisch importieren.

Listing 6.1: DTD für CMDB-Intelligence

```
<!--General-SMS-Import DTD Version 2 -->
<!ELEMENT root (CI*)>
  <!ELEMENT CI ( (PARENT? | RELATION*), ATTR* , COMP*) >
    <!ATTLIST CI  key CDATA     #REQUIRED>
    <!ATTLIST CI  ci_type CDATA     #REQUIRED>
    <!ATTLIST CI  product CDATA #REQUIRED>
    <!ATTLIST CI  account CDATA #IMPLIED>
    <!ATTLIST CI  contact CDATA #IMPLIED>
    <!ATTLIST CI  location  CDATA #IMPLIED>


  <!ELEMENT COMP (COMPATTR*)>
    <!ATTLIST COMP    key  CDATA #REQUIRED>
    <!ATTLIST COMP    category CDATA #IMPLIED>
    <!ATTLIST COMP    product  CDATA #IMPLIED>

  <!ELEMENT ATTR (#PCDATA)>
    <!ATTLIST ATTR  name  CDATA #REQUIRED>

  <!ELEMENT COMPATTR (#PCDATA)>
    <!ATTLIST COMPATTR  name      CDATA  #REQUIRED>

  <!ELEMENT PARENT (#PCDATA)>
    <!ATTLIST PARENT  relation_type CDATA #REQUIRED>

  <!ELEMENT RELATION  (#PCDATA)>
    <!ATTLIST RELATION  relation_type CDATA #REQUIRED>
    <!ATTLIST RELATION  key2 CDATA  #IMPLIED>
    <!ATTLIST RELATION  mac_address CDATA #IMPLIED>
    <!ATTLIST RELATION  port CDATA  #IMPLIED>
```

Der prinzipielle Aufbau des Java-Codes zur Erzeugung einer Liste von virtuellen Ma-

schinen mit deren jeweiligen Konfigurationsinformationen entspricht im Wesentlichen dem Code, der in Kapitel 6.3.1 verwendet wurde. Für den XML-Export werden die „Xerces“-Bibliotheken aus dem Apache XML Project [The05] verwendet. Der komplette Quellcode zur Erstellung von XML-Dateien, die mit Hilfe der CMDB Intelligence-Schnittstelle eingelesen werden können, befindet sich in Anhang E.1. Listing 6.2 zeigt ein Beispiel für den Import einer Maschine.

Listing 6.2: Beispiel-Import Datei

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE root SYSTEM "file:///C:/iET/CMDBIntelligence/iET.CMDB.Xml2InvSvr.dtd">
<root>
   <CI ci_type="Virtuelles Produkt" key="BADWLRZ-TWIET6" product="BADWLRZ-TWIET6">
         <ATTR name="Dienst-Nutzer">M. Gillmeister</ATTR>
         <ATTR name="Verwendungungszweck">IET-Test</ATTR>
         <ATTR name="RAM">1024</ATTR>
         <ATTR name="BS-Admin">PC-Gruppe</ATTR>
         <ATTR name="AnzahlCPUs">1</ATTR>
         <ATTR name="OS">Microsoft Windows Server 2003, Standard Edition (32-bit)</
             ATTR>
         <ATTR name="ResourcePool">LRZ-Test</ATTR>
         <ATTR name="Dienst-Admin">PC-Gruppe</ATTR>
    </CI>
</root>
```

7. Zusammenfassung, Fazit und Ausblick

Der Prozess des Configuration Managements zählt zu den wichtigsten bei der Einführung von IT-Servicemanagement. Er bildet die Grundlage für die anderen Prozesse und liefert mit der Configuration Management Database (CMDB) ein Werkzeug für den Mittelpunkt der prozessorientierten IT. Mit der Qualität der CMDB steht oder fällt die Qualität der Prozesse und damit die Qualität der erbrachten IT-Dienstleistung. Somit obliegt es dem Configuration Management, ein strukturiertes und vollständiges Informations- und Datenmodell mit den Prozess-Verantwortlichen zu erarbeiten, dass allen Ansprüchen genügt. Auf der anderen Seite sollte es auch nicht zu viele Details und Modellierungen enthalten, die später nicht gebraucht und somit nicht gepflegt werden. Zudem muss sich Configuration Management um Schnittstellen zu bestehenden Tools und Datenquellen, sogenannten Management Data Repositories oder Daten-Silos, kümmern.

In Kapitel 2 wurden die Grundlagen des Configuration Managements diskutiert. Anhand des Deming-Kreislaufs Plan, Do, Check, Act wurde das prinzipielle Vorgehen zum Aufbau und Betrieb einer CMDB erläutert. Zudem wurden Qualitäts- und Reifegradstufen des Configuration Managements vorgestellt. Schließlich wurde auf den Betrieb mit verteilten CMDBs eingegangen und die Architektur des CMDBf-Normvorschlags analysiert.

Wie der Aufbau des Configuration Managements in der Praxis ablaufen kann, wurde im zweiten Teil dieser Arbeit am Beispiel des Leibniz-Rechenzentrums (LRZ) gezeigt. Dazu wurden zunächst drei ausgewählte Verfahren analysiert (Kapitel 3) und anschließend die beteiligten Datenquellen in Kapitel 4 auf ihre Koexistenz mit einer CMDB untersucht. Hierfür wurde ein Bewertungsschema entwickelt, um Aussagen über Migrations- oder Integrationstauglichkeit zu treffen. Für jedes untersuchte potenzielle MDR wird anhand eines Kriterienkatalogs ein Integrations- und Migrationswert errechnet. Das Verhältnis der Werte zueinander ergibt schließlich eine Empfehlung für das weitere Vorgehen.

Auf Basis aller erhobenen Daten wurde in Abschnitt 5.1 in mehreren Phasen ein Informationsmodell erzeugt, dass das analysierte Szenario am LRZ abbildet. Wie so oft geben erstellte Modelle Anlass zu Diskussionen und Verbesserungsvorschlägen. Das hier erstellte Informationsmodell erhebt nicht den Anspruch vollständig für das gesamte LRZ zu sein. Es stellt aber eine ausreichend gute und detaillierte Grundlage dar, um das Vorhaben am LRZ, eine gesamtheitliche CMDB einzuführen, unterstützt. Während der Entwicklung wurde das Modell so gestaltet, dass Erweiterungen jederzeit schnell und mit geringem Aufwand möglich sind.

Nach einer Untersuchung der zukünftigen ITSM-Software „iET ITSM“ (Abschnitt 5.2) am LRZ wurde die Umsetzung des Informationsmodells innerhalb der Software aufgezeigt und ein Datenmodell-Konzept erstellt (Abschnitt 5.4). Abschließend wurde dieses Datenmodell prototypisch umgesetzt und eine Anbindung der „LRZ Netzdoku“ sowie der „VMware Infrastructure“ durchgeführt (Kapitel 6).

Dabei hat sich gezeigt, dass man ITSM-Tools (egal von welchem Hersteller) immer an das jeweilige Unternehmen anpassen muss, da das vom Hersteller vorgesehene CMDB-Datenmodell selten die Bedürfnisse des Unternehmens vollständig erfüllt. Somit ist es wünschenswert,

im Datenmodell aber auch in den entsprechenden Formularen Anpassungen vornehmen zu können. Dies sollten optimalerweise die hauseigenen Entwickler des Unternehmens durchführen können, da diese als interne Mitarbeiter ein gutes Verständnis für die Prozesse besitzen. Für die Entwicklung und Anpassung der ITSM-Software ist entscheidend, wie gut der Hersteller Einblick in sein CMDB-Datenmodell gewährt. Im Falle der am LRZ eingesetzten Software „iET ITSM“ wurde dies hervorragend gelöst. So stehen einem Entwickler alle Möglichkeiten offen, das System beliebig zu erweitern und anzupassen, wie es im Kapitel 6 gezeigt wurde.

Eine Einschränkung besitzen allerdings die viele großen ITSM-Tools. Die Hersteller sehen ihre CMDB meist als den einzigen Ort der Wahrheit an („single source of truth“). Das hat zur Folge, dass es in der Regel keinen Export-Mechanismus von CMDB-Daten gibt. Allerdings vergessen sie dabei, dass es durchaus wünschenswert wäre, Daten auch in die Gegenrichtung zu synchronisieren, da es immer Software geben wird, die weiterhin im Einsatz bleibt. Jeder Hersteller entwickelt stattdessen ausschließlich Lösungen, um Informationen verschiedener Datenquellen einzulesen. Der im Entwurf befindliche Standard „CMDBf“ [CMD08] zum Aufbau föderierter Datenbasen, der in Abschnitt 2.4.2 diskutiert wurde, könnte hier Abhilfe schaffen. Wann und ob der Standard jemals kommen wird, ist fraglich. Nach eigener Einschätzung wird es in den kommenden Jahren keinen gemeinsamen Standard für föderierte CMDBs gemäß den Plänen der CMDBf Workgroup geben. Sollte der Entwurf der Arbeitsgruppe zum Standard erklärt werden, ist dieser in der vorliegenden Version derart ungenau formuliert, dass letztenendes jeder Hersteller wieder eine eigene Implementierung entwickeln könnte. Diese wären nur sehr begrenzt kompatibel zueinander. Unter Umständen wäre es vielleicht möglich, dass einige ITSM-Software-Hersteller Informationen untereinander austauschen können, aber in aller Regel wird ein Unternehmen exakt ein Tool eines Herstellers einsetzen. Viel wichtiger ist die Synchronisation von Informationen mit Datenquellen wie z.B. Office-Dokumente, Monitoring-Systeme (HP Openview, Nagios) und Asset-Management-Software, die als permanenter Datenlieferant für die CMDB in Frage kommen. Für diese müssten eigene CMDBf-Adapter und Konnektoren entwickelt werden, um einen Austausch mittels SOAP-Schnittstelle zu gewährleisten. Allerdings scheinen nur wenige Hersteller Interesse an dem CMDBf-Konzept zu haben.

Obwohl die Vorgehensweise in dieser Arbeit auf das LRZ ausgerichtet war, ist es dennoch möglich, das grundsätzliche Vorgehen zum Aufbau eines Daten- und Informationsmodells für das Configuration Management auf andere Unternehmen zu übertragen, da es sich um ein generisches Vorgehen handelt.

Diese Arbeit bietet zahlreiche Möglichkeiten daran anzuknüpfen. Während der intensiven Tests der Software zusammen mit dem ITSM-Arbeitskreis am LRZ haben sich zahlreiche Ideen für die Weiterentwicklung der Unternehmensprozesse ergeben, die aufbauend auf dieser Arbeit umgesetzt werden können. So wird derzeit an der vollständigen Anbindung von iET an „VMware Infrastructure“ entwickelt. Dies ermöglicht den kompletten Lifecycle einer virtuellen Maschine innerhalb von iET zu managen. Kunden können über das Self-Service-Portal neue virtuelle Maschinen bestellen. Nach der Bestellfreigabe kann die Maschine anschließend direkt via Java-API erzeugt und eingerichtet werden. Auch der Prozess der Software-Lizenzbestellung von Kunden könnte über das Self-Service-Portal abgewickelt und in iET integriert werden. Durch das offene Datenmodell und die Java API für eigene Entwicklungen stehen prinzipiell alle Möglichkeiten offen. Dadurch ergeben sich vollständig integrierte IT-Servicemanagement Prozesse am LRZ.

A. CMDBf QueryService-Beispiel

Mit Hilfe der Anfrage A.1 an ein MDR werden alle Computer abgefragt, die von Personen benutzt werden, die dem Standort „Garching“ zugeordnet sind. Dazu werden die itemTemplates „user“ und „computer“ angefragt. Für „user“ wird die Einschränkung nach dem Ort mit angegeben. Beide itemTemplates sind über ein relationshipTemplate („usage“) miteinander verbunden. Als Ergebnis werden nun Benutzer mit ihren jeweiligen Computern zurückgegeben. Sollte ein Benuzter am Standort keinen PC nutzen, so taucht er in der Antwort nicht auf.

Listing A.1: Anfrage

```
<?xml version="1.0" encoding="utf-8"?>

<query>

  <itemTemplate id="user">
    <recordConstraint>
      <recordType namespace="http://example.com/people" localName="person"/>
      <propertyValue namespace="http://example.com/people" localName="city">
        <equal>Garching</equal>
      </propertyValue>
    </recordConstraint>
  </itemTemplate>

  <itemTemplate id="computer">
    <recordConstraint>
      <recordType namespace="http://example.com/computer" localName="computer"/>
    </recordConstraint>
  </itemTemplate>

  <relationshipTemplate id="usage">
    <recordConstraint>
      <recordType namespace="http://example.com/computer" localName="uses"/>
    </recordConstraint>

    <sourceTemplate ref="user"/>
    <targetTemplate ref="computer"/>
  </relationshipTemplate>

</query>
```

B. Skript: Import von Stammdaten aus der LRZ Netzdoku

Listing B.1: Import-Skript

```
<?PHP

// Verbindungsaufbau iET ITSM-Datenbank (MS SQL)
$serverName = "(local)";
$connectionInfo = array(  "UID"=>"XXX",
                          "PWD"=>"XXX",
                          "Database"=>"itsm");
$db_itsm = sqlsrv_connect($serverName, $connectionInfo);
if($db_itsm === false) {
  echo "Kann nicht mit ITSM-DB verbinden";
  die( print_r( sqlsrv_errors(), true));
}

//Verbindungsaufbau LRZ Netzdoku (Oracle)
$db_nd = oci_connect("XXX", "XXX", "XXXX/oracle9i");
if($db_nd === false) {
  echo "Kann nicht mit Netzdoku verbinden.</br>"; die();
}

//*****************************************************************************
//Organisation importieren
$s = oci_parse($db_nd, 'select distinct organisation from netzdoku.institut');
oci_execute($s);
while ($row = oci_fetch_array($s, OCI_NUM+OCI_RETURN_NULLS)) {
  $c0 = "SELECT CASE WHEN MAX(account_id) IS NULL THEN '1' ELSE MAX(account_id)+1 END
       FROM dbo.account";
  $stmt = sqlsrv_query( $db_itsm, $c0);
  sqlsrv_fetch($stmt);

  $in = "INSERT INTO dbo.account (account_id,name,status,type)
  VALUES (" . sqlsrv_get_field($stmt, 0) . ", '" . $row[0] . "','Kunde','Firma')";

  $stmt = sqlsrv_query( $db_itsm, $in);
  if( $stmt === false ) {
    echo "Error in Query.<br> " . $in . "<br>";
    die( print_r( sqlsrv_errors(), true));
  }
}

//*****************************************************************************
//Kunden importieren
$s = oci_parse($db_nd, 'select * from netzdoku.institut');
oci_execute($s);
while ($row = oci_fetch_array($s, OCI_RETURN_NULLS)) {

  //ID von übergeordneter Organisation holen
  $q = "SELECT account_id FROM dbo.account WHERE name='" . $row['ORGANISATION'] . "'"
      ;
  $sql_parent = sqlsrv_query( $db_itsm, $q);
  sqlsrv_fetch($sql_parent);
  $parent = sqlsrv_get_field($sql_parent,0);
```

```
//IDs für die entsprechenden Tabellen holen
$q = "SELECT CASE WHEN MAX(account_id) IS NULL THEN '1' ELSE MAX(account_id)+1 END
    FROM dbo.account";
$sql_account = sqlsrv_query( $db_itsm, $q);
sqlsrv_fetch($sql_account);
$max_account=sqlsrv_get_field($sql_account,0);

$q = "SELECT CASE WHEN MAX(acct_location_id) IS NULL THEN '1' ELSE MAX(
    acct_location_id)+1 END FROM dbo.account_location";
$sql_account_location = sqlsrv_query( $db_itsm, $q);
sqlsrv_fetch($sql_account_location);
$max_account_location=sqlsrv_get_field($sql_account_location,0);

$q = "SELECT CASE WHEN MAX(account_contact_id) IS NULL THEN '1' ELSE MAX(
    account_contact_id)+1 END FROM dbo.account_contact";
$sql_account_contact = sqlsrv_query( $db_itsm, $q);
sqlsrv_fetch($sql_account_contact);
$max_account_contact=sqlsrv_get_field($sql_account_contact,0);

//Query1: Kundenanlage
$sql1 = "INSERT INTO dbo.account (account_id,name,parent_id,phone,status,type,
    institutskennung)
VALUES(" . $max_account . ",'" . str_replace("'", "", $row['NAME']) . "',"
    $parent . ",'" . $row['TELEFON'] . "','Kunde','Firma','" . $row['ID'] . "')";

// Aus Ortsfeld PLZ,Ort,Land extrahieren
if (is_numeric(substr($row['ORT'], 0, 5)) ) {
  $plz = substr($row['ORT'], 0, 5);
  $ort = substr($row['ORT'], 5);
  $land="GERMANY";
  $landcode=276;
} else {
  $plz=NULL;
  $ort=$row['ORT'];
  $land=NULL;
  $landcode='NULL';
}

//Query2: Standortanlage
$sql2 = "INSERT INTO dbo.account_location
(acct_location_id,account_id,address1,city,country,country_id,it_location_desc,
    postal_code_id,type)
VALUES (
" . $max_account_location . "," . $max_account . ",'" . $row['STRASSE'] . "','" .
    $ort . "','" . $land . "'," . $landcode . ",'Hauptsitz','" . $plz . "','
    Unternehmen')";

//Queries 1 und 2 ausführen
$stmt = sqlsrv_query( $db_itsm, $sql1);
if( $stmt === false ) { echo "Error in Query.<br>" . $sql1 . "<br>"; die( print_r(
    sqlsrv_errors(), true)); }

$stmt = sqlsrv_query( $db_itsm, $sql2);
if( $stmt === false ) { echo "Error in Query.<br>" . $sql2 . "<br>"; die( print_r(
    sqlsrv_errors(), true)); }

//falls Masteruser angegeben, kann dieser eingetragen werden
if ($row['MASTERUSER']!="") {
  $anrede="";$first="";$last="";$title="";

  //aus Feld Masteruser werden Anrede,Titel,Vorname und Nachname extrahiert
  $teil = explode(" ", $row['MASTERUSER']);
  $anrede=$teil[0];

  $counter=0;
  for ($i=1;$i<count($teil);$i++) {
```

```
      if ($teil[$i]=="Dr." || $teil[$i]=="Prof." || $teil[$i]=="Dr.-Ing." || $teil[$i
          ]=="habil." || $teil[$i]=="PH") { $counter=$i; $title .= $teil[$i] . " "; }
    }
    for ($i=$counter+1;$i<count($teil)-1;$i++) {
      $first .= $teil[$i] . " ";
    }
    $last=$teil[count($teil)-1];
    $first=trim($first);
    $last=trim($last);
    $title=trim($title);

    //Query 3: Kontaktanlage von Masteruser
    $sql3 = "INSERT INTO dbo.account_contact
    (account_contact_id,account_id,acct_location_id,active,phone,first_name,last_name
        ,salutation,title,type)
    VALUES (
    " . $max_account_contact . "," . $max_account . "," . $max_account_location . ",'
        Y','" . $row['TELEFON'] . "','" . $first . "','" . $last . "','" . $anrede .
        "','" . $title . "','Entscheider')";

    //Query 3 ausführen
    $stmt = sqlsrv_query( $db_itsm, $sql3);
    if( $stmt === false ) { echo "Error in Query.<br>" . $sql3 . "<br>"; die( print_r
        ( sqlsrv_errors(), true)); }
  }


  //Weitere Kontakte importieren
  $t = oci_parse($db_nd, "select * from netzdoku.person where institut_id='" . $row['
      ID'] . "'");
  oci_execute($t);
  while ($row2 = oci_fetch_array($t, OCI_RETURN_NULLS)) {

    $q = "SELECT CASE WHEN MAX(account_contact_id) IS NULL THEN '1' ELSE MAX(
        account_contact_id)+1 END FROM dbo.account_contact";
    $sql_account_contact = sqlsrv_query( $db_itsm, $q);
    sqlsrv_fetch($sql_account_contact);
    $max_account_contact=sqlsrv_get_field($sql_account_contact,0);

    $sql4 = "INSERT INTO dbo.account_contact
    (account_contact_id,account_id,acct_location_id,active,cell_phone,email,phone,
    home_phone,first_name,last_name,salutation,title,type) VALUES(
    " . $max_account_contact . "," . $max_account . "," . $max_account_location .
    ",'Y','" . $row2['MOBIL'] . "','" . $row2['EMAIL'] . "','" . $row2['TELEFON'] . "
        ',
    '" . $row2['TELEFON2'] . "','" . $row2['VORNAME'] . "','" . $row2['NACHNAME'] . "
        ',
    '" . $row2['ANREDE'] . "','" . $row2['TITEL'] . "','Primärer Anwender')";

    $stmt = sqlsrv_query( $db_itsm, $sql4);
    if( $stmt === false ) { echo "Error in Query.<br>" . $sql4 . "<br>"; die( print_r
        ( sqlsrv_errors(), true)); }

  }

}



?>
```

C. Live-Zugriff auf vCenter innerhalb von iET

Listing C.1: xxx

```
import java.net.URL;
import com.vmware.vim25.*;
import com.vmware.vim25.mo.*;
import com.vmware.vim25.mo.util.*;

public class vmware1 extends EntCtrlBtn {

    int OnChange() {
        String urlStr = "https://xxxxx/sdk";
        String username = "xxxx";
        String password = "xxxx";

        int i, j;
        Integer countVM=0;

        EntCtrl resspool;
        EntCtrl number;
        EntForm formObj;

        try {
            //vCenter Connection
            ServiceInstance si = new ServiceInstance(new URL(urlStr), username,
                password, true);
            Folder rootFolder = si.getRootFolder();

            //Cluster auslesen
            ManagedEntity[] me = new InventoryNavigator(rootFolder).
                searchManagedEntities("ComputeResource");
            for (i = 0; i < me.length; i++) {
                if (me[i].getName().equals("ESX-Testcluster-1")) {
                    // Im Falle des Testclusters alle Resource-Pools durchlaufen
                    // und die virtuellen Maschinen zählen
                    ResourcePool[] rp = ((ComputeResource) me[i]).getResourcePool().
                        getResourcePools();
                    for(j=0; j< rp.length;j++) {
                        countVM += rp[j].getVMs().length;
                    }
                }
            }
            si.getServerConnection().logout();

        } catch (Exception e) {return 0;}

        //Ergebnis in zwei Formularfelder eintragen
        formObj = this.m_Form;
        resspool = formObj.GetCtrlByTableColumn("temp.ex_str64_1", EntCtrl.
            TEXT_CONTROL);
        number = formObj.GetCtrlByTableColumn("temp.ex_str64_2", EntCtrl.TEXT_CONTROL
            );
        resspool.setValue("LRZ-Testcluster");
        number.setValue(countVM.toString());
        return 0;
    }
}
```

D. Live-Zugriff auf vCenter zur Abfrage aller VMs

Listing D.1: xxx

```
import java.net.URL;
import com.vmware.vim25.*;
import com.vmware.vim25.mo.*;
import com.vmware.vim25.mo.util.*;

public class vmware2 extends EntCtrlBtn {

    int OnChange() {
        String urlStr = "https://xxxx/sdk";
        String username = "xxx";
        String password = "xxx";

        int i, j, k;
        int countVM = 0;
        int r = 0;

        EntData ed;
        EntData row;
        EntCtrlGrid xy;
        ed = EntData.NewArray(1);

        xy = m_Form.getGridCtrl("LRZ VM1TEST");
        if (xy == null) {
            return 0;
        }


        VirtualMachine vm;
        VirtualMachineConfigInfo vminfo;
        VirtualMachineCapability vmc;

        try {
            //vCenter Connection
            ServiceInstance si = new ServiceInstance(new URL(urlStr), username,
                password, true);
            Folder rootFolder = si.getRootFolder();

            //Cluster auslesen
            ManagedEntity[] me = new InventoryNavigator(rootFolder).
                searchManagedEntities("ComputeResource");
            for (i = 0; i < me.length; i++) {
                if (me[i].getName().equals("ESX-Testcluster-1")) {
                    ResourcePool[] rp = ((ComputeResource) me[i]).getResourcePool().
                        getResourcePools();
                    for (j = 0; j < rp.length; j++) {
                        countVM = rp[j].getVMs().length;
                        for (k = 0; k < countVM; k++) {
                            //Informationen jeder virtuellen Maschine holen und dem
                                Grid hinzufügen
                            vm = rp[j].getVMs()[k];
                            vminfo = vm.getConfig();
                            vmc = vm.getCapability();
```

```
                                row = new EntData();
                                row = EntData.NewArray(1);
                                row.AddDataToArray(0, vm.getName());
                                row.AddDataToArray(1, ((Integer) vminfo.getHardware().
                                    getNumCPU()).toString());
                                row.AddDataToArray(2, ((Integer) vminfo.getHardware().
                                    getMemoryMB()).toString());
                                row.AddDataToArray(3, vminfo.getGuestFullName());
                                ed.AddDataToArray(r, row);
                                r++;
                            }
                        }
                    }
                }
                si.getServerConnection().logout();

            } catch (Exception e) { return 0; }

            xy.AsyncSetGridDisplay(ed);
            return 0;
        }
}
```

E. Programm: Import virtueller Maschinen über CMDB Intelligence

Listing E.1: Import-Skript

```
import java.net.URL;
import java.util.*;
import java.io.*;

//VM Infrastructure Java API
//http://vijava.sourceforge.net/
import com.vmware.vim25.*;
import com.vmware.vim25.mo.*;
import com.vmware.vim25.mo.util.*;

// Xerces classes.
//http://xerces.apache.org/xerces-j/
import org.apache.xerces.dom.DocumentImpl;
import org.apache.xml.serialize.*;
import org.w3c.dom.*;

public class Main {

    public static void main(String[] args) throws Exception {
        String urlStr = "xxx";
        String username = "xxx";
        String password = "xxx";

        int i, j, k, l, countVM;
        VirtualMachine vm;
        VirtualMachineConfigInfo vminfo;
        VirtualMachineCapability vmc;
        CustomFieldStringValue cfsv;

        // Document (Xerces implementation only).
        Document xmldoc = new DocumentImpl();
        Element root = xmldoc.createElement("root");

        //vCenter Connection
        ServiceInstance si = new ServiceInstance(new URL(urlStr), username, password,
            true);
        Folder rootFolder = si.getRootFolder();

        //Cluster auslesen
        ManagedEntity[] me = new InventoryNavigator(rootFolder).searchManagedEntities
            ("ComputeResource");

        for (i = 0; i < me.length; i++) {
            //für jeden Cluster ResourcePools holen
            ComputeResource cr = (ComputeResource) me[i];

            //ResourcePools
            ResourcePool[] rp = cr.getResourcePool().getResourcePools();
            for (j = 0; j < rp.length; j++) {
                //für jeden ResourcePool die VMs auslesen
                countVM = rp[j].getVMs().length;
                for (k = 0; k < countVM; k++) {
```

```
                    vm = rp[j].getVMs()[k];
                    vminfo = vm.getConfig();
                    vmc = vm.getCapability();

                    //Informationen holen und XML Node erstellen
                    Element ci = xmldoc.createElement("CI");
                    ci.setAttribute("key", vm.getName());
                    ci.setAttribute("ci_type", "Virtuelles Produkt");
                    ci.setAttribute("product", vm.getName());

                    HashMap<String, String> att = new HashMap<String, String>();
                    att.put("ResourcePool", rp[j].getName());
                    att.put("OS", vminfo.getGuestFullName());
                    att.put("AnzahlCPUs", ((Integer) vminfo.getHardware().getNumCPU()
                        ).toString());
                    att.put("RAM", ((Integer) vminfo.getHardware().memoryMB).toString
                        ());
                    try {
                        for (l = 0; l < vm.getCustomValue().length; l++) {
                            cfsv = (CustomFieldStringValue) vm.getCustomValue()[l];
                            switch (cfsv.key) {
                                case 9:
                                    att.put("BS-Admin", cfsv.value);
                                    break;
                                case 10:
                                    att.put("Dienst-Nutzer", cfsv.value);
                                    break;
                                case 11:
                                    att.put("Verwendungungszweck", cfsv.value);
                                    break;
                                case 13:
                                    att.put("Dienst-Admin", cfsv.value);
                                    break;
                            }
                        }
                    } catch (Exception e) {}

                    Set entries = att.entrySet();
                    Iterator it = entries.iterator();
                    while (it.hasNext()) {
                        Map.Entry entry = (Map.Entry) it.next();
                        Element attr = xmldoc.createElement("ATTR");
                        attr.setAttribute("name", (String) entry.getKey());
                        attr.appendChild(xmldoc.createTextNode((String) entry.
                            getValue()));
                        ci.appendChild(attr);
                    }

                    root.appendChild(ci);
                }
            }
        }
        xmldoc.appendChild(root);

        FileOutputStream fos = new FileOutputStream("C:\\iET\\CMDBIntelligence\\
            Import\\vm.xml");
        OutputFormat of = new OutputFormat("XML", "ISO-8859-1", true);
        of.setIndent(1);
        of.setIndenting(true);
        of.setDoctype(null, "file:///C:/iET/CMDBIntelligence/iET.CMDB.Xml2InvSvr.dtd"
            );
        XMLSerializer serializer = new XMLSerializer(fos, of);
        serializer.asDOMSerializer();
        serializer.serialize(xmldoc.getDocumentElement());
        fos.close();
```

```
            si.getServerConnection().logout();
        }
    }
```

Abbildungsverzeichnis

Abkürzungsverzeichnis

API Application Programming Interface

ARS Action Request System

CI Configuration Item

CMDB Configuration Management Database

CMDBf Federated Configuration Management Database

CobiT Control Objectives for Information and Related Technology

DTD Document Type Definition

IIS Internet Information Server

ITIL IT Infrastructure Library

ITSM IT-Servicemanagement

LRZ Leibniz-Rechenzentrum

MDR Management Data Repositories

MOF Microsoft Operations Framework

MWN Münchner Wissenschaftsnetz

ODBC Open Database Connectivity

OLA Operational Level Agreement

OSI Open Systems Interconnection

PDU Power Distribution Unit

RfC Request for Change

SDK Software Development Kit

SLA Service-Level Agreement

SOAP Simple Object Access Protocol

SSID Service Set Identifier

SAN Storage Area Network

UC Underpinning contract

XML Extensible Markup Language

W3C World Wide Web Consortium

WSDL Web Services Description Language

Literaturverzeichnis

[AC07] Adams, Patricia und Ronni J. Colville: *Is a CMDB Standard on the Horizon?* Technischer Bericht, Gartner, November 2007. http://www.gartner.com/DisplayDocument?doc_cd=152615&ref=g_rss.

[AH07] Apostolescu, Victor und Mandana Hassanloo: *Das Dokuticket - Dokumentation von Hardware am LRZ*, Februar 2007.

[And09] Andresen, Kai: *ITIL, Projektmanagement & Qualifizierungen.* IT Service Management Forum (itSMF) Deutschland e.V., Februar 2009.

[BD07] Brügge, Bernd und Allen H. Dutoit: *Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java.* Pearson, 2007.

[BEK+00] Box, Don, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte und Dave Winer: *Simple Object Access Protocol (SOAP) 1.1.* World Wide Web Consortium (W3C), Mai 2000. http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

[Ber09] Berner, Stefan: *UML-Modell der virtuellen Infrastruktur*, 2009.

[BR09] Biardzki, Christoph und Martin Roll: *Einrichten einer neuen virtuellen Maschine mit Linux-Betriebssystem*, 2009.

[Bre07] Brenner, Michael: *Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement: Ein modellbasierter Ansatz.* Doktorarbeit, 2007. http://www.nm.ifi.lmu.de/~brennera/files/presentation_2007-07-27.pdf.

[BT09] Brauer, Steffi und Romano Tesone: *ITIL v3 und der Service Catalog von CA*, 2009. http://ca.com/Files/Presentations/ie09_itsm04_208013.pdf.

[CB08] Coyle, David M. und Kris Brittain: *Magic Quadrant for the IT Service Desk.* Technischer Bericht, Gartner, November 2008.

[CCMW01] Christensen, Erik, Francisco Curbera, Greg Meredith und Sanjiva Weerawarana: *Web Services Description Language (WSDL) 1.1.* World Wide Web Consortium (W3C), März 2001. http://www.w3.org/TR/wsdl.html.

[CMD08] CMDB Federation Workgroup: *CMDB Federation (CMDBf) Version 1.0b*, Januar 2008. http://cmdbf.org/schema/1-0-0/CMDBf_v1.0b.pdf.

[cmm09a] *CMMI for Services, Version 1.2*, Februar 2009. http://www.sei.cmu.edu/pub/documents/09.reports/09tr001.pdf.

[cmm09b] *Published Appraisal Results*, 2009. http://sas.sei.cmu.edu/pars/pars.aspx.

[cmt] *Configuration Management Toolkit*. http://www.configurationkit.com.

[Col06] Colville, Ronni J.: *CMDB or Configuration Database: Know the Difference*. Technischer Bericht, Gartner, Juni 2006.

[Det08] Dettmer, Klaus: *In 5 Schritten zur CMDB - Projekt-Leitfaden zum Aufbau einer Configuration Management Database*, 2008. http://tinyurl.com/5steps-cmdb.

[DLM+09] Doherty, Peter, Randal Locke, Ram Melkote, David Messineo und Marv Waschke: *The Value of Standards-based CMDB Federation*. CA, April 2009. http://www.ca.com/files/WhitePapers/031909-cmdb-fed-wp-final_204178.pdf.

[Eng09a] England, Rob: *CA CMDBf paper oversells the CMDB Federation standard, as predicted*, Mai 2009. http://www.itskeptic.org/ca-cmdbf-paper-oversells-cmdb-federation-standard-.

[Eng09b] England, Rob: *The CMDB Federation is a brilliant piece of vendor marketing smokescreen*, Mai 2009. http://www.itskeptic.org/cmdb-federation-brilliant-piece-vendor-marketing-s.

[iET08] iET Solutions: *iET Enterprise Application Programming Interface*, 2008. Version #11.1.

[Inf07] Information Systems Audit and Control Association (ISACA): *Cobit 4.1*, 2007. www.isaca.org/cobit.

[ISO05a] ISO/IEC: *Information Technology - Service Management - Part 1: Specification*, Dezember 2005. International Standard ISO/IEC 20000-1:2005.

[ISO05b] ISO/IEC: *Information Technology - Service Management - Part 2: Code of Practice*, Dezember 2005. International Standard ISO/IEC 20000-1:2005.

[Jak08] Jakobs, Dr Ing Roger H.: *Advanced CMDB*, 2008. http://www.comconsult.de/cckt_pub/advanced_cmdb.pdf.

[Jin08] Jin, Steve: *Object model of VI Java API*, 2008. https://vijava.svn.sourceforge.net/svnroot/vijava/trunk/docs/Object%20model%20of%20VI%20Java%20API.pdf.

[Jin09] Jin, Steve: *VMware VI (vSphere) Java API*, 2009. http://vijava.sourceforge.net/.

[KE09] Kemper, Alfons und Andre Eickler: *Datenbanksysteme. Eine Einführung*. Oldenbourg, 2009.

[Kem09] Kempter, Andrea Dr.: *Service Catalogue Management*, 2009. http://wiki.de.it-processmaps.com/index.php/Service_Catalogue_Management.

[Lei09] Leibniz Rechenzentrum: *Das LRZ in Kürze*, Juni 2009. http://www.lrz-muenchen.de/wir/lrz-flyer/de/index.html.

[MB08] Meisel, Alexander und Ralf Buchsein: *IT-Compliance: Die praktische Umsetzung von Compliance-Anforderungen mit Hilfe von Best Practices.* iET Solutions GmbH, 2008.

[Mic08] Microsoft: *Microsoft Operations Framework 4.0*, 2008. `http://www.microsoft.com/MOF`.

[OGC05] OGC (Office of Government Commerce): *Introduction to ITIL. IT Infrastructure Library. The Stationary Office*, 2005.

[OGC07] OGC (Office of Government Commerce): *The Official Introduction to the ITIL Service Lifecycle. IT Infrastructure Library. The Stationary Office*, 2007.

[Ric09] Richter, Christian: *Remedy Nutzungsstatistik*, 2009.

[Rol08] Roll, Martin: *VMware Gesamtstruktur*, 2008.

[Ros08a] Roselieb, Christian: *CMDB Anforderungen - Anforderungen an das Configuration Management und die CMDB.* santix AG, Oktober 2008.

[Ros08b] Roselieb, Christian: *CMDB im Projekt - Erfahrungen bei der CMDB-Realisierung aus verschiedenen santix-Projekten.* santix AG, Oktober 2008.

[RSW+09] Runge, Roland, Christian Sturm, Stefan Wisskirchen, Nadin Ebel, Joachim Groh, Oliver Höller und Carsten Mewes: *VMware Infrastructure 3 im Business Umfeld.* Addison-Wesley, 2009.

[Sch08] Schaal, Johannes: *Entwicklung eines Werkzeugkonzepts zur Unterstützung des Change-Managements gemäß ISO/IEC 20000.* Diplomarbeit, Dezember 2008.

[Sur08] Surholt, Dirk: *Mit dem IT-Kunden in einem Boot - IT-Services für zufriedene Kunden*, 2008. `http://www.helbling.de/hol/publikationen/prozessoptimierung-mit-dem-kunden-in-einem-boot`.

[Tag05] Tague, Nancy R.: *The Quality Toolbox.* Amer Society for Quality, 2005.

[The05] The Apache Software Foundation: *Apache Xerces Project*, 2005. `http://xerces.apache.org/xerces-j/`.

[Win08] Winkelmann, Axel: *Stammdaten in der Warenwirtschaft*, 2008. `http://tinyurl.com/stammdaten`.