



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Unterstützungssystem zur Lizenzverwaltung“

verfasst von / submitted by

Michael Kirschberger Bakk.rer.soc.oec.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Diplom-Ingenieur (Dipl.-Ing.)

Wien, 2018 / Vienna, 2018

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 066 926

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Master Wirtschaftsinformatik (UG2002)

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Dr. Gerald Quirchmayr

Danksagung

Ich möchte mich bei meinen Eltern, meiner restlichen Familie, sowie meinen Freunden, Studien- und Arbeitskollegen bedanken, welche mich auf meinem Weg durchs Studium begleitet und unterstützt haben.

Des Weiteren bedanke ich mich bei meinem Betreuer Univ.-Prof. Dipl.-Ing. Dr. Dr. Gerald Quirchmayr, der es mir nach einer längeren arbeitsbedingten Pause ermöglicht hat, meine Masterarbeit bei ihm abzuschließen und mich mit wertvollen Ratschlägen während meines gesamten Studiums unterstützt hat.

Abschließend möchte ich mich noch bei der Firma DBConcepts GmbH bedanken, die mir den Anstoß zu dieser Arbeit lieferte und es mir zeitlich ermöglichte, diese umzusetzen.

Inhalt

1. Einleitung.....	9
2. Literaturanalyse	11
2.1 Grundlagen.....	11
2.1.1 Lizenzformen.....	12
2.1.2 Lizenzarten	15
2.1.3 Lizenzklassen	16
2.1.4 Lizenztypen und Metriken	17
2.1.5 Kaufmännische Einteilung der Lizenzen	18
2.1.6 Neue Lizenzmodelle	19
2.2 Lizenzverwaltungsansätze und Problemstellen	20
2.2.1 Nutzen von Lizenzverwaltung.....	21
2.2.2 Lizenz Life Cycle.....	24
2.2.3 Phasen der Lizenz.....	26
2.2.4 Überlizenzierung	28
2.2.5 Unterlizenzierung.....	28
2.2.6 Probleme und Folgen von fehlender Lizenzverwaltung	29
2.3 Rechtliche Sicht.....	31
2.3.1 Vervielfältigungsrecht	33
2.3.2 Verbreitungsrecht	33
2.3.3 Bestimmungsmäßige Benutzung.....	34
2.3.4 Sicherungskopie.....	35
2.3.5 Recht zum Beobachten, Untersuchen und Testen des Funktionierens	35
2.3.6 Recht auf Dekompilieren	35
2.3.7 Vervielfältigung zum eigenen Gebrauch.....	35
2.3.8 Sanktionen.....	36
2.3.9 Schutzdauer	37

2.4	Rahmenbedingung für Lizenzverwaltung	37
2.5	Systeme	40
2.5.1	Inbetriebnahme.....	40
2.5.2	Funktionsumfang.....	41
3.	Ausgangssituation / Use Case	43
3.1	Das Unternehmen.....	43
3.2	Aufgaben des Unternehmens.....	43
3.2.1	Datenbank- und Systemadministration.....	43
3.2.2	Entwicklung von Individualsoftware und Consultant.....	44
3.2.3	Cloud-Lösungen	44
3.2.4	Vertrieb von Hardware	44
3.2.5	Vertrieb von Lizenzen.....	45
3.2.6	Weitere Tätigkeitsfelder	45
3.3	Aktuelles System.....	45
3.4	Use Case für das neue Unterstützungstool.....	46
4.	Anforderungsdefinition	51
5.	Modell	55
5.1	Oracle Application Express - Framework.....	55
5.2	Ablaufmodelle.....	60
5.2.1	Übersichtsseite Zusammenfassung	60
5.2.2	Kunden und User verwalten.....	61
5.2.3	Infrastruktur verwalten	63
5.2.4	Software verwalten.....	65
5.2.5	Lizenzen verwalten	66
5.2.6	Lizenzverwendung verwalten	69
5.2.7	Softwareanalyse	71
5.2.8	Anforderung erstellen.....	72

5.2.9	System-Scan.....	73
5.3	Datenmodell	75
5.3.1	Kundenbereich	78
5.3.2	Infrastrukturbereich	87
5.3.3	Softwarebereich.....	91
5.3.4	Lizenzbereich	93
5.3.5	Bereich für dynamische Bedingungen.....	100
5.3.6	Loggingbereich	103
6.	Implementierung	105
6.1	Anforderungen an den Prototypen	105
6.2	Abgrenzungen.....	107
6.3	Umsetzungen der GUI (Graphical User Interface)	108
6.4	Codeumsetzung.....	114
6.4.1	Hilfsskripte und Entwicklungsrichtlinien	114
6.4.2	Ergebnisermittlung der Lizenzanforderungen	117
6.5	Überarbeitungen.....	125
7.	Funktionaler Test und Diskussion der Ergebnisse	130
7.1	Lizenzfassung	130
7.2	Infrastrukturfassung.....	134
7.3	Anforderungstests	136
7.3.1	Testcase 1 – Datenbank für einen User auf einem Client.....	136
7.3.2	Testcase 2 – Datenbank für einen RAC mit zwei Rechnern mit unterschiedlicher CPU-Threadanzahl	137
7.3.3	Testcase 3 – Datenbankanforderung mit fehlenden Informationen.	139
7.3.4	Testcase 4 – Anforderung einer Lizenz mit Addon-übergreifenden Featureabdeckungen	140
8.	Zusammenfassung und Conclusio	144
9.	Literaturverzeichnis	148

10.	Abbildungsverzeichnis	151
11.	Tabellenverzeichnis.....	153
12.	Anhang: Abstract.....	155

1. Einleitung

In der heutigen Zeit findet man kaum noch Prozesse in Unternehmen, die nicht durch eine entsprechende Software unterstützt werden. Bei vielen Prozessen kann es durch die Menge unterschiedlichster Programme sehr leicht dazu kommen, dass Firmen den Überblick über die eingesetzte Software verlieren und unwissentlich unberechtigte Programme ausführen.

Um eine Applikation ausführen zu dürfen, benötigt man Software-Lizenzen, die es den Unternehmen erlauben, bestimmte Programme zu verwenden. Fehlen solche Lizenzen oder sind nicht genügend vorhanden, um die Menge der eingesetzten Software zu decken, so begeht das Unternehmen eine Straftat. Die Unkenntnis über den Umfang der eingesetzten Programme schützt die Unternehmensleitung allerdings nicht vor der Verantwortung, die interne IT-Struktur ausreichend zu lizenzieren.

Um den Überblick über die interne IT-Infrastruktur eines Unternehmens zu behalten, gibt es das Lizenzmanagement, das für die Verwaltung der einzelnen Lizenzen verantwortlich ist. Die Aufgabe des Lizenzmanagements ist es, einen genauen Überblick der im Unternehmen verwendeten Programme zu besitzen, sowie den Lizenzhaushalt zu kontrollieren. Dies ist keine leichte Aufgabe, da man einerseits darauf achten muss, dass jegliche im Unternehmen eingesetzte Software ausreichend lizenziert ist, um keine teuren strafrechtlichen Konsequenzen erwarten zu müssen, andererseits aber auch, dass es beim Einkauf von Lizenzen nicht zu einem unnötigen und kostspieligen Überschuss von Lizenzen kommt.

Lizenzmanagement kann nun entweder direkt vom jeweiligen Unternehmen durchgeführt werden oder aber auch durch externe Dienstleister in beratender Funktion in Anspruch genommen werden. So ein Dienstleister ist das Unternehmen DBConcepts GmbH, das sich unter anderem als Aufgabe vorgenommen hat, den Kunden bestmöglich im Bereich Lizenzmanagement zu beraten und die idealste Lösung für einen etwaigen Zukauf von Lizenzen anbieten zu können. Dies geschieht vor allem im Bereich von Oracle-Lizenzen, da hier auch die anderen zwei Kernbereiche, das Development und die Datenbankadministration, tätig sind.

Um die Eruierung und Verwaltung von Kundenlizenzen zu optimieren, ist es geplant, im Rahmen dieser Master Thesis ein Programm zu entwickeln, welches es ermöglicht, die

beste und sparsamste Lösung für den Lizenzkauf von Kunden zu ermitteln, sowie eine historische Verwaltung zu ermöglichen, welche es auch erlaubt, den Kunden rechtzeitig vor fehlenden Lizenzen zu warnen und optimal zu beraten. Ebenfalls soll das Programm das firmeneigene Lizenzmanagement abdecken und unterstützen.

Es gibt bereits unzählige Lizenzmanagementprogramme, die allerdings meistens konkret auf die aktive Lizenzverwaltung von einem Unternehmen ausgerichtet sind. Das durch diese Master Thesis ausgearbeitete Unterstützungssystem zur Lizenzverwaltung hat zusätzlich aber noch den Fokus auf die vor der Anschaffung benötigte Softwareanalyse, welche dem Unternehmen hilft, geeignete Software zu klassifizieren und im nächsten Schritt zu ermitteln, welche Lizenz am besten und kosteneffizientesten dazu geeignet ist, die jeweilige Softwareverwendung im Unternehmen abzudecken. Das Tool soll ermöglichen, die individuellen Bedürfnissen und Anforderungen der einzelnen Kunden zu erfassen, diese zu verarbeiten und entsprechende Resultate zur Unterstützung zu ermitteln.

Hierfür soll ein Datenmodell entwickelt werden, das auf dem theoretisch erarbeiteten Wissen über das Lizenzmanagement aufbaut, um möglichst flächendeckend alle Arten und Variationen von Lizenzen und deren Restriktionen darstellen, sowie die zeitlich veränderten Bedürfnisse der unterschiedlichsten Kunden festhalten zu können und auf Basis dieser die entsprechenden Empfehlung zu optimalen Lizenzkäufen zu ermöglichen.

Aufbauend auf das Datenmodell soll ein Prototyp dieses Lizenzverwaltungssystems entwickelt werden. Um die Arbeit einzuschränken wird das Hauptaugenmerk auf Oracle-Datenbank-Lizenzen liegen, wobei das System darauf aufbaut, auch andere Lizenzen verwalten zu können.

2. Literaturanalyse

In den folgenden Kapiteln soll ein Einblick über das Thema im Allgemeinen sowie ein Einblick in die aktuelle Situation zum Thema Lizenzmanagement gegeben werden. In erster Folge werden also Grundthematiken beschrieben, welche notwendig sind, um die Grundlage für ein Lizenzmanagement zu bieten und die zu Grunde liegenden Begriffe zu verstehen.

Daraufhin soll ein Einblick über den aktuellen Stand des Lizenzmanagements geschaffen werden, um die Lage besser einschätzen zu können und etwaige Probleme aufzuzeigen.

In weiterer Folge betrachten wir die rechtliche Grundlage von Lizenzen und deren Management, um zu verstehen, warum es das Lizenzmanagement überhaupt gibt und warum eine vernünftige Verwaltung von diesem in heutigen Betrieben eigentlich unabdingbar sind.

Weiters wollen wir die Rahmenbedingungen für den Einsatz eines Lizenzmanagements in Organisationen aufstellen, um eine Abgrenzung dieser Thematik zu erreichen.

Als letzten Punkt der Analyse befassen wir uns mit den vorhandenen Tools, die bereits am Markt erhältlich sind und schaffen einen Überblick über deren definierten Usecase.

2.1 Grundlagen

In diesem Kapitel soll ein näherer Einblick über Lizenzen im Allgemeinen gegeben werden. Es soll erklärt werden, warum es Lizenzen gibt, wozu diese nötig sind und wie diese kategorisiert werden können.

Die Grundlage zum Lizenzmanagement schafft das Urheberrecht. Dieses bestimmt, wer über die Verbreitung und Nutzung von einem urheberrechtlich geschütztem Gut entscheidet. Im Grunde sagt das Urheberrecht aus, dass der Schöpfer eines Werkes über dessen Einsatz beziehungsweise dessen Nutzung entscheiden kann. Seit 1993 gelten in Österreich auch Computerprogramme, im Weiteren als Software bezeichnet, als urheberrechtlich geschützte Werke. Mehr Details zum Einblick in die rechtliche Lage wird später im Kapitel 2.3 Rechtliche Sicht dargestellt.

Um nun die Nutzung und die Verbreitung von urheberrechtlich geschützten Werken zu verwalten, gibt es Lizenzen. Laut Duden ist eine Lizenz eine rechtskräftige Genehmigung

zur Nutzung, Übersetzung und Übernahme eines Werkes.¹

Über eine Lizenz wird also bestimmt, wie ein urheberrechtlich geschütztes Werk, in unserem Fall Software, von anderen natürlichen oder juristischen Personen verwendet werden darf. Es gibt mehrere Möglichkeiten, die ein Urheber hat, Softwarelizenzen anzubieten.

Im ersten Schritt gilt vom Urheber zu überlegen, wie er seine Software vertreiben will, hier stehen ihm verschiedene Formen zur Verfügung. Die meistgenutzten Formen sind Open Source oder proprietäre Software, wobei sich letztere noch einmal in drei Kategorien unterteilt: Freeware, Shareware und kommerzielle Software.²

Ein wichtiger Punkt ist noch, dass die folgenden Kategorisierungen nicht standardisiert sind und somit deren Bezeichnungen in unterschiedlichen Literaturen voneinander abweichen können. Die hier herangezogenen Kategorien sind Großteils aus den Vorschlägen von (Groll, 2016) abgeleitet.

2.1.1 Lizenzformen

In der ersten Kategorisierung unterscheiden wir Lizenzen also nach deren Formen. Der Urheber muss sich Gedanken darüber machen, wie er seine Software anbieten will, also ob beispielsweise Änderungen an deren Quellcode erwünscht sind. Hier unterscheidet man zwischen zwei verschiedenen Ansätzen, einerseits des offenen Standards Open Source und der proprietären Software, bei welcher der Sourcecode nicht zugänglich ist und eine Veränderung der Software damit ausgeschlossen wird.

2.1.1.1 Open Source

Open Source bezeichnet hinlänglich Software, bei der Veränderung und Vervielfältigung erlaubt sind. Unter anderem gibt es hier die GNU General Public License (kurz GNU GPL) oder der European Public License (kurz EUPL), welche kompatibel zur GNU GPL ist. Durch diese Lizenz erlaubt der Urheber (oder Lizenzgeber) den uneingeschränkten Nutzen der Software, die Erlaubnis auf Vervielfältigung und Bearbeitung der Software, sowie deren Verbreitung. Es wird also ausdrücklich erlaubt, den Quellcode zu

¹ (Duden, 2018)

² (Groll, 2016)

analysieren und zu studieren, ihn gegebenenfalls nach eigenen Wünschen anzupassen und weiter zu verbreiten.³

Die bekanntesten Vertreter von Open Source sind das Betriebssystem Linux, die Bürossoftware Libre Office und der Browser Firefox.

2.1.1.2 Proprietäre Software

Unter proprietärer Software versteht man Programme, deren Quellcode nicht einsehbar ist und damit auch Veränderungen an diesem nicht genehmigt sind. Änderungen erfolgen ausschließlich über den Urheber oder von ihm berechtigten Personen. Im Gegensatz zu Open Source ist eine Anpassung von einem Lizenznehmer nicht möglich. Natürlich gibt es noch die Möglichkeit der Anpassung über eine zur Verfügung gestellte Schnittstelle in der jeweiligen Software, diese wird aber nicht als Teil der zu lizenzierenden Software angesehen.

Die proprietäre Software wird weiter in verschiedene Kategorien unterteilt, die diese auf Hinblick der Kosten und Verbreitungsmöglichkeit definiert. Wir gehen hier auf die drei generellen Formen Freeware, Shareware und kommerzielle Software ein. Es gibt noch einige andere Vertriebsmöglichkeiten, wie beispielsweise Donationware, bei welcher sich der Lizenznehmer entscheidet, ob er eine Gebühr zahlt und Mindware, bei welcher der Lizenznehmer über die Höhe der Gebühr entscheiden kann.⁴ Allerdings sind diese Varianten eigentlich Unterkategorien von Freeware oder Shareware.

2.1.1.2.1 Freeware

Bei Freeware handelt es sich um proprietäre Software, welche allerdings kostenlos genutzt werden kann. Im Gegensatz zu Open Source ist der Quellcode hier nicht offen zugänglich und damit weder die Analyse noch die Veränderung des Quellcodes erlaubt. Der Vertrieb ist allerdings kostenlos und die Verbreitung der unveränderten Software ist erlaubt. Bekannte Beispiele für Freeware sind unter anderem der Adobe Acrobat Reader sowie die Office Viewer Programme, die es ermöglichen Office-Dateien zu lesen, aber nicht zu bearbeiten.

³ (Casares, 2016)

⁴ (Groll, 2016)

2.1.1.2.2 Shareware

Als Shareware wird proprietäre Software bezeichnet, welche frei vertrieben werden kann, allerdings im Gegensatz zu Freeware nicht vollständig kostenlos ist. Wieder gilt dabei, dass es sich um proprietäre Software handelt und der Quellcode nicht offen ist. Meistens geht es hier um Programme, bei denen der Urheber sich nicht um den Vertrieb kümmern kann oder will. Allerdings wird die Nutzung der Software meist technisch, zumindest aber vertraglich in der Lizenzvereinbarung (beispielsweise der EULA – End User License Agreement) beschränkt. Diese Einschränkungen sind in den meisten Fällen funktionale oder zeitliche (meist 30 Tage) Beschränkungen. Aber auch die Variante, nur im kommerziellen Umfeld Lizenzgebühren zu verlangen, ist möglich, sodass beispielsweise Privatpersonen die Software unentgeltlich nutzen können, Selbstständige und Unternehmen diese hingegen lizenzieren müssen.

Die häufigste Verwendung von Shareware ist aber für Demoversionen von Programmen, welche nur im eingeschränkten Umfang zur Verfügung stehen. Der Endnutzer muss eine Lizenz erwerben, um das Programm im vollen Umfang (je nachdem zeitlich oder funktional) nutzen zu können.

Eines der bekanntesten Shareware ist wohl WinZip, allerdings gibt es hier auch unzählige andere Beispiele, welche meist auf Onlineplattformen verbreitet werden.

2.1.1.2.3 Kommerzielle Software

Die kommerzielle Software ist neben der Shareware ein wichtiger Kernpunkt des Lizenzmanagements. Diese ist, wie auch schon die Freeware und Shareware, nicht veränderbar, da ihr Quellcode nicht offen ist. Eine Verbreitung außerhalb der in den Lizenzvereinbarungen (zum Beispiel EULA – End User License Agreement) definierten Möglichkeiten ist untersagt. Die Nutzung von kommerzieller Software ist, wie der Name schon sagt, gebührenpflichtig und ist nur unter den im Lizenzvertrag angeführten Bedingungen nutzbar. Welche das sein können erfahren wir in den folgenden Kapiteln.

Shareware und kommerzielle Software sind für das Lizenzmanagement deshalb so interessant, da beispielsweise bei einem Lizenz-Audit die entsprechenden Lizenzverträge zur legalen Nutzung der eingesetzten Software vorgelegt werden müssen.

Unter Lizenz-Audit versteht man die Prüfung eines Lizenzgebers bei einem Lizenznehmer, zur Kontrolle, ob dieser die auch tatsächlich im Einsatz befindliche Software des Lizenznehmers korrekt und ausreichend lizenziert hat.

Dies ist ein wichtiger Punkt im Lizenzmanagement, auf den noch genauer im Kapitel 2.2 Lizenzverwaltungsansätze und Problem eingegangen wird, um aber erst einmal einen Überblick zu schaffen, welche Lizenz- Kategorisierungen es gibt, werden wir im folgenden Kapitel die Arten, Klassen und Typen von Lizenzen genauer analysieren.

2.1.2 Lizenzarten

Die Lizenzarten werde allgemein in zwei Möglichkeiten unterteilt, die Einzelplatzlizenz und die Mehrplatzlizenz. Wie der Name schon sagt, ist erstere Lizenzart für ein System gedacht, während Mehrplatzlizenzen auf mehrere Systeme ausgerollt werden darf.

2.1.2.1 Einzelplatzlizenz

Die bekanntesten Einzelplatzlizenzen sind die Boxversionen, welche von Einzelhändlern erwerbbar sind. Diese enthalten in den meisten Fällen nur die Möglichkeit auf einem System installiert und verwendet zu werden. Wichtig hierbei ist, dass eine Einzelplatzversion im Normalfall nicht zum Ausrollen in Unternehmen geeignet ist, da hier die Nutzung laut Lizenzvereinbarung tatsächlich nur für ein System ausgelegt sind. Eine Vervielfältigung ist in den meisten Fällen untersagt. Ein Unternehmen müsste also genaugenommen auf jedem System ein separates Installationsmedium aus einer neuen Box verwenden. Als Installationsmedium dienen in den meisten Fällen CDs, DVDs, Speicherkarten oder Speichersticks, in manchen Fällen enthalten die Boxen aber auch einen Key zur Onlineaktivierung mit anschließendem Download der Software. Es gibt allerdings auch Ausnahmen, bei denen die Installation auf meist mobilen Zweitgeräten erlaubt ist. Ein Beispiel hierfür wären diverse Microsoft Office Versionen.

2.1.2.2 Mehrplatzlizenz

Bei Mehrplatzlizenzen wird es erlaubt, eine bestimmte Software auf mehreren Geräten zu vervielfältigen. Die Anzahl der erlaubten Kopien der Software wird im Normalfall im Lizenzvertrag festgelegt. Diese Lizenzart ist für Unternehmen mit einigen Arbeitsplätzen ausgelegt und erleichtern meist das Ausrollen der entsprechenden Software. Ab einer bestimmten Größe rentieren sich die Mehrplatzlizenzen in mehrfacher Hinsicht, da

einerseits die Verwaltung der Lizenzen erleichtert wird und weiters auch ein Mengenrabatt möglich ist.

2.1.3 Lizenzklassen

Als Lizenzklasse wird definiert, welche Voraussetzungen an Vorgängerversionen für eine Lizenz nötig sind. Es folgt ein kurzer Überblick über die einzelnen Klassen.

2.1.3.1 Vollversion

Die Lizenzklasse, die keine Bedingungen an eine Vorgängerversion knüpft ist die Vollversion. Zwar kann diese eventuell eine funktionelle oder zeitliche Einschränkung zu anderen Versionstypen haben, allerdings wird keine zusätzliche Version benötigt, um die enthaltenen Funktionen nutzen zu können.

2.1.3.2 Upgrade

Ein Upgrade bedeutet einen Versionssprung einer Vollversion. Im Normalfall wird eine vorhergegangene Vollversion benötigt. Es kann aber auch sein, dass ein Upgrade auf ein älteres Upgrade aufbaut, welches wiederum eine vorangehende Vollversion oder ein weiteres Upgrade benötigt. Welche vorangehenden Versionen für ein Upgrade berechtigt sind, definiert der Lizenzgeber. Am Anfang der Kette steht aber in allen Fällen eine Vollversion.

Dies muss aber nicht zwingend die Vollversion des eigenen Produktes sein. In so genannten Cross-Upgrades wird die Möglichkeit geboten, von einem Konkurrenzprodukt kostengünstiger auf das Produkt eines anderen Anbieters zu wechseln. Diese Art eines Upgrades dient dazu, Benutzer zum Wechsel von einem Fremdprodukt auf das eigene zu animieren.

In allen Fällen gilt, bei Upgrades ist es notwendig, alle in der Upgradekette benötigten Lizenzen vorweisen zu können, was unter Umständen bedeutet, dass sehr alte Vollversionslizenzen weiter verwahrt und bei Lizenz Audits vorgewiesen werden müssen.

2.1.3.3 Update

Bei einem Update handelt es sich meist um eine kleinere Aktualisierung innerhalb einer Version, wie beispielsweise Sicherheitsupdates oder Bugfixes. Updates sind in den meisten Fällen kostenlos, sind aber teilweise zeitlich beschränkt, sodass beispielsweise Softwareupdates nur einen definierten Zeitraum nach dem Kauf einer Vollversion oder

eines Upgrades zur Verfügung gestellt werden. In solchen Fällen werden meist Wartungsverträge für eine Verlängerung dieses Zeitraumes angeboten.

2.1.3.4 Addon

Ein Addon wird meistens mit einer Funktionserweiterung der Vollversion verbunden. Bei Oracle beispielsweise werden solche Addons Optionen genannt, welche die Funktionalität der Software um weitere Bereiche, wie beispielsweise performanceoptimierende Maßnahmen, erweitert. Auch Addons können wiederum mit Addon-Updates auf die nächste Version gehoben werden.

2.1.3.5 Client Access License

Die Client Access License (kurz CAL) ist eine spezielle Lizenz für die Nutzung von Server-Architekturen von mehreren Benutzern. Sie ist beispielsweise notwendig, um Software auf einem Server abzurufen und verwenden zu dürfen. Auch hier besteht meist die Möglichkeit eines Upgrades zu neueren Versionen.

2.1.4 Lizenztypen und Metriken

Der nächste wichtige Faktor einer Lizenz ist der Lizenztyp. Die bekanntesten Typen von Lizenzen sind „pro Gerät“, „pro User“ oder „pro CPU“. Damit ist gemeint, wie die Unterteilung einer Lizenz aussieht. Wie genau diese Aufgliederung aussieht, wird in den Lizenzmetriken festgelegt, welche im Endeffekt eine Auflistung nach Kriterium und Wert ist, also beispielsweise ist eine Lizenz für 10 User oder eine andere für 20 Geräte gültig.

Eine „pro Gerät“ Lizenz erlaubt beispielsweise das Verwenden einer Software auf einem System für unterschiedliche User, solange diese nicht gleichzeitig darauf zugreifen.

Eine „pro User“ Lizenz erlaubt das Installieren der Software auf mehreren Systemen, wie beispielsweise dem fixen Arbeitsplatzrechner und einem mobilen Gerät. Der User muss sich in einer Art identifizieren und kann das Produkt auf mehreren Systemen nutzen. Öfters kommt auch eine Mischform „pro User pro Gerät“ vor, hier muss tatsächlich für jeden User und jedes Gerät eine separate Lizenz ausgewiesen werden.

Eine „pro CPU“ Lizenz ist meist in Serverbereichen erforderlich. Die Lizenzierung nach Usern ist in den Jahren immer schwieriger geworden, da die Feststellung der tatsächlichen User, die auf den Server zugreifen nicht so trivial ist. Vor allem in Anbetracht der Tatsache, dass viele Systeme mittlerweile aus dem Internet zugänglich sind, ist eine Userbasierte Lizenz am Server nicht mehr tragbar. Die Lizenzgeber haben

deshalb neue Formen gesucht, um Server nach Leistungsklassen einzuteilen und sind aktuell im Bereich der Lizenzierung pro CPU angelangt. Je nach Hersteller kann dies aber weitere Kriterien beinhalten, IBM beispielsweise gliedert einzelne CPUs in sogenannte PVUs (Process Value Units), um ganz genaue Angaben über die Leistung der jeweiligen CPU abzurechnen. Oracle verlangt meist die Anzahl der Kerne pro CPU und auch die Sockelanzahl der Mainboards spielt in einigen Fällen eine wichtige Rolle, teilweise auch dann, wenn diese gar nicht mit einer zweiten CPU bestückt sind.

2.1.5 Kaufmännische Einteilung der Lizenzen

Die letzte Unterscheidung der Lizenzen ist die kaufmännische Sicht auf eben diese. Sie stellt eine Einschränkung in der Verwendbarkeit von Software dar. Sie ist vor allem relevant beim Thema Weiterverkauf von Lizenzen. Wir unterscheiden hier zwischen dem Full Package Product, einer System Builder Software und einer OEM-Version, welche nun im näheren beschrieben werden.

2.1.5.1 Full Package Product

Die erste Kategorie ist das Full Package Product, auch gerne Box-Version genannt. Sie stellt das beim Einzelhändler bekannte Produkt dar, welches in teilweise aufwendigen Verpackungen erhältlich ist, zumeist ein gedrucktes Handbuch und eine End User License Agreement beinhaltet. Diese Variante stellt aufgrund der aufwendigen Aufmachung auch gleichzeitig die teuerste Variante dar.

2.1.5.2 System Builder Software

Eine weitere Möglichkeit eines Software-Lizenzzerwerbs stellt die System Builder Software dar. Sie dient primär dazu, dass Verkäufer fertige Systeme vorbereiten und die Software-Lizenz in einem Paket, meist mit dazu passender Hardware, verkaufen können. Da in solchen Fällen die Software bereits fertig installiert geliefert wird, wird hier auf die Box-Aufmachung verzichtet. Meistens erhält man einen Lizenzschlüssel und in manchen Fällen noch eine gedruckte Kurzanleitung. Auf ein aufwendiges Boxdesign, sowie ein ausführlich bedrucktes Handbuch wird oft verzichtet. Das Handbuch liegt häufig in digitaler Form der Softwareinstallation bei, ebenso wie das EULA, welches im Allgemeinen im Setup oder erstmaligem Start der Software digital präsentiert und akzeptiert werden muss, um die Software nutzen zu können.

2.1.5.3 Original Equipment Manufacturer (kurz OEM)

Die letzte Variante ist die OEM-Version. Sie ist für Hardwarehersteller gedacht, die deutlich kostengünstiger deren Systeme mit vorinstallierter Software, beispielsweise einem Betriebssystem, ausstatten und ausliefern können. Die Lizenz ist hier stark mit der Hardware verknüpft und es wird meist auf ein eigenes Setup für die Software verzichtet. Der Hardwarehersteller liefert in solchen Fällen meist nur ein komplettes Recovery-Image, mit dem alle beim Verkauf vorinstallierten Programme zusammen wiederhergestellt werden können, nicht aber eine konkrete Software davon.

Ein Weiterverkauf der Software ohne die dazugehörige Hardware ist vom Lizenzgeber eigentlich nicht beabsichtigt, laut österreichischem Recht aber unter Umständen erlaubt, nämlich dann, wenn die Software auf einem Datenträger in Verkehr gebracht wurde. Hier darf der Käufer jegliche Daten von seinem System entfernen und die Software-Lizenz mitsamt Installationsmedium, Handbuch, EULA und weiteren mitgelieferten Bestandteilen weiterverkaufen.

2.1.6 Neue Lizenzmodelle

Die bisherigen Kriterien entsprechen dem klassischen Lizenzmodell, bei dem die Software noch in einer konkreten Version für die Systeme in einem Unternehmen gedacht sind. Mittlerweile entwickelt sich aber bereits das Modell „Software as a Service“ (kurz SaaS) immer weiter und wird auch immer attraktiver für die Geschäftswelt. Da die Software hier direkt vom Anbieter geliefert wird und teilweise bis größtenteils auf dessen Server laufen, ist ein genauere Einsatz von Lizenzangaben möglich. Die Lizenz zur Verwendung der Software wird also nicht mehr für eine bestimmte Version verkauft, sondern vermietet. Der Lizenznehmer zahlt also eine regelmäßige Gebühr auf Zeit. Im Umkehrschluss darf der Kunde aber meist, solange der Mietvertrag läuft, auf die aktuelle Version der Software zugreifen. Außerdem erspart man sich die Verwaltung der Applikationen, da diese meist in einer zentralen Verwaltung des Anbieters von statten geht. Ein solches Modell wäre beispielsweise Microsofts Office 365, bei der die gewohnten Clientapplikationen der Office-Welt über das Onlinesystem aktiviert und verwaltet werden können.

Dies klingt erstmal sehr positiv, die neuen Modelle beherbergen aber auch neue Risiken. Ein wichtiger Punkt, warum SaaS noch nicht flächendeckend genutzt wird, ist einerseits die Hoheit über die eigenen Daten, die man bei so einer Lösung meist aus der Hand gibt.

Außerdem ist die Verlässlichkeit des Lizenzgebers ein entscheidendes Kriterium, da die internen Prozesse meist eng mit der eingesetzten Software verknüpft sind. Strauchelt das Unternehmen des Lizenzgebers, kann auch eine ernsthafte Gefahr für den Lizenznehmer entstehen, wenn plötzlich nicht mehr auf die benötigten Dienste zugegriffen werden kann.

Ein weiterer Punkt ist die Verfügbarkeit der Software, denn je mehr die Software vom Zugriff auf die vom Lizenzgeber zur Verfügung gestellten Serverprozesse abhängt, desto schmerzhafter ist ein Stillstand bei einem Ausfall der Verbindung zu eben jenem Server. Ein Ausfall der Internetleitung könnte also ein Stillstand im Unternehmen bedeuten.

Da nun ein genereller Überblick über die verschiedensten Kategorisierungen von Lizenzen gegeben wurde, wird im nächsten Schritt ein tieferer Einblick über den Sinn, die Umsetzung und die Probleme eines Lizenzmanagements gewährt.

2.2 Lizenzverwaltungsansätze und Problemstellen

Softwareentwicklung ist heutzutage kaum mehr wegzudenken, sehr viele Prozesse in heutigen Unternehmen beinhalten IT-unterstützte Prozessschritte und diese sind in vielen Fällen gar nicht mehr ohne Software durchführbar. Dies erkannten diverse Softwarehersteller schon sehr früh und konzentrierten sich darauf, in den Markt mit ihrer Software durchzudringen. Der Vorteil von Softwarevertrieb ist die geringe Markteintrittsbarriere. Hat man einmal ein funktionierendes und sinnvolles Produkt kreiert, benötigte die Vervielfältigung nur wenig Produktionskosten, die Marktdurchdringung ist beispielsweise durch den Sharewarevertrieb über Downloadportale möglich. Hat sich eine Software erst einmal etabliert, ist die Fluktuation sehr gering, denn die internen Prozesse von Unternehmen, die eben jene Software einsetzen, ist nach einiger Zeit so sehr an das Softwareprodukt gekoppelt, dass eine Umstellung auf ein Konkurrenzprodukt oft auch eine aufwendige Umstellung von internen Prozessen nach sich ziehen würde.

Bis zum Punkt des Softwarevertriebs müssen allerdings sehr viele Arbeitsstunden investiert werden, um eine Software zu erstellen, die auch tatsächlich vom Markt angenommen wird. Genau diesen Aufwand wollen Softwareanbieter sich im Nachhinein von den Anwendern derer Software vergüten lassen. Seit 1993 gilt Software in Österreich als urheberrechtlich geschütztes Gut, wie auch in Kapitel 2.3 Rechtliche Sicht näher beschrieben. Das heißt der Softwarehersteller ist unabdingbar als Urheber mit

seinem Softwareprodukt verbunden. Nur er darf bestimmen, wie seine Software verbreitet und eingesetzt werden darf. Die Regelung, wie Software von Kunden verwendet, verbreitet und eventuell vertrieben werden darf, wird über entsprechende Lizenzen festgelegt. Im Lizenzvertrag werden die Anforderungen und Bedingungen für die Nutzung einer Software geregelt.

Der Lizenznehmer ist also verpflichtet, bei Verwendung einer bestimmten Software auch über die nötigen Softwarelizenzen zu verfügen, die es ihm erlauben, die Software im jeweiligen Rahmen auch tatsächlich einzusetzen. Das heißt also, dass der Softwarenutzer über eine Lizenz verfügen muss, die ihn dazu berechtigt, eine bestimmte Software im jeweiligen Rahmen auszuführen. Diese Softwarelizenz muss der Nutzer der Software aufbewahren und beispielsweise im Falle eines Lizenz-Audits vorweisen können. Macht er dies nicht, betreibt er die Software illegal und dies kann, wie später noch beschrieben, drastische Konsequenzen nach sich ziehen.

Da nun klar ist, dass für die Verwendung von Software Lizenzen notwendig sind, um deren Nutzungsbestimmungen zu vereinbaren und das Recht auf die Nutzung von Software vorweisen zu können, blicken wir im folgenden Kapitel auf den Nutzen, den eine ordentliche Lizenzverwaltung mit sich bringt.

2.2.1 Nutzen von Lizenzverwaltung

Für kleine Betriebe mit einer Hand voll von Rechnern wird der Überblick über die Lizenzen für die Softwarenutzung auf den einzelnen Geräten noch halbwegs überschaubar ausfallen. Wächst das Unternehmen aber, wird unweigerlich neben der wachsenden Anzahl an Systemen, die lizenziert werden müssen, auch das IT-Portfolio mitwachsen. Hier einen Überblick zu bewahren wird immer aufwendiger und irgendwann wird es nicht mehr reichen, beispielsweise über Exceltabellen die laufende Software und deren Lizenzverträge zu erfassen und zu verwalten. Es wird ein Punkt erreicht werden, an dem ein vernünftiges softwareunterstütztes Programm aushelfen muss, den Überblick über die zum Teil komplexen Nutzungsbedingungen einzelner Lizenzen, vor allem im Serverbereich, zu bewahren. Ebenfalls zählt der Überblick über und der Einsatz von Software zu wichtigen Punkten des Risikomanagements, vor allem bei Aktiengesellschaften. Eine nicht ausreichend lizenzierte Systemlandschaft kann fatale Folgen für ein Unternehmen haben, die noch im Kapitel 2.2.6 Probleme und Folgen von fehlender Lizenzverwaltung näher ausgeführt werden.

Neben den erwähnten Gefahren, die es mit Hilfe von Lizenzverwaltung zu minimieren gilt, entstehen aber bei Einführung eines vernünftigen Lizenzmanagement auch zusätzlich diverse Vorteile, nämlich die Transparenz der eingesetzten Produkte, eine mögliche Kostensenkung in der Systemlandschaft, die Compliance gegenüber dem Lizenzgeber und die Rechtmäßigkeit der geltenden Gesetze.⁵⁶ In den folgenden Unterkapiteln werden diese ausführlicher beschrieben.

2.2.1.1 Transparenz

Durch das Einsetzen von Lizenzverwaltung ergibt sich automatisch eine Transparenz über die eingesetzte Software im Unternehmen. Wenn man weiß, wo welche Software verwendet wird, kann man leichter analysieren, welche Software sinnvoll zum Einsatz kommt und wie gut sie die Anforderungen abdeckt, die für die jeweiligen Prozesse im Unternehmen entscheidend sind. Der Überblick über die Softwarelandschaft im eigenen Unternehmen fördert auch die Kontrolle und Einschätzung über die Notwendigkeiten von spezifischen Programmen, was auch gleich zum nächsten Punkt führt, der Kostensenkung.

2.2.1.2 Kostensenkung

Oftmals werden von verschiedenen Abteilungen unterschiedlichste Software für deren Prozesse eingesetzt. Eventuell übersieht man dabei, dass eine Vereinheitlichung von bestimmten Programmen eine deutliche Kostenminimierung darstellen könnte.

Auch könnte es sein, dass innerhalb einer Abteilung zwei verschiedene Programme zur Durchführung diverser Tätigkeiten eingesetzt werden, wo auch ein komplett anderes Produkt direkt alle unterstützenden Funktionen aufweisen könnte. Man würde sich damit eventuell unnötige Lizenzkosten ersparen.

Weiters könnten Programm-Schnittstellen unterschiedlicher Abteilungen besser koordiniert und aufeinander abgestimmt werden, sodass Prozessschritte vereinfacht oder sogar direkt eingespart werden könnten.

Ein ordentlich geführtes Lizenzmanagement kann also direkt förderlich für die Analysen dieser Fragen sein.

⁵ (Härtel, 2015)

⁶ (Groll, 2016)

2.2.1.3 Compliance

Ein weiterer Vorteil ist der Überblick über die Einhaltung der Lizenzbestimmungen. Es wird also die Überprüfung der Einhaltung der Compliance (also zu Deutsch „Erfüllung“) der vom Lizenzgeber veranlassten Bedingungen erleichtert. Wie bereits erwähnt, bestimmt der Lizenzgeber, in welchem Umfang seine Software eingesetzt werden darf. Durch unterschiedlichste Lizenzierungen können unterschiedliche Bedingungen entstehen, die für eine rechtmäßige Ausführung der Software von Nöten sind. Um den Überblick zu behalten, ob die Softwarebestimmungen, beispielsweise bei Hardwareänderungen, noch korrekt erfüllt werden, ist es gut zu wissen, was die im Lizenzvertrag ausgeführten Einschränkungen sind und ob diese noch mit den bestehenden Systemen übereinstimmen. Die Compliance ist vor allem wichtig, um auf Software Audits vorbereitet zu sein. Hat man den Überblick über die im Unternehmen vorhandene Systemlandschaft, sollte der Nachweis der benötigten Lizenzen leicht abrufbar sein und erreicht somit eine deutliche Risikominimierung.

Ist gewährleistet, dass man sich an die Bestimmungen des Lizenzgebers hält, ist man auch rechtlich gesehen auf der sicheren Seite, denn ein Verstoß gegen diese, wäre ein Verstoß gegen das Urheberrechtsgesetz.

2.2.1.4 Rechtmäßigkeit

Das Lizenzmanagement dient also dazu, auch die rechtliche Seite abzudecken. Einerseits hat man sich an die Bestimmungen zu halten, die vom Lizenzgeber erwartet werden, andererseits werden auch rechtliche Anforderungen damit abgedeckt. Beispielsweise ein korrektes Einbinden der System- und Softwarelandschaft des eigenen Unternehmens im Risikomanagement, welches teilweise gesetzlich vorgeschrieben ist. Auch soll es zur Absicherung der Einhaltung von nationalen und auch internationalen Gesetzen dienen. Denn wie (Härtel, 2015) schon treffend schreibt:

„Die Unkenntnis über die Abweichung vom geltenden Nutzungsrecht, vom UrhG und von den zahlreichen internationalen Regularien (SOX, Basel II) entbindet die Geschäftsführung nicht von der Verantwortung lizenzrechtliche Nutzungsbestimmungen einzuhalten. Die Wichtigkeit einer intakten IT-Compliance wird

durch diese Regularien noch stärker betont und führt zu wachsenden Anforderungen an ein funktionierendes Software-Asset-Management.“⁷

Wurden nun also die Vorteile für das Lizenzmanagement angeführt, soll im nächsten Schritt auf den Vorgang zur Lizenzvergabe und -verwaltung in Unternehmen ein wenig genauer eingegangen werden.

2.2.2 Lizenz Life Cycle

Wie in den vorangegangenen Kapiteln bereits erwähnt, ist der Einsatz von einem ordentlich geführten Lizenzmanagement in Unternehmen in mehreren Aspekten äußerst vorteilhaft. Doch wie wird nun eigentlich das Lizenzmanagement ordentlich geführt und gelebt?

Nach (Groll, 2016) gliedert sich das Lifecyclemodell von Software in zwei Sparten, dem kaufmännischen Prozess und dem technischen Prozess (siehe Abbildung 1: Software-Lifecyclemodell nach (Groll, 2016)). Während sich der kaufmännische Prozess um die Beschaffung der Lizenz kümmert, ist der technische Prozess für den Einsatz verantwortlich.

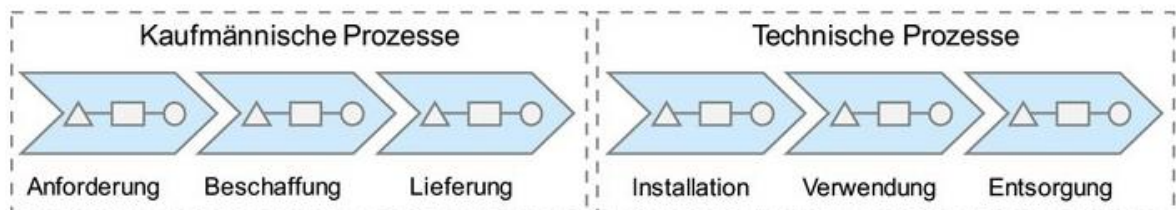


Abbildung 1: Software-Lifecyclemodell nach (Groll, 2016)⁸

2.2.2.1 Kaufmännische Prozesse

Sieht man sich den kaufmännischen Prozess an, untergliedert sich dieser wiederum in drei Teile, die Anforderung, Beschaffung und Lieferung. Im Folgenden gehen wir näher auf die entsprechenden Phasen ein.

2.2.2.1.1 Anforderung

Die Definierung der Anforderung ist ein erster wichtiger Schritt, denn je genauer man diesen durchführt, desto weniger wird man mit unangenehmen Situationen im

⁷ (Härtel, 2015)

⁸ (Groll, 2016)

produktiven Einsatz mit Software erleben, wenn beispielsweise bestimmte Anforderungen nur marginal von einem Softwareprodukt abgedeckt werden.

Es wird empfohlen, sich hier erst einmal einen Softwarekatalog anzulegen, der alle Softwareprodukte beinhaltet, die möglicherweise im Unternehmen eingesetzt werden könnten. In diesem analysiert und definiert man, welche Softwareprodukte welche Anforderungen erfüllen, um so ermitteln zu können, welche Programme am geeignetsten die Prozesse im Unternehmen unterstützen.

Erweist sich eine Software als ausreichend relevant, wird sie in den nächsten Topf der „genehmigten Software“ geschoben, in dem nur noch alle als tauglich erwiesenen Programme gesammelt werden. Gibt es Programme, die sich im ersten Schritt als unzulänglich erwiesen haben, verbleiben diese im ersten Topf des Softwarekatalogs. Es empfiehlt sich auch die Schwachpunkte und Vorteile solcher Applikationen zu dokumentieren, um eventuell zu einem späteren Zeitpunkt eruieren zu können, ob sich die Software zu Gunsten der internen Bedürfnissen des Unternehmens weiterentwickelt hat.

Der letzte Topf stellt dann den „Softwarewarenkorb“ dar. Hierbei handelt es sich um Software, die tatsächlich bestellt werden kann und soll, womit wir zum nächsten Schritt des Lifecycles übergehen, der Beschaffung.

2.2.2.1.2 Beschaffung

Die Beschaffung wird meist in der kaufmännischen Abteilung eines Unternehmens eingeleitet. Durch vernünftiges Lizenzmanagement können also die kaufmännische Abteilung, welche für den Erwerb der Softwarelizenzen zuständig ist und die technische Abteilung, welche für die Ausrollung der Software im Unternehmen verantwortlich ist, vernünftig zusammenarbeiten. Nachdem das Software-Portfolio festgelegt wurde, können der kaufmännischen Abteilung die genauen Stückzahlen für die benötigten Lizenzen übergeben werden und diese kann sich um die Beschaffung beim entsprechenden Anbieter kümmern.

2.2.2.1.3 Lieferung

Nach der Bestellung durch das Unternehmen erfolgt üblicherweise die Lieferung der Software durch den Anbieter. Der Lizenznehmer erhält eine Bestätigung in Form eines

Lizenznutzungsvertrages, in dem genau definiert ist, wie die Software im Unternehmen eingesetzt werden darf.

Nach diesem Schritt endet der kaufmännische Prozess und es erfolgt der technische Prozess, der für die Verbreitung der Software im Unternehmen sorgt.

2.2.2.2 Technische Prozesse

Der technische Prozess erfolgt laut (Groll, 2016) in der Installation der Software, also der Verbreitung der Software im Unternehmen im Hinblick auf die erlaubte Nutzung laut Lizenzvertrag.

Während der Verwendung ist zu beachten, dass jede aktive Verwendung der Software auch eine gültige Lizenz dafür voraussetzt, andernfalls macht man sich strafbar.

Wird die Software irgendwann nicht mehr benötigt, erfolgt die Entsorgung. Hier sei zu erwähnen, dass oft übersehen wird, dass Software auch von ausgelagerter Hardware wieder korrekt entfernt werden muss, wenn man die entsprechende Lizenz auf einer anderen Hardware wiederverwenden will. Ein oft auftretender Fehler ergibt sich meist bei einer Neuanschaffung von Clientgeräten, um beispielsweise ein Hardwareupgrade durchzuführen. Erhalten nun die Angestellten im Unternehmen neue, aktuellere Hardware, wird die alte Arbeitsumgebung meist durch das Klonen der alten Maschine auf das neue System direkt übernommen. Der alte Rechner wird unverändert ins Lager gestellt, bleibt aber unberührt. In diesem Zustand befindet sich aber noch die Software auf diesem Gerät und muss somit durch den Lizenzvertrag abgedeckt werden. Bei Software-Audits werden also sehr gerne Rechner aus den Lagerbeständen kontrolliert. Es empfiehlt sich vor dem Einlagern alter Maschinen, diese gründlich von jedweder Software zu bereinigen.

Damit wäre das Lifecycle-Modell von Lizenzen näher erläutert. Im nächsten Schritt wollen wir nun auch noch die einzelnen Phasen der Lizenz analysieren, da diese entscheidend für ein Lizenzmanagementtool sind.

2.2.3 Phasen der Lizenz

Haben wir nun näher beschrieben, wie der Lifecycle einer Lizenz aussieht, wollen wir im Folgenden etwas konkretisieren, wie die einzelnen Phasen der Lizenz aussehen, um sie später im Lizenzmanagementtool einsetzen zu können.

In Abbildung 2: Phasen einer Lizenz ist ersichtlich, wie solche Phasen aussehen können.

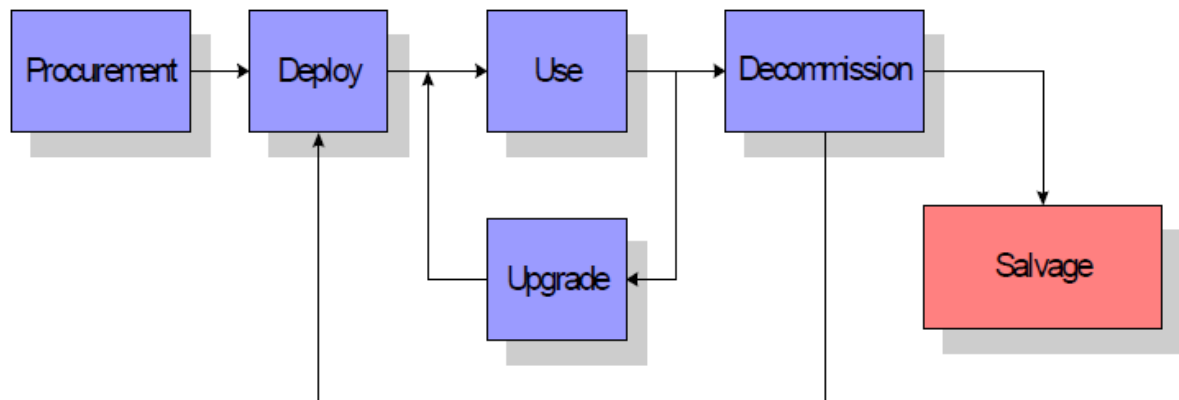


Abbildung 2: Phasen einer Lizenz

Wir haben hier als ersten Punkt die Beschaffung, welche im Lifecycle ausführlich abgehandelt wurde. Dies stellt die Eintrittsphase der Lizenz dar.

Darauf folgt der sogenannte IMAC-Prozess, wobei die Buchstaben für Install, Move, Add und Change stehen. Mit anderen Worten, die Lizenz wird im Unternehmen eingesetzt, um den Softwarebetrieb korrekt abzudecken. Im ersten Schritt kommt es zum Ausrollen der Software, wodurch die Lizenz aktiviert wird und sich in den Status „Use“ begibt. Sobald eine Software auf einem System ausgeführt wird, wird eine entsprechende Nutzungslizenz dafür verlangt und muss im Falle eines Audits vorgewiesen werden können.

Nach längerer Laufzeit ergeben sich mehrere Möglichkeiten: Entweder wird die Software auf einen neuen Stand gehoben und benötigt deshalb eventuell eine aktualisierte Lizenz, beispielsweise für eine höhere Versionsnummer eines Programmes, oder die Software wird nicht mehr benötigt und somit vom entsprechenden Gerät entfernt. Dann kann die Lizenz außer Betrieb (Decommission) genommen und wieder in den Pool der verfügbaren Softwarelizenzen aufgenommen werden. Hier entscheidet sich dann, ob die Software für einen anderen Einsatzzweck, beziehungsweise ein anderes Gerät, verwendet und deshalb neu „Deployed“ wird, oder aber die Lizenz ausgemustert wird und in die letzte Phase übergeht.

Die letzte Phase ist die Salvage-Phase (zu Deutsch Verwertung), die entweder in Ausscheidung der Lizenz oder in deren Weiterverkauf endet. Hier sei noch einmal

erwähnt, dass die Software vom ursprünglichen Gerät komplett entfernt werden muss, bevor die Lizenz veräußert oder wiederverwendet werden darf.

Da nun der Lifecycle und die Phasen einer Lizenz bekannt sind, betrachten wir in den kommenden Kapiteln wichtige Punkte, auf die im Lizenzmanagement zu achten ist.

2.2.4 Überlizenzierung

Betrachtet man die Sachlage, dass jede eingesetzte Software auch eine entsprechende Lizenz benötigt, die den Anwender dazu berechtigt, die Software im entsprechenden Rahmen auszuführen, so ist es notwendig, einen entsprechenden Pool an Lizenzen bereitzustellen. Es ergibt sich nun also, dass in einem Unternehmen eine gewisse Anzahl an Programmen auf den entsprechenden Systemen installiert ist und dem gegenüber eine entsprechende Anzahl an Lizenzen im Unternehmen zur Verfügung stehen muss, die eben den Einsatz der jeweiligen Software erlauben. Ist nun die Anzahl der Lizenzen, die ein Unternehmen für ein bestimmtes Programm besitzt höher, als die Anzahl der eingesetzten Software, so spricht man von Überlizenzierung.

Überlizenzierung bedeutet also, dass ein Unternehmen zu viele Lizenzen besitzt, die es zum aktuellen Zeitpunkt gar nicht benötigt. Es kann hier also durch die Ermittlung des Verhältnisses von eingesetzter Software zur verfügbaren Softwarelizenz eine Einsparung erfolgen, indem man dieses Verhältnis möglichst gering hält. Idealerweise hat ein Unternehmen genauso viele Lizenzen erworben, wie es für die im Einsatz befindlichen Softwareprodukte benötigt. Da aufgrund von Fluktuationen, sowohl beim Personal, als auch bei der Hardware nicht dauerhaft möglich ist, immer genau die richtige Anzahl an Softwarelizenzen zu besitzen, sollte darauf geachtet werden, dass die Überlizenzierung sich im Rahmen hält. Man sollte also versuchen, immer eine möglichst geringe Überlizenzierung anzustreben.

2.2.5 Unterlizenzierung

Von Unterlizenzierung spricht man, wenn die Anzahl der eingesetzten Programme die Anzahl der Lizenzen, die für diese Programme zur Verfügung stehen, übersteigt. Im Gegensatz zur Überlizenzierung fehlt also dem Anwender die Berechtigung, die entsprechende Software in diesem Ausmaß einzusetzen. Dieser Fall sollte in einem Unternehmen unbedingt vermieden werden, denn ansonsten ist die Sachlage der Unterlizenzierung gegeben und man macht sich strafbar, da man Software betreibt, für

die man keine, oder zumindest nicht die in dem entsprechenden Ausmaß, benötigte Berechtigung hat. Die Compliance ist also nicht gegeben, denn man betreibt mehr Programme, als es durch den Lizenznutzungsvertrag erlaubt ist. Der Anteil der Software, für die es also keine verfügbaren Lizenzen gibt, werden illegal betrieben und dies kann äußerst unangenehme Folgen haben, wie im nächsten Kapitel 2.2.6 Probleme und Folgen von fehlender Lizenzverwaltung beschrieben wird.

Im Allgemeinen gilt, der Zustand einer Unterlizenzierung sollte in jedem Fall vermieden werden, denn spätestens bei einem Software-Audit kann das schwerwiegende Folgen nach sich ziehen. Es empfiehlt sich also eine leichte Überlizenzierung im Unternehmen aufzuweisen, um beispielsweise im Falle einer dringenden zusätzlichen Workspace-Installation auf keinen Fall eine Unterlizenzierung zu riskieren. Je vorausschauender man Softwareinstallationen plant, umso geringer kann man die Überlizenzierung halten.

2.2.6 Probleme und Folgen von fehlender Lizenzverwaltung

Da nun die Sachlage von Softwarelizenzen und deren Verwaltung bekannt sind, soll ein Ausblick darauf gegeben werden, was für Probleme bei schlechter oder gar fehlender Lizenzverwaltung auftreten können und welche Konsequenzen diese mit sich ziehen.

Ein gelebtes Lizenzmanagement ist vor allem deshalb entscheidend, um Verstöße gegen das Urheberrecht vorzubeugen und zu verhindern. Ist es bei einem kleinen Unternehmen noch möglich, die einzelnen Softwarelizenzen zu überschauen, so wird es bei einem stetig wachsenden Betrieb immer komplexer, den Überblick zu behalten.

Allerdings ist es wichtig, dass es auf keinen Fall zur Unterlizenzierung im eigenen Unternehmen kommt, da dies bedeutet, dass nicht ausreichend Softwarelizenzen für die eingesetzte Software zur Verfügung stehen. Um diesen Zustand zu überwachen ist es notwendig, eine Momentaufnahme der Lizenzsituation im Unternehmen herstellen zu können. Dies ist allerdings ein aufwendiger Prozess, wenn man seine Lizenzen nicht bereits vorsorglich verwaltet hat.

Aus diesem Grund entstand der Bedarf nach einem Lizenzmanagement in Unternehmen. Durch das saubere Verwalten von Lizenzen kann jederzeit die aktuelle Situation im Unternehmen analysiert und notfalls reagiert werden. Außerdem erleichtert es die Arbeit im Falle eines Software-Audits, bei welchem das Unternehmen beweisen muss, dass es ausreichend Lizenzen für die aktuelle Softwarenutzung aufweisen kann.

Zu einem vernünftigen Lizenzmanagement gehört aber auch die Kontrolle über die einzelnen Rechner. Während bei einem Server nur wenige Personen eingreifen und somit Software installieren dürfen und können, kann dies auf Clientrechnern schon ganz anders aussehen. Vergibt man nämlich jedem User Adminrechte, läuft man Gefahr, dass diese sich widerrechtlich Software auf einem Firmengerät installieren und damit eine Straftat begehen, welche auch auf das Unternehmen zurückfallen. Zu einem geeigneten Lizenzmanagement gehört also auch die Einteilung der Zuständigkeiten, wer tatsächlich berechtigt ist, neue Software zu installieren und diese auch korrekt dokumentiert beziehungsweise verwaltet.

Neben der Verwaltung der eingesetzten Lizenzen ist es auch wichtig, einen Überblick über die Lizenz-Versionen zu behalten, beispielsweise verlangen Upgrade-Lizenzen eine vorangegangene Vollversionslizenz, welche bei einem Audit ebenfalls vorgewiesen werden muss. Eine Upgrade-Lizenz alleine erlaubt noch nicht den Einsatz der Software. Nur mit dem Vorweis der gesamten Kette bis zur ersten Vollversionslizenz ist die aktuelle Upgrade-Lizenz auch tatsächlich gültig. Auch die Verwaltung dieser ist ein Teil des Lizenzmanagements.

Die Konsequenzen bei einem Verstoß können fatal sein. So ist es unter anderem möglich, dass es neben diversen Strafzahlungen auch zu einem Stopp des Geschäftsbetriebs kommen kann. Auch ein negativer Ruf, durch das Aufscheinen des Verstoßes in der Öffentlichkeit, kann enorme Konsequenzen mit sich ziehen. Es ist beispielsweise möglich, dass aufgrund des Verstoßes mit Vergabesperren auf künftige Aufträge zu rechnen ist. Auch ist es möglich, dass es zu Untersuchungshaft und Freiheitsstrafen von verantwortlichen Managern kommen kann. Bis zur Betriebsstilllegung und einer beruflichen Existenzbedrohung ist alles möglich und muss auch beim Risikomanagement berücksichtigt werden. Eine Auflistung der gesetzlich festgelegten Sanktionierungsmöglichkeiten wird im Kapitel 2.3.8 Sanktionen beschrieben.

Betrachtet man nun das Angebot von Software as a Service, so werden einem einige Aufgaben der Lizenzverwaltung abgenommen. Häufig werden die verwendeten Lizenzen in Übersichtsseiten der SaaS-Anbieter dargestellt und über eine entsprechende Oberfläche verwaltbar. Dies führt dazu, dass die Verwaltungskosten gesenkt werden können und durch die Kernkompetenzfokussierung der Anbieter effizient Prozesse abgedeckt werden können. Neben diesen Vorteilen birgt der Bereich SaaS noch diverse

Nachteile, weshalb sich das Konzept noch nicht vollständig durchgesetzt hat. Einerseits wäre zu erwähnen, dass die Datenhoheit und deren Sicherheit aus der Hand gegeben werden, was auch im Hinblick auf das neue Datenschutzgesetz bedacht werden sollte. Ebenfalls macht man sich hierdurch abhängig von einem Anbieter und muss auf dessen finanzielle Stabilität hoffen, denn fällt der Anbieter weg, kann es sein, dass auch die nötigen Prozesse im Unternehmen nicht mehr durchgeführt werden können. Ein weiterer Punkt, der bei SaaS bedacht werden muss, ist das Problem der Performance und der Verfügbarkeit, welche vor allem von den eingesetzten Internet Service Providern (kurz ISP) abhängt. Die stabile Verbindung und ein entsprechend verfügbarer Datendurchsatz sind zwingend notwendig, um SaaS vernünftig betreiben zu können.

2.3 Rechtliche Sicht⁹

Für den Einsatz und Handhabung von Lizenzen ist auch die rechtliche Sicht entscheidend, welche über (BMDW, 2018) einsehbar ist und vor allem von (Jahnel, Schramm, & Staudegger, 2002) im Hinblick auf die Informatik gut heruntergebrochen wird. Wie schon einige Male erwähnt ist der Ausgangspunkt von Softwarelizenzen erst einmal das Urheberrecht. Doch warum wird dieses überhaupt benötigt?

Ein Kernpunkt der Softwareentwicklung ist die aufwendige Planung und Umsetzung der Software, die einen Mehrwert für potentielle Käufer darstellen soll. Es ist also mit erhöhten Kosten in der Anfangszeit zu rechnen, einerseits weil man genügend Personal braucht, um komplexe Software entwickeln zu können, andererseits auch, weil man die nötigen Spezialisten dafür finden oder einschulen muss.

Im Gegensatz dazu ist die Vervielfältigung von fertiger Software meist leicht umzusetzen, was dafür sorgt, dass Software widerrechtlich verbreitet und eingesetzt werden kann, beispielsweise über die heutzutage sogenannten Raubkopien. Dieser Umstand führt dazu, dass die maßgeblich am Schaffen neuer Software beteiligten Personen einem existenzgefährdenden Entlohnungsentzug gegenüberstehen.

Um dieses Verhalten gesetzlich unter einen sinnvollen Rahmen zu bringen, wurde vom Rat der europäischen Gemeinschaft 1991 die „Richtlinie über den Rechtsschutz von Computerprogrammen“ beschlossen, in welcher unter anderem festgelegt wurde, dass

⁹ (Jahnel, Schramm, & Staudegger, 2002)

es sich bei Software um urheberrechtlich zu schützendes Gut handelt. Diese Richtlinie musste bis 1993 im innerstaatlichen Recht der Gemeinschaftsländer umgesetzt werden.

Im Hinblick auf den anstehenden EG-Beitritt von Österreich am 01.01.1995 wurde schon vorab von der Regierung am 01.03.1993 eine Urheberrechts-Novelle (kurz UrhGNov 1993) beschlossen, die die von der EG vorgeschriebene Richtlinie umsetzt. Es stand im Raum, ob es eventuell sinnvoller wäre, den Urheberrechtsschutz für Software als eigenen Sonderschutz umzusetzen, um Gesetzestexte differenzierter zu anderen urheberrechtlich geschützten Werken formulieren zu können, man entschied sich allerdings zu Gunsten des internationalen Schutzes dagegen. Die Abweichung vom vorhandenen System aus Konventionen wäre international vor allem wegen Ländern der Dritten Welt kaum erreichbar gewesen.

Wie in § 40a Abs 1 UrhG beschrieben, ist das Ergebnis aus eigener geistiger Schöpfung entstandene Werk ein urheberrechtlich zu schützendes Gut. Im Falle eines Computerprogrammes heißt dies also, dass dieses urheberrechtlich geschützt ist, ganz unabhängig von qualitativen oder ästhetischen Kriterien. Auf eine genaue Definition von „Computerprogramm“ wurde bewusst verzichtet, um für etwaige Entwicklung der IT offen zu bleiben, denn unter urheberrechtlich zu schützende Werke fallen laut § 40a Abs 2 UrhG unter anderem auch Maschinencode und jeglichem weiteren Material zur Entwicklung eines Computerprogrammes.

Wichtig hierbei ist allerdings, dass nur die konkrete Umsetzung, inklusive vorhandener Ablaufdiagramme, Schrittfolgenbeschreibungen, sowie Pseudo-Code urheberrechtlich geschützt sind, nicht aber die Idee. So ist unter anderem das Pflichtenheft oder die Schritte zur Systemanalyse, die für eine etwaige Lösungsfindung von Problemstellungen dienen, nicht urheberrechtlich geschützt.

Auch der in der IT oft vorkommende „Algorithmus“ ist urheberrechtlich geschützt, sofern er über allgemeine oder bereits bekannte Grundsätze, Regeln oder Anweisungen hinausgeht. Wie auch schon beim Computerprogramm ist aber nicht der Entwurf eines Algorithmus geschützt, sondern nur die Implementierung eines solchen.

Als Urheber und damit Rechteinhaber von Software gilt eine natürliche Person, die an der Umsetzung des Computerprogrammes maßgeblich beteiligt ist oder war. Sind mehrere Personen an der Entwicklung beteiligt, gilt die Miturheberschaft. Diese tritt

dann in Kraft, wenn die Kreationen der einzelnen beteiligten untrennbar mit dem Gesamtwerk zusammenhängen, was in der IT oft der Fall ist. Jeder Miturheber hat damit dieselben Rechte am Werk und kann Verletzungen des Urheberrechtes gesetzlich verfolgen.

Als Zusatz sei hier angemerkt, dass Software, welche im Sinne einer Anstellung in einem Unternehmen, entwickelt wurde als ein „Dienstnehmerwerk“ im Sinne von § 40b gilt. Der Dienstgeber hat in so einem Fall ein unbeschränktes Werknutzungsrecht, sofern nichts anderes vereinbart worden ist.

Im Folgenden wird näher auf die Rechte von Urhebern eingegangen, sowie die möglichen Sanktionen bei widerrechtlicher Verwendung von Software.

2.3.1 Vervielfältigungsrecht

Das Vervielfältigungsrecht wird in § 15 UrhG näher behandelt. Ausschließlich der Urheber ist befugt, seine Software selbst zu Vervielfältigen oder es Dritten zu gestatten, dies zu tun. Zu erwähnen ist hier, dass im Gesetzestext nicht genau definiert ist, ob das Ausführen einer Software bereits unter den Verstoß des Vervielfältigungsrechtes fällt. Dies begründet sich damit, dass die EU-Richtlinie zwar davon ausgeht, dass dies so sei, allerdings in der Umsetzung des österreichischen Gesetzestextes nicht explizit darauf eingegangen wird, wodurch die Auslegung des Textes Sache der Rechtsprechung ist.

2.3.2 Verbreitungsrecht

Unter Verbreitungsrecht versteht man die Weitergabe an die Öffentlichkeit. Dies kann beispielsweise durch den Verkauf, beziehungsweise Weiterverkauf der Software, durch Tauschen, Schenken, Vermieten oder Verleasen der Fall sein.

In § 16 UrhG wird definiert, dass der Urheber das ausschließliche Recht hat, darüber zu bestimmen, ob sein Computerprogramm kopiert, vermietet oder verliehen werden darf.

Anders sieht dies allerdings beim Weiterverkauf von Software aus, hier greift nämlich § 16 Abs 3 UrhG „Erschöpfung des Verbreitungsrechts“. Das ausschließliche Verbreitungsrecht des Urhebers ist also nicht mehr gültig, wenn es sich um einen Weiterverkauf einer konkreten Computerprogrammkopie handelt. Sofern diese vom System des ursprünglichen Besitzers komplett entfernt wird, ist der Weiterverkauf zulässig und kann nicht vom Urheber verhindert werden. Bei einem Hardwareverkauf

mit mitgelieferter Software, wird diese natürlich bei dem entsprechenden Hardwareverkauf mitübertragen und darf nicht mehr beim ursprünglichen Besitzer auf einer anderen Hardware weiterverwendet werden.

Eine Möglichkeit für Urheber einen Weiterverkauf dennoch einzuschränken, wäre ein mögliches Nutzungsrecht nur für ein bestimmtes Gebiet. Solche etwaigen Einschränkungen sind aber nur über den gesamten EWR- und EFTA-Raum zulässig, eine Beschränkung auf einzelne Mitgliedsstaaten ist ungültig.

Ursprünglich bezog sich die „Erschöpfung des Verbreitungsrechts“ auch auf das Vermieten und Verleihen. Da dies aber als existenzgefährdend für den Softwareentwickler erachtet wurde, wurde im November 1993 der § 16a UrhG eingeführt, welcher das Vermieten von Computerprogrammen vom „Erschöpfen des Verbreitungsrechts“ ausnahm. Es bleibt also dem Urheber vorbehalten, ob er das Vermieten seiner Software durch Dritte erlaubt. Das Verleihen ist allerdings weiterhin auch ohne urheberrechtlichem Einverständnis zulässig, allerdings steht in so einem Fall dem Urheber ein Vergütungsanspruch zu.

2.3.3 Bestimmungsmäßige Benutzung

Nach § 40d Abs 2 UrhG ist nach dem Erwerb einer Softwarelizenz nur die bestimmungsmäßige Benutzung des Programmes erlaubt, allerdings einschließlich der Anpassungen an die eigenen Bedürfnisse. Einschränkende Kriterien der bestimmungsmäßigen Benutzung müssen vorhersehbar sein, ansonsten sind diese bei einem Weiterverkauf der Lizenz ungültig. Beispielsweise kann eine Softwarenutzung auf eine Seriennummer einer bestimmten CPU in einem Unternehmen eingeschränkt werden. Solange die Lizenz im Unternehmen bleibt, ist auch nur diese bestimmungsmäßige Benutzung der Software gültig. Im konkreten Beispiel darf die Software also nicht auf eine neue Hardware mit einer neuen CPU installiert und verwendet werden. Wird die Softwarelizenz allerdings aus dem Unternehmen ausgeschieden und an einen Dritten weiterverkauft, wäre die Beschränkung auf die CPU-Seriennummer nicht vorhersehbar und damit nichtig.

2.3.4 Sicherungskopie

Das Recht auf eine Sicherungskopie war vor 1993 in den meisten Fällen im Nutzungsvertrag bereits inkludiert. Seit 1993 kann laut § 40d UrhG dieses Recht vom Urheber aber auch gar nicht mehr ausgeschlossen werden.

2.3.5 Recht zum Beobachten, Untersuchen und Testen des Funktionierens

Laut § 40d Abs 3 Z 2 UrhG hat der Lizenznehmer das Recht auf die Analyse der zugrundeliegenden Idee und der technischen Besonderheiten der entsprechenden Software. Dies stellt bei Software einen Spezialfall dar, da die Funktionalität nicht durch die menschlichen Sinne erfasst werden können. Das Blackbox-Test-Verfahren, in dem man bestimmte Parameter an eine Funktion übergibt und danach kontrolliert, was die Software wieder zurückliefert, muss also erlaubt sein. Da dieser Test aber nicht ohne die Vervielfältigung möglich wäre und auch die bestimmungsgemäße Benutzung bei so einem Test nicht gegeben wäre, wird im § 40d Abs 3 Z 2 UrhG explizit angeführt, dass dies bei Computerprogrammen erlaubt ist.

2.3.6 Recht auf Dekompilieren

Das Dekompilieren von Software, also das Rückführen von Maschinencode in einen von Menschen lesbaren Quellcode ist unter bestimmten Umständen erlaubt, allerdings nur dann, wenn es sich als unerlässlich erweist. Ein Fall könnte beispielsweise eine Interoperabilität mit einem anderen, unabhängigen Softwareprodukt sein, welches auf die Schnittstellen des Computerprogrammes angewiesen ist. Wird vom Urheber keine andere Möglichkeit gegeben, diese Schnittstellen zu identifizieren, beispielsweise durch eine Schnittstellenoffenlegung), so ist die Dekompilierung der Software erlaubt. Dies ist unter anderem einer der Gründe, warum viele Softwarehersteller seitdem deutlich bessere Schnittstelleninformationen zur Verfügung stellen. Ist eine solche Dokumentation der Schnittstellen vorhanden, darf im angegebenen Beispiel der Programmcode nicht dekompiert werden.

2.3.7 Vervielfältigung zum eigenen Gebrauch

Ein wichtiger Punkt ist noch § 40d UrhG, welcher das Privileg aus § 42, welches die „Erlaubnis zur Vervielfältigung zum eigenen Gebrauch“ darstellt, bei Computerprogrammen untersagt. Das heißt, auch für den eigenen Gebrauch von

Computersoftware ist eine Vervielfältigung nur mit Einverständnis des Urhebers erlaubt.

2.3.8 Sanktionen¹⁰

Ein Verstoß gegen das Urheberrecht kann unterschiedliche Sanktionen nach sich ziehen, im Folgenden werden die relevanten Sanktionen im Bereich der Computerprogramme näher beschrieben:

Unterlassungsanspruch (§81) und Beseitigungsanspruch (§82)

Das Verlangen nach sofortiger Unterlassung vom Einsatz unlizenzierter Software sollte für jeden nachvollziehbar sein. Was man allerdings bedenken muss, ist, dass beim Verlangen einer vollständigen Beseitigung der Software, es zur Folge haben kann, dass ein Unternehmen nicht mehr in der Lage ist, seinen Geschäftstätigkeiten nachzugehen. Dies bewirkt also ein gefährliches Risikopotential, wodurch dieser Umstand auf jeden Fall vermieden werden sollte.

Urteilsveröffentlichung (§ 85)

Der Urheber hat den Anspruch auf eine Urteilsveröffentlichung, welche einen enormen Imageschaden des Unternehmens nach sich ziehen kann. Dieser kann sich einerseits auf die direkten Kunden auswirken, welche von da an die Produkte meiden, es kann aber auch erhebliche Probleme bei Unternehmenspartnern oder Geldgebern hervorrufen.

Anspruch auf angemessenes Entgelt (§ 86)

Ebenfalls kann es zu enormen Nachzahlungen wegen einem angemessenen Entgelt kommen, um den Lizenzschaden zu kompensieren. Dies stellt allerdings nicht die einzige finanzielle Belastung dar.

Anspruch auf Schadenersatz und auf Herausgabe des Gewinnes (§ 87)

Der Urheber hat auch das Recht auf Schadenersatz, sowie eine mögliche Herausgabe des erwirtschafteten Gewinns eines Unternehmens, welches sich strafbar gemacht hat.

¹⁰ (Dietmar Jahnelt, 2003), (BMDW, 2018)

Eingriff (§ 91)

Neben den finanziellen Aspekten kann es sogar zu Freiheitsstrafen von bis zu sechs Monaten kommen, oder eine Geldstrafe von bis zu 360 Tagessätzen.

Freiheitsstrafe bis zu 2 Jahren

Durch die Urhebergesetzesnovelle 1996 wurde ein weiterer Punkt durch Abs 2a hinzugefügt. Und zwar handelt es sich hierbei um eine Freiheitsstrafe von bis zu 2 Jahren, welche bei einer gewerbsmäßigen Begehung der Straftat zum Tragen kommt.

Vernichtung und Unbrauchbarmachung von Eingriffsgegenständen und Eingriffsmitteln (§ 92) und Beschlagnahme (§ 93)

Ebenfalls kann es bis zur Vernichtung oder Unbrauchbarmachung von entsprechenden Eingriffsgegenständen kommen, was wiederum ebenfalls zu einem erheblichen bis existenzgefährdenden Schaden des Unternehmens führen kann.

Es sei auch erwähnt, dass eine Haftung durch den Inhaber eines Unternehmens existiert, wenn dieser über einen Urheberrechtsverstoß seiner Mitarbeiter Bescheid wissen konnte.

2.3.9 Schutzdauer

Die Schutzdauer von Computerprogrammen ist seit der Urheberrechtsgesetz-Novelle 1996 bis 70 Jahre nach Tod des Urhebers aufrecht. Sollte es sich um ein Werk von mehreren Beteiligten handeln, so dauert die Schutzdauer bis 70 Jahre nach dem Tod des letztlebenden Miturhebers an.

2.4 Rahmenbedingung für Lizenzverwaltung

Um ein Lizenzmanagement zu etablieren, müssen gewisse Rahmenbedingungen erfüllt werden, damit das Durchführen der Verwaltung, sowie das Einsetzen einer Toolunterstützung vernünftig funktionieren kann.

Das wichtigste Kriterium für ein funktionierendes Lizenzmanagement ist der Überblick über die vorhandenen Systeme in einem Unternehmen. Oft kommt es vor, dass verschiedene Abteilungen auf unterschiedliche Software setzen, die deren Prozesse unterstützen. Deshalb ist es wichtig, dass beim Erarbeiten eines firmenweiten Lizenzmanagements auch alle Abteilungen miteinbezogen werden. Es empfiehlt sich

hierbei oft, pro Abteilung ein paar Verantwortliche zu ernennen, welche die aktuelle Situation beurteilen, sowie die zukünftig benötigten Anforderungen erarbeiten können. Ist der erste Schritt hierfür getan und die eingesetzten Systeme bekannt, kann versucht werden, etwas mehr Vereinheitlichung in das Unternehmen zu bringen, um eventuelle Softwareersparnisse zu erwirken (beispielsweise durch Mengenrabatte oder aber auch durch bessere Prozessabdeckung von bisher nicht im Einsatz befindlicher Software, sei es aus mangelnder Kenntnis des Produkts oder wegen fehlender Tests, die den Umfang eines Softwareproduktes ermitteln).

Bei der Analyse der vorhandenen Systeme ist ebenfalls wichtig, dass die Klassifizierung der Lizenzbedingungen erfasst werden, um sie vernünftig in einem Unterstützungstool erfassen und verwalten zu können. Hiermit sind beispielsweise die Auslegung der Stückzahlen (also beispielsweise pro User, pro Gerät oder pro CPU) aus den Nutzungsbedingungen der jeweiligen Softwarelizenzen gemeint.

Ist die erstmalige Erfassung erledigt, gilt es organisierte Softwarerichtlinien zu erarbeiten. Nicht jede Software hat eine Daseinsberechtigung, beispielsweise kann Software eingekauft worden sein, von der bestimmte Mitarbeiter das Knowhow ins Unternehmen mitbrachten, und um dieses zu nutzen, wurde in einer Abteilung eben jene Software ausgerollt. Im Nachhinein stellt sich dann aber vielleicht heraus, dass die Software nicht mehr wirklich den Prozessen des Unternehmens dienlich ist und mehr Aufwand als Nutzen produziert, dass aber beispielsweise andere Abteilungen bereits eine Software im Einsatz haben, die viel besser die Bedürfnisse abdeckt. Das Erarbeiten von Softwarerichtlinien, nach denen die einzelnen Softwareprodukte kategorisiert und bewertet werden können, kann dabei helfen, später objektiver an die Selektion von bestimmter Software heranzugehen, um, wie im vorigen Beispiel erwähnt, auch die Mitarbeiter zu überzeugen, dass ein neues Softwareprodukt eventuell sinnvoller für die Abteilung und das Unternehmen wäre, auch wenn diese stark am alten Produkt hängen. Dies soll aber im Umkehrschluss nicht heißen, dass man zwingend unternehmensweit auf ein Softwareprodukt zu setzen hat, manchmal sind unterschiedliche Produkte besser auf die einzelnen Bedürfnisse der unterschiedlichen Abteilungen angepasst. Es gilt hier einen optimalen Einsatz von Software zu ermitteln und nicht auf Biegen und Brechen eine globale Vereinheitlichung zu erreichen.

Genügend Unternehmen arbeiten neben Standardprodukten auch mit Individualsoftware, welche hausintern entwickelt wird oder aber auch an Drittfirmen beauftragt wurde. Da das Unternehmen im Normalfall im Besitz der Individualsoftware ist, ist eine Erfassung hier nicht zwingend notwendig. Es kann sich aber lohnen auch diese Anwendungen zu erfassen, beispielsweise, wenn es einmal zur Entscheidung des Unternehmens kommt, die Individualsoftware durch eine Standardsoftware zu ersetzen. Verwaltet man diese mit, so ist es bei einem Wechsel deutlich leichter zu identifizieren, welche Abteilungen welche Funktionen durch eine neue Standardsoftware ersetzt bekommen müssen, um die auszuscheidende Individualsoftware zu kompensieren.

Ein wichtiger Punkt beim Analysieren von Software ist das Einsatzgebiet. Soll also die Software auf Client-Rechner ausgerollt oder auf einem Server zur Verfügung gestellt werden? Meist sind Serverlizenzenbedingungen deutlich komplexer als jene von Clientlizenzen. Es wird also bei Serverlizenzen eher nötig sein, Details von Hardware und Einsatzgebiet zu ermitteln, denn oft ist die Serverlizenz von den zugreifenden Usern oder der CPU abhängig, während Clientlizenzen meist auf eine pro Client oder pro User Bedingung fokussiert sind. Außerdem werden für Client-Serverzugriffe gerne spezielle Lizenzen erforderlich (siehe Kapitel 2.1.3.5 Client Access License). Im Clientbereich sei weiters zu beachten, dass durch das Vergeben von Adminrechten an die Client-User, den Mitarbeitern ein Weg geboten wird, unerlaubte Software im Unternehmen zu installieren und in Betrieb zu nehmen. Es sollte also darauf geachtet werden, dass die Mitarbeiter entsprechend geschult werden, welche Software sie einsetzen dürfen, oder nur zuständiges Personal Adminrechte erhalten, welches dann die benötigte Software für die Mitarbeiter ausrollt und entsprechend deren Einsatz protokolliert.

Die Protokollierung kann entweder so erfolgen, dass dem gesonderten Lizenzmanagement Bescheid gegeben wird, damit dieses entsprechend den Einsatz der Lizenzen dokumentiert. Es empfiehlt sich aber, die Verantwortlichen für Softwareinstallationen gleich im Lizenzmanagement mit zu integrieren, so können diese direkt den Eintrag beispielsweise in einem möglichen Unterstützungstool durchführen.

Der Einsatz eines Unterstützungstools für das Lizenzmanagement ist äußerst hilfreich, da damit einerseits ein Überblick über die aktuelle Systemlandschaft und den Einsatz von Software in dieser gegeben ist, weiters aber auch die Bestandsaufnahmen neuer Software durchgeführt werden können. Hinzu kommt noch die Möglichkeit, diverse

Aktualisierungen von Software und deren Lizenzen übersichtlich festzuhalten und beispielsweise bei einem Audit direkt die benötigten Daten zur Hand zu haben.

2.5 Systeme¹¹

In diesem Kapitel soll ein Überblick über die bereits auf dem Markt befindlichen Systeme geschaffen werden. Hierzu soll ein Funktionsüberblick gegeben werden und vereinzelt auf gewisse Stärken von unterschiedlichen Produkten hingewiesen werden.

Die unterschiedlichen Asset Management Systeme verfolgen vor allem zwei Ziele. Einerseits gibt es einfacher zu bedienende Versionen, welche darauf abzielen, die Bestände von Lizenzen in Unternehmen abzubilden und übersichtlich zentral zu überwachen. Andererseits gibt es Tools, mit denen sehr detailliert der gesamte Lizenz-Lifecycle abbildbar ist, also neben den aktuellen im Einsatz befindlichen Softwarelizenzen, auch jene darzustellen, welche aktuell nicht eingesetzt werden und zur Verfügung stehen, bis dahin, Funktionalität zum Managen der Ausmusterung von Lizenzen oder der Beendigung von gemieteter Software zu bieten.

2.5.1 Inbetriebnahme

Die komplexeren Tools haben meist das Problem der umfangreichen Erstkonfiguration und Inbetriebnahme. Hier wird empfohlen, einen entsprechenden Techniker mit gegebenen Know-How als Unterstützung bei der Installation heranzuziehen. Es gibt aber auch simplere Tools, wie das „Maximo Asset Tracker“ von MRO Software, welches über eine einfache Installationsroutine ausgerollt werden kann.

Ein weiterer Punkt nach der Installation eines Servers, der für die entsprechende zentrale Schnittstelle sorgt, müssen meist die entsprechenden Analysewerkzeuge auf die einzelnen Clients ausgerollt werden. Dies kann meist auf unterschiedlichste Art erfolgen, die verbreitetsten sind hier das Versenden per Mail, das Herunterladen der Software vom zentralen Server oder das Ausrollen mit entsprechenden Unternehmensroutinen.

¹¹ (Terzievski, 2004), (Creamer, 2017), (Gartner Peer Insights, 2018), (Microsoft, 2018)

2.5.2 Funktionsumfang

Bei der Bestandsverwaltung werden meist konkrete Darstellungen von Unterteilungen geboten, wo, welche Lizenz wie oft zum Einsatz kommt, um darüber auch eine Kosten-Nutzen-Analyse fahren zu können.

Das Auslesen der im Einsatz befindlichen Software erfolgt über diverse Hilfsmittel. So wird unter anderem die Windows Management Instrumentation verwendet, um Software zu identifizieren. Weiters tragen Einträge in der Registry und das Auslesen von *.ini-Dateien dazu bei, die konkret im Einsatz befindlichen Softwareinstallationen, sowie deren Konfiguration und damit deren Einsatzgebiet zu identifizieren. Das Tool „Microsoft Distribution and Asset Management Services“ bietet hier noch zusätzlich die Möglichkeit die Active Directory System Discovery zu verwenden, um entsprechende Programme ausfindig zu machen.

Die meisten Tools verfügen zusätzlich über eine Netzwerkanalyse, bei der das Firmennetzwerk gescannt wird, um automatisiert ein Abbild der Systemlandschaft erstellen zu können, sowie eine Übersicht über die verwendete Hardware zu bieten, inklusive derer Firmwareversion.

Einige Tools, unter anderem die „Symantec Asset Management Suite“ bieten die Möglichkeit an, genaue Trackings durchzuführen, wie oft und wie lange eine bestimmte Software verwendet wird, um deren Nutzen genauer analysieren zu können.

Ebenfalls gemeinsam haben die meisten Tools, dass sie ein entsprechendes Reporting und einen Export der Analysen ermöglichen, sowie zumindest alternativ eine Webübersicht zur Verwaltung zur Verfügung stellen.

Die „Microsoft Distribution and Asset Management Services“ bietet unter anderem noch ein Zusatz-Feature, nämlich das Ausrollen von Patches für veraltet Software. Auch Remote-Unterstützungen werden von mehreren Tools zusätzlich angeboten. Zu Bedenken ist, dass diese Zusatzfeatures meist separat zu erwerben sind.

Um nun einen Vergleich zwischen den auf dem Markt vorhandenen Systemen und der hier ausgearbeiteten Umsetzung näher zu erklären, wird die aktuelle Situation im folgenden Kapitel 3 Ausgangssituation / Use Case näher beschrieben. Des Weiteren wird in Kapitel 3.4 Use Case für das neue Unterstützungstool ein Vergleich über die Use Cases

der hier erwähnten Programme, sowie der in der Masterthesis ausgearbeiteten Umsetzung näher erläutert.

3. Ausgangssituation / Use Case

Da nun ein Überblick über die Theorie gegeben wurde, erfolgt im folgenden Kapitel eine kurze Beschreibung des Unternehmens DBConcepts GmbH, für welches das Unterstützungstool verwirklicht werden soll. Es wird ein Einblick über die Tätigkeiten und Produkte des Unternehmens gegeben, sowie eine Ausführung des Einsatzgebietes des Unterstützungstools zum Lizenzmanagement.

3.1 Das Unternehmen

Das Unternehmen DBConcepts GmbH ist in Österreich seit dem Jahr 2000 im Bereich Datenbanksupport tätig. Es umfasst über 20 Mitarbeiter und ist stetig am Wachsen. Mittlerweile haben sich die Aufgabengebiete um einige Bereiche erweitert. Neben der Wartung, sind unter anderem auch die Erstinstallation von Systemen, Erstellung von Individualsoftware und der Vertrieb von Hardware und Softwarelizenzen wichtige Themenbereiche geworden. Das Unternehmen gliedert sich in mehrere Abteilungen, in denen es aber zu Überschneidungen der Tätigkeiten kommt, weswegen hier nun mehr auf die Aufgabenbereiche anstatt auf die Abteilungen eingegangen werden soll.

3.2 Aufgaben des Unternehmens

Als Oracle Partner bietet das Unternehmen nicht nur den Vertrieb von Lizenzen und Hardware, sondern auch kompetenten Support für den Betrieb von Oracle-Datenbanken und Systemen, sowie auch die Erstellung von Individualsoftware im Bereich der Oracle-Infrastruktur sowie ein Cloud-basiertes Angebot. Im Folgenden wird ein kurzer Überblick über die wichtigsten Aufgaben des Unternehmens gegeben, wodurch anschließend die Basis der Anforderung in Kapitel 4 Anforderungsdefinition für ein Lizenzmanagement-Unterstützungstool ersichtlicher wird.

3.2.1 Datenbank- und Systemadministration

Ein wichtiges Kerngebiet ist das Überwachen von Oracle-Datenbanken und -Infrastruktur, sowie von Systemen, die meist für den Betrieb von Oracle-Architektur verwendet werden. Hierfür werden spezielle Überwachungsmethoden verwendet, die darauf ausgelegt sind, den Zustand von in Wartung befindlichen Datenbanken und Servern zu ermitteln und ständig zu monitoren, um rasch auf etwaige Engpässe oder kritische Zustände reagieren zu können.

Ein weiteres Fachgebiet ist auch das Aufsetzen einer solchen Infrastruktur inklusive der Erstinstallation und Grundkonfiguration von Oracle-Systemen, die daraufhin vom Kunden in die eigene Umgebung eingebunden und in Betrieb genommen werden. Ebenfalls werden die Bereiche Wartung und Upgrade von Server und Datenbanksystemen von DBConcepts GmbH angeboten und durchgeführt.

3.2.2 Entwicklung von Individualsoftware und Consultant

Ein weiterer Aufgabenbereich ist das Erstellen von Individualsoftware, welche auf Kundenwünsche hin optimiert und ständig erweitert, sowie gewartet werden. Diese bauen meist auf Basis von Oracle-Architekturen auf, wie beispielsweise Oracle Forms, Oracle BI Analytics und Oracle Apex. Des Weiteren werden hier Schwerpunkte auf Consultant gesetzt, um unterstützend bei Projekten von Partnern mitwirken zu können. Ein Schwerpunkt hierbei ist die Performanceoptimierung von Datenbank-Prozessen.

Neben den applikatorischen Expertisen, liefert man auch Empfehlungen und Unterstützung aus dem Themenbereich Business Intelligence. Einerseits ist hier der Bereich von Migration von BI-Lösungen auf aktuelle Systeme gefragt, aber auch die Umsetzung von Datenintegrationsfunktionalität, die für ein darauf aufgesetztes Data-Warehouse aufbereitet werden sollen.

3.2.3 Cloud-Lösungen

Eine Verschmelzung aus den beiden Bereichen Administration und Entwicklung bildet der Bereich für Cloudlösungen, bei dem die Wartung der von den Kunden benötigten Systeme vollends durch DBConcepts GmbH übernommen, durchgeführt und gehostet wird. Hierbei werden für Kunden maßgeschneiderte Lösungen mit dessen Zusammenarbeit ausgearbeitet und danach vollständig von DBConcepts GmbH betreut. Diese Anforderungen sind für jeden Kunden unterschiedlich, so werden unter anderem nur das Datenbanksystem gehostet, oder wiederum Individualsoftwarelösungen inklusive deren Server-Architektur, oder auch komplette Webauftritte von Kunden.

3.2.4 Vertrieb von Hardware

Ein weiteres Aufgabengebiet ist der Vertrieb von Hardware, welche sich meist auf Serverlösungen bezieht. Hier werden mit dem Kunden mögliche Lösungen von bestimmten Anforderungen, wie beispielsweise Backup-Server und Ausfallsicherheitskonzepte durchbesprochen, um eine vernünftige Lösung zur

Hardwareanschaffung zu ermöglichen. Meist erfolgt nach einem Hardwarekauf auch eine entsprechende Grundkonfiguration der Server, um sie direkt nach Erhalt in Betrieb nehmen zu können.

3.2.5 Vertrieb von Lizenzen

Der letzte und wahrscheinlich wichtigste Aufgabenbereich für diese Masterthesis ist der Vertrieb von Lizenzen. Meist in Verbindung mit Hardwareverkäufen und mitinstallierter Software, aber auch unabhängig davon, werden Kunden beraten, welche Lizenz am sinnvollsten für den gewünschten Einsatz der Software wäre. Im Konkreten handelt es sich hier hauptsächlich, aber nicht ausschließlich, um Oracle-Lizenzen. Die entsprechenden Informationen werden hier von Oracle in PDF oder Excel-Sheet-Form geliefert und es gilt als Aufgabenbereich der Vertriebsabteilung, hier die geeigneten Tätigkeitsfelder der einzelnen Angebote zu identifizieren.

3.2.6 Weitere Tätigkeitsfelder

Zuletzt sei hier der Vollständigkeit halber noch erwähnt, dass natürlich auch weitere Aufgaben, wie das Marketing, diverse Schulungsangebote, die Buchhaltung und weitere verwaltungstechnische Bereiche in den jeweiligen Abteilungen abgedeckt werden. Da diese aber nicht im direkten Zusammenhang mit dieser Masterthesis stehen, wird auf die ausführliche Aufgliederung hierbei verzichtet.

3.3 Aktuelles System

Die internen Arbeiten werden durch eine in Eigenentwicklung befindliche Software unterstützt. Diese Software umfasst sowohl ein CRM für die Kundenwartung, eine Abrechnungsschnittstelle für diverse Verrechnungen und ein Ticketsystem, bei dem Kunden mittels Tickets Fehler melden können und diese dann von der entsprechenden Abteilung entgegengenommen und bearbeitet werden. Alternativ wird das Ticketsystem ebenfalls dazu verwendet, den Kunden zu informieren, wenn Probleme an den zu überwachenden Systemen festgestellt wurden, um die weiteren Schritte zu klären, oder die bereits durchgeführte Bereinigung der Fehler zu protokollieren.

Da Kunden oftmals mehrere Server, ob virtuelle oder reale Maschinen, im Einsatz haben, auf denen unterschiedlichste Software in Betrieb ist, wurde im System bereits eine Abbildung der Systemlandschaft ermöglicht, sofern der Kunde dies wünscht. Diese grobe Systemabbildung hilft dabei, die entsprechenden Server, auf denen gerade Fehler

auftreten oder bei denen ein Eingriff erforderlich ist, möglichst effizient zu identifizieren. So ist es nicht notwendig erst mit dem Kunden abzuklären, um welche Systeme es sich konkret handelt, bei denen weitere Schritte erforderlich sind, sondern es werden konkret die betroffenen Infrastrukturen ins Ticket eingebunden, wodurch die Kommunikation zwischen Dienstleister und Kunden erleichtert wird.

3.4 Use Case für das neue Unterstützungstool

Da im Unternehmen auch Lizenzen vertrieben und verwaltet werden, gibt es nun die Überlegung, diese Verwaltung an das vorhandene Infrastrukturmodell anzuknüpfen. Da die aktuelle Situation bei einer Neuanschaffung von Software (sowohl inklusive als auch exklusive entsprechender Hardware) für Lizenzen die ist, dass von Oracle zur Verfügung gestellte PDFs durchgearbeitet, und dann die geeignetsten Lizenzen ermittelt werden müssen, kam die Idee auf, diese Schritte zu automatisieren.

Im ersten Schritt wäre es also wünschenswert, dass die Lizenzbedingungen und Kosten von Oracle-Lizenzen im System erfasst werden können, um zu einem Katalog aufgebaut zu werden. Diese sollen anschließend dazu dienen, bei Neuanschaffungen gezielter entsprechende Lizenzen, die die Anforderungen abdecken, finden zu können, um so kosteneffiziente Angebote erstellen zu können. Dieser Prozess wird im folgenden Kundenanforderung oder Anforderungsanalyse genannt.

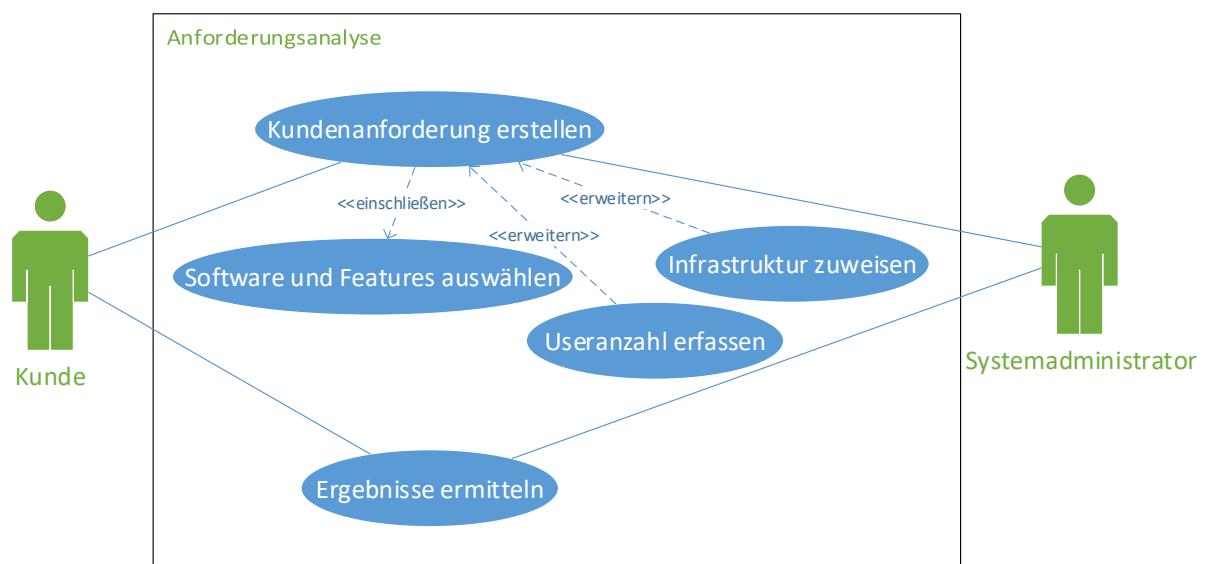


Abbildung 3: Use Case der Anforderungsanalyse

Wie in Abbildung 3: Use Case der Anforderungsanalyse ersichtlich, ist die Anforderungsanalyse ein wichtiger Bestandteil der Implementierung. Die in Kapitel 2.5

Systeme analysierten Tools am Markt unterstützen diesen Schritt nicht oder nur in eingeschränktem Umfang. Es stellt aber eine bedeutende Unterstützung zwischen der Anforderungs- und Beschaffungsphase dar. Nachdem ein Kunde ermittelt hat, welche Software am sinnvollsten die Unternehmensprozesse unterstützt und somit eingesetzt werden sollte, gilt es zu ermitteln, wie der Softwareeinsatz am kosteneffizientesten mit den am Markt verfügbaren Lizenzangebot abzudecken ist.

Es soll dem Kunden ermöglicht werden, eine Anforderung zu erstellen, in der er angibt, welche Software er anschaffen möchte, welche Features dieser Software für das Unternehmen und deren Prozesse relevant sind, sowie wo und in welchem Umfang die Software in Betrieb genommen werden sollte.

Anhand der vom Kunden eingegebenen Eckdaten soll nun vom System ermittelt werden, welche Lizenzoptionen zur Verfügung stehen, und welche davon am sinnvollsten und kostengünstigsten wären.

In weiterer Folge soll natürlich auch die bereits erfasste Infrastruktur der Kunden um ein mögliches Lizenzmanagement, vorerst für die von DBConcepts GmbH angebotenen Lizenzen, erweitert werden, damit diese einen möglichst guten Überblick über die Leistungen von DBConcepts GmbH aufgelistet bekommen. Zukünftig soll auch der Kunde selbst Lizenzen erfassen können, um die im Kundenunternehmen bereits vorhandenen Lizenzen diverser Anbieter zu erfassen und diese auch verwalten zu können, wie es auch die in Kapitel 2.5 Systeme erwähnten Tools unterstützen.

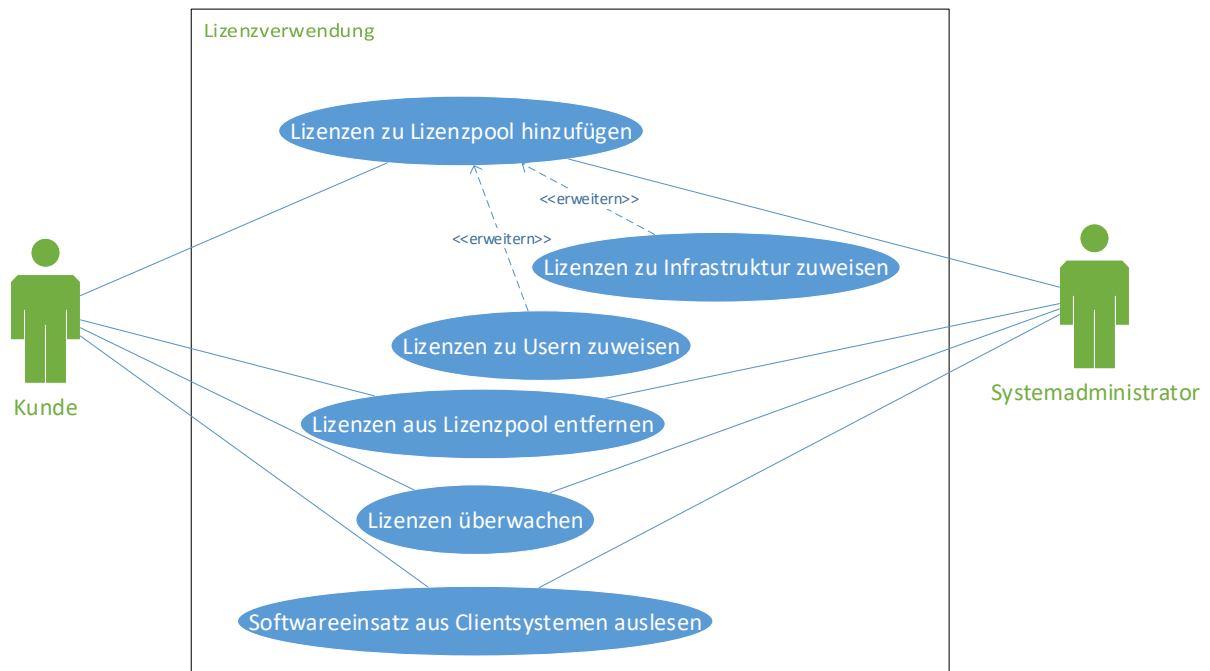


Abbildung 4: Use Case der Lizenzverwaltung

Der in Abbildung 4: Use Case der Lizenzverwaltung dargestellte Use Case zeigt die Standardaufgabe eines Lizenzmanagementtools. Es sollen Lizenzen zu einem Lizenzpool eines Kunden zugewiesen, verwaltet und auch wieder entfernt werden können. Des Weiteren sollen die eingesetzten Lizenzen überwacht werden, um Unter- und Überlizenzierung zu vermeiden und bei etwaigen Lizenz-Audits vorbereitet zu sein. Was die meisten analysierten Tools auch aufweisen, ist eine automatisierte Analyse der auf den einzelnen Clientsystemen tatsächlich eingesetzten Software, um die Erfassung und Verwaltung der Lizenzen teilweise automatisieren und erleichtern zu können.

Die durch die Lizenzverwaltung bereits erfassten Lizenzen sollen auch dazu dienen, um bei einer möglichen Überlizenzierung, die freien Lizenzen direkt mit einzukalkulieren und so eventuell eine kostengünstigere Neuanschaffung bei der Anforderungsanalyse zu ermitteln. Andere Anwendungsfälle wären ein Upgrade von vorhandenen Lizenzen auf neuere Versionen. Hier können die vorhandenen Informationen helfen, zu identifizieren, ob ein Upgrade aktuell sinnvoll erscheint, oder ob dies einen Feature-Verlust von in Verwendung befindlichen Features zur Folge hätte. Ebenfalls soll ermittelt werden können, wie die durch die Anschaffung von neueren Versionen weggefallenen Features wieder abgedeckt werden könnten.

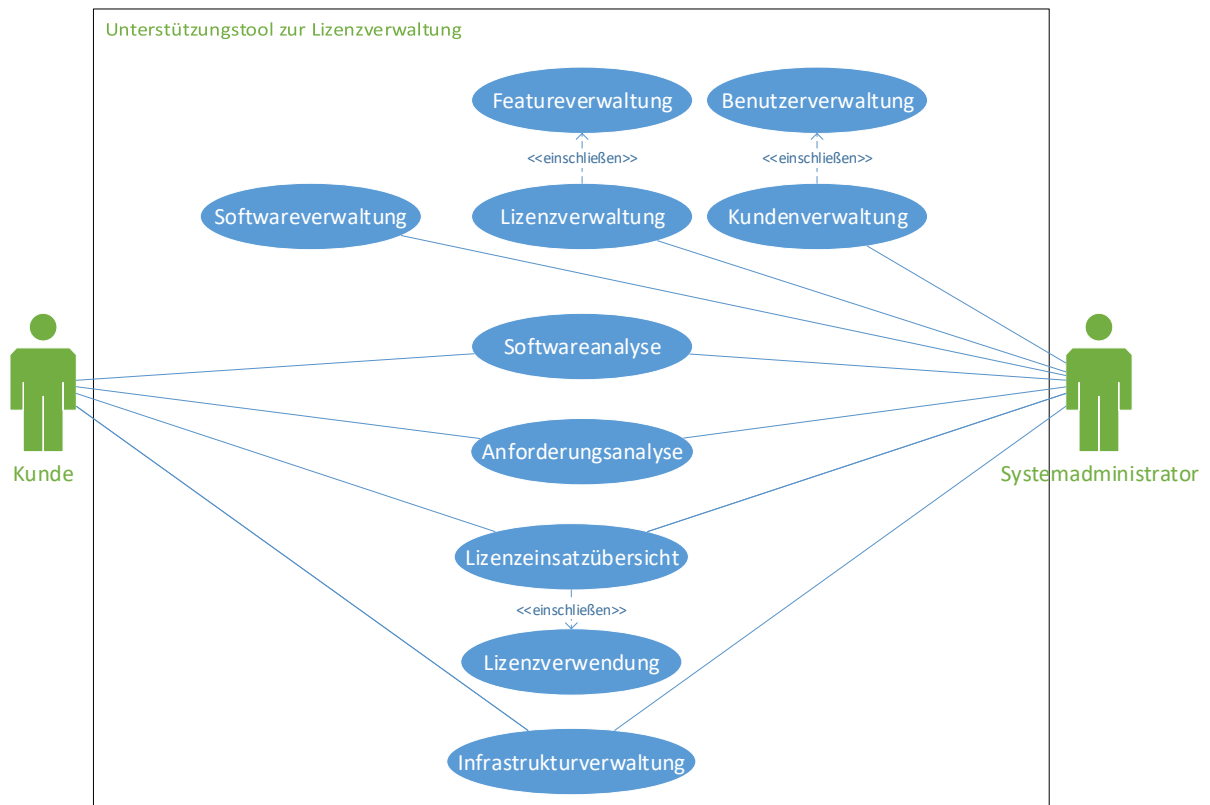


Abbildung 5: Use Case des Unterstützungstools zur Lizenzverwaltung

In Abbildung 5: Use Case des Unterstützungstools zur Lizenzverwaltung ist ersichtlich, in welche Teilkomponenten die in dieser Arbeit ausgearbeitete Implementierung gegliedert werden soll. Für die Systemadministratoren steht die Verwaltung der Stammdaten wie die Software, Features und Lizenzen zur Verfügung. Ebenfalls in den Aufgabenbereich der Systemadministratoren fallen die Verwaltung der Kunden und deren Benutzer.

Allgemein zugänglich sind dann die Bereiche, die die Unternehmen beim Einsatz mit Lizenzen in den einzelnen Phasen der Lizenzverwaltung (wie in Kapitel 2.2.2 Lizenz Life Cycle beschrieben) unterstützen sollen. Für die Anforderungsphase soll die Softwareanalyse zur Verfügung stehen, um mögliche Softwarekandidaten auswerten und bewerten zu können. Anschließend soll in der Anforderungsanalyse eine passende Lizenz zur Abdeckung des geplanten Softwareeinsatzes ermittelt werden können, welche dann in der Beschaffungsphase erworben werden kann. Der tatsächliche Einsatz, welcher die technischen Prozesse abdeckt, soll mit der Lizenzverwendung abgedeckt werden, welche auch eine Übersicht über die aktuelle Lizenzsituation im jeweiligen Unternehmen liefern soll. Um die Anforderungsanalyse und Lizenzverwendung voll

nutzen zu können, wird die Abbildung der jeweiligen Infrastruktur über eine Infrastrukturverwaltung ermöglicht.

Durch die geschilderte Darstellung der Use Cases der Implementierung, können im folgenden Kapitel die entsprechenden Anforderungen an das System definiert werden.

4. Anforderungsdefinition

An das bestehende Infrastruktursystem von DBConcepts GmbH, welches einen Teil des Ticketsystems darstellt, soll ein Unterstützungstool zur Lizenzverwaltung angeknüpft werden, um den Lizenzvertrieb künftig effizienter zu unterstützen, indem ein besserer Überblick über die vorhandenen Lizenzvariationen geboten wird, sowie eine umfangreiche Filterung von in Frage kommenden Lizenzen, auch in Anbetracht der bereits vorhandenen Lizenzen, ermöglicht wird.

Um dies zu ermöglichen, soll im ersten Schritt die Infrastrukturerfassung erweitert werden, um neue benötigte abstraktere Typen wie „Abteilungen“ zuzulassen, da diese benötigt werden, um Mehrplatzlizenzen effektiver an entsprechende Infrastrukturen zu koppeln.

Außerdem muss es Systemadministratoren (derzeit ausschließlich DBConcepts-Mitarbeiter geplant, theoretisch kann man die Berechtigung aber an jeden Benutzer vergeben) ermöglicht werden, diverse Stammdaten zu erfassen, zu bearbeiten und zu löschen. Darunter zählen die Infrastrukturkomponenten, die Softwareangebote, die Lizenzen, sowie dynamische Werte zur erweiterten Anforderungsdefinition von Lizenzen. Die genaue Erfassung wird im Folgenden beschrieben. Diese soll zudem auch die Bearbeitung und Löschung von den angeführten Datensätzen inkludieren:

Um einen Softwarepool aufzubauen, soll es den Systemadministratoren ermöglicht werden, neue Software zu erfassen und ihr entsprechende Versionsnummern zuzuweisen.

Ebenfalls benötigt wird die Erfassung von Lizenzen, deren Form, Klasse, Typ, Art und deren Anforderungen. Diese Anforderungen bestehen aus Minimal- und Maximal-Werten, sowie die exakte Abdeckung der Lizenz zum angegebenen Preis. Die fest definierten abzubildenden Werte, die für die einzelnen Lizenzen vergeben werden können sollen, sind im Kapitel 5.3.4 Lizenzbereich in der Tabelle 29: Spaltendefinition der Tabelle LMT_LIZENZEN detailliert beschrieben.

Etwaige weitere Daten für Lizenzbedingungen, die von den üblichen Angaben abweichen, sollen dynamisch erweitert und erfasst werden können. Simple Vergleiche sollten direkt umsetzbar sein. Handelt es sich hierbei aber um komplexere

Anforderungen, müssen diese durch eine Codeerweiterung in den entsprechenden Funktionen stetig erweitert werden.

Anschließend soll es ermöglicht werden, die im System erfassten Lizenzen einer entsprechenden Software zuzuweisen, dessen Einsatz sie abdeckt. Zusätzlich soll hierbei noch die Information mitgespeichert werden, welche Features dieser Software durch die Lizenz ermöglicht werden. Ebenfalls soll deklariert werden können, welche Infrastrukturtypen bei der Anforderungsanalyse durch diese Lizenz abgedeckt werden, damit in späterer Folge auch nur die entsprechenden Infrastrukturen bei einer Kundenbedingung erfasst werden können.

Auch soll es möglich sein, den einzelnen Lizenzen Preistabellen zuzuordnen, um entsprechende Mengenrabatte abbilden zu können. Ab dem jeweils konfigurierten Wert wird statt des original eingetragenen Preises, der Preis aus der Preistabelle herangezogen. Hierbei zählt immer der Preis der nächst niedrigeren Mengenangabe in der Preistabelle. Der angegebene Preis stellt immer den Stückpreis dar und muss mit der entsprechenden Stückzahl multipliziert werden.

Es soll außerdem ermöglicht werden, die Lizenzen miteinander zu Verknüpfen. Dies soll einerseits dazu dienen, gleiche Lizenzen mit unterschiedlichen Bedingungen als gegenseitig konvertierbar zu markieren, andererseits sollen so Upgradelizenzen mit deren Vorgängerlizenz verknüpft werden, um feststellen zu können, welche Lizenzen durch die Upgradlizenzen auch wirklich upgegradet werden dürfen. Ebenfalls sollen Lizenzen für Addons mit der entsprechend benötigten Vollversionslizenz verknüpft werden können.

Für die unterschiedlichen Kunden, die auf das System zugreifen, soll es nun möglich sein, für sein jeweiliges Unternehmen, dem sie zugewiesen sind, Softwarebewertungen durchführen zu können. Dazu ist eine Maske erforderlich, in der ein Kunde seine gewünschte Software aus dem verfügbaren Softwarepool des Tools auswählen und dieser Software nun Bewertungen sowie Kommentare und einen Status anfügen kann, um im späteren Verlauf besser entscheiden zu können, ob die gewählte Software sinnvoll in das jeweilige Unternehmen eingegliedert werden kann.

Als nächsten Schritt soll es dem Kunden ermöglicht werden, eine entsprechende Kundenanforderung zu stellen, um ermitteln zu können, welche Lizenz am

kosteneffizientesten die Bedürfnisse des Kunden, sowie den Einsatz der Software im Unternehmen abdecken kann. Hierzu kann der Kunde Eckdaten wie die gewünschte Benutzer- sowie Geräteanzahl angeben und einen Zeitraum definieren, in der die Software eingesetzt werden soll. All diese Angaben sind optional. Verpflichtend ist hingegen, dass der Kunde als nächstes eine gewünschte Software auswählt und eventuell direkt die entsprechende Version. Anschließend muss er noch mindestens ein Feature auswählen, welches von der Software abgedeckt werden soll. Des Weiteren kann optional noch eine passende Infrastruktur zur Kundenanforderung hinzugefügt werden, um die Ermittlung von passenden Lizenzen mit entsprechenden Kennzahlen des Unternehmens zu optimieren. Der Pool der möglichen Infrastrukturtypen ergibt sich aus den möglichen Zuweisung der Lizenzen (welche die gewünschte Software abdecken) zu den Infrastrukturtypen, die bei der Erfassung der Lizenzen im System vom Systemadministrator zugewiesen wurden. Als letzten Punkt können noch dynamische Werte zusätzlich erfasst werden, falls die Lizenzen (welche die gewünschte Software abdecken) diese benötigen.

Sind nun alle Kundenbedingungen zur entsprechenden Anforderung erfasst, kann das Ergebnis abgerufen werden, welches alle Lizenzen ermittelt, die die gewünschten Features unter den von der Lizenz verlangten Bedingungen abdeckt. Diese werden nach entsprechender Featureabdeckung (das heißt die Lizenz deckt das Feature ab, welches vom Kunden gewünscht ist) sowie nach dem resultierenden Preis sortiert und ausgegeben, sodass die idealste und kostengünstigste Lösung an erster Stelle steht. Das Ergebnis kann anschließend vom Kunden beliebig gefiltert und sortiert werden.

Nachdem nun eine Lizenz vom Kunden ermittelt ist, kann diese erworben werden und entsprechend dem Kundenlizenzpools im System hinzugefügt werden. Etwaige Bestätigungen zum Erwerb der Lizenz können im entsprechenden Pooleintrag hinzugefügt werden. Im Kundenlizenzpools kann nun die aktuelle Lizenzverwendung verwaltet werden. Einerseits können die Lizenzpools durch Upgrade- oder Addon-Lizenzen erweitert werden und eine entsprechende Beziehung zur vorangegangenen Lizenz erstellt werden. Andererseits können die im Pool befindlichen Lizenzen nun verwendet werden, indem man entweder eine bestimmte Anzahl an Lizenzen aus dem Pool als verwendet deklariert, oder aber direkt eine Infrastruktur der entsprechenden Lizenz zuweist, wodurch das System anhand der Infrastrukturkonstellation

automatisiert die jeweilige Menge an Lizenzen im Lizenzpool als in Verwendung deklariert.

Zum Schluss soll noch eine Übersichtsseite geschaffen werden, auf der der Kunde direkt die Compliance des eigenen Unternehmens ablesen kann, um schnell feststellen zu können, ob eine Unterlizenzierung vorhanden ist, oder ob es eventuell Überlizenzierungen von bestimmten Lizenzen gibt. Hierzu sollen die einzelnen Infrastrukturen und die darauf zugewiesenen Lizenzen als Tree dargestellt und farblich angezeigt werden, um eine fehlende oder noch nicht zugewiesene Infrastrukturkomponente direkt erkennen zu können. Der Lizenzpool und eine eventuelle Knappheit von Lizenzen soll ebenfalls farblich dargestellt werden, damit rechtzeitig entsprechend reagiert werden kann und eventuell neue Lizenzen zeitgerecht nachbestellt werden können.

Abseits der manuellen Bedienung der Applikation soll es auch noch ermöglicht werden, Software auf den einzelnen im Unternehmen verteilten Systemen zu ermitteln und zu protokollieren, um eine entsprechende Lizenzabdeckung nachträglich zu kontrollieren. Hierzu soll ein Tool erstellt werden, welches auf die einzelnen Systeme im Unternehmen verteilt werden kann. Über die Systemdaten sollen mittels dieses Tools entsprechende Infos über die installierten und verwendeten Anwendungen auf den einzelnen Geräten gesammelt und entweder direkt per Webservice an das zentrale Lizenzmanagementtool übertragen, oder aber bei nicht vorhanden sein einer Verbindung eine XML-Datei erstellt werden, welche nachträglich über die Benutzeroberfläche importiert werden kann.

5. Modell

Aufgrund der definierten Anforderungen folgen nun diverse Modelle zur Verwirklichung des Unterstützungstools für das Lizenzmanagement. Im Nachfolgenden soll der Ablauf des Nutzungsverhaltens der Applikation erläutert werden. Außerdem soll eine Übersicht über die Funktionalität des eingesetzten Frameworks nähergebracht werden, um die Umsetzung näher zu beschreiben. Das Kernstück der Applikation bilden die Datenhaltung und -abbildung, welche mittels ER-Diagramm beschrieben werden.

5.1 Oracle Application Express - Framework

Das Unterstützungstool für das Lizenzmanagement soll mit Oracle Application Express (kurz APEX) umgesetzt werden. Es handelt sich hierbei um ein Webapplikationsframework, welches ein kostenloses Feature einer Oracle Datenbank (welches auch die kostenfreie Oracle Express Edition, kurz Oracle XE, sein kann) darstellt. Der Vorteil dieses Frameworks ist, effizient diverse Datenbank Anwendungen erstellen zu können.

APEX wird direkt bei der Installation einer Oracle Datenbank bereitgestellt, kann aber auch nachträglich auf Oracle-Datenbanken installiert oder auch auf eine neuere Version aktualisiert werden. Der Download steht im Oracle Technology Network unter der URL <http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html> bereit.

Die Installation erfolgt über die Ausführung eines SQL-Skripts, welches in der Datenbank das benötigten Schema erzeugt und in diesem diverse benötigte Objekte anlegt, unter anderem Tabellen zur Speicherung der Applikationsstrukturen und Konfigurationen, sowie Packages zur Abhandlung von auszuführendem Code.

APEX selbst läuft in der Datenbank und kreiert mittels des Standardpackages "http" und "htf" HTML-Code. Dieser wird mittels einer JDBC-Connection über die Oracle REST Data Services (kurz ORDS) an einen Applikationsserver übermittelt. Als Applikationsserver können sowohl Oracle Weblogic, als auch Glassfish oder Apache Tomcat verwendet werden. Auf diese wird das entsprechend konfigurierte ORDS-File eingespielt und als Applikation zur Verfügung gestellt. Nun kann über den entsprechend konfigurierten Port am Applikationsserver die APEX-Applikation über einen Browser aufgerufen werden.



Abbildung 6: Systemabbildung Oracle Apex¹²

Üblicherweise werden die entsprechenden Web-Applikationen hinter einem Applikationsserver noch über einen Webserver, wie den Apache HTTP Server oder Oracle HTTP Server geführt. Diese Konfiguration ist allerdings optional. In der folgenden Abbildung ist beispielsweise ersichtlich, wie die Umsetzung von APEX in einer reinen Oracle-Architektur aussehen könnte.

¹² (Saygili, 2015)

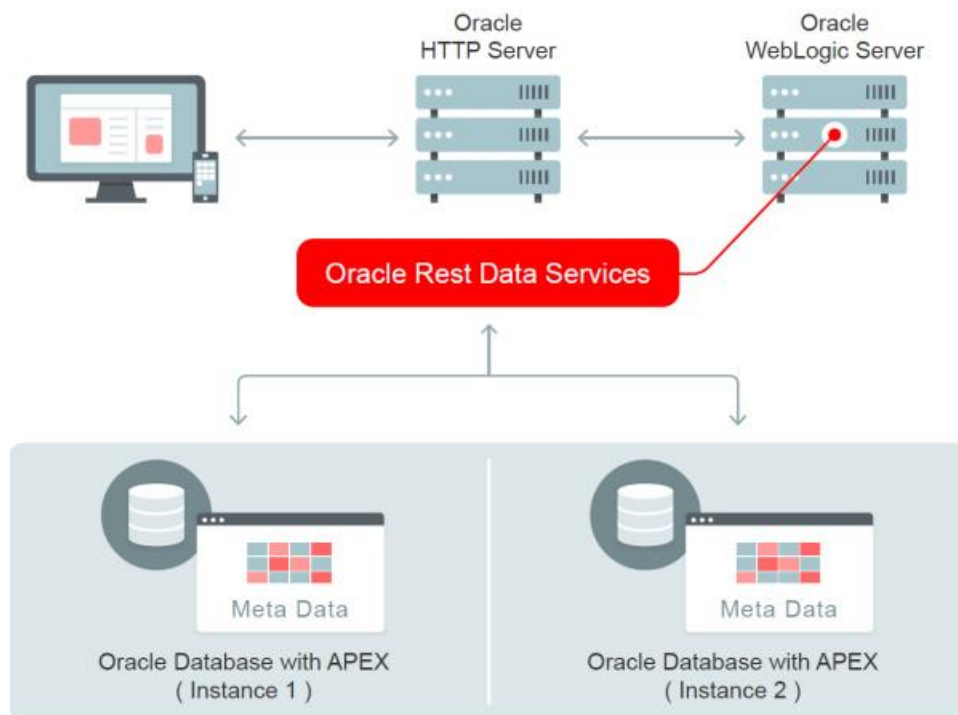


Abbildung 7: Oracle Apex Konfiguration mit Oracle-Komponenten¹³

Wie in der Abbildung ersichtlich, kann man nicht nur eine APEX-Instanz über ORDS laufen lassen, sondern beliebig viele. So kann beispielsweise sowohl das Test als auch das Produktivsystem über denselben Applikationsserver laufen, während man aber im Hintergrund auf verschiedene Datenbankinstanzen zugreift.

Ist nun APEX installiert und konfiguriert, so kann über eine entsprechende URL im Browser auf das Framework zugegriffen werden. Über die Default-Startseite gelangt man auf einer frischen Oracle APEX-Installation auf die Entwicklungsoberfläche des Frameworks. Denn auch die Entwicklungsoberfläche ist eine APEX-Applikation, welche im Framework eingebunden ist. Diese muss nicht zwingend installiert sein, beispielsweise wird in Produktivsystemen davon abgeraten, die Entwicklungsumgebung zur Verfügung zu stellen, um Angriffsflächen, welche unter anderem durch schlechte Entwicklerpasswörter entstehen könnten, zu minimieren und die Sicherheit der Applikation zu erhöhen.

In APEX können nun Workspaces definiert werden, welche wiederum auf ein entsprechendes Nutzdatenschema zugreifen. Workspaces dienen vor allem zur Trennung von Abteilungs- und Entwicklerzugriffen, denn man kann in einem Workspace

¹³ (Koratamaddi, 2015)

beliebig viele Applikationen erstellen und auf beliebig viele Schemata zugreifen. Man benötigt pro Workspace aber einen separaten Benutzeraccount.

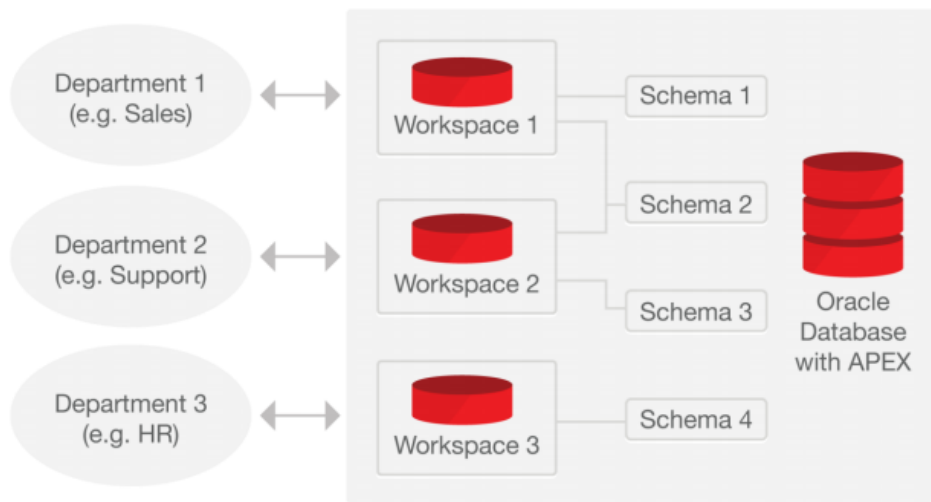


Abbildung 8: Oracle Apex - Schema und Workspace¹⁴

Ist ein entsprechender Workspace angelegt, können in diesem eine oder mehrere gewünschte Applikation erstellt und weiterentwickelt werden. Diese Applikationen können auch exportiert und importiert werden. Der Export besteht aus einer SQL-Datei, die die entsprechenden Befehle zur Importierung der Applikation beinhalten. Dieser SQL-Export kann auch direkt über die Datenbank eingespielt werden. Dies ist beispielsweise dann sinnvoll, wenn wie bereits erwähnt, in Produktivsystemen auf die Installation der Entwicklungsumgebung verzichtet wird. Hier würde dann nämlich die Möglichkeit eines Imports über die GUI fehlen, die SQL-Datei kann allerdings problemlos direkt in die Datenbank, die das APEX-Schema beinhaltet, importiert werden.

Ist nun eine Applikation erstellt, kann man diese in der Entwicklungsumgebung bearbeiten. Man kann beliebige Seiten erstellen und diese mit Hilfe von so genannten Branches miteinander verbinden. Auf den einzelnen Seiten ist es wiederum möglich diverse Regionen zu erstellen, die entweder die bekannten HTML-Items oder aber auch Reports (eine tabellarische Anzeige von Daten) darstellen können. Ein weiteres Feature ist die Darstellungsmöglichkeit eines Grids, einer Kombination aus Reports und Input-Items, welche vor allem im Bereich Multirow-Update herangezogen werden kann.

¹⁴ (Koratamaddi, 2015)

Die Darstellung erfolgt im Allgemeinen über HTML und CSS mit der Unterstützung von JavaScript. Im Hintergrund verarbeitet das Framework aber auch SQL-Befehle, welche vor allem zum Aufbau von Reports und Grids herangezogen werden, sowie PLSQL, um Verarbeitungslogik abbilden zu können.

Weitere Features des Frameworks sind unter anderem:

- die Erstellung eigener Objekte (sowohl Reports als auch Itemtypes) über sogenannte Plugins
- die zentrale Konfiguration der Navigation für Menüpfade und Breadcrumbs
- die Anpassung von HTML-Strukturen einzelner Items, Reports, Regionen oder Seiten mittels Templates
- Mehrsprachenfähigkeit
- diverse Sicherheitsfeatures
 - Seiten können Public oder User-Restricted zur Verfügung gestellt werden
 - der Authentifizierungsprozess kann angepasst und erweitert werden
 - Berechtigungen können definiert und einzelnen Funktionalitäten zugewiesen werden, auch kann darüber konfiguriert werden ob bestimmte Objekte eventuell nur gelesen werden können oder überhaupt nicht sichtbar sind, ebenso wie ganze Seiten
 - durch die Möglichkeit der Aktivierung einer Session State Protection kann mittels einer automatisch implementierten Hashwert-Validierung verhindert werden, dass User Requests manipulieren, um sich so Zugang zu nicht erlaubten Bereichen oder den Zugriff auf fremde Daten erschleichen
- Erstellung von REST und SOAP Web Service-Schnittstellen

Der allgemeine Aufbau der einzelnen Seiten besteht üblicherweise aus einer Übersichtsseite, in der mittels eines Reports alle vorhandenen Objekte dargestellt werden. Hier kann man mittels eines Buttons einen neuen Eintrag erfassen, mit Klick auf ein bestehendes Objekt das entsprechende Objekt bearbeiten oder löschen. Die Erfassungs- und Bearbeitungsseite ist im Regelfall dieselbe. Sie unterscheidet sich nur darin, dass bei Neuerfassungen noch keine ID vergeben ist und diese erst bei der Verarbeitung erstellt wird, während bei einem Update die entsprechende Objekt-ID dem Seitenaufruf mitgegeben wird und die entsprechenden Daten geladen und nach der

Bearbeitung wieder abgespeichert werden. Nach der Verarbeitung landet man im Regelfall wieder auf der Übersichtsseite.

5.2 Ablaufmodelle

Um die Vorgangsweise der Applikation zu konkretisieren, sollen im nächsten Schritt die einzelnen Abläufe anhand von entsprechenden Modellen erläutert werden. Es folgt die grobe Übersicht, über das System, sowie die Beschreibung der einzelnen Teilprozesse.

5.2.1 Übersichtsseite Zusammenfassung

Auf der Übersichtsseite werden die von einem Kunden erfassten Infrastrukturen, deren Lizenzen und die Lizenzverwendung dargestellt. Zudem wird hier ersichtlich, wo im Unternehmen eine Über- und eine Unterlizenzierung vorherrscht, um gegebenenfalls eingreifen zu können. Es ist der zentrale Ausgangspunkt, den man nach dem Login erreicht. Durch die Darstellung der einzelnen Objekte kann einfach zu konkreten Anzeigen gewechselt und Bearbeitungen vorgenommen werden.

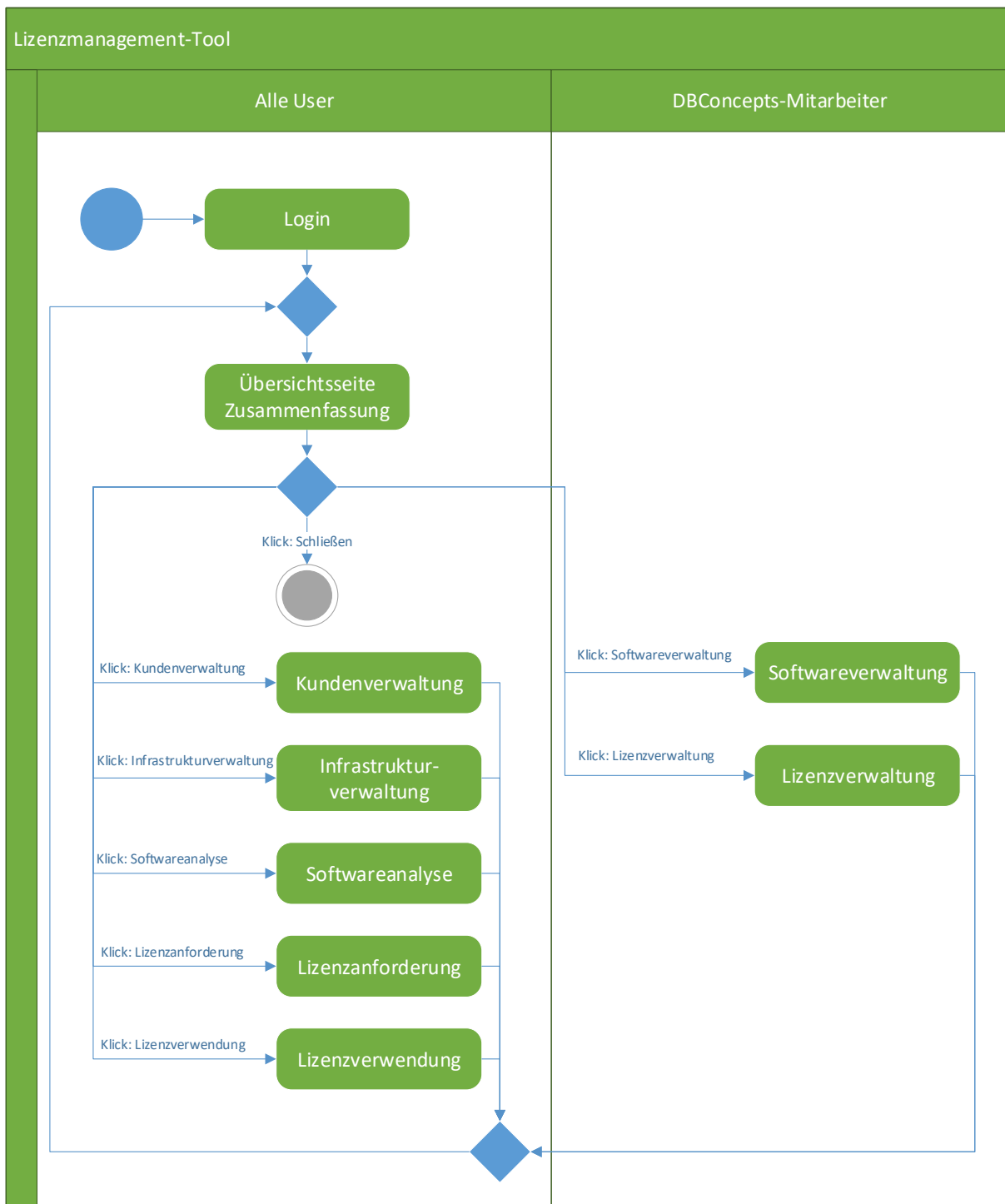


Abbildung 9: Globale Navigation

5.2.2 Kunden und User verwalten

Die Kundenübersicht kann nur von Systemadministratoren (in erster Linie DBConcepts-Mitarbeiter) aufgerufen werden, da die Userverwaltung aktuell nur von diesen vorgenommen wird. Eine vollständig automatisierte Registrierung ist aktuell nicht geplant, da die entsprechenden Tools vorrangig den existierenden Wartungsvertragskunden zur Verfügung gestellt werden sollen. Die Usererstellung wird

also bei Vertragsabschluss eingeleitet, weitere Nutzer werden nach Anfrage des Kunden erstellt oder ausgeschiedene User gelöscht. Dieser Vorgang ist auch wichtig, da in diversen Systemen kostenpflichtige Prozesse angestoßen werden können und deshalb nur ein entsprechend berechtigter Personenkreis des Kunden den Zugang zum System erhalten sollte.

Auf der Kundenübersichtsseite sind alle Kunden gelistet. Hier kann über einen Button ein neuer Kunde erstellt oder mit dem Klick auf einen vorhandenen Kunden die Bearbeitungsseite des jeweiligen Kunden aufgerufen werden, auf der einerseits die Kundendaten angepasst und andererseits auch neue Userlogins erstellt, alte Userzugänge gelöscht oder Passwörter zurückgesetzt werden können.

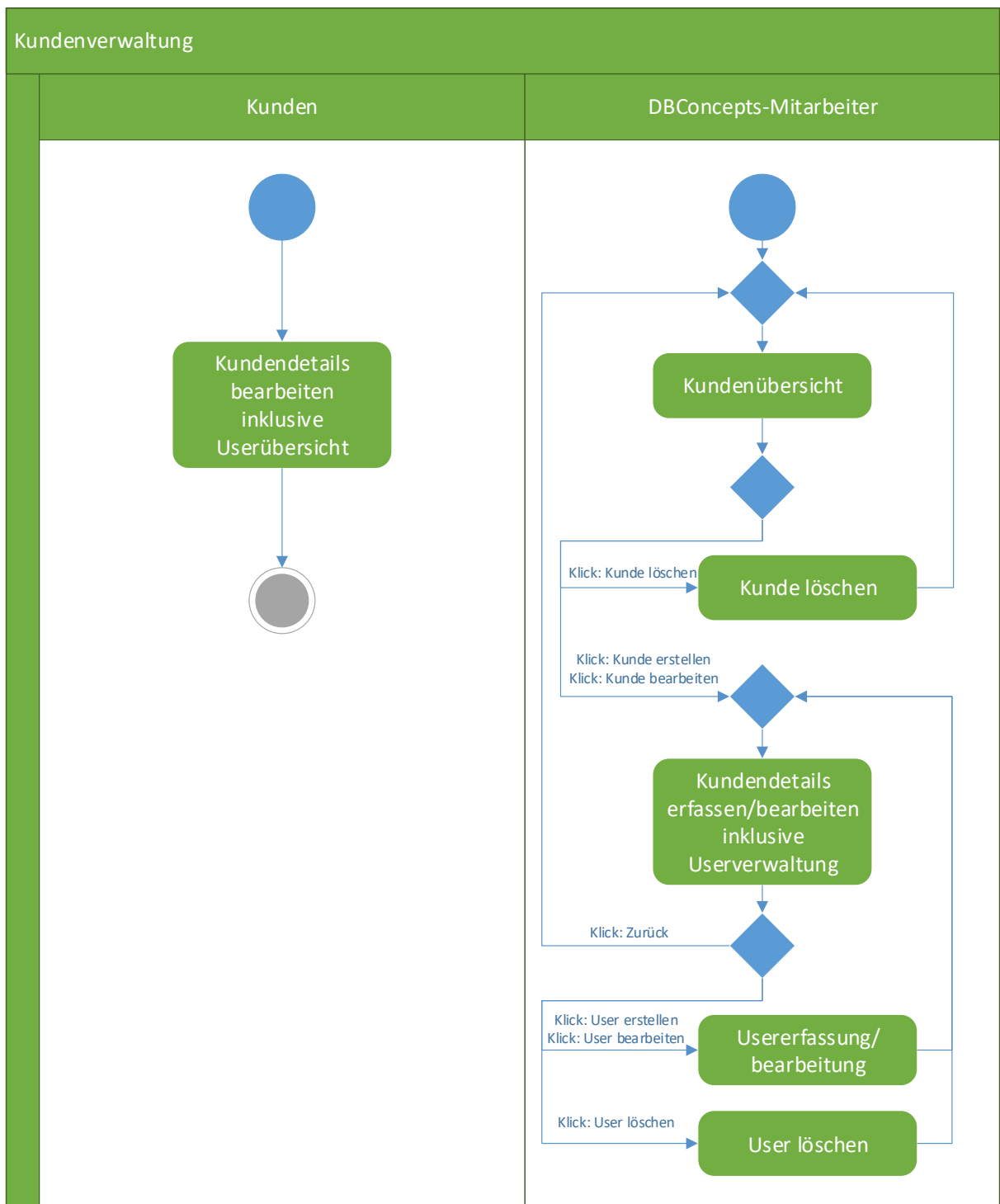


Abbildung 10: Kundenverwaltung

5.2.3 Infrastruktur verwalten

Auf der Infrastrukturübersichtsseite werden alle vorhandenen Infrastrukturobjekte eines Kunden angezeigt. Hier kann man mittels Button eine neue Infrastruktur erstellen oder mit Klick auf eine bestehende Infrastruktur diese bearbeitet oder gelöscht werden.

Gelangt man auf die Bearbeitungsseite kann man entsprechende Daten für ein Infrastrukturobjekt erfassen, den gewünschten Infrastruktortypen auswählen und falls gewünscht das aktuelle Objekt mit einem bereits vorhandenen Infrastrukturobjekt verknüpfen. Wird ein zu verknüpfendes Objekt ausgewählt, muss auch ein entsprechender Verknüpfungstyp ausgewählt werden.

Ebenso ist es möglich auf der Übersichtsseite eine Detailansicht einzelner Infrastrukturen aufzurufen, welche beispielsweise alle zugewiesenen Lizenzen zu einer Infrastrukturinstanz auflistet, sowie alle darüber liegenden Zuweisungen, sofern die Beziehung zu einer übergeordneten Instanz konfiguriert wurde.

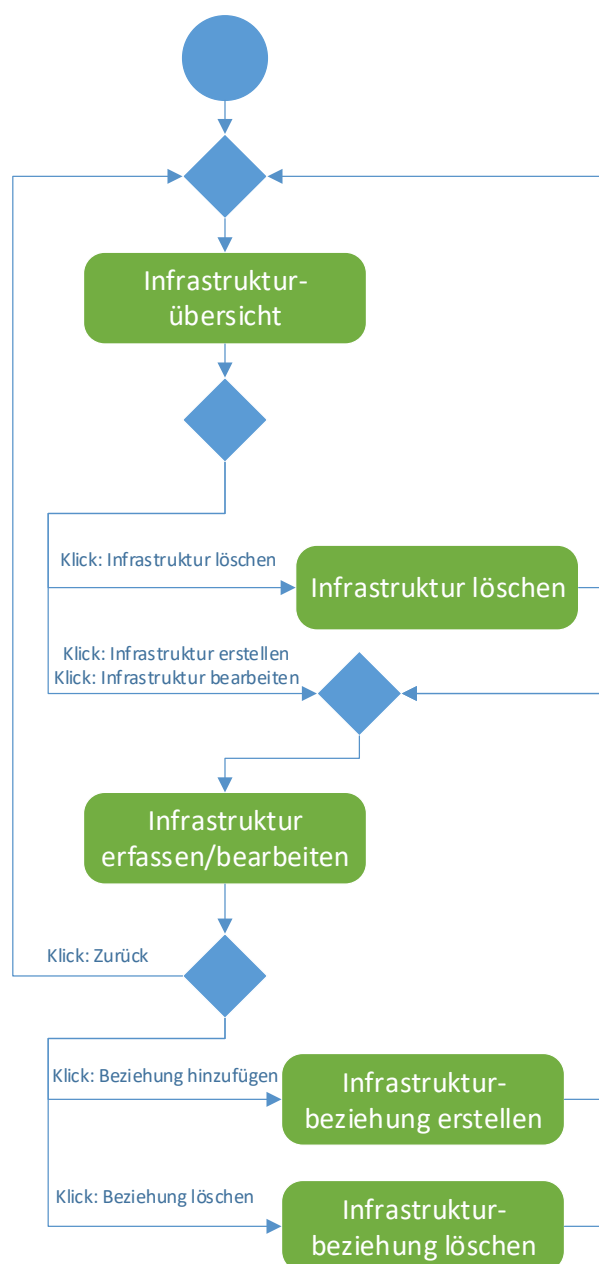


Abbildung 11: Infrastrukturverwaltung

5.2.4 Software verwalten

Auf der Übersichtsseite der Software werden alle vorhandenen Softwarepakete inklusive deren Edition und Version dargestellt. Mittels Button kann eine neue Softwareversion erstellt werden, mit einem Klick auf eine vorhandene Softwareversion kann diese bearbeitet werden. Auf der Bearbeitungsseite kann mittels eines Autovervollständigungs-Items ein Softwarepaket ausgewählt werden, oder ein neues erfasst werden. Bei der Erstellung wird automatisch geprüft, ob das Softwarepackage bereits existiert oder ob dieses angelegt werden muss. Des Weiteren wird in der Bearbeitungsmaske noch die Version angegeben, sowie ein Startdatum gesetzt und die restlichen Informationen erfasst.

Auf der Übersichtsseite der Software kann man mittels separatem Button auch noch die Übersichtsseite der Softwarepakete erreichen auf der man erfasste Softwarepakete bearbeiten oder löschen kann. Die Erfassung neuer Softwarepakete erfolgt über die bereits beschriebene Softwareversionsbearbeitungsseite.

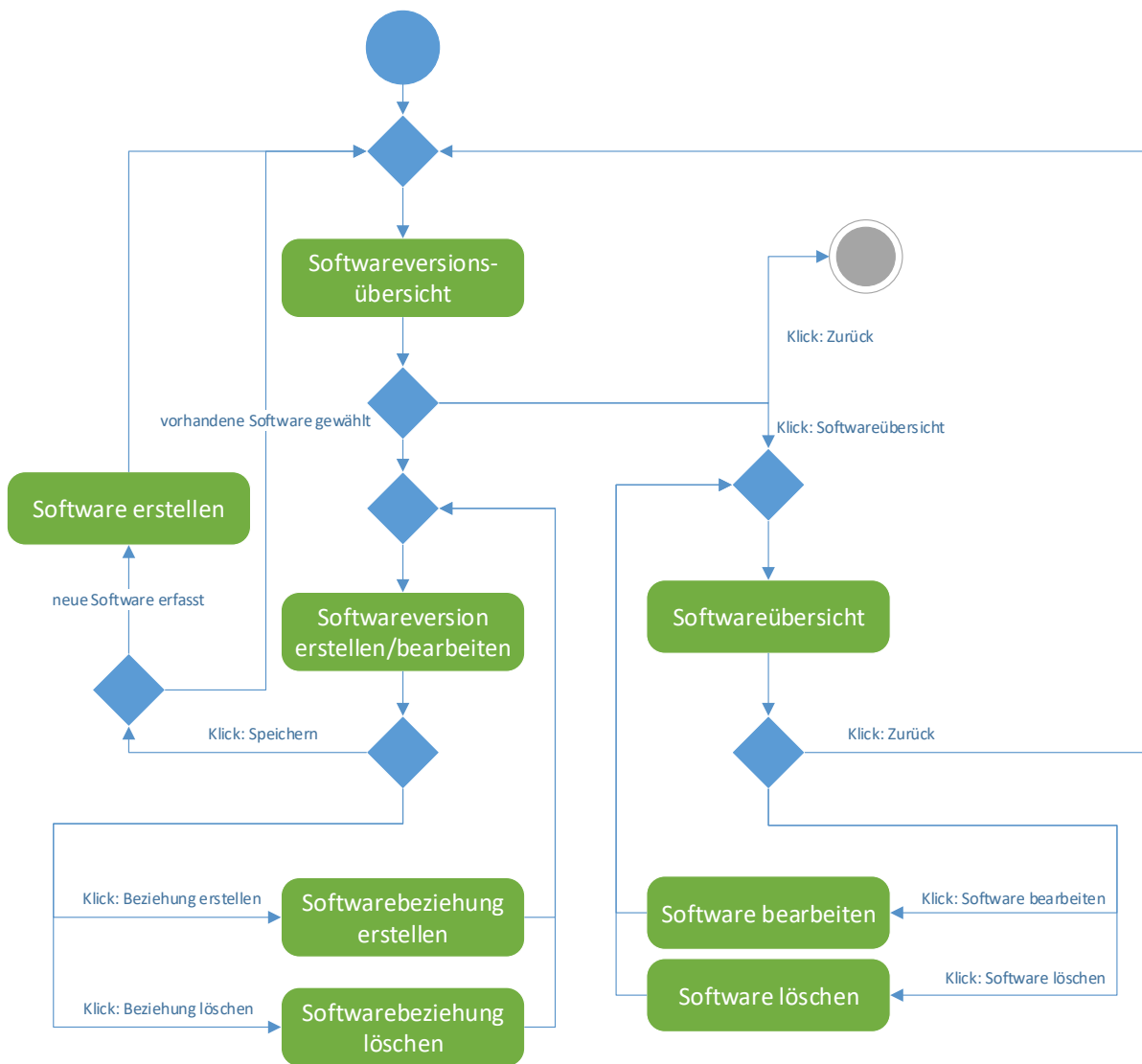


Abbildung 12: Softwareverwaltung

5.2.5 Lizenzen verwalten

Auf der Übersichtsseite werden alle vorhandenen Lizenzen aufgelistet. Wie gewohnt kann man durch entsprechende Buttons oder Links auf die Erfassen- und Bearbeiten-Seite gelangen, um neue Lizenzen hinzuzufügen, zu bearbeiten oder zu entfernen. Neben der Lizenzbezeichnung wird hier auch der Preis für die Lizenz erfasst, sowie die Lizenzanforderung und auch die Einteilung der Klassifizierung der Lizenzen in Form, Klasse, Typ und Art.

Ebenfalls wird hier die entsprechende Lizenzabdeckung definiert, welche die entsprechende Software, sowie die abgedeckten Features beinhaltet.

Handelt es sich um eine Upgrade-Lizenz, so müssen auch die möglichen Vorgängerylizenzen als Lizenzbeziehung definiert werden, bei Addons muss

entsprechend die benötigte Vollversionslizenz zugewiesen werden. Ebenfalls über die Beziehung können etwaige Wartungsvertragskosten zu einer Lizenz in Beziehung gestellt werden. Diese Wartungsvertragskosten werden meist über einen gewissen Zeitraum definiert, welcher in der Lizenzfassung mitangegeben werden kann. Selbiger dient bei SaaS-Lizenzen für die Konfiguration für die entsprechende zeitliche Einteilung und Beschränkung.

Des Weiteren muss definiert werden, für welche Infrastrukturtypen die jeweilige Lizenz angewandt werden kann, um bei der Lizenzanforderung und Lizenzverwendung eine entsprechend vorgefilterte Auswahlmöglichkeit zu bieten.

Neben einem konkreten Preis besteht auch die Möglichkeit eine Preisliste via Grid-Darstellung zu erfassen, also in Form einer bearbeitbaren Tabelle. Es wird die Mindeststückzahl sowie der danach gültige Einzelpreis abgespeichert.

Des Weiteren können zusätzlich zu den häufig verwendeten Bedingungen auf der Lizenzbearbeitungsseite auch die dynamischen Bedingungen bearbeitet werden, welche wiederum als Übersicht dargestellt sein sollen. Neue Bedingungen können erfasst und bearbeitet und alte Bedingungen entfernt werden.

Von der Lizenzübersicht kann man außerdem auf die Übersichtsseite der Features wechseln. Hier werden alle bisher erfassten Features aufgelistet und können entsprechend bearbeitet und gelöscht werden. Außerdem ist es hier natürlich auch möglich neue Features anzulegen.

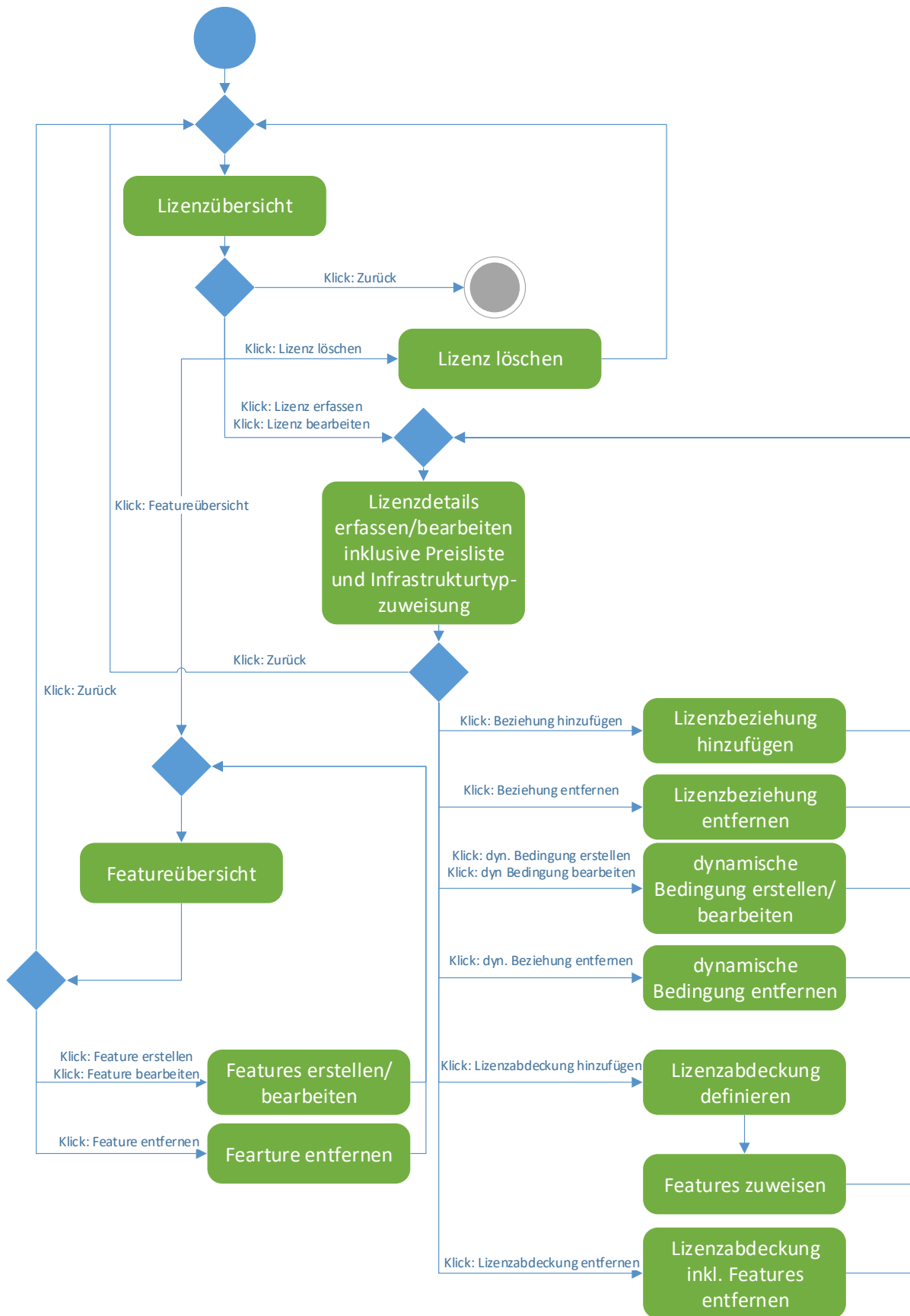


Abbildung 13: Lizenzverwaltung

5.2.6 Lizenzverwendung verwalten

Auf der Übersichtsseite der Lizenzverwendungen werden alle Lizenzen gelistet, die einem Kunden zugewiesen sind, sowie deren Einsatz im Unternehmen, sofern hierzu konkrete Angaben erfasst wurden, wie beispielsweise die Zuweisung zu einer Infrastrukturkomponente.

Mittels Button kann eine neue Lizenz dem jeweiligen Kundenunternehmen zugefügt werden. Durch einen Klick auf eine bestimmte Lizenz gelangt man zur Detailseite, auf welcher man sieht, wo die Lizenz überall eingesetzt wird und kann den Einsatz der Lizenz entsprechend verändern.

Unter anderem kann die im Unternehmen vorhandene Menge einer Lizenz bearbeitet werden, ebenso kann eine bestimmte Menge dieser Lizenz einer bestimmten Infrastruktur zugewiesen oder entzogen werden.

So ist es beispielsweise möglich, einen Lizenzpool jedem einzelnen Clientrechner zuzuweisen. Alternativ kann aber auch in der Infrastruktur ein Pool aus Clientrechnern zu einer Gruppe (beispielsweise einer Abteilung) zusammengefasst werden und ein entsprechend großer Lizenzpool dieser Gruppen-Infrastrukturinstanz zugewiesen werden. Dies erleichtert das Management bei vielen gleichen Systemkonfigurationen, sofern es sich nicht um konkrete Einzelplatzlizenzen handelt, die direkt einem Rechner zugewiesen werden müssen. In einem solchen Fall muss die Zuweisung pro Infrastruktur-Instanz erfolgen.

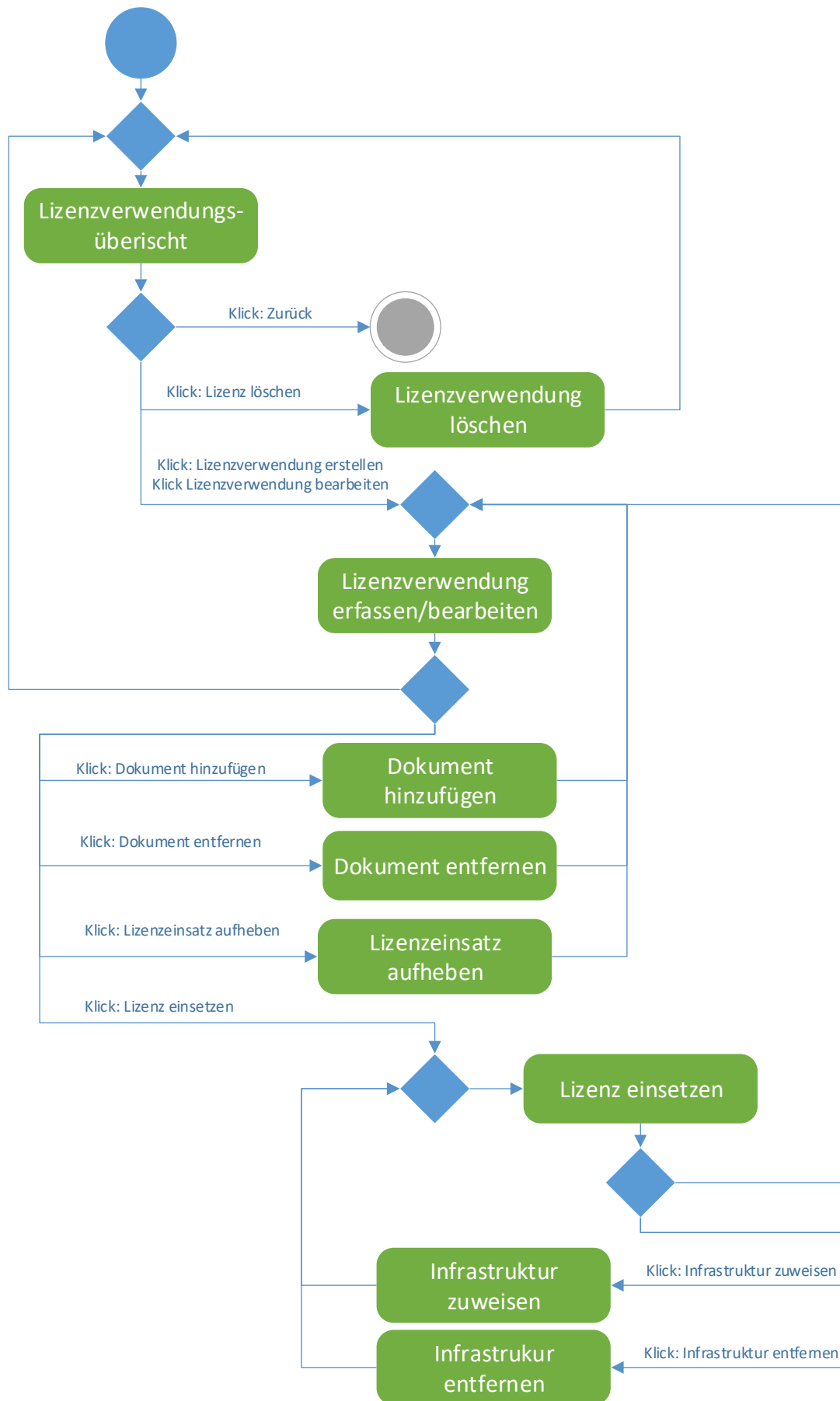


Abbildung 14: Lizenzverwendung

5.2.6.1 Dokumente hinzufügen

Einzelnen Lizenzverwendungen können in der Detailseite, welche auch zur Lizenzbearbeitung dient, beliebig viele Dokumente angehängt werden. Diese Dokumente dienen unter anderem als Ablage für die entsprechende Bestätigung für einen Lizenzzerwerb. Wird beispielsweise bei einem Auditing nun der Nachweis für bestimmte Lizenzen verlangt, können diese im Tool leicht ermittelt und heruntergeladen werden.

5.2.7 Softwareanalyse

Die Softwareanalyse stellt einen eigenen Bereich dar. Jeder Kunde hat eine Übersicht über seinen Softwarepool, kann über einen Button eine neue Software aus den Softwareversionen hinzufügen und mittels Klick auf eine Software diese bewerten, sowie einen Kommentar hinzufügen, um später die Information wieder einsehen zu können, warum eine Software aufgenommen werden sollte oder was deren Schwachstellen waren. Über einen Status kann festgelegt werden, ob die Software erst analysiert werden muss, ob sie bereits analysiert wurde, ob sie als untauglich abgelehnt wurde oder ob sie sich für einen Einsatz im Unternehmen qualifiziert hat. Auf der Übersichtsseite können die Daten entsprechend sortiert und gefiltert werden.

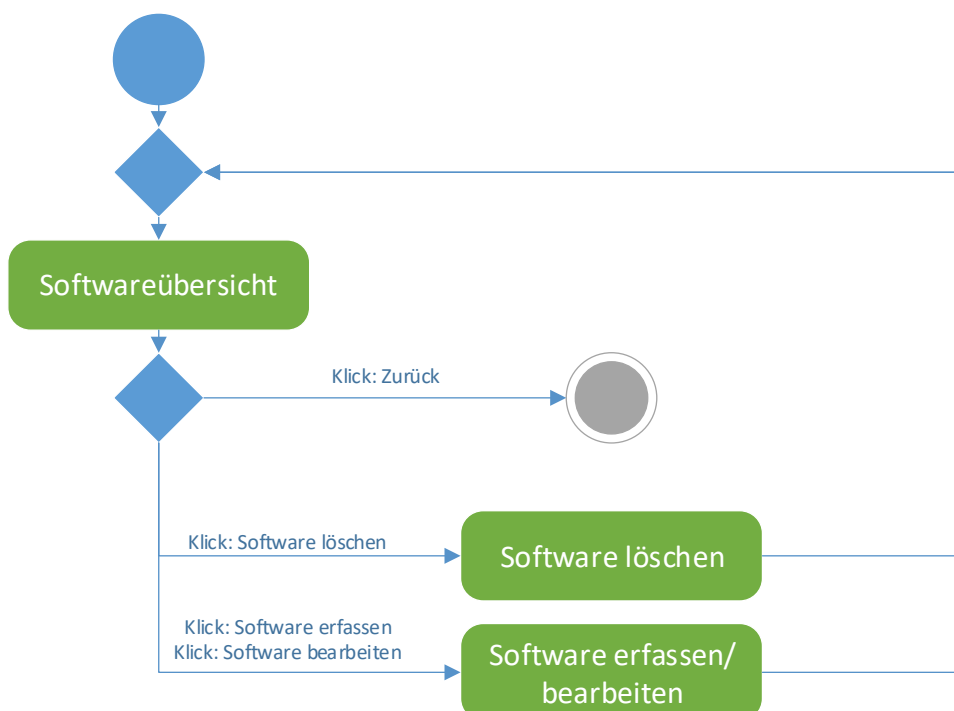


Abbildung 15: Softwareanalyse

5.2.8 Anforderung erstellen

Einen wichtigen Bereich stellt noch die Anforderungserfassung dar. In einer Übersichtsseite kann ein Kunde die erstellten Anforderungen einsehen, mittels Klick auf eine Zeile die entsprechende Anforderung abändern oder löschen, oder über einen eigenständigen Button eine neue Anforderung erstellen.

Für eine neue Anforderung wählt der Kunde zuerst die Software und eventuell deren Version, die er sich wünscht. Danach wird ihm ein Pool an Features, den die unterschiedlichen Softwareversionen bieten, zur Auswahl angeboten. Hier kann der Kunde nun auswählen, welche Features einer Software von ihm tatsächlich gebraucht werden. Des Weiteren können diverse Anforderungskriterien erfasst werden, wie beispielsweise die Benutzeranzahl. Zusätzlich kann der Kunde bestimmte Infrastrukturkomponenten der Anforderung hinzufügen, um Detaildaten zu CPU und RAM zu liefern.

Werden von den Lizenzen, die die ausgewählte Softwareversion unterstützen, aufgrund der Komplexität der Lizenzanforderung noch zusätzliche Bedingungen verlangt, so werden diese ebenfalls angezeigt und der Kunde kann entsprechende Detailwerte eintragen.

Wurde alles erfasst kann der Kunde die Anforderung abspeichern und sich das Ergebnis anzeigen lassen. Die Ergebnisanzeige kann über die Übersichtsseite der Anforderungen auch zu einem späteren Zeitpunkt wieder aufgerufen werden. Sie listet alle Lizenzen, die die vom Kunden nachgefragten Features unterstützen. Fehlen bestimmte Features, da beispielsweise keine Lizenz alle benötigten Features anbietet, so wird dies in der Übersicht ebenfalls ausgewiesen. Die Übersicht wird im Standardfall nach deren Kosten beginnend mit dem billigsten sortiert, wobei Lizenzen die alle benötigten Features abdecken zuerst gereiht werden. Die Sortierung kann aber im Nachhinein noch vom User verändert werden.

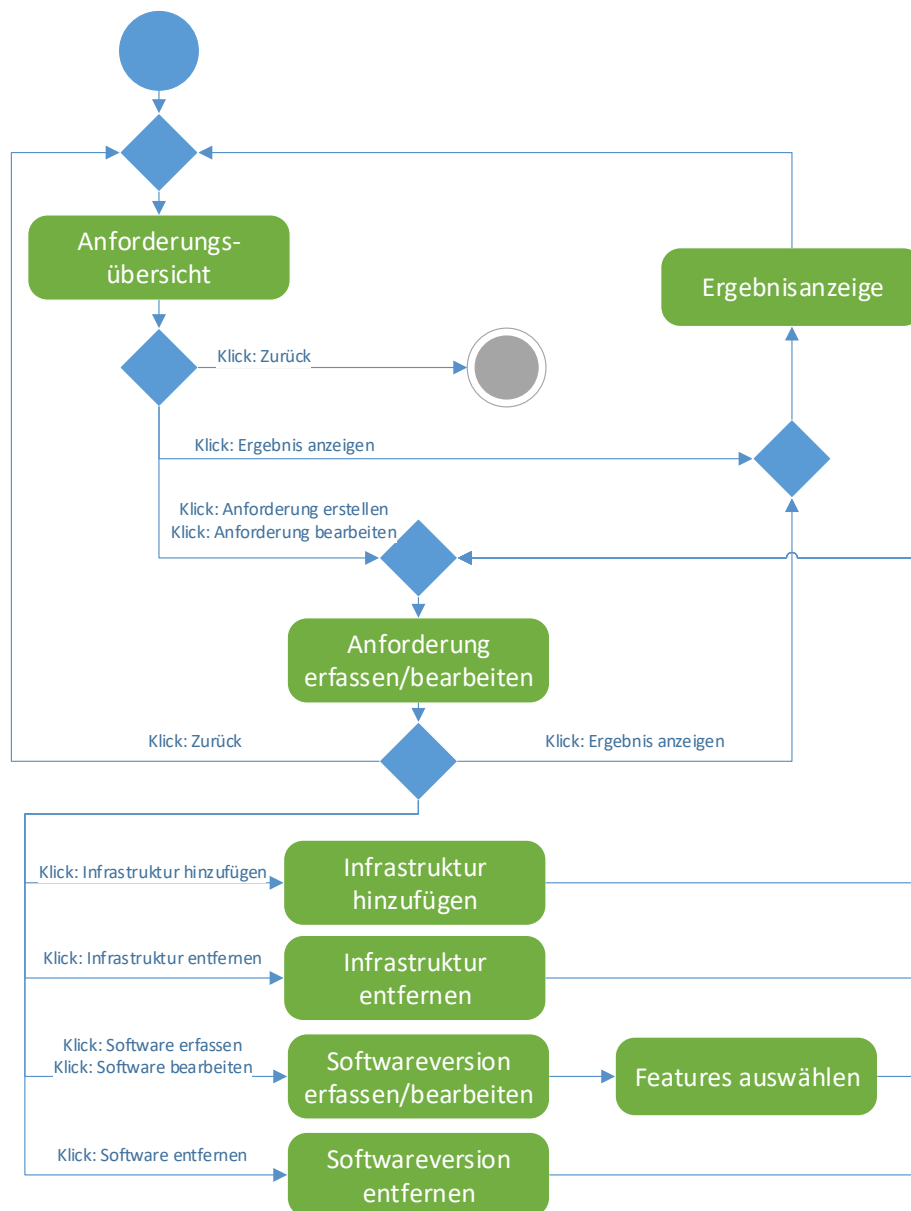


Abbildung 16: Lizenzanforderung

5.2.9 System-Scan

Der System-Scan soll zur Ermittlung der bereits im Einsatz befindlichen Software auf den diversen verteilten Systemen dienen. Die Abfrage von entsprechender Software erfolgt über einen separaten Client, der auf die einzelnen Zielsysteme verteilt und auf denen ausgeführt werden kann. Mittels dieses Tools sollen das AD, die Registry, sowie entsprechende Windows-Analyse-Dateien ausgelesen werden können, um dokumentieren zu können, welche Software auf dem jeweiligen System installiert ist und ausgeführt wird. Die Analyse erstellt ein entsprechendes XML-Dokument, welches in die Lizenzmanagement-Datenbank auf zwei Wege importiert werden kann. Gibt es

einen direkten Zugriff auf die Webservices des Tools, so können direkt über dieses die entsprechenden Dateien übertragen werden.

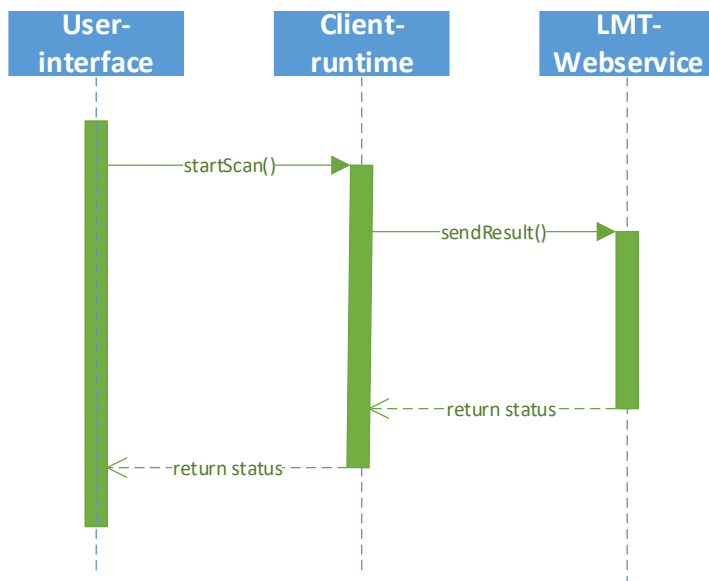


Abbildung 17: System-Scan über Webservice

Besteht aktuell keine Verbindung zum Server-System, kann alternativ das resultierende XML-File auch am jeweiligen System erstellt und gespeichert werden und sobald ein entsprechender Zugriff zum Lizenzmanagementtool hergestellt werden kann, über eine dafür vorgesehene Importmaske in die Applikation, beziehungsweise in weiterer Folge in die Datenbank, importiert werden.

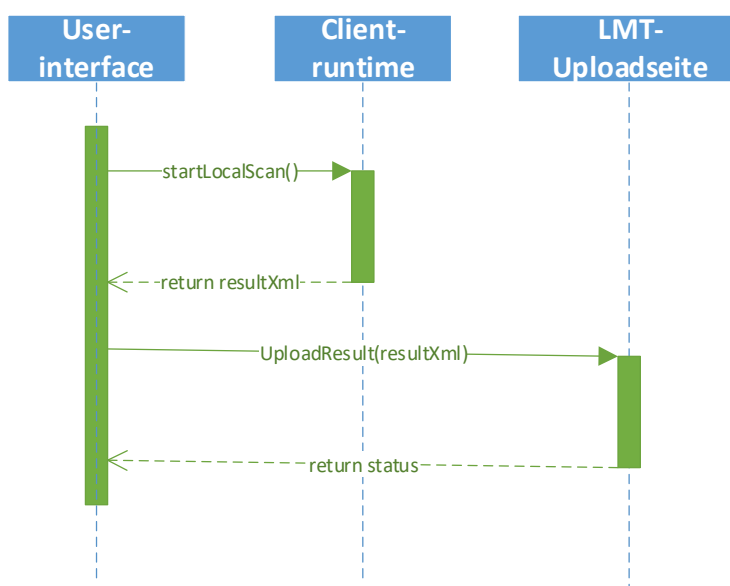


Abbildung 18: Lokaler System-Scan mit anschließendem Upload

Da nun die Prozesse genauer definiert und geschildert wurden, soll im nächsten Schritt ein Überblick über die Datengrundlage gegeben werden.

5.3 Datenmodell

Die Organisation der Daten ist entscheidend für die funktionale Umsetzung des Unterstützungstools. Zur Übersichtlichkeit werden die einzelnen Komponenten des Modells in sechs Teilstücke unterteilt, welche in Abbildung 19: Datenmodell LMT als gesamtes ersichtlich sind. Da die Komplexität mit einer gewissen Unübersichtlichkeit hergeht, werden die einzelnen Teilstücke nachträglich separat genauer aufgeführt und beschrieben. Für eine bessere optische Darstellung wurden die Teilstücke farblich unterteilt, im Nachfolgenden wird beschrieben, was die einzelnen Teilbereiche abdecken:

- **Rot:** Dieser Bereich dient zur Kundendarstellung. Er dient einerseits zur Identifikation der einzelnen Kunden, um die Datenanzeige entsprechend einschränken zu können, bildet andererseits aber auch die Bedürfnisse und Anforderungen der Kunden ab, also für welche Software Lizenzen gesucht und welche Bedürfnisse mit der entsprechenden Software abgedeckt werden sollen.
- **Orange:** Der Bereich rund um die Infrastruktur bildet eine Darstellung von Hardwaresystemen dar, die einem Kunden zugewiesen werden können. Es kann sich hier einerseits um eine konkrete Hardware handeln, die hier definiert werden kann, aber auch um komplexere Konstrukte, wie beispielsweise Cluster oder auch virtuelle Serverlandschaften.
- **Blau:** Dieser Bereich dient zur Festlegung der Software, die mittels des Lizenzmanagementtools gewartet werden kann. Jegliche Software, die von einem Kunden angefordert oder mittels einer bestimmten Lizenz abgedeckt wird, wird in diesem Bereich erfasst. Auch die Version einzelner Softwareprodukte, sowie deren Funktionsumfang wird hier festgelegt.
- **Grün:** Dieser Bereich stellt die Lizenzen in all ihren Ausprägungen dar. Außerdem umfasst er die Preislisten, sowie Upgrade-Möglichkeiten einzelner Lizenzangebote.
- **Gelb:** Der gelbe Bereich dient zur Umsetzung von dynamischen Anforderungswerten. Die Lizenzanforderungen werden immer komplexer und Lizenznehmer lassen sich immer wieder neue Methoden einfallen, um Lizenzen

zu kategorisieren und auf bestimmte Anforderungen hin anzupassen, wie dies vor allem bei Serverlizenzen bereits ersichtlich ist. Um auch zukünftige komplexere Anforderungen darstellen zu können, wird hiermit die dynamische Erfassung von zusätzlich angeforderten Daten ermöglicht.

- Grau: Dieser Bereich umfasst das Logging, um beispielsweise bei Fehlerfällen ein effizienteres Bugfixing zu ermöglichen.

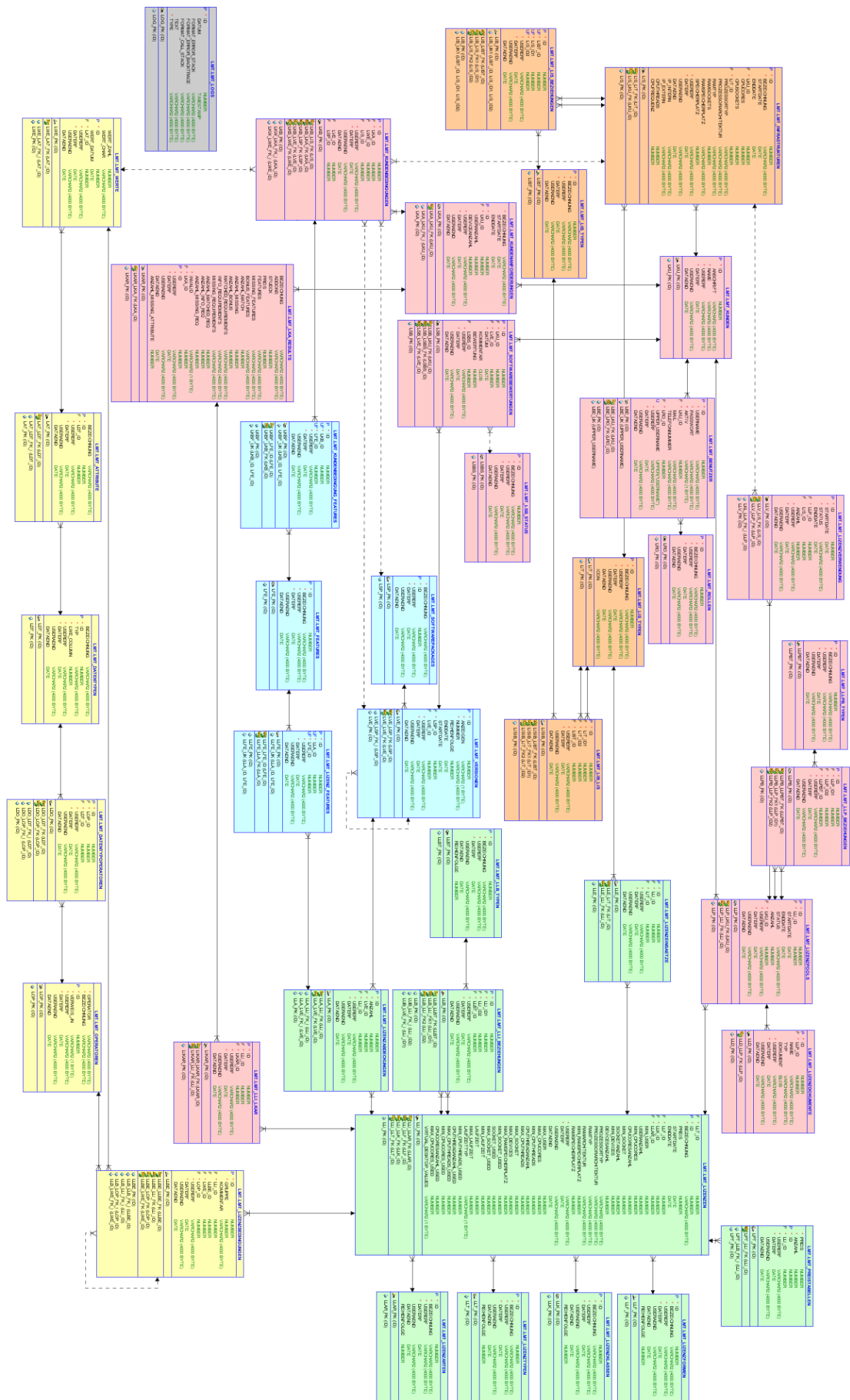


Abbildung 19: Datenmodell LMT

Wie man erkennen kann (oder auch nicht), besitzt das Datenmodell einen ordentlichen Umfang. Das Gesamtmodell soll nur als Veranschaulichung dienen, wie die einzelnen Verknüpfungen in diesem Modell angedacht und umgesetzt sind. Im nächsten Schritt sollen nun die einzelnen Bereiche genauer erläutert werden und damit einen Überblick über die Datenlandschaft und deren Sinn bieten.

5.3.1 Kundenbereich

Ein wichtiger Bereich, und gleichzeitig auch der umfangreichste, stellt die Abbildung der Kunden dar. Die wesentlichen Komponenten sind die Tabellen LMT_BENUTZER, sowie LMT_KUNDEN. Wie der Name der Tabelle LMT_KUNDEN schon sagt, dient diese als Abbildung der eigentlichen Kunden, um einen Datenschutz zu gewährleisten. So sollen unterschiedliche Kunden natürlich nicht in die Planungen und Systemabbildungen anderer Kunden einsehen können. Die Tabelle beschreibt also eine große Gruppierungsrichtlinie, denn nur Benutzer, welche in Tabelle LMT_BENUTZER erfasst werden und einem Kunden zugeordnet werden können, dürfen die entsprechenden Erfassungen des jeweiligen Kunden sehen und je nach Rolle auch bearbeiten. Daten von anderen Kunden sollen nicht einsehbar und manipulierbar sein. Auch DBConcepts GmbH selbst wird hier als Kunde erfasst und genauso behandelt, wie jeder andere Kunde auch.

Tabelle 1: Spaltendefinition der Tabelle LMT_KUNDEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Kunden
NAME	VARCHAR2(4000)	Name eines Kunden
ANSCHRIFT	VARCHAR2(4000)	Anschrift eines Kunden

Tabelle 2: Spaltendefinition der Tabelle LMT_BENUTZER

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation des Benutzers
USERNAME	VARCHAR2(4000)	Username des Benutzers
PASSWORT	VARCHAR2(4000)	Passwort-Hash des Benutzers (keine Klartextablage des Passwortes)
AKTIV	VARCHAR2(1)	Y/N-Flag, ob User aktiv ist. Nur aktive User können sich anmelden.
LKU_ID	NUMBER	Referenz zum Kunden dem der Benutzer zugewiesen ist.
MAIL	VARCHAR2(4000)	Mailadresse des Benutzers
TELEFONNUMMER	VARCHAR2(4000)	Telefonnummer des Benutzers
LRO_ID	NUMBER	Referenz auf Rollenzuweisung
UPPER_USERNAME		Virtuelle Spalte, die sich aus upper(username) errechnet, dient als Vereinfachung des Unique-Key-Check auf den caseinsensitiven Usernamen

Den Usern kann nun noch eine entsprechende Rolle zugewiesen werden, welche wiederum die Berechtigungen im System widerspiegeln.

Aktuell gibt es drei Rollen im System:

- **Kunde-Readonly:** darf die Daten eines bestimmten Kunden abfragen
- **Kunde:** darf die Daten eines bestimmten Kunden abfragen und bearbeiten
- **Systemadministrator:** darf die Daten eines bestimmten Kunden abfragen und bearbeiten und zusätzlich noch Stammdaten für Lizenzen, Software und Features erfassen

Diese werden in der Tabelle LMT_ROLLEN erfasst und über die Tabelle LMT_USER einer Person zugewiesen.

Tabelle 3: Spaltendefinition der Tabelle LMT_ROLLEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Rolle
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung einer Rolle

Neben der Benutzer- und Kundendarstellung beinhaltet dieser Bereich auch die tatsächlichen Anforderungen und Ausgangskriterien der Kunden. In der Tabelle LMT_LIZENZPOOLS werden alle Lizenzen historisierend erfasst, welche ein Kunde besitzt, oder besessen hat, wieviel Stück dieser Lizenz im Unternehmen existieren und um welche Lizenz es sich konkret handelt. Über die Beziehungstabelle LMT_LLBP_BEZIEHUNGEN werden Lizenzen im Lizenzpool miteinander verknüpft. Dies ist vor allem wichtig, um Upgrades, Addons und Wartungsverträge zur jeweiligen zugehörigen ausgehenden Lizenz im Lizenzpool zuordnen zu können. In der Tabelle LMT_LLBPB_TYPEN gibt es also unter anderem folgende Einträge:

- Addonzuweisung
- Upgradezuweisung
- Wartungsvertragzuweisung

Tabelle 4: Spaltendefinition der Tabelle LMT_LIZENZPOOLS

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Lizenzpools eines Kunden
LLI_ID	NUMBER	Referenz auf eine Lizenz
STARTDATE	DATE	Datum ab wann die Lizenz dem Lizenzpool hinzugefügt wurde
ENDDATE	DATE	Datum bis wann sich die Lizenz im Lizenzpool befand.
STATUS	VARCHAR2(4000)	Status der Lizenz im Lizenzpool
ANZAHL	NUMBER	Anzahl der Lizenz im Lizenzpool
LKU_ID	NUMBER	Referenz auf einen Kunden

Tabelle 5: Spaltendefinition der Tabelle LMT_LL_PBEZIEHUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzpoolbeziehung, um Upgrades oder Addons einer anderen Lizenz im Lizenzpool zuweisen zu können.
LLP_ID1	NUMBER	Referenz auf einen Master-Lizenzpooleintrag einer hierarchischen Beziehung oder ein erster Lizenzpooleintrag einer gleichwertigen Beziehung
LLP_ID2	NUMBER	Referenz auf einen Detail- Lizenzpooleintrag einer hierarchischen Beziehung oder ein zweiter Lizenzpooleintrag einer gleichwertigen Beziehung
LLPBT_ID	NUMBER	Referenz auf einen Lizenzpoolbeziehungstyp

Tabelle 6: Spaltendefinition der Tabelle LMT_LL_PBTYPEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Lizenzpoolbeziehungstyps
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung eines Lizenzpoolbeziehungstyps

Über die Tabelle LMT_LIZENZVERWENDUNG wird nun definiert, welche Lizenzen aus dem Lizenzpool aktiv verwendet werden und falls es gewünscht ist, können die entsprechenden Infrastrukturen, auf denen sie aktiviert wurden, zugewiesen werden. Etwa können Mehrplatzlizenzen den einzelnen Clients zugewiesen werden oder Serverlizenzen dem entsprechenden Server.

Tabelle 7: Spaltendefinition der Tabelle LMT_LIZENZVERWENDUNG

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzverwendung aus einem Lizenzpool
STARTDATE	DATE	Datum ab wann eine Lizenz eingesetzt wird
STATUS	VARCHAR2(4000)	Status der Lizenzverwendung
ENDDATE	DATE	Datum bis wann eine Lizenz eingesetzt wird
LLP_ID	NUMBER	Referenz auf einen Lizenzpooleintrag
LIS_ID	NUMBER	Referenz auf eine Infrastruktur
ANZAHL	NUMBER	Anzahl der eingesetzten Lizenzen, wenn keine Infrastruktur zugewiesen wurde

Über die LMT_LIZENZDOKUMENTE können den entsprechenden Lizenzpooleinträgen etwaige Bestätigungen für den Erwerb der Lizenzen angehängt werden, um sie bei einem Lizenz-Audit zentral abgelegt zu haben.

Tabelle 8: Spaltendefinition der Tabelle LMT_LIZENZDOKUMENTE

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Dokumentes zu einer Lizenz in einem Lizenzpool eines Kunden
LLP_ID	NUMBER	Referenz auf eine Lizenz im Lizenzpool eines Kunden
NAME	VARCHAR2(4000)	Name des Dokuments
TYP	VARCHAR2(4000)	Dateityp des Dokuments (PDF, TXT, ...)
DOKUMENT	BLOB	Ablage einer binären Datei als Oracle Secure File

Abseits der aktiven Verwendung und Zuweisung von im Unternehmen gehandhabten Lizenzen, gibt es auch den Bereich der Planung für die jeweiligen Kunden. Die Tabelle LMT_SOFTWAREBEWERTUNGEN unterstützt beispielsweise den Schritt der Eruierung einzelner Softwareangebote (Referenz auf LMT_VERSIONEN). Durch den Status aus LMT_LSB_STATUS soll kategorisiert werden, in welchem Zustand sich die Softwareeruierung befindet, also ob die Software noch zu testen ist, ob sie bereits erfolgreich getestet wurde und eine Anschaffung in Betracht gezogen werden soll, oder ob die Softwareeruierung für einen späteren Zeitpunkt zurückgestellt werden soll, da das aktuelle Ergebnis nicht zufriedenstellend für den Einsatz im Unternehmen ist. Ebenso kann eine Punktebewertung vorgenommen werden und als Kommentar die Kritikpunkte erfasst werden, die eventuell noch abzuklären sind, oder bei

Zurückstellung als Hinweis dienen, was an der entsprechenden Software nicht zufriedenstellend war.

Tabelle 9: Spaltendefinition der Tabelle LMT_SOFTWAREBEWERTUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Softwarebewertung
LKU_ID	NUMBER	Referenz auf einen Kunden
LVE_ID	NUMBER	Referenz auf eine Softwareversion
DATUM	DATE	Datum einer Bewertung
KOMMENTAR	CLOB	Kommentare, Hinweise und Infos zu einer Bewertung
BEWERTUNG	NUMBER	Bewertung einer Softwareversion
LSBS_ID	NUMBER	Referenz auf einen Softwarebewertungsstatus

Tabelle 10: Spaltendefinition der Tabelle LMT_LSB_STATUS

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Status einer Softwarebewertung
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung eines Status

Wurde eine Anschaffung einer bestimmten Software in Betracht gezogen, ist es im nächsten Schritt entscheidend, welche Lizenz nun am besten den Softwareeinsatz abdeckt. Um dies zu ermitteln gibt es die Tabelle LMT_KUNDENANFORDERUNGEN, welche dazu dient, die Eckpunkte für den Softwareeinsatz aus Lizenznehmerseite festzuhalten. Unter anderem sind das die vermutete Useranzahl sowie eine generische Geräteanzahl. Letztere ist allerdings nicht erforderlich, wenn stattdessen eine detailliert erfasste Infrastruktur mit der Anforderung verknüpft wird. Diese Verknüpfung geschieht über die Tabelle LMT_KUNDENBEDINGUNGEN (mit Referenz auf LMT_INFRASTRUKTUREN). Eine Infrastrukturzuweisung wird einer generischen Geräteanzahl stets bevorzugt.

Tabelle 11: Spaltendefinition der Tabelle LMT_KUNDENANFORDERUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Kundenanforderung
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung einer Kundenanforderung
STARTDATE	DATE	Beginn eines Zeitraumes einer Kundenanforderung, dient zur Ermittlung von zukünftigen Lizenzen

ENDDATE	DATE	Ende eines Zeitraumes einer Kundenanforderung, dient zum Abgrenzen von kommenden Lizenzen
LKU_ID	NUMBER	Referenz auf Kunden, von dem die Anforderung kommt
USERANZAHL	NUMBER	Useranzahl, die die jeweilige Software verwenden soll.
DEVICEANZAHL	NUMBER	Anzahl der Geräte, auf denen die Software eingesetzt werden soll, falls keine Infrastrukturzuweisung durchgeführt wird.

Tabelle 12: Spaltendefinition der Tabelle LMT_KUNDENBEDINGUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Kundenbedingung
LKA_ID	NUMBER	Referenz auf eine Kundenanforderung
LWE_ID	NUMBER	Referenz auf einen dynamischen Wert für eine Kundenanforderung
LIS_ID	NUMBER	Referenz auf eine Infrastruktur für eine Kundenanforderung
LSP_ID	NUMBER	Referenz auf eine Software für eine Kundenanforderung, wird durch eine explizite Softwareversion überlagert
LVE_ID	NUMBER	Referenz auf eine Softwareversion für eine Kundenanforderung

Des Weiteren ist es notwendig in den Kundenbedingungen die gewünschte Software (mit Referenz auf die Tabelle LMT_SOFTWAREPACKAGES oder LMT_VERSIONEN) zu erfassen, sowie zumindest ein benötigtes Feature zuzuweisen (über die Tabelle LMT_KUNDENBESIEHUNGEN_FEATURES). Über die Features wird im Endeffekt ermittelt, welche Lizenz am geeignetsten und kostengünstigsten die Anforderungen abdeckt.

Sollten noch weitere Werte von Relevanz sein, welche nicht im Modell inkludiert sind, können über LMT_WERTE noch dynamische Bedarfswerte für Spezialfälle erfasst werden.

Der letzte Teil des Kundenbereichs stellt das Speichern von Kundenanforderungsergebnissen dar. Da die Berechnung dieser eine erhöhte Durchlaufzeit aufweist, werden die ermittelten Ergebnisse in der Tabelle LMT_LKA_RESULTS gespeichert, damit nach einer Berechnung nur mehr die Daten aus

der Ergebnistabelle geladen werden müssen. Ändert sich die Ausgangssituation der Kundenanforderung (inklusive der Infrastruktur der Kunden), so werden die Ergebnisse entsprechend auf invalide gesetzt und bei einem erneuten Aufruf neu berechnet. Da auch Änderungen an Lizenzen zu einer Änderung der Ergebnisse führen kann, wird in der Tabelle LMT_LLI_LKAR mitdokumentiert, welche Lizenzen bei welchen Ergebnissen herangezogen wurden. Ändert sich an entsprechenden Lizenzen etwas, so werden die zugewiesenen Ergebnisse ebenfalls auf invalide gesetzt und müssen bei einem erneuten Aufruf neu berechnet werden.

Tabelle 13: Spaltendefinition der Tabelle LMT_LKA_RESULTS

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Kundenanforderungsergebnisses
BEZEICHUNG	VARCHAR2(4000)	Bezeichnung der Grundlizenz
ADDONS	VARCHAR2(4000)	Bezeichnung der zusätzlichen Addons (in HTML-Struktur)
STUECK	NUMBER	Stückzahl
PREIS	NUMBER	Gesamtpreis der Lizenzanschaffung in Berücksichtigung der Stückzahl
FEATURES	VARCHAR2(4000)	Abgedeckten Features einer Lizenzanforderung (in HTML-Struktur)
MISSING_FEATURES	VARCHAR2(4000)	Fehlende Features einer Lizenzanforderung (in HTML-Struktur)
BONUS_FEATURES	VARCHAR2(4000)	Zusätzliche, nicht angeforderte Features zu einer Lizenzanforderung (in HTML-Struktur)
ANZAHL_MATCH	NUMBER	Anzahl an abgedeckten Features
ANZAHL_MISSING	NUMBER	Anzahl an fehlenden Features
ANZAHL_BONUS	NUMBER	Anzahl an zusätzlichen, nicht angeforderten Features
MATCHED_REQUIREMENTS	VARCHAR2(4000)	Liste der abgedeckten Lizenzanforderungen (in HTML-Struktur)
INFO_REQUIREMENTS	VARCHAR2(4000)	Liste der abzudeckenden Lizenzanforderungen, bei denen nicht eindeutig definiert werden kann, ob sie abgedeckt sind (in HTML-Struktur)
MISSING_REQUIREMENTS	VARCHAR2(4000)	Liste der nicht erfüllten Lizenzanforderungen (in HTML-Struktur)
ANZAHL_MATCHED_REQ	NUMBER	Anzahl der abgedeckten Lizenzanforderungen

ANZAHL_INFO_REQ	NUMBER	Anzahl der abzudeckenden Lizenzanforderungen
ANZAHL_MISSING_REQ	NUMBER	Anzahl der nicht abgedeckten Lizenzanforderungen
INVALID	VARCHAR2(1)	Kennzeichen, ob der Datensatz invalide ist und neu berechnet werden muss.
LKA_ID	NUMBER	Referenz auf eine Kundenanforderung
ANZAHL_MISSING_ATTRIBUTE	NUMBER	Anzahl der fehlenden Attribute einer Kundenanforderung, die zur Ermittlung der Erfüllung einer Lizenzanforderung erforderlich wären.

Tabelle 14: Spaltendefinition der Tabelle LMT_LLI_LKAR

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenz- Kundenanforderungsergebniszuweisung, um identifizieren zu können, welche Lizenzen bei welchem Ergebnis eingebunden wurden, damit das Ergebnis bei der Änderung von Lizenzdaten auf invalid gesetzt werden kann.
LLI_ID	NUMBER	Referenz auf eine Lizenz
LKAR_ID	NUMBER	Referenz auf ein Kundenanforderungsergebnis

5.3.2 Infrastrukturbereich

Der nächste entscheidende Teil für die Systemabbildung einzelner Kunden ist der Infrastrukturbereich, welcher vor allem die Hardwareanforderungen abbilden soll, dieser kann allerdings auch deutlich detaillierter verwendet werden, wenn dies gewollt ist.

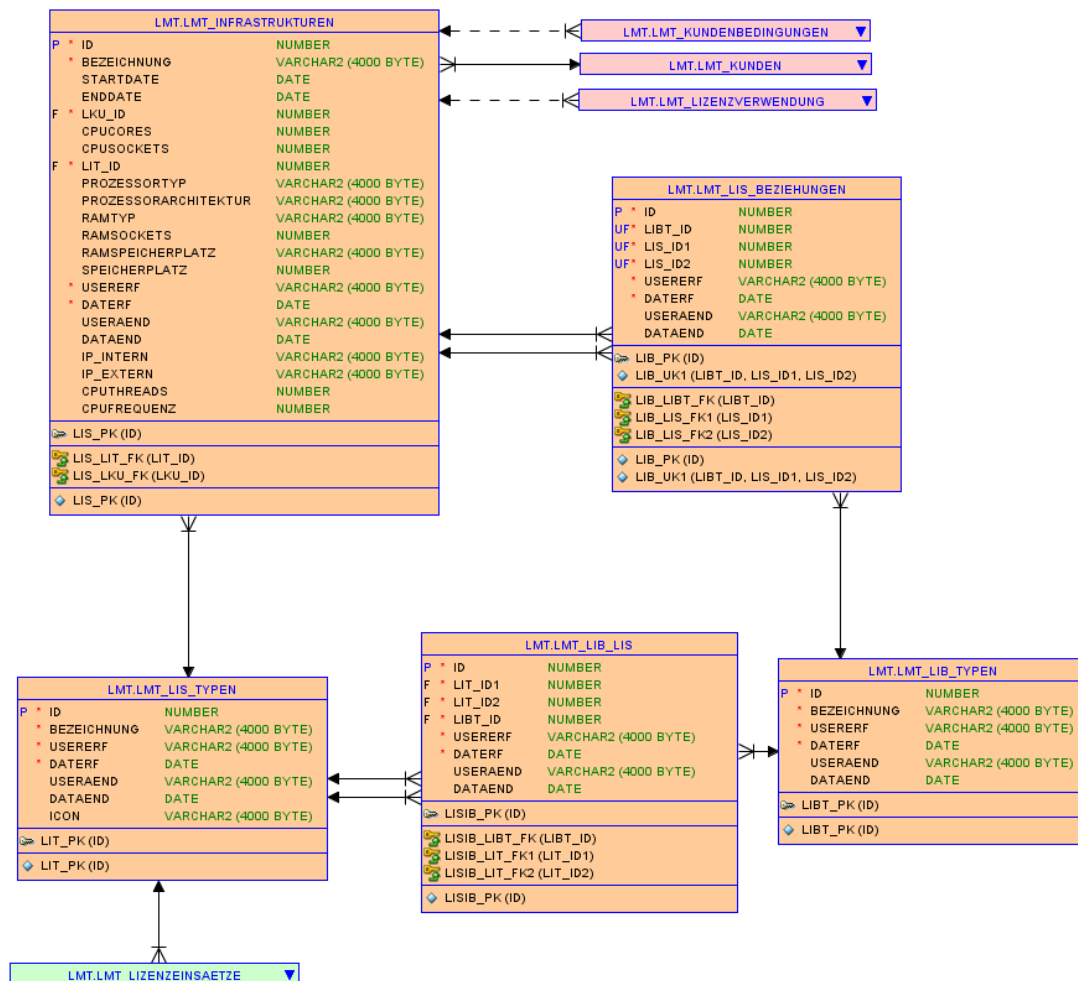


Abbildung 21: Datenmodell Infrastrukturbereich

Der Aufbau ist sehr dynamisch durch flexibel hinzufügbare Typen gestaltet, wodurch einer stetigen Erweiterung der Abbildung der Systemlandschaft nichts im Wege steht. Die wichtigste Tabelle in diesem Bereich stellt die LMT_INFRASTRUKTUR-Tabelle dar. Hier werden die eigentlichen Komponenten gespeichert und sowohl dem Kunden zugewiesen, als auch bei einer Lizenzanfrage zu den Kundenbedingungen hinzugefügt. Ebenfalls können eingesetzte Lizenzen bestimmten Infrastrukturteilen zugewiesen werden. Über die Tabelle LMT_LIS_TYPEN können alle möglichen Arten von Typen den einzelnen Infrastrukturkomponenten zugewiesen werden, aber über dies hinaus, können dadurch auch Gruppierungen, wie beispielsweise Cluster oder Abteilungen ermöglicht werden. Ebenfalls möglich ist dadurch die Abbildung von virtuellen Maschinen.

Tabelle 15: Spaltendefinition der Tabelle LMT_INFRASTRUKTUREN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation des Infrastrukturobjektes
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung der Infrastruktur
STARTDATE	DATE	Datum, ab wann Infrastruktur aktiv genutzt wird
ENDDATE	DATE	Datum, bis wann die Infrastruktur aktiv verwendet wurde
LKU_ID	NUMBER	Referenz auf den Kunden, dem die Infrastruktur zugewiesen ist
CPUCORES	NUMBER	Core-Anzahl einer CPU
CPUSOCKETS	NUMBER	CPU-Socketanzahl eines Mainboards/Rechners
LIT_ID	NUMBER	Referenz auf den Infrastrukturtyp
PROZESSORTYP	VARCHAR2(4000)	Typ eines Prozessors (bspw. Core i7-8000)
PROZESSORARCHITEKTUR	VARCHAR2(4000)	Architektur eines Prozessors (x86, x64, ...)
RAMTYP	VARCHAR2(4000)	Typ eines RAMs (DDR3, DDR4, ...)
RAMSOCKETS	NUMBER	RAM-Socketanzahl eines Mainboards/Rechners
RAMSPEICHERPLATZ	VARCHAR2(4000)	Speichergröße eines RAM-Riegels
SPEICHERPLATZ	NUMBER	Festplattenspeicher eines Rechners
IP_INTERN	VARCHAR2(4000)	Kundeninterne IP eines Rechners
IP_EXTERN	VARCHAR2(4000)	Öffentliche IP eines Rechners
CPUTHREADS	NUMBER	Threads einer CPU
CPUFREQUENZ	NUMBER	Taktfrequenz einer CPU

Tabelle 16: Spaltendefinition der Tabelle LMT_LIS_TYPEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Infrastrukturtyps
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung eines Infrastrukturtyps
ICON	VARCHAR2(4000)	Apex-Item-Code für die Darstellung im Infrastruktur-Tree

Um die einzelnen Komponenten zu verknüpfen, gibt es die Tabelle LMT_LIS_BEZIEHUNGEN, mit deren Hilfe man beispielsweise diverse virtuelle Maschinen mit einer realen Hardware verknüpfen kann oder aber auch mehrere reale Maschinen zu einem Cluster zusammenfügen kann. Auch möglich ist es, eine große Anzahl an Clientrechner zu bestimmten Abteilungen zusammenzufassen, um diese globaler verwalten zu können. Über die Tabelle LMT_LIB_TYPEN werden die einzelnen unterschiedlichen Beziehungstypen festgelegt, welche durch die eben beschriebenen

verschiedenen Arten von Beziehungen auftreten können. Diese können zukünftig leicht erweitert werden, zum aktuellen Stand handelt es sich aber konkret um folgende:

- Abteilungszuweisung
- Clusterzuweisung
- Serverzuweisung
- CPU-Zuweisung

Tabelle 17: Spaltendefinition der Tabelle LMT_LIS_BEZIEHUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Infrastrukturbeziehung
LIBT_ID	NUMBER	Referenz auf einen Infrastrukturbeziehungstypen
LIS_ID1	NUMBER	Referenz auf eine Master-Infrastruktur einer hierarchischen Beziehung oder eine erste Infrastruktur einer gleichwertigen Beziehung
LIS_ID2	NUMBER	Referenz auf eine Detail-Infrastruktur einer hierarchischen Beziehung oder eine zweite Infrastruktur einer gleichwertigen Beziehung

Tabelle 18: Spaltendefinition der Tabelle LMT_LIB_TYPEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Infrastrukturbeziehungstypen
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung eines Infrastrukturbeziehungstypen

Über die Tabelle LMT_LIB_LIS wird festgelegt, welche Infrastrukturtypen über welche Beziehungstypen miteinander verknüpft werden können, um eine entsprechende Vorauswahl beim Erstellen der jeweiligen Verbindung durchführen zu können und die Eingabe für den Endbenutzer zu erleichtern und Fehleingaben zu unterbinden.

Tabelle 19: Spaltendefinition der Tabelle LMT_LIB_LIS

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Infrastrukturbeziehungsvariante, welche definiert, welche Infrastrukturtypen in welcher Beziehung zueinander stehen dürfen
LIT_ID1	NUMBER	Referenz auf einen Master-Infrastrukturtyp einer hierarchischen Beziehung oder einen ersten Infrastrukturtyp einer gleichwertigen Beziehung

LIT_ID2	NUMBER	Referenz auf einen Detail-Infrastrukturtyp einer hierarchischen Beziehung oder einen zweiten Infrastrukturtyp einer gleichwertigen Beziehung
LIBT_ID	NUMBER	Referenz auf einen Infrastrukturbeziehungstypen

5.3.3 Softwarebereich

Der nächste Bereich dient der Erfassung und Wartung von Software, deren Nutzung im System mittels Lizenzen abgedeckt werden soll. In der Tabelle

LMT_SOFTWAREPACKAGE werden die Bezeichnungen diverser Softwareprodukte erfasst. So ist hier unter anderem „Oracle Database“ oder „Microsoft Office“ ein mögliches Beispiel für so ein Softwarepackage.

Tabelle 20: Spaltendefinition der Tabelle LMT_SOFTWAREPACKAGES

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Softwarepackages/einer Software
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung des Softwarepackages/der Software

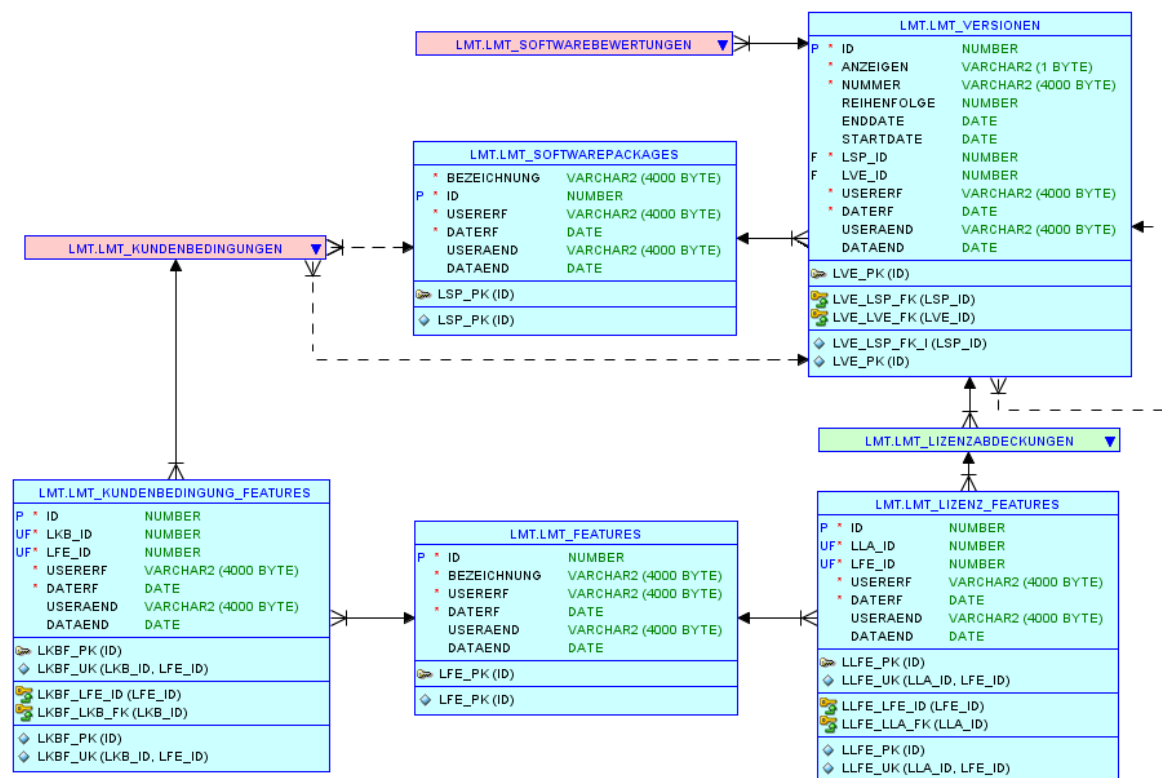


Abbildung 22: Datenmodell Softwarebereich

Die einzelnen Versionen werden in einer separaten Tabelle LMT_VERSIONEN gewartet, bei Oracle Datenbanken könnte das beispielsweise die Version 11g oder 12c sein, während es bei Office unter anderem Version 2013 oder 2016 wäre. Durch das Startdate kann bestimmt werden, ab wann entsprechende Versionen verfügbar sind, während das optionale Attribut Enddate das Ende der Erwerbsmöglichkeit entsprechender Software entspricht.

Tabelle 21: Spaltendefinition der Tabelle LMT_VERSIONEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Softwareversion
ANZEIGEN	VARCHAR2(1)	Kennzeichen, ob die Version angezeigt werden soll, aktuell noch nicht implementiert
NUMMER	VARCHAR2(4000)	Die Versionsnummer einer Softwareversion
REIHENFOLGE	NUMBER	Sortierreihenfolge der Version beim Auflisten aller Versionen einer Software
STARTDATE	DATE	Datum ab wann eine Version verfügbar ist
ENDDATE	DATE	Datum bis wann eine Version verfügbar war
LSP_ID	NUMBER	Referenz auf eine Software
LVE_ID	NUMBER	Referenz auf eine Softwareversion, aktuell nicht verwendet

Die Tabelle LMT_LIZENZ_FEATURES stellt eine m:n-Beziehung von Lizenzabdeckungen zu deren Features aus Tabelle LMT_FEATURES dar. Zu so einer Lizenzabdeckung gehört auch die Zuweisung zu einer Softwareversion. Die Lizenzabdeckung gibt also an, welche Softwareversion eine Lizenz abdeckt und welche Features davon sie unterstützt.

Tabelle 22: Spaltendefinition der Tabelle LMT_LIZENZ_FEATURES

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation der Lizenz-Feature-Zuweisung
LLA_ID	NUMBER	Referenz auf eine Lizenzabdeckung
LFE_ID	NUMBER	Referenz auf ein Feature

Tabelle 23: Spaltendefinition der Tabelle LMT_FEATURES

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation des Software-Features
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung des Features

Die Features dienen zur Darstellung diverser Softwarefunktionalitäten. Einerseits können sie beispielsweise bei Office-Paketen die unterschiedlichen enthaltenen Programme, wie Texterstellungs- oder Präsentationsprogramme sein, während sie bei Datenbanken beispielsweise die enthaltenen Optionen auflisten können. Sie dienen dazu, dass Kunden entsprechend die benötigten Features herauspicken können, die sie im Unternehmensalltag benötigen und somit eine Eingrenzung der möglichen Lizenzvarianten durchgeführt werden kann. Die Zuweisung zur Kundenanforderung beziehungsweise der konkreten Kundenbedingung einer Kundenanforderung geschieht über die Tabelle LMT_KUNDENBEDINGUNG_FEATURES. Die Kundenbedingungen stellen das Gegenstück zur Lizenzabdeckung dar, sie geben an welches Softwarepackage oder welche Softwareversion sich der Kunde wünscht und gleichzeitig, welche konkreten Features einer Software benötigt werden.

Tabelle 24: Spaltendefinition der Tabelle LMT_KUNDENBEDINGUNG_FEATURES

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Feature-Zuweisung zu einer Kundenbedingung, welche eine entsprechende Referenz auf die gewünschte Software(version) beinhaltet.
LKB_ID	NUMBER	Referenz auf eine Kundenbedingung
LFE_ID	NUMBER	Referenz auf ein Feature

Wie ausführlich die Featureliste erfasst wird, hängt vor allem von der Variation der entsprechenden Lizenzbedingungen ab. Wichtig ist, dass Features, welche nur bei bestimmten Lizenzversionen vorhanden sind, aufgelistet werden, damit beim Ermitteln von möglichen Lizenzen, welche erworben werden können, berücksichtigt werden kann, ob der Kunde ein entsprechendes Feature tatsächlich benötigt, und in Folge dessen der Pool an möglichen Lizenzen einschränkt werden kann.

5.3.4 Lizenzbereich

Der Lizenzbereich dient dazu, Lizenzen und deren Anforderungen abzubilden. In der Tabelle LMT_LIZENZEN werden die einzelnen verfügbaren Lizenzen mit ihren Anforderungen erfasst. Die Tabellen LMT_LIZENZFORMEN, LMT_LIZENZARTEN, LMT_LIZENZKLASSEN und LMT_LIZENZTYPEN bilden die in Kapitel 2 Literaturanalyse

beschriebene Klassifizierungen von Lizenzen ab. Weiters können über die LMT_LIZENZBEDINGUNGEN noch zusätzliche dynamisch generierte Attribute abgefragt werden.

Tabelle 25: Spaltendefinition der Tabelle LMT_LIZENZFORMEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzform
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung einer Lizenzform
REIHENFOLGE	NUMBER	Sortierreihenfolge bei der Auflistung der Lizenzformen

Tabelle 26: Spaltendefinition der Tabelle LMT_LIZENZARTEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzart
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung einer Lizenzart
REIHENFOLGE	NUMBER	Sortierreihenfolge bei der Auflistung der Lizenzarten

Tabelle 27: Spaltendefinition der Tabelle LMT_LIZENZKLASSEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzklasse
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung einer Lizenzklasse
REIHENFOLGE	NUMBER	Sortierreihenfolge bei der Auflistung der Lizenzklassen

Tabelle 28: Spaltendefinition der Tabelle LMT_LIZENZTYPEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Lizenztyps
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung eines Lizenztyps
REIHENFOLGE	NUMBER	Sortierreihenfolge bei der Auflistung der Lizenztypen

Die in LMT_LIZENZEN abgebildeten Anforderungen stellen eine Grundmenge an klassischen Kriterien dar, die von Lizenzen häufig verwendet werden. Diese werden deshalb aktuell nicht als dynamische Werte dargestellt, um schneller auf alle entsprechenden Daten einer Lizenz zugreifen und diese effizienter ermitteln zu können. Einen Überblick über die möglichen Anforderungen und wie diese genau interpretiert werden, wird in Kapitel 6 Implementierung geboten. Grundlegend kann man hier User-, Geräte- und CPU-Mengen erfassen. Genauer gesagt die tatsächliche Anzahl von der

Lizenz abgedeckten User, Geräte oder CPUs, sowie deren Minimal- und Maximal-Mengen, für die die jeweilige Lizenz zugelassen ist.

Das Weitern können Laufzeiten erfasst werden (ebenfalls wieder eine konkrete Laufzeit, sowie eine Minimallaufzeit und Maximallaufzeit und auch die Intervalleinheit (Stunde, Tag, Woche, Monat, Quartal, Jahr) der Laufzeitangabe). Diese dienen zwei Zwecken, einerseits können damit bestimmte SaaS-Anforderungen dargestellt werden, andererseits können so Wartungsverträge für bestimmte Lizenzen erfasst werden. Wartungsverträge stellen separate Einträge in die LMT_LIZENZEN-Tabelle dar, welche dann über den entsprechenden Beziehungstyp mit der jeweilig zugehörigen Lizenz verknüpft werden.

Tabelle 29: Spaltendefinition der Tabelle LMT_LIZENZEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenz
LLT_ID	NUMBER	Referenz auf einen Lizenztypen
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung einer Lizenz
PREIS	NUMBER	Preis für die Wertmenge einer Lizenz (abhängig vom Lizenztyp und der jeweiligen zugehörigen Spalte USERANZAHL, DEVICEANZAHL, CPUCORESANZAHL, SOCKETANZAHL, CPUTHREADANZAHL, CPUCORESANZAHL_USED, SOCKETANZAHL_USED, CPUTHREADANZAHL_USED)
STARTDATE	DATE	Datum ab wann eine Lizenz verfügbar ist
ENDDATE	DATE	Datum bis wann eine Lizenz verfügbar war
LLF_ID	NUMBER	Referenz auf eine Lizenzform
LLK_ID	NUMBER	Referenz auf eine Lizenzklasse
LLAR_ID	NUMBER	Referenz auf eine Lizenzart
MIN_USER	NUMBER	Minimal verlangte Useranzahl
USERANZAHL	NUMBER	Useranzahl auf die skaliert wird (Preis oder Maximalwerte)
MIN_CPUCORES	NUMBER	Minimal verlangte CPU-Core-Anzahl
CPUCORESANZAHL	NUMBER	CPU-Core-Anzahl auf die skaliert wird (Preis oder Maximalwerte)
MIN_SOCKET	NUMBER	Minimal verlangte Socketanzahl
SOCKETANZAHL	NUMBER	Socketanzahl auf die skaliert wird (Preis oder Maximalwerte)

MIN_DEVICES	NUMBER	Minimal verlangte Geräteanzahl
DEVICESANZAHL	NUMBER	Geräteanzahl auf die skaliert wird (Preis oder Maximalwerte)
PROZESSORTYP	VARCHAR2(4000)	Erlaubter Prozessortyp (bspw. Core i7-8000), aktuell noch nicht implementiert
PROZESSORARCHITEKTUR	VARCHAR2(4000)	Erlaubte Prozessorarchitektur (x86, x64, ...), aktuell noch nicht implementiert
RAMTYP	VARCHAR2(4000)	Erlaubter RAM-Typ (DDR3, DDR4, ...), aktuell noch nicht implementiert
RAMARCHITEKTUR	VARCHAR2(4000)	Erlaubte RAM-Architektur, aktuell keine Verwendung
MIN_RAMSPEICHERPLATZ	NUMBER	Minimal verlangte RAM-Speichergröße, aktuell noch nicht implementiert
RAMSPEICHERPLATZ	NUMBER	RAM-Speichergröße, aktuell noch nicht implementiert
MAX_USER	NUMBER	Maximal erlaubte Useranzahl
MAX_CPUCORES	NUMBER	Maximal erlaubte CPU-Core-Anzahl
MIN_CPUTHREADS	NUMBER	Minimal verlangte CPU-Thread-Anzahl
CPUTHREADANZAHL	NUMBER	CPU-Thread-Anzahl auf die skaliert wird (Preis oder Maximalwerte)
MAX_CPUTHREADS	NUMBER	Maximal erlaubte CPU-Thread-Anzahl
MAX_SOCKET	NUMBER	Maximal erlaubte Socketanzahl
MAX_DEVICES	NUMBER	Maximal erlaubte Geräteanzahl
MAX_RAMSPEICHERPLATZ	NUMBER	Maximal erlaubte RAM-Speichergröße, aktuell noch nicht implementiert
MIN_SOCKET_USED	NUMBER	Minimal verlangte Anzahl an Sockets in Verwendung
SOCKET_USED	NUMBER	Anzahl der Sockets in Verwendung auf die skaliert wird (Preis oder Maximalwerte)
MAX_SOCKET_USED	NUMBER	Maximal erlaubte Anzahl der Sockets in Verwendung
MIN_LAUFZEIT	NUMBER	Minimal verlangte Laufzeitdauer, aktuell noch nicht implementiert
LAUFZEIT	NUMBER	Laufzeitdauer auf die skaliert wird (Preis oder Maximalwerte), aktuell noch nicht implementiert
MAX_LAUFZEIT	NUMBER	Maximal erlaubte Laufzeitdauer, aktuell noch nicht implementiert
LAUFZEITTYP	VARCHAR2(1)	Laufzeittyp, mögliche Werte: Y...Year, M...Month, Q...Quartal, W...Week, D...Day, H...Hours

MIN_CPUTHREADS_USED	NUMBER	Minimal verlangte Anzahl an CPU-Threads in Verwendung
CPUTHREADANZAHL_USED	NUMBER	Anzahl der CPU-Threads in Verwendung auf die skaliert wird (Preis oder Maximalwerte)
MAX_CPUTHREADS_USED	NUMBER	Maximal erlaubte Anzahl der CPU-Threads in Verwendung
MIN_CPUCORES_USED	NUMBER	Minimal verlangte Anzahl an CPU-Cores in Verwendung
CPUCORESANZAHL_USED	NUMBER	Anzahl der CPU-Cores in Verwendung auf die skaliert wird (Preis oder Maximalwerte)
MAX_CPUCORES_USED	NUMBER	Maximal erlaubte Anzahl der CPU-Cores in Verwendung
VIRTUAL_DESKTOP_VALUES	VARCHAR2(1)	Angabe, ob auf virtuelle Werte geachtet wird: Y...Werte werden von virtuellen Umgebungen herangezogen, N...Werte werden von der realen Hardware bezogen, auf der die virtuellen Maschinen laufen (aktuell noch nicht implementiert)

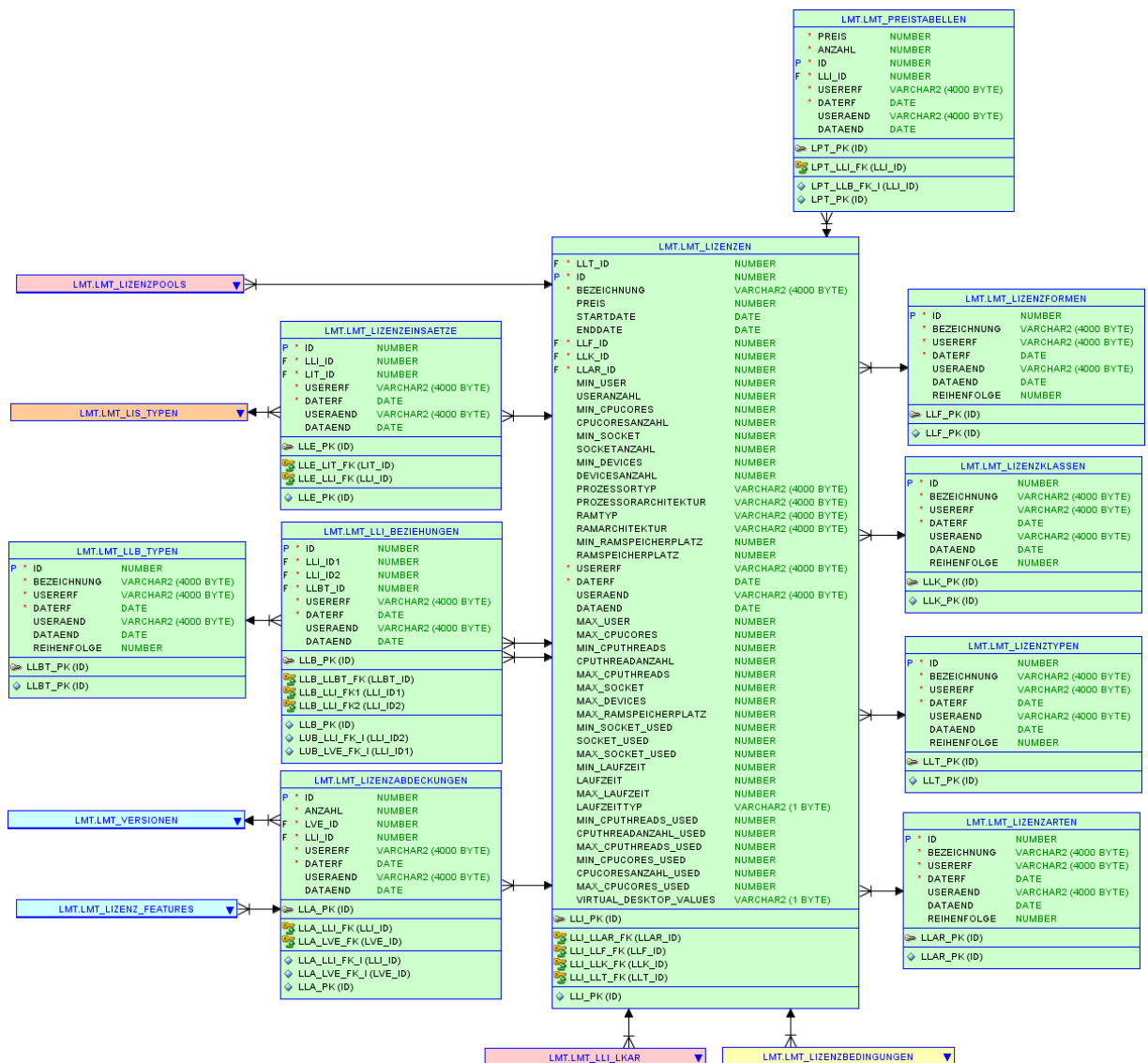


Abbildung 23: Datenmodell Lizenzbereich

Über die Tabelle **LMT_LIZENZEINSAETZE** werden die Auswahlmöglichkeiten der Lizenz zu bestimmten Infrastrukturen gespeichert, welche wiederum bei der Lizenzverwendung, sowie der Lizenzanforderung ausgewählt werden können. Nur Infrastrukturen mit den hier zugewiesenen Infrastrukturtypen können bei der jeweiligen Zuweisung ausgewählt werden. Dies dient wiederum zur Erleichterung der Erfassung von Daten für den Endbenutzer und minimiert Falscheingaben.

Tabelle 30: Spaltendefinition der Tabelle **LMT_LIZENZEINSAETZE**

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Lizenzesinsatzes, welcher definiert, welche Lizenzen welchem Infrastrukturtypen zugewiesen werden können.
LLI_ID	NUMBER	Referenz auf eine Lizenz
LIT_ID	NUMBER	Referenz auf einen Infrastrukturtypen

Um die Kosten zu ermitteln gibt es einerseits die direkte Preisangabe über die Lizenz selbst, also auch eine weitere Tabelle LMT_PREISTABELLEN, in denen Mengenrabatte und andere Preisnachlässe abgebildet werden können.

Tabelle 31: Spaltendefinition der Tabelle LMT_PREISTABELLEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Preistabelleneintrages
PREIS	NUMBER	Preis pro Lizenz zu einer bestimmten Mindestmenge, welche in Spalte ANZAHL definiert wird.
ANZAHL	NUMBER	Eine Mindestmenge, ab der der aktuelle Preis aus Spalte PREIS pro Lizenz gilt.
LLI_ID	NUMBER	Referenz auf eine Lizenz

Über die Tabelle LMT_LIZENZABDECKUNGEN wird definiert, welche Softwareversion eine Lizenz abdeckt, sowie welche Features von der entsprechenden Lizenz unterstützt werden.

Tabelle 32: Spaltendefinition der Tabelle LMT_LIZENZABDECKUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzabdeckung
ANZAHL	NUMBER	Anzahl der abgedeckten Software bei Multisoftwareabdeckungen, aktuell nicht in Verwendung
LVE_ID	NUMBER	Referenz auf eine Softwareversion
LLI_ID	NUMBER	Referenz auf eine Lizenz

Um Upgradelizenzen, Addonlizenzen und Wartungsverträge zu einer entsprechenden Lizenz zuweisen zu können, gibt es die Tabelle LMT_LLI_BEZIEHUNGEN. Außerdem kann hier angegeben werden, wenn zwei Lizenzen mit unterschiedlichen Bedingungen dieselbe Grundlizenz darstellen (wie dies beispielsweise bei der Oracle Database Standard Edition 2 der Fall ist, dies wird in Kapitel 6.5 Überarbeitungen noch genauer beschrieben). Ist dies der Fall, kann man die entsprechenden Lizenzen mit dem Beziehungstyp „Konvertierbar“ miteinander verknüpfen.

Tabelle 33: Spaltendefinition der Tabelle LMT_LLI_BEZIEHUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzbeziehung
LLI_ID1	NUMBER	Referenz auf eine Master-Lizenz einer hierarchischen Beziehung oder eine erste Lizenz einer gleichwertigen Beziehung
LLI_ID2	NUMBER	Referenz auf eine Detail- Lizenz einer hierarchischen Beziehung oder eine zweite Lizenz einer gleichwertigen Beziehung
LLBT_ID	NUMBER	Referenz auf einen Lizenzbeziehungstyp

Tabelle 34: Spaltendefinition der Tabelle LMT_LLB_TYPEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Lizenzbeziehungstypen
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung eines Lizenzbeziehungstyps
REIHENFOLGE	NUMBER	Sortierreihenfolge bei der Auflistung der Lizenzbeziehungstypen

Interessant zu erwähnen ist vielleicht noch, dass Upgradelizenzen natürlich auch als Vorgänger Upgradelizenzen haben können, welche dann wiederum weitere Vorgängerlizenzen voraussetzen. Wichtig ist nur, dass am Anfang des Pfades eine Vollversion steht. Ist der gesamte Upgradepfad erfüllt, kann auch eine Upgrade-Lizenz eingesetzt werden.

5.3.5 Bereich für dynamische Bedingungen

Der nächste Bereich ist für komplexe Lizenzen gedacht. Reichen die gängigsten Restriktionen nicht aus, um Lizenzbedingungen vollständig darstellen zu können, so kann dies möglicherweise durch eine Erstellung einer dynamischen Bedingung erfolgen.

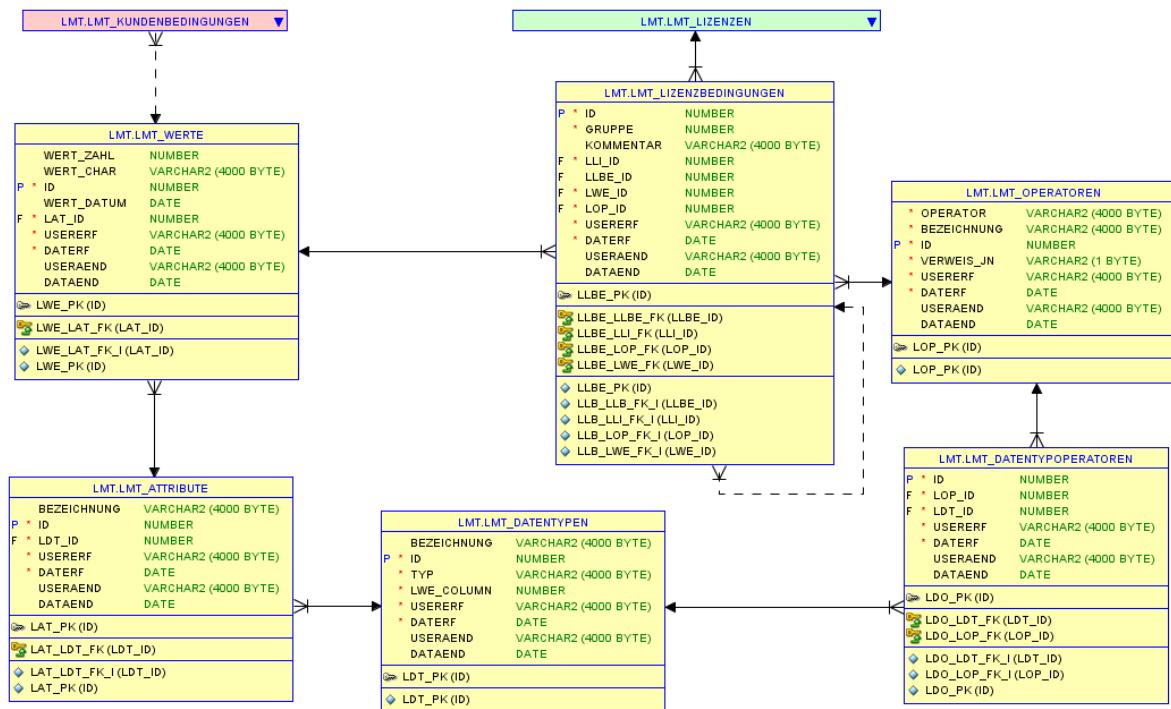


Abbildung 24: Datenmodell dynamische Bedingungen

Über die Tabelle LMT_LIZENZBEDINGUNG wird eine zusätzliche Bedingung, die nicht bereits im Datenmodell (vor allem in der Tabelle LMT_LIZENZEN) enthalten ist, angelegt. Die benötigten Werte für diese Bedingung werden in der Tabelle LMT_WERTE erfasst. Um das Attribut zu benennen, wird die Tabelle LMT_ATTRIBUTES herangezogen, damit das Programm beim Kunden nach einer konkreten Detailinformation nachfragen kann. Um nach der Kundeneingabe zu überprüfen, ob die Werte des Kunden den Anforderungen der Lizenz entsprechen, wird über den Datentyp in Tabelle LMT_DATENTYPEN (unter anderem die Typen Character, Number, Date) der entsprechende Wert gespeichert.

Tabelle 35: Spaltendefinition der Tabelle LMT_LIZENZBEDINGUNGEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Lizenzbedingung
GRUPPE	NUMBER	Gruppierung von Lizenzbedingung, aktuell nicht in Verwendung, unterschiedliche Bedingungen werden durch separate Lizenzen dargestellt
KOMMENTAR	VARCHAR2(4000)	Kommentar zur Lizenzbedingung
LLI_ID	NUMBER	Referenz auf eine Lizenz
LLBE_ID	NUMBER	Selbstreferenz auf eine vorhandene Lizenzbedingung bei Gruppen, aktuell nicht in Verwendung, unterschiedliche Bedingungen werden durch separate Lizenzen dargestellt

LWE_ID	NUMBER	Referenz auf einen dynamischen Wert
LOP_ID	NUMBER	Referenz auf den eingesetzten Operator

Tabelle 36: Spaltendefinition der Tabelle LMT_WERTE

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines dynamischen Wertes
WERT_ZAHL	NUMBER	Spalte für einen Zahlenwert
WERT_CHAR	VARCHAR2(4000)	Spalte für einen Text
WERT_DATUM	DATE	Spalte für einen Datumswert
LAT_ID	NUMBER	Referenz auf ein Attribut

Tabelle 37: Spaltendefinition der Tabelle LMT_ATTRIBUTE

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation von Attributen
BEZEICHNUNG	VARCHAR2(4000)	Name des Attributs
LDT_ID	NUMBER	Referenz auf Datentyp des Attributs

Tabelle 38: Spaltendefinition der Tabelle LMT_DATENTYPEN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation des Datentyps
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung des Datentyps
TYP	VARCHAR2(4000)	Tatsächliche Datenbanktypbezeichnung
LWE_COLUMN	NUMBER	Datentypspalte aus LMT_WERTE (1 ... WERT_ZAHL, 2 ... WERT_CHAR oder 3 ... WERT_DATUM)

Über die Tabelle LMT_DATENTYPOPERATOREN werden dem Erfasser der Lizenzbedingung alle möglichen Vergleichsoperatoren angezeigt, welche für den entsprechenden Datentyp möglich sind. So kann beispielsweise ein Character-Wert mit „like“ verglichen werden, ein Datumswert mit „between“ analysiert werden und eine Zahl auf andere Vergleichsoperatoren, wie beispielsweise „größer-gleich“, zurückgreifen. Der benötigte Vergleichsoperator wird also beim Erstellen der Lizenzbedingung ausgewählt und in der Tabelle LMT_OPERATOREN hinterlegt.

Tabelle 39: Spaltendefinition der Tabelle LMT_DATENTYPENOPERATOREN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation der möglichen Datentypoperatoren, welche beim referenzierten Datentyp möglich sind
LOP_ID	NUMBER	Referenz auf den Operator
LDT_ID	NUMBER	Referenz auf den Datentyp

Tabelle 40: Spaltendefinition der Tabelle LMT_OPERATOREN

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation eines Operators
OPERATOR	VARCHAR2(4000)	Darstellung eines Operators im Quellcode
BEZEICHNUNG	VARCHAR2(4000)	Bezeichnung eines Operators
VERWEIS_JN	VARCHAR2(1)	Kennzeichen, ob ein Verweis auf einen weiteren Operator erfolgt, wird aktuell nicht verwendet

5.3.6 Loggingbereich

Der letzte Bereich ist nicht direkt mit den anderen Tabellen verknüpft. Er dient dem Logging, allen voran dem Exception-Handling. Tritt eine Exception im Programmcode auf, wird diese in der Tabelle LMT_LOGS abgespeichert.

LMT.LMT_LOGS	
P * ID	NUMBER
DATUM	TIMESTAMP
FORMAT_ERROR_STACK	VARCHAR2 (4000 BYTE)
FORMAT_ERROR_BACKTRACE	VARCHAR2 (4000 BYTE)
FORMAT_CALL_STACK	VARCHAR2 (4000 BYTE)
TEXT	VARCHAR2 (4000 BYTE)
* TYPE	VARCHAR2 (4000 BYTE)
LOG_PK (ID)	
LOG_PK (ID)	

Abbildung 25: Datenmodell für Logtabelle

Durch den Error-Backtrace kann anschließend ermittelt werden, wie und wo es zu besagter Exception kam, um einen etwaigen Bugfix erstellen und einspielen zu können.

Tabelle 41: Spaltendefinition der Tabelle LMT_LOGS

NAME	TYPE	KOMMENTAR
ID	NUMBER	Identifikation einer Logzeile
DATUM	TIMESTAMP(6)	Zeitpunkt des Logeintrags
FORMAT_ERROR_STACK	VARCHAR2(4000)	Information von DBMS_UTILITY.FORMAT_ERROR_BACKTRACE

FORMAT_ERROR_BACKTRACE	VARCHAR2(4000)	Information von DBMS_UTILITY. FORMAT_ERROR_BACKTRACE
FORMAT_CALL_STACK	VARCHAR2(4000)	Information von DBMS_UTILITY. FORMAT_CALL_STACK
TEXT	VARCHAR2(4000)	Optionaler Text beim Erfassen einer Logzeile
TYPE	VARCHAR2(4000)	Typ des Logeintrags (LOG, ERROR)

Damit sind alle Tabellen des Datenmodells näher beschrieben worden, die für die Implementierung des Systems benötigt werden. Da die grundlegende Planung damit abgeschlossen ist, kommen wir im nächsten Kapitel zur Beschreibung der konkreten Implementierung eines Prototyps für das in dieser Arbeit beschriebene Unterstützungstool zur Lizenzverwaltung.

6. Implementierung

Da nun die Anforderungen definiert sind, sowie die geplante Umsetzung ausmodelliert und erklärt ist, soll im nächsten Schritt ein Prototyp entwickelt werden. Hierzu wurde ein Teilabschnitt des geplanten Systems ausgewählt und implementiert, um dessen Funktionalität veranschaulichen zu können, sowie den Aufbau der Applikation ersichtlich zu machen. Der ausgewählte Abschnitt ist die Anforderungsanalyse, welche den Kunden dabei unterstützen soll, die passende Lizenz zu seinen Anforderungen zu finden.

Die genauen Anforderungen und Abgrenzungen werden in den folgenden Kapiteln beschrieben. Anschließend erfolgt eine Beschreibung der Umsetzung, sowie eine Analyse der Problemstellen und eine Auflistung der Änderungen zum ursprünglichen Konzept, die sich während der Implementierung ergaben.

6.1 Anforderungen an den Prototypen

Der Prototyp soll einen Überblick über das System geben, sowie eine Teilimplementierung gewisser Funktionen darstellen. Dies soll autark geschehen und nicht bereits in Verbund mit dem aktuell im Einsatz befindlichen Kundensystem von DBConcepts GmbH. Im ersten Schritt soll die Erfassung von benötigten Daten zur Umsetzung weiterer Funktionalität erfolgen. Die Einteilung der zu erfassenden Daten, welche für den Prototyp relevant sind, erfolgen in administrativen und kundenspezifischen Daten.

Administrative Daten:

- Kundenverwaltung
- Softwareverwaltung
- Lizenzverwaltung

Kundenspezifischen Daten:

- Infrastrukturverwaltung
- Softwareanalyse

Es sollen entsprechende Masken/Seiten erstellt werden, welche die Erfassung, Bearbeitung und Löschung dieser Daten zulässt. Entscheidend ist, die Möglichkeit zu schaffen, die benötigten Daten für den Test zu erfassen, die Validierung der

Sinnhaftigkeit von Eingaben ist in diesem ersten Prototyp nur bedingt relevant. Es wird zwar bereits ermöglicht, die meisten Daten, die das System abbildet, zu erfassen. Es wird aber keine Kontrolle über die Sinnhaftigkeit der Eingabe gewährleistet, sowie fehlt noch eine vom System geleitete Eingabe, sodass nur Werte, die für die jeweiligen Typen relevant sind, erfassbar sind. Dies hat zwei Gründe: Einerseits soll Spielraum für Testabbildungen gewährleistet bleiben, um die Eingabe von komplexeren Lizenzanforderungen und -abdeckungen nicht zu unterbinden. Andererseits soll durch zusätzliche Eingabefelder ermöglicht werden, für den späteren Echtzeitbetrieb noch weitere Anregungen für eine erleichterte Bedienung ermitteln zu können. (Beispielsweise könnte man bei einer Infrastrukturerfassung bei Systemen mit nur einem Prozessorsockel auf dem Mainboard direkt am System die Prozessordetaildaten angeben, anstatt dass diese, wie aktuell funktional erforderlich, nur von separat zugewiesenen Prozessoren ausgewertet werden können.) Des Weiteren werden im Prototypen noch keine Massenerfassungen ermöglicht, welche unter anderem das Erstellen von vielen Daten automatisiert.

In der Anforderungsanalyse soll es möglich sein, alle Lizenzvariationen von Oracle Datenbanken zu suchen und zu ermitteln, welche davon die beste Kaufentscheidung darstellt. Es werden aktuell nur die Datenbanken eingepflegt, da diese die herausfordernderen Lizenzbedingungen und mit allen inkludierten Optionen bereits einen großen Anteil an zur Verfügung stehenden Lizenzen darstellen (eine genaue Auflistung der Implementierten Lizenzen findet man in Kapitel 7.1 Lizenzfassung). Das Einpflegen weiterer Lizenzen durch den Benutzer soll aber möglich gemacht werden.

Wird eine Kundenanforderung erstellt, so können die gewünschte Software mitsamt den gewünschten Features ausgewählt werden, sowie ein Infrastrukturobjekt hinzugefügt werden, welches Zusatzinformationen zum Einsatzgebiet der benötigten Lizenz darstellt. Entscheidend für die Lizenzanalyse ist das auswählen von mindestens einem Feature. Die Zuweisung einer Infrastruktur ist optional, alternativ kann auch eine Geräteanzahl eingetragen werden, allerdings ist auch diese optional. Die Infrastrukturzuweisung wird höher priorisiert als die separat angebbare Geräteanzahl. Zusätzlich kann noch die benötigte Enduseranzahl mitangegeben werden.

Aufgrund dieser Angaben soll nun eine Filterung und Auflistung aller infrage kommender Lizenzen erfolgen, welche nach Preis und abgedeckter Featureanzahl, sowie dem Erfüllen der nötigen Bedingungen für einen Lizenzeinsatz sortiert werden können soll.

6.2 Abgrenzungen

Um die Grenzen des Prototyps zu definieren, wird im Folgenden noch ein Überblick gegeben, was zu dem Bereich der umgesetzten Funktionalitäten des Prototyps zählt und was bereits darüber hinaus geht. So sind beispielsweise diverse Datenerfassungen zwar bereits auf der Oberfläche ersichtlich und deren Erfassung auch implementiert, sie wirken sich aber funktional noch nicht auf die Umsetzung aus. Unter anderem gibt es in vielen Bereichen die Möglichkeit zur Angabe eines Zeitraumes, welcher aber aktuell noch nicht berücksichtigt wird. Des Weiteren ist zu erwähnen, dass erst einmal nur eine reine Erfassungsmöglichkeit geschaffen werden soll, die auch für bestimmte Typen irrelevante Einträge ermöglicht. So wird es möglich sein für Abteilungen eine Socketanzahl zu erfassen, obwohl diese hier keinerlei Bedeutung hat. Die eingeschränkte Darstellung bei der Erfassung, auf der entsprechende Eingaben nur dem jeweiligen gerade zu erfassenden Typ ermöglicht werden, ist nicht Teil des Prototyps, sondern ist als Komfortfeature nachträglich zu entwickeln, etwa durch das Ausblenden von irrelevanten Datenfeldern. Auch soll die uneingeschränkte Erfassung von Daten die Möglichkeit bieten, komplexere Abbildungen von Lizenzen, Infrastruktur und deren Einsatz zu testen, welche momentan noch gar nicht benötigt werden.

Infrastrukturen können aktuell nur mit einer einzigen anderen Komponente verknüpft werden, da weitere Beziehungen reine informative Erfassungen darstellen, welche zur Lizenzermittlung nicht relevant sind. So kann ein virtueller Server nur einem tatsächlichen Server zugewiesen werden, nicht aber einer Abteilung. In Zukunft soll dies natürlich schon möglich sein, da aber die Lizenzermittlung nur für einen Server oder für eine Abteilung ermittelt werden kann, wird entsprechend auch nur eine Beziehung zugelassen, um nicht bei der Anforderung noch weitere Eingaben verlangen zu müssen. Man muss sich also aktuell bei der Erstellung der Infrastruktur darauf festlegen, was man nun bei der Kundenanforderung ermitteln will. In der fertigen Applikation kann die Auswahl natürlich auch erst bei der Lizenzermittlung erfolgen, wodurch dann auch mehrere Beziehungszuweisungen erlaubt sein werden.

In der Anforderungsanalyse kann aktuell nur eine Software pro Analyse ermittelt werden, ebenso wie nur eine Infrastrukturkomponente hierfür zugewiesen werden kann. Das Konstrukt soll aber bereits so aufgebaut werden, dass eine Mehrfachauswahl ermöglicht wird. Ob dies im Endeffekt umgesetzt wird, wird sich erst bei der Performancemessung zeigen. Sollte es zu hohe Aufwände erzeugen, mehrere Softwarepakete auf einmal zu ermitteln, so wird es sinnvoller sein, auch in Zukunft für jede Software separat eine entsprechende Lizenzermittlung durchzuführen.

Preislisten werden aktuell nicht berücksichtigt. Der Prototyp geht aktuell von einem Fixpreis aus, welchen er linear mit der benötigten Stückzahl erhöht, da Oracle im Shop keine ersichtlichen Rabatte gewährt, ist die Preistabelle für den Prototyp auch nicht relevant.

Aktuell werden bei Lizenzanforderungen noch keine bereits im Unternehmen vorhandenen Lizenzen berücksichtigt, da die Erfassung dieser im Prototyp nicht vorhanden ist. Der Kunde kann noch keinen eigenen Lizenzpool aufbauen und einsetzen und somit auch nicht angeben, ob er bestimmte bereits vorhandene Lizenzen für eine neue Lizenzanforderung mit einbeziehen will. Daraus resultierend wird auch die Startseite, welche eine Audit-Zusammenfassung der jeweiligen Kunden darstellt, leer bleiben, da entsprechende Informationen noch nicht eingepflegt werden können. Damit die Applikationsstruktur aber schon korrekt ersichtlich ist, wird das Navigieren auf diese Seite möglich sein. Selbiges gilt für die Seite zur Lizenzverwendung. Beide Bereiche werden aber im aktuellen Prototypen nur eine leere Seite darstellen.

Da keine vorhandenen Lizenzen berücksichtigt werden, ist es auch nicht möglich Upgrades zu verwenden. Diese wurden deshalb im Prototyp ebenfalls außer Acht gelassen. Im Regelfall verhält sich aber ein Upgrade wie eine Vollversion, die lediglich als zusätzliche Prüfung das Vorhandensein einer möglichen Vorgängerversion (welche über die Lizenzbeziehungen dargestellt werden wird) benötigt. Die Prozesse zur Ermittlung der passenden Lizenzen ist ansonsten ident zu der einer Vollversionslizenz.

6.3 Umsetzungen der GUI (Graphical User Interface)

Um den Prototypen zu erstellen wurde zunächst eine entsprechende Applikation im APEX-Framework angelegt und dem darunterliegenden Datenbankschema zugewiesen. Anschließend wurden die ersten Seiten für die einzelnen Bereiche der Applikation

angelegt, welche zu diesem Zeitpunkt allesamt noch leer waren. Darauf aufbauend wurde die Navigationsleiste erstellt und Schritt für Schritt die einzelnen Seiten umgesetzt. Die Datenerfassung gliedert sich im Normalfall immer auf dieselbe Weise auf: Man landet auf einer Übersichtsseite, welche eine Auflistung von allen erfassten Daten anzeigt und kann hier mittels Klick auf einen Eintrag auf die entsprechende Bearbeitungsseite gelangen. Selbiges geschieht beim Hinzufügen neuer Datensätze mithilfe eines entsprechenden Buttons, welcher sich rechts oberhalb der aufgelisteten Daten befindet, wie in Abbildung 26: Beispiel der Übersichtsseite anhand der Kundenverwaltung ersichtlich ist.

Name	Anschrift	Löschen
DBConcepts GmbH	-	
Kunde 1	-	

Abbildung 26: Beispiel der Übersichtsseite anhand der Kundenverwaltung

Die entsprechende Bearbeitungsseite für den Eintrag eines Kunden (in unserem Beispiel „DBConcepts GmbH“) bietet nun wiederum eine Dateneingabemöglichkeit, sowie eine Übersicht über weitere Details, sofern dies notwendig ist. In diesem konkreten Beispiel handelt es sich um die einzelnen User, die einem Kunden zugewiesen sind. Diese sind in Abbildung 27: Darstellung einer Datenerfassungsseite mit einer weiteren Detailübersicht anhand der Kundenbearbeitung ersichtlich. Auch hier hat man bei der Detailanzeige die Möglichkeit einen existierenden Datensatz zu bearbeiten, mittels des oben rechts befindlichen Buttons „User erstellen“ einen neuen Datensatz hinzuzufügen oder mittels Klick auf das Mistkübelsymbol in der Spalte „Löschen“ den Datensatz zu entfernen.

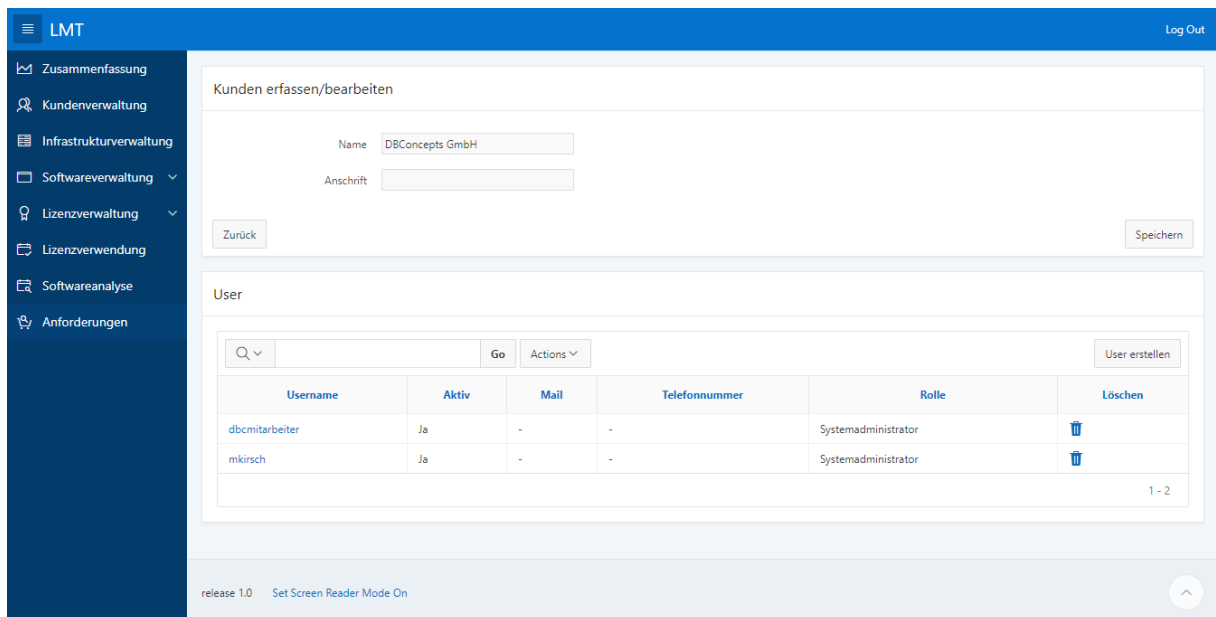


Abbildung 27: Darstellung einer Datenerfassungsseite mit einer weiteren Detailübersicht anhand der Kundenbearbeitung

Im Normalfall ist die tiefste Ebene der Datenerfassung ein modales Fenster, also ein Popup-Bereich über der Übersichtsseite, welchen man auch innerhalb des Browserfensters verschieben kann, um eventuell noch Informationen aus der Übersicht ablesen zu können. Dies ist beispielsweise in der Userbearbeitung der Fall, deren Bearbeitungsseite keine weiteren Detailseiten mehr aufweist, wie in Abbildung 28: Darstellung einer modalen Bearbeitungsseite anhand der Userbearbeitung ersichtlich.

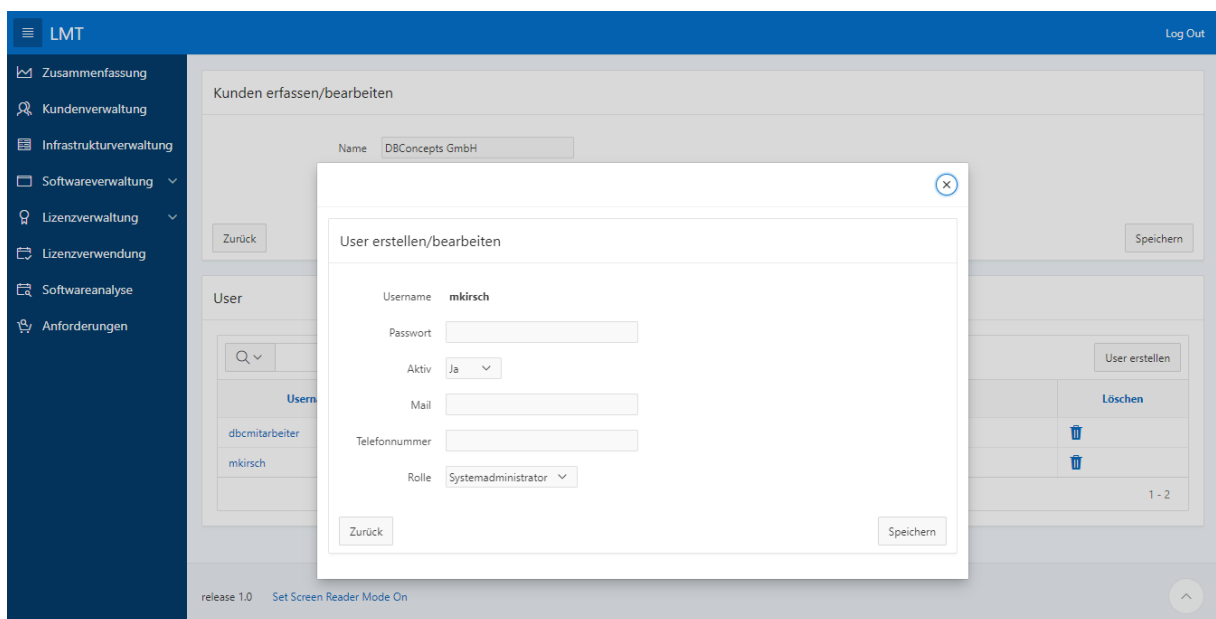


Abbildung 28: Darstellung einer modalen Bearbeitungsseite anhand der Userbearbeitung

Das Fenster ist wie erwähnt verschiebbar und die Übersichtsseite im Hintergrund einsehbar, eventuell entfallene Informationen können also überprüft werden, ohne die

Erfassungsmaske verlassen zu müssen. Da es sich aber um ein modales Fenster handelt, ist die Funktionalität der im Hintergrund befindlichen Übersichtsseite gesperrt und kann erst wieder bedient werden, wenn man das Bearbeitungsfenster über den Schließen-, den Zurück-, den Erstellen- oder den Speicher-Button verlässt, wie man in Abbildung 29: Abbildung der Features eines modalen Erfassungsfensters anhand der Userbearbeitung sieht.

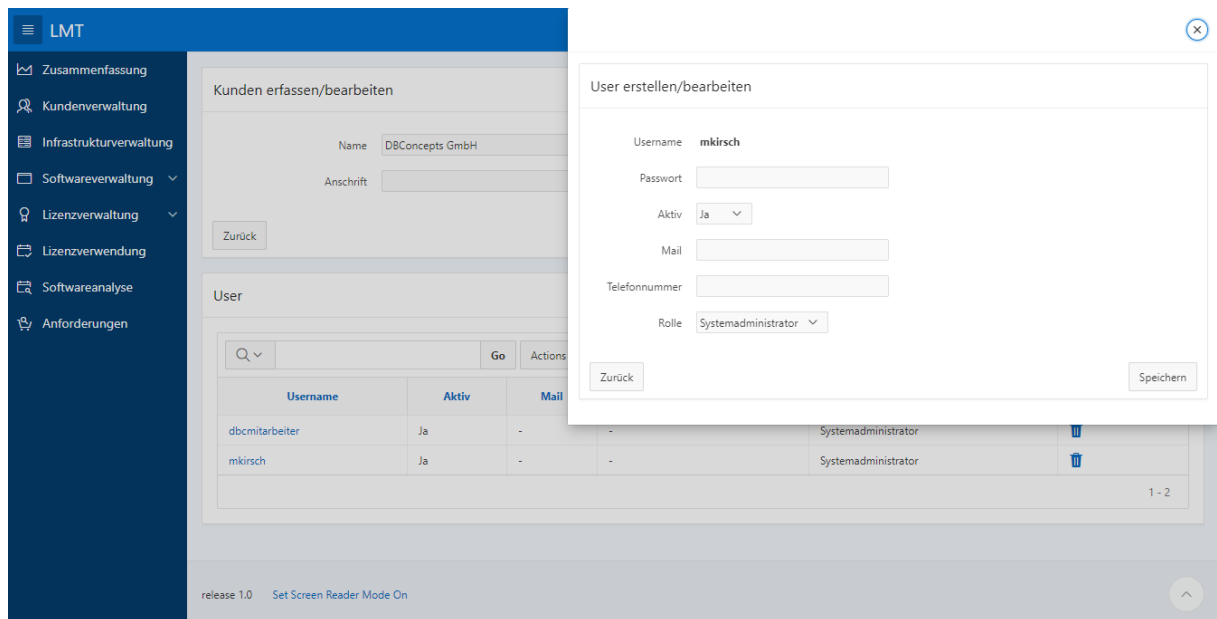


Abbildung 29: Abbildung der Features eines modalen Erfassungsfensters anhand der Userbearbeitung

Die Infrastrukturverwaltung und deren Bearbeitungsseite weisen eine äquivalente Seitenstruktur auf. Eine Besonderheit dieser Seite ist das Ergänzen um ein Komfortfeature. Zusätzlich zur tabellarischen Auflistung können die Komponenten auch über eine Baumstruktur, welche anhand der Infrastrukturbeziehungen aufgebaut wird, angezeigt und ausgewählt werden. Dies ist in Abbildung 30: Darstellung des Komfortfeatures bei der Infrastrukturübersicht ersichtlich.

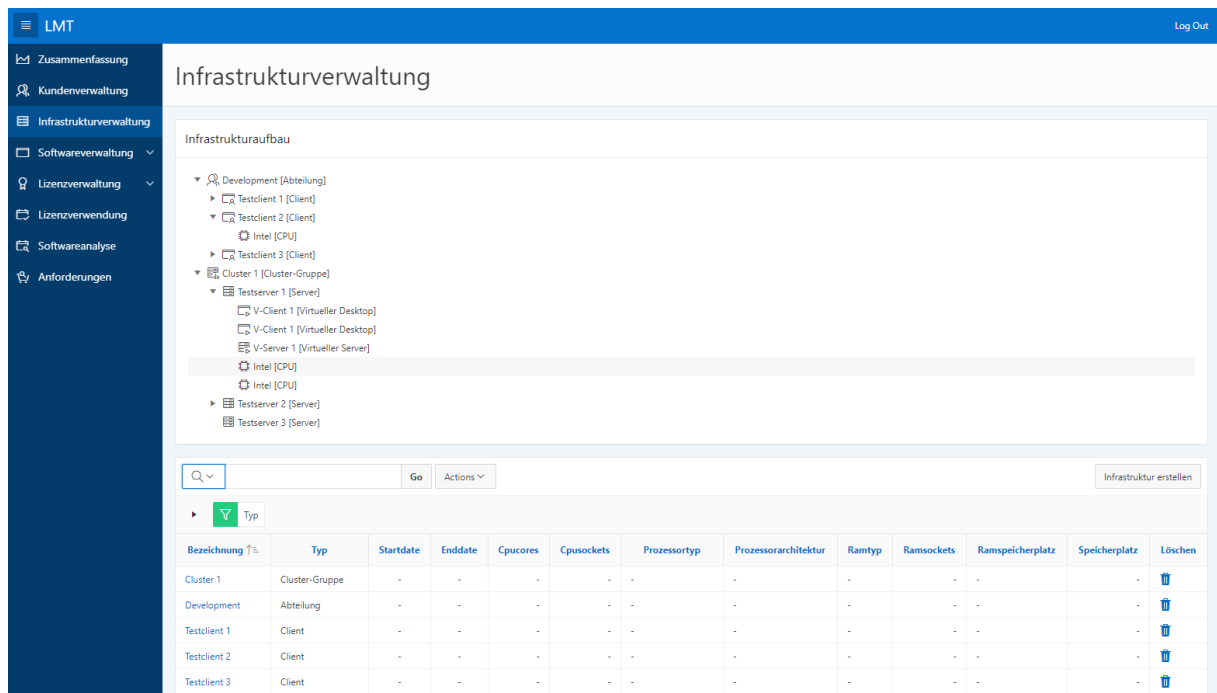


Abbildung 30: Darstellung des Komfortfeatures bei der Infrastrukturübersicht

Bei der Softwareverwaltung gibt es als Besonderheit einen Untermenüpunkt für die Erfassung und Bearbeitung von Programmen. Hier kann man die einzelnen Softwarepakete, welche dann bei der Softwareverwaltung mit Versionsnummern und weiteren Details ergänzt werden, erstellen.

Selbiges ist bei der Lizenzverwaltung der Fall. Hier können wiederum neue Features erfasst werden, welche anschließend den einzelnen Lizenzversionen hinzugefügt werden können. Das ausgeklappte Menü ist in Abbildung 31: Darstellung der Untermenüpunkte ersichtlich.

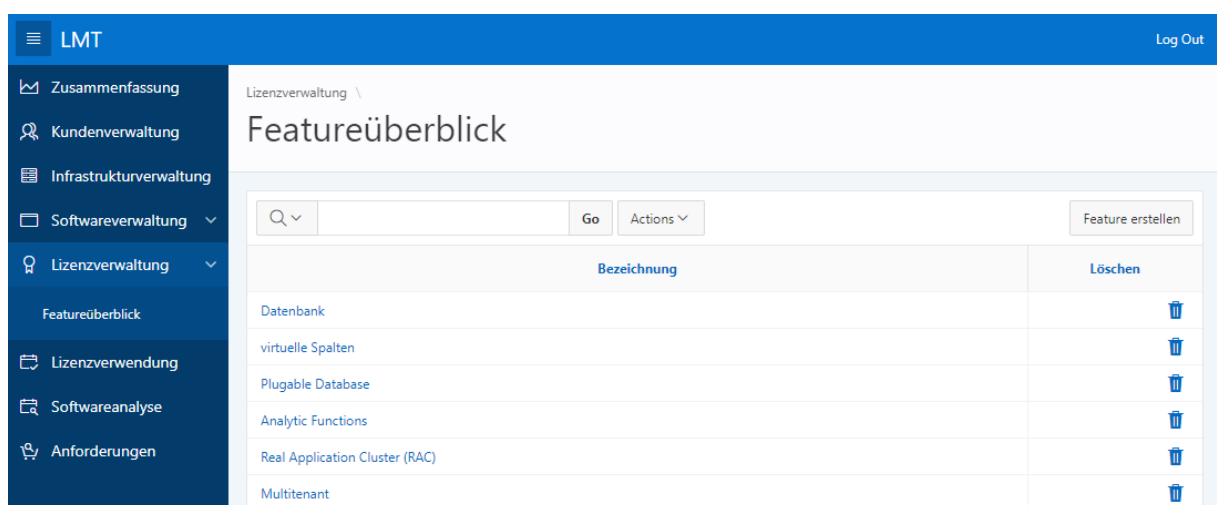


Abbildung 31: Darstellung der Untermenüpunkte

Den letzten Punkt der Implementierung des Prototyps stellt die Anforderungsanalyse dar. Sie ist der umfangreichste Teil, da hier die zuvor erfassten Daten miteinander verglichen und ausgewertet werden. Hierfür gibt es wieder die bereits bekannte Struktur zur Erfassung der Anforderungsdaten. Wurden diese erstellt, kann man über ein weiteres Feld in der Übersichtsseite das Ergebnis der Analyse aufrufen, wie in Abbildung 32: Darstellung der Anforderungsübersicht mit Ergebnisschaltfläche ersichtlich.

Bezeichnung	Startdate	Enddate	Useranzahl	Deviceanzahl	Löschen	Ergebnis
Test mit offenem Webzugriff	-	-	-	-		
Test	-	-	20	-		
Testanforderung Addons All	-	-	5	-		
Testanforderung Addon 235	-	-	5	-		
Testanforderung Addons 15	-	-	5	-		
Datenzugriff Web Testserver 2	-	-	-	-		
Datenzugriff Web Testserver 3	-	-	-	-		

Abbildung 32: Darstellung der Anforderungsübersicht mit Ergebnisschaltfläche

Klickt man nun auf das Listensymbol in der Ergebnisspalte, gelangt man zur Übersicht über die möglichen Lizenzen, die zur Auswahl stehen, wie in Abbildung 33: Darstellung der ermittelten Ergebnisse zu einer konkreten Anforderung ersichtlich. Sortiert sind diese erstmal nach der Anzahl der fehlenden Features (aufsteigend), der unerfüllten Anforderungen (aufsteigend), des Preises (aufsteigend), der abgedeckten Features (absteigend) und der erfüllten Anforderungen (absteigend). Die Sortierung kann nachträglich nach Belieben geändert werden.

LMT Log Out										
Zusammenfassung Kundenverwaltung Infrastrukturverwaltung Softwareverwaltung Lizenzverwaltung Lizenzverwendung Softwareanalyse Anforderungen										
Anforderung \ Ergebnisse										
Go Actions										
Lizenz	Addons	Stück	Preis	Abgedeckte Features	Fehlende Features	Zusätzliche Features	Anzahl Featureabdeckung	Anzahl fehlende Features	Anzahl zusätzliche Features	
Testlizenz [pro User] (10 EUR)	Testaddon 4 [pro User] (7 EUR)	5	85	Testfeature 1 Testfeature 2 Testfeature 3 Testfeature 4 Testfeature 5	-	-	5	0	0	
Testlizenz [pro User] (10 EUR)	Testaddon 1 [pro User] (5 EUR) Testaddon 2 [pro User] (3 EUR)	5	90	Testfeature 1 Testfeature 2 Testfeature 3 Testfeature 4 Testfeature 5	-	-	5	0	0	
Testlizenz [pro User] (10 EUR)	Testaddon 1 [pro User] (5 EUR) Testaddon 3 [pro User] (4 EUR)	5	95	Testfeature 1 Testfeature 2 Testfeature 3 Testfeature 4 Testfeature 5	-	-	5	0	0	
Testlizenz [pro User] (10 EUR)	Testaddon 2 [pro User] (3 EUR) Testaddon 4 [pro User] (7 EUR)	5	100	Testfeature 1 Testfeature 2 Testfeature 3 Testfeature 4 Testfeature 5	-	-	5	0	0	
Testlizenz [pro User] (10 EUR)	Testaddon 3 [pro User] (4 EUR) Testaddon 4 [pro User] (7 EUR)	5	105	Testfeature 1 Testfeature 2 Testfeature 3 Testfeature 4 Testfeature 5	-	-	5	0	0	

Abbildung 33: Darstellung der ermittelten Ergebnisse zu einer konkreten Anforderung

Die hier erwähnten Seiten sind Teil des Prototyps und stellen die grafische Oberfläche dar. Im nächsten Schritt wollen wir uns mit der funktionalen Codeumsetzung beschäftigen.

6.4 Codeumsetzung

Da nun die Darstellung der Seiten bekannt ist, soll nun ein Einblick in den dahinterliegenden Code gewährt werden. Im ersten Schritt soll ein kurzer Einblick über ein paar verwendete Hilfsskripts beschrieben werden, welche die Erstellung des Codes vereinfacht haben. Anschließend erfolgt eine Analyse über die Umsetzung der Ergebnisermittlung von Lizenzanforderungen, da dies den spezifischsten Teil der Umsetzung darstellt.

6.4.1 Hilfsskripte und Entwicklungsrichtlinien

Für die Erfassung der Daten wurde ein generisches Skript erstellt, um die Implementierung der einzelnen Datenerfassungen komfortabler umsetzen zu können. Das Skript benötigt zwei Parameter, einerseits den gewünschten Tabellennamen, andererseits die APEX-Page, in der die Erfassung stattfindet. Mithilfe dieser zwei Parameter erzeugt das Skript diversen Quellcode, welcher dann mittels Copy-Paste an die jeweiligen Stellen übertragen werden kann.

Das Skript holt sich über das Oracle Data Dictionary alle Spalteninformationen der im Parameter angegebenen Tabelle und erzeugt dafür drei Funktionen (in der Oracle-Welt unterscheidet man zwischen Funktionen, welche Werte zurückliefern und Prozeduren, welche keinen Return-Value aufweisen, hier wären es also eine Funktion und zwei Prozeduren), welche die Datenverwaltung ermöglichen.

Die erste Funktion ist ein Upsert für eine Zeile der entsprechenden Tabelle. Als Upsert bezeichnet man die Kombination aus Insert und Update, je nachdem, ob der Datensatz bereits vorhanden ist, oder nicht (ähnlich einem Merge-Statement). Ob ein Datensatz vorhanden ist, wird über die Primary-Key-Spalte ermittelt. Eigentlich hätte man dies über die jeweilige PK-Definition der Tabelle holen können, da bei der angewandten Codekonvention aber diese Spalte immer „ID“ heißt, wurde dieser Schritt vereinfacht. Das Resultat der Funktion ist also eine Prüfung, ob die mitgelieferte ID bereits vorhanden ist oder nicht. Ist sie vorhanden, wird ein Update auf die entsprechende Spalte ausgeführt. Ist die ID hingegen nicht vorhanden, wie das bei neuen Datensätzen der Fall wäre (denn hier würde die Applikation noch keine ID übergeben, da noch keine zugewiesen wurde), so wird ein Insert-Statement für diese Zeile ausgeführt. Als Return-Value erhält man anschließend die neu vergebene oder bereits vorhandene ID zurückgeliefert.

Die zweite Funktion (in der Oracle-Welt eigentlich eine Prozedur) stellt eine Wrapper-Funktion der Upsert-Funktion dar, welche keinen Return-Value liefert. In den meisten Fällen ist die vergebene ID nämlich egal, da man nach dem Erfassen des Datensatzes sowieso wieder zurück zur Übersichtsseite gelangt, bei der kein konkreter Datensatz erforderlich ist. Werden allerdings noch weitere Detaildatensätze verarbeitet ist die Retournierung der ID erforderlich.

Die dritte Funktion (wieder eine Prozedur) dient zum Löschen eines Datensatzes.

Die Codegenerierung der Datenverwaltung kann nun in ein gewünschtes Package in der Datenbank eingepflegt werden. Ab und an sind noch Modifikationen für entsprechende Detailanpassungen notwendig, allerdings erspart man sich einige Schreibarbeit durch die automatische Codegenerierung.

Des Weiteren erzeugt das Skript nun noch die jeweiligen Funktionsaufrufe aus dem APEX-Framework heraus, die entsprechenden Quellen für die Daten sind APEX-Items

und diese werden üblicherweise aus dem Zusammenschluss der Seitennummer und des jeweiligen Spaltennamens generiert. Das Skript erzeugt also entsprechend dieser Konvention die Aufrufe der einzelnen Funktionen und Prozeduren.

Es gibt noch zwei weitere Skripte, die die Erstellung der Tabellen und Handhabung der Datenerfassung erleichtern. Das erste der beiden Skripte erstellt vordefinierte Logspalten in allen Tabellen des Schemas (mit Ausnahme der Exception/Log-Tabelle). Die vordefinierten Spalten sind reine Logspalten und werden im aktiven Programmcode nicht verwendet, sie dienen nur dazu, zu protokollieren, wer zu welchem Zeitpunkt einen Datensatz erfasst beziehungsweise zuletzt bearbeitet hat. Die daraus resultierenden vier Spalten, die alle Tabellen (mit Ausnahme der Log-Tabelle) aufweisen, sind:

Tabelle 42: Logspalten

Spalte	Beschreibung
USERERF	Erfasser des Datensatzes
DATERF	Datum der Erfassung
USERAEND	Erfasser der letzten Änderung am Datensatz
DATAEND	Datum der letzten Änderung

Das zweite kleinere Skript dient zum Erstellen von grundlegenden Triggern für alle Tabellen. Diese Trigger haben einerseits die Aufgabe die Logspalten entsprechend zu befüllen und andererseits die ID mit einer eindeutigen Identifikation zu versehen, sofern keine im entsprechenden Insert-Statement mitangegeben wurde.

Die IDs werden nicht wie üblich aus einer Sequenz erzeugt, sondern mithilfe des sogenannten SYS_GUID() (Globally Unique Identifier), welcher es ermöglicht, einen weltweit einzigartigen Schlüssel mit einer Kombination aus Host-ID (Prozessor-ID, Thread-ID oder sonstiges) in Kombination mit einem nicht wiederholenden Wert zu erzeugen. Der Vorteil dieser ID ist, wenn zukünftig viele Instanzen der Software betrieben werden, kann beim Zusammenführen der Daten (beispielsweise in ein BI-System) problemlos ein Merge der Daten durchgeführt werden, da es zu keiner ID-Überschneidung kommen kann.

Mit diesen Skripts wurde die Datenbasis geschaffen, sowie eine Vereinfachung der Datenerfassung ermöglicht. Im nächsten Schritt soll nun die konkrete Umsetzung der Ergebnisermittlung von Lizenzanforderungen beschrieben werden.

6.4.2 Ergebnisermittlung der Lizenzanforderungen

Der wohl komplexeste Teil der Prototyprealisierung ist die Ermittlung der sinnvollen Ergebnisse für eine Lizenzanforderung. Als Ausgangssituation hat ein Kunde eine Software oder eine konkrete Softwareversion ausgewählt, sofern ihm möglich eine Anzahl der User definiert, die auf das System zugreifen können sollen, und eine entsprechende Infrastrukturkomponente hinzugefügt. Nicht immer kann eine Useranzahl angegeben werden, denn wenn beispielsweise die Daten in der Datenbank im Internet zum Aufbau einer Webseite verwendet werden, so ist keine genaue Useranzahl verfügbar, deshalb ist dieses Feld optional. Hat man aber ein abgeschlossenes System in einem Unternehmen, ist es zum Ermitteln der kostengünstigsten Lizenzvariante natürlich von Vorteil, eine entsprechende Useranzahl zu erfassen.

Die Prozedur zur Ermittlung der Ergebnismenge (Prozedur „calc_lka_result“ ersichtlich im Quellcode des Packages „lmt_anforderung.pkg“) speichert die Daten in eine Tabelle, welche ähnlich einem Result-Cache agiert. Die entsprechenden Lizenzen werden ermittelt und in die Tabelle LMT_LKA_RESULTS gespeichert. Dies geschieht beim ersten Aufruf der Ergebnismenge in der Applikation. Solange sich die darauf basierenden Grunddaten (sowohl lizenz-, als auch kundenseitig) nicht ändern, kann die Ergebnismenge danach immer direkt abgerufen und angezeigt werden. Das heißt lediglich, der erste Aufruf des Ergebnisses benötigt etwas Zeit zur Berechnung der passenden Lizenzen. Über entsprechende Trigger (TR_RES_<Tabellenname>) in der Datenbank werden bei einer Änderung der Daten in der entsprechenden Tabelle alle zugehörigen Ergebnisse in der Ergebnistabelle auf ungültig gesetzt. Beim erneuten Aufruf der Ergebnismenge wird kontrolliert, ob die Resultate noch valide sind. Sind sie es nicht mehr, wird eine erneute Ermittlung der möglichen Ergebnisse der Lizenzanforderung angestoßen.

Die Ermittlung erarbeitet nun die möglichen Kombinationen von Lizenzen, die herangezogen werden können, um den benötigten Softwareeinsatz zu decken. Dazu

werden im ersten Schritt alle bereits vorhandenen Resultate in der zuvor erwähnten Ergebnistabelle gelöscht, danach erfolgt die eigentliche Berechnung.

Je nachdem, ob nun bei der Anforderung eine Software oder eine Softwareversion angegeben wurde (wurden beide angegeben zählt die restriktivere, in diesem Fall also die Softwareversion), werden entsprechend alle Lizenzen (Vollversion und Upgrades) ermittelt, die die gewünschten Features der entsprechenden Software oder Softwareversion abdecken. Addons haben hier eine besondere Handhabung, da diese allein nicht verwendet werden können, allerdings spezielle Features für eine Software abdecken. Es wird also zusätzlich jede Vollversion ermittelt, die im Zusammenspiel mit einem oder mehreren Addons (definiert über die Lizenzbeziehungen) die gewünschten Features abdecken.

Im Konkreten würde das entsprechende Query also folgendermaßen aussehen, wenn bei der Anforderung die Softwareversion angegeben wurde, wobei der rot gekennzeichnete Wert einen Parameter („p_lkb_id“) darstellt, der die Ergebnismenge auf eine bestimmte Kundenbedingung einschränkt.

```
select a.lli_id, a.lve_id
  from lmt_lizenzabdeckungen a
 join lmt_lizenz_features f
    on a.id = f.lla_id
 join lmt_lizenzen l
    on a.lli_id = l.id
 where l.llk_id in (lmt_const.cf_lmt_llk_vollversion,
                  lmt_const.cf_lmt_llk_upgrade)
    and (a.lve_id, f.lfe_id) in
      (select b.lve_id, kf.lfe_id
        from lmt_kundenbedingungen b
        join lmt_kundenbedingung_features kf
          on b.id = kf.lkb_id
        where b.id = p_lkb_id)
union
select a.lli_id, a.lve_id
  from lmt_lizenzabdeckungen a
 join lmt_lllbeziehungen b
    on b.lli_id1 = a.lli_id
    and b.llbt_id = lmt_const.cf_lmt_llbt_addon
 join lmt_lizenzen la
    on b.lli_id2 = la.id
 join lmt_lizenzabdeckungen aa
    on la.id = aa.lli_id
 join lmt_lizenz_features f
    on aa.id = f.lla_id
 where la.llk_id = lmt_const.cf_lmt_llk_addon
    and (a.lve_id, f.lfe_id) in
      (select kb.lve_id, kf.lfe_id
        from lmt_kundenbedingungen kb
        join lmt_kundenbedingung_features kf
          on kb.id = kf.lkb_id
        where kb.id = p_lkb_id);
```

Wir ermitteln hier also zwei unterschiedliche Blöcke, die durch das Union in der Mitte des Queries kombiniert werden. Der obere Teil ermittelt die direkte Featureabdeckung einer Vollversion oder eines Upgrades, während der untere Teil die Ermittlung der Addons darstellt, von denen wiederum die entsprechenden Vollversionen oder Upgrades ermittelt werden. Aktuell wird davon ausgegangen, dass ein Addon direkt an einer Vollversion oder einem Upgrade hängt. Sollte es einmal den Fall geben, dass ein Addon über ein weiteres Addon erst einer Vollversion oder einem Upgrade zugewiesen wird, so muss das Query dahingehend erweitert werden, dass ein Hierarchical Query erstellt wird, von dem alle Vorgängerknoten, welche Vollversionen oder Upgrades sind, ermittelt werden. Da wir bei Oracle aber nur direkte Addon-Verknüpfungen zu Vollversionen oder Upgrades haben, wurde dieser Fall aktuell noch nicht implementiert.

Wäre in der Anforderung nicht die Softwareversion, sondern nur die Software angegeben worden, würde ein ähnliches Query die entsprechenden Lizenzen ermitteln, es wäre nur eine entsprechende Verknüpfung zu der Software enthalten. Die Ergebnismenge würde aber weiterhin alle Lizenzen und Softwareversionen liefern, die die gewünschten Features abdecken, es würden dann im Ergebnis nur mehrere Versionen und deren Lizenzen separat ausgewiesen werden.

Durch das Union in der Abfrage wird das Ergebnis auch gleich distincted, also es wird darauf geachtet, dass jede Lizenz-Softwareversion-Kombination nur einmalig im Ergebnis auftaucht.

Da nun alle betroffenen Lizenzen ermittelt sind, werden diese anschließend im Einzelnen analysiert. Im ersten Schritt wird ermittelt, ob die Vollversion oder das Upgrade die einzelnen angeforderten Features abdeckt. Ist dies der Fall wird das nächste Feature analysiert, trifft dies nicht zu, kommen wir zu einem komplexen Teil der Anforderungsanalyse.

Ursprünglich war geplant, dass Addons nur modular Features abdecken und es zu keiner Überschneidung kommt, was bei Oracle aktuell der Fall ist. Allerdings gab es früher Featurepacks, welche mehrere Features kombinierten und zusätzlich zu den modularen Optionen (also Addons) angeboten wurden. Grundsätzlich kann es also zu Überschneidungen bei Feature-Abdeckungen von Addons kommen.

Tabelle 43: Addon-Feature-Abdeckung soll ein Beispiel für so eine Überschneidung vorführen, um das Problem besser verständlich zu machen. Die Tabelle zeigt an, welches Feature von welchem Addon abgedeckt ist. Wie wir sehen, ist Addon 4 ein Paket aus Addon 1 und Addon 3. Wenn also ein Kunde unbedingt Feature 1, 3, 4 und 5 benötigt, hat dieser die Wahl, entweder Addon 1 und Addon 3 zu erwerben oder Addon 4, natürlich könnte er auch alle drei Addons (1, 3 und 4) anschaffen, es wäre nur nicht so kosteneffizient, soll aber auch nicht ausgeschlossen werden.

Tabelle 43: Addon-Feature-Abdeckung

	Addon 1	Addon 2	Addon 3	Addon 4
Feature 1	X			X
Feature 2		X		
Feature 3			X	X
Feature 4			X	X
Feature 5	X			X

Nun war die Überlegung, wie diese Problemstellung am sinnvollsten umzusetzen ist. Eine Idee war, das kartesische Produkt aller vier Addons zu bilden um alle möglichen Kombinationen zu ermitteln und abzudecken. Anhand der danach tatsächlich abgedeckten Features und der Preiskalkulation wäre gewährleistet, dass die sinnvollste Kombination auf jeden Fall mitaufscheint und der bevorzugte beziehungsweise empfohlene Vorschlag zur Anschaffung wäre (also auf Platz 1 der Default-Sortierreihenfolge der Ergebnismenge). Das Problem daran ist allerdings, dass der Berechnungsaufwand exponentiell steigen würden, je mehr Addons in der kompletten Anforderung auftauchen. Da es sehr unwahrscheinlich ist, dass viele Addons viele idente Features aufweisen, würde der Ansatz in den meisten Fällen über das Ziel hinausschießen.

Deshalb wurde eine weitere Variante überlegt, welche nun umgesetzt wurde. Es werden nur die Kombinationen ermittelt, die anhand der vom Kunden gewünschten Features möglich sind (in Kombination mit der jeweilig benötigten Vollversion oder des benötigten Upgrades). Es werden also Feature-Container gebildet, welche die Summe aller gewünschten Features beinhalten. Würde in unserem Beispiel sich ein Kunde also Feature 1 und 2 wünschen, so würden zwei mögliche Ergebniscontainer ermittelt, nämlich einer mit Addon 1 und 2 und ein weiterer Container mit Addon 4 und Addon 2.

Diese Container werden anschließend noch bereinigt, da es ja vorkommen kann, dass Addons mehrmals in einem Container vorkommen, da sie mehrere Features abdecken. Auch übergreifend wird dann noch einmal kontrolliert, dass nicht zwei idente Container mit denselben Addons existieren.

Dieser Vorgang wird mit jeder weiteren möglichen Vollversion und jedem weiteren Upgrade wiederholt, um zum Schluss alle möglichen Kombinationen aller Lizenzen zu erhalten, die die benötigten Features abdecken, wenn auch nicht zwingend zur Gänze. Es kann natürlich vorkommen, dass einige oder in seltenen Fällen sogar alle Lizenzkombination nicht alle Features abdecken, dies wird im Ergebnis aber mitangegeben, wenn ein gewünschtes Feature nicht vorhanden ist.

Ist nun also die Grundmenge der verschiedenen Lizenzkombinationen ermittelt, wird im nächsten Schritt kontrolliert, ob die Lizenzen die Infrastrukturanforderungen erfüllen und ob bestimmte Mindestmengen nicht unterschritten werden. Sollten sie unterschritten werden, wird die Stückzahl auf die Mindestmenge angehoben.

Der bei den Lizenzen definierte Lizenztyp entscheidet, wie das Handling der einzelnen Bedingungen der Lizenz abläuft. Handelt es sich um eine „pro User“ Lizenz, so wird die Useranzahl ausschlaggebend, bei einer „pro Device“ Lizenz werden die entsprechenden Server oder Clientrechner der zugefügten Infrastruktur ermittelt, ist keine angegeben, kann die allgemeine Geräteanzahl aus der Anforderungsangabe herangezogen werden. Ist der Lizenztyp hingegen „per CPU“, so ist entscheidend, welcher Wert in der Lizenzbedingung gesetzt wurde. Hier gibt es die Unterscheidung von tatsächlich eingesetzten CPUs („CPU-Sockets in Verwendung“), aktiven CPU-Cores („CPU-Cores in Verwendung“) oder auch CPU-Threads („CPU-Threads in Verwendung“). Alternativ kann es auch Lizenzen geben, die nicht die tatsächlich eingesetzten Komponenten voraussetzen, sondern zum Beispiel, wie viele Prozessoren theoretisch eingesetzt werden könnten (anhand der Socketanzahl auf dem Mainboard). Dafür dienen die Angaben der jeweiligen CPU-Werte ohne den Beisatz „in Verwendung“. Als Beispiel sei hier eine Restriktion bei Oracle erwähnt, die besagt, dass die „Standard Edition 2“ nur auf Mainboards mit maximal 2 CPU-Sockets verwendet werden darf, ganz unabhängig davon, wie viele tatsächlich in Verwendung sind.

Außerdem gibt es öfters die Restriktionen von Mindeststückzahlen von Lizenzen, so ist beispielsweise eine Bedingung von Oracle, dass bei NUP-Lizenzen (Named User Plus,

was dem Lizenztyp „per User“ entspricht) eine Mindestmenge bestellt werden muss, welche ebenfalls in den Lizenzbedingungen erfasst werden kann. Bei den Restriktionen der „Enterprise Edition“ wird es dann noch komplexer, da die Mindeststückzahl ebenfalls abhängig von der Anzahl der verwendeten Prozessoren ist. Es gibt also eine Mindestmenge von 25 NUPs pro eingesetztem Prozessor. Um dies berechnen zu können ist es wichtig, dass der korrekte Lizenztyp angegeben wurde. Dadurch weiß das System, welcher Grundwert die Abdeckung der Lizenz angibt. Alle anderen Grundwerte gelten als Multiplikator der sonstigen angegebenen Mindestmengen.

Wenn wir uns das Beispiel der „Enterprise Edition“ genauer ansehen, so sehen wir bei der „pro User“ Lizenz folgende Einträge:

Tabelle 44: Lizenzanforderung für Oracle Database Enterprise Edition

Spalte	Wert
Minimale Usermenge	25
Useranzahl	1
Sockets in Verwendung	1

Da der Lizenztyp „pro User“ ist, ist dem System klar, dass die Useranzahl die abgedeckte Menge der Lizenz darstellt, in unserem Fall also ein User (dies ist ausschlaggebend für die Preisberechnung). Die minimale Usermenge ist 25, allerdings gilt diese für einen „Socket in Verwendung“. Hat das Mainboard nun beispielsweise zwei Prozessoren zugewiesen, so steigt die minimale Usermenge auf 50. Es ergibt sich in diesem konkreten Fall (mit zwei verwendeten Prozessoren) also folgende Formel:

$$User_{min} = Lizenz[User_{min}] * \frac{Infrastruktur[CPU_{Sockets}]}{Lizenz[CPU_{Sockets}]} = 25 * \frac{2}{1} = 50$$

Die entsprechende Formel zieht sich natürlich auch mit den anderen Werten noch durch, also Gerätemenge, CPU-Threads und CPU-Cores, sowohl in Verwendung als auch schlicht vorhanden, sowie auch für die Maximalwerte der einzelnen Lizenzbestimmungen, und ist im Quellcode im Package „lmt_lizenz“ in der Funktion „get_lizenz_anforderung“ einsehbar.

Durch die Berechnung der entsprechenden Grenzwerte und Stückzahlen, kann nun der Preis der entsprechenden Lizenzen berechnet werden. Des Weiteren wird ermittelt, welche Features, die nachgefragt wurden, von den einzelnen Containern auch

tatsächlich abgedeckt werden, sowie welche Features fehlen und auch welche zusätzlich enthalten sind, da sie durch die Lizenz und eventuell deren Addons mitangeboten werden.

Wenn dies ermittelt ist, gilt es noch festzustellen, ob auch tatsächlich die Rahmenbedingungen der Lizenz eingehalten werden. Hierzu zählen vor allem die Minimal- und Maximalwerte der numerischen Attribute der Lizenzbedingungen (also beispielsweise die Useranzahl, Geräteanzahl oder CPU-Anzahl). Diese werden für die Gesamtstruktur ermittelt. Wird also ein Cluster oder eine Abteilung als Infrastruktur der Anforderung hinzugefügt, so werden die Komponenten, die über die entsprechende Beziehung diesen zugewiesen sind, aufsummiert. Handelt es sich um virtuelle Umgebungen, so wird die reale Hardwarekomponente ermittelt, auf der die virtuelle Umgebung aufgesetzt wurde, und dessen Bestückung herangezogen. Die entsprechenden Lizenzanforderungen werden in drei Kategorien unterteilt:

- **Erfüllte Anforderung** – Diese Anforderungen werden von den betroffenen Infrastruktur-Komponenten eingehalten.
- **Vorhandene Anforderung** – Diese Anforderungen können nicht geprüft werden, da eine entsprechende Angabe fehlt. Sie werden zu informativen Zwecken angegeben, damit der Kunde diese berücksichtigen kann.
- **Unerfüllte Anforderung** – Diese Anforderungen werden von der aktuell zugewiesenen Infrastruktur nicht erfüllt. Beispielsweise kann eine Maximalmenge von CPU-Sockeln überschritten werden, die für eine vergünstigte Lizenz zugelassen sind. Existiert mindestens ein nicht erfülltes Kriterium, so ist die Lizenz nicht anwendbar, sofern man auf die bei der Anforderung zugewiesene Infrastruktur besteht.

Im Anschluss wird noch ermittelt, ob einzelne Teilkomponenten die Anforderung nicht erfüllen, um dies ebenfalls festzuhalten. Dies gilt dann, wenn als Infrastrukturtyp eine Abteilung oder ein Cluster gewählt wurden. Hier muss anschließend überprüft werden, ob die Lizenz auch für die einzelnen Komponenten eine Bedingung stellt und diese von jedem einzelnen untergeordneten Gerät erfüllt sind. Ist dies nicht der Fall, wird ein entsprechender Eintrag mit Verweis auf die entsprechende Teilkomponente in die unerfüllten Anforderungen mitaufgenommen.

Danach ist die Ergebnisermittlung abgeschlossen, die entsprechenden Daten werden für die Anzeige aufbereitet. In der GUI ist dann noch eine sinnvolle Standarddarstellung konfiguriert, welche die Lizenzen sinnvoll sortiert und markiert.

Die Sortierung erfolgt nach folgender Logik in der angeführten Reihenfolge:

- **Anzahl fehlende Features (aufsteigend)** – denn je weniger Features fehlen, die der Kunde anfordert, desto sinnvoller ist die Lizenz geeignet (ideal ist es, wenn kein Feature fehlt)
- **Anzahl unerfüllte Anforderungen (aufsteigend)** – denn je weniger Lizenzanforderungen unerfüllt sind, desto leichter kann die Infrastruktur an die Bedingungen angepasst werden (ideal ist es, wenn keine Anforderung unerfüllt bleibt, denn dann ist keine Änderung der Infrastruktur notwendig)
- **Preis (aufsteigend)** – je günstiger die Gesamtlizenzkosten sind, desto effizienter ist der Lizenzeinsatz
- **Anzahl Featureabdeckung (absteigend)** – je mehr Features der Kundenanforderung abgedeckt werden, desto besser ist die Lizenz für den Softwareeinsatz geeignet (ideal ist es, wenn alle benötigten Features abgedeckt werden)
- **Anzahl erfüllte Anforderungen (absteigend)** – es sind Lizenzen zu bevorzugen, mit denen mehr Lizenzanforderungen erfüllt werden (ideal ist es, wenn alle Anforderungen erfüllt sind)

Zusätzlich werden Ergebnis-Einträge, deren genaue Ermittlung nicht möglich war, da entsprechende Anforderungen nicht definiert wurden, gelb eingefärbt. Dies dient als Hinweis für den User, dass hier noch Attribute bei der Kundenanforderung hinzugefügt werden sollten, um sicher zu gehen, dass die Resultate die gewünschten Rahmenbedingungen abdecken (beispielsweise bei einer nicht eingetragenen/angegebenen Useranzahl).

Ebenfalls werden Ergebnisse, die unerfüllte Lizenzanforderungen besitzen oder nicht alle angeforderten Features aufweisen, rot eingefärbt, um zu kennzeichnen, dass der entsprechende Lizenzkorb nicht alle Kriterien der aktuellen Anforderungen erfüllt. Eventuell ist es aber einem Kunden aus Kostengründen lieber, die Anforderungen anzupassen, als zur teureren Lizenz zu greifen, welche alle Anforderungen erfüllt.

Ein Beispiel der Darstellung kann in Abbildung 34: Beispiel einer Ergebnisanzeige eingesehen werden.

Name	Adresse	Stück	Preis	Zugeordnete Features	Fehlende Features	Zusätzliche Features	Erfüllte Anforderungen	Anforderungen	Unerfüllte Anforderung
Oracle Database Personal Edition (per User) (2019 EUR)		1	399	• SQL Datenbank		• Active Data Guard • Advanced Analytics • Advanced Compression • Advanced Security • Analytics Functions • Analytics SQL Model • Cloud Management Pack for Oracle Database • Communications Data Model • Data Masking and Subsetting Pack • Database In-Memory • Database Lifecycle Management Pack • Database Vault • Diagnostic Pack • Label Security • Multitenant • OLAP • Partitioning • Plugable Database • Real Application Cluster (RAC) • Real Application Testing • Retail Data Model • Spatial and Graph • TimesTen Application Tier Database Cache • Tuning Pack • Unlimited Data Model • virtuelle Spalten		• Minimale Useranzahl: 1	
Oracle Database Enterprise Edition (per User) (825 EUR)		20	2025	• SQL Datenbank		• Analytics Functions • Plugable Database • virtuelle Spalten		• Minimale Useranzahl: 25	
Oracle Database Enterprise Edition (per CPU) (41340 EUR)		1	41340	• SQL Datenbank		• Analytics Functions • Data Guard/HA aus offiziellem Netz (unbestimmbar) • RUPP • Plugable Database • virtuelle Spalten			
Oracle Database Standard Edition 2 (per User) (204 EUR)		10	2040	• SQL Datenbank		• Analytics Functions • Plugable Database • virtuelle Spalten	• Garantenanzahl: 1 • CPU-Socketanzahl in Verwendung: 1	• Minimale Useranzahl: 10 • Minimale CPU-Threads in Verwendung:	• Minimale CPU-Socketanzahl: 2 (vorhanden: 4)
Oracle Database Standard Edition 2 (RAC) (per User) (204 EUR)		20	4080	• SQL Datenbank		• Analytics Functions • Plugable Database • Real Application Cluster (RAC)	• CPU-Socketanzahl in Verwendung: 2	• Minimale Useranzahl: 20 • Minimale Garantenanzahl: 2 • Minimale CPU-Threads in	• Minimale CPU-Socketanzahl: 2 (vorhanden: 4)

Abbildung 34: Beispiel einer Ergebnisanzeige

Wie bei allen anderen tabellarischen Darstellungen gilt, dass die Sortierung und Darstellung vom User nachträglich bearbeitet werden kann. Die angegebene Standardansicht dient nur als empfohlene Darstellungsform, es ist aber möglich, die Sortierung und Markierung (sowie auch weitere Eigenschaften, wie beispielsweise die Gruppierung) jederzeit anzupassen und auf Userebene abzuspeichern.

Über den Button „Ergebnis neu ermitteln“ rechts oberhalb der Ergebnisauflistung kann eine manuelle Neuberechnung der Kundenanforderungsergebnisse angestoßen werden. Dieser Button wird im Normalfall nicht benötigt, da bei einer Änderung der Grunddaten automatisch eine Neuberechnung erfolgt. Er dient aber dazu, sicherzugehen, dass das Resultat auch tatsächlich das aktuelle Ergebnis widerspiegelt.

6.5 Überarbeitungen

Wir haben nun die Umsetzung der Ergebnisermittlung der Lizenzanforderungen näher betrachtet. Diese war nicht von Anfang an so konkret geplant, durch die Implementierung des Prototypen haben sich einige Optimierungen am Datenmodell ergeben, außerdem wurden Beziehungen erweitert oder neu strukturiert, um die Anforderungen sinnvoller abdecken zu können. Im folgenden Abschnitt wird ein wenig Einblick gewährt, welche Stolpersteine die Entwicklung in der zuvor festgelegten

Planung hatte und wie diese gelöst und damit auch die entsprechenden Modelle angepasst wurden.

Der ursprüngliche Plan der Einteilung von Software und deren Funktionsumfang war es, dass Softwarepakete (also der Name der Software) und deren Funktionsumfang (nun Features genannt) über die Softwareversion verknüpft werden. Softwarepakete hatten also bestimmte Versionsnummern und haben bestimmte Features abgedeckt. Betrachtet man beispielsweise MS Office, da dies wohl eine der bekanntesten Softwareprodukte ist, so wären für Office mit der Versionsnummer 2016 einerseits die Features Word, Excel, Powerpoint und Onenote in der „Home & Student“ Edition gegeben, während die „Professional“ Edition noch weitere Features wie Outlook, Publisher und Access beinhalten würde. Man merkt also, dass die Version nicht ausreichte, um die einzelnen Produktpakete zu schnüren, also war als Folgeidee aufgekommen, auch Editionen in die Softwareversionstabelle aufzunehmen. Bei näherer Betrachtung wurde aber erkannt, dass die Editionen ja eigentlich die Lizenzen darstellen, welche erworben werden können. Eine Abbildung der Editionen in den Softwareversionen machte also keinen Sinn, da man dadurch nur redundante Pakete schnürte, die in der Lizenzabbildung bereits dargestellt werden. Es lief also darauf hinaus, dass die Feature-Zuweisung keine direkte Softwarerelevanz mehr hatte, sondern deren Zuweisung in die Lizenzabdeckung fiel. Da die Feature aber Teile der eigentlichen Software abbilden, blieben sie schematisch weiterhin in der Softwarekategorie des Datenmodells.

Des Weiteren wurden die Features ursprünglich alle einzeln sowohl zur Kundenanforderung als auch zur Lizenzabdeckung hinzugefügt, was darin resultierte, dass diese zwei Zuweisungen zusätzlich gruppiert werden mussten. Um diesen Überschuss an Gruppierungen in der Zuweisungserfassung zu minimieren, wurden die zwei Zwischentabellen „LMT_KUNDENBEDINGUNG_FEATURES“ und „LMT_LIZENZ_FEATURES“ eingeführt, welche jeweils als m:n-Verknüpfung der Kundenbedingung oder der Lizenzabdeckung zu den entsprechenden Features (Tabelle „LMT_FEATURES“) fungieren. Dadurch wurde es auch leichter möglich, bestimmte Features direkt einer gewünschten Software und eventuell deren Version zuzuweisen, was die Abfrage nach relevanten Lizenzen für Anforderungen vereinfachte.

Ein weiterer Punkt waren die Darstellungen der Lizenzbedingungen. Immer noch vorhanden ist die Möglichkeit, diese Bedingungen dynamisch zu erstellen (über den

Weg der Tabelle „LMT_WERTE“), die soll im Endmodell auch tatsächlich zum Einsatz kommen, es wurden aber zusätzlich fixe vordefinierte Angaben für Lizenzanforderungen definiert, welche tatsächlich in eigene Spalten der Tabelle „LMT_LIZENZEN“ eingetragen werden. Diese Entscheidung fiel aus der Überlegung heraus, da eine reine dynamische Darstellung es erschwerte, performante Abfragen zu generieren, weil der Pool an Bedingungen für jede Abfrage neu gebildet werden muss. Im aktuellen Prototyp ist eine solche konkrete Abfrage nicht implementiert, da die Werte aktuell sowieso einzeln analysiert werden müssen. Es steht also die Überlegung im Raum, diese Werte zukünftig doch wieder in die dynamischen Wertemengen mit aufzunehmen. Allerdings soll dies erst nach einer Probelaufphase entschieden werden, da es sehr wohl zu Performanceeinbußen beim Vorhandensein vieler Lizenzdefinitionen kommen kann. Es wird sich erst zeigen müssen, in welcher Relation der Speicherplatzverlust zur daraus resultierenden Performance steht, wenn mehr Produktivlizenzen eingespielt wurden.

Bezüglich Performance gab es noch einen wichtigen Punkt, nämlich die Berechnung der Ergebnisse von bestimmten Kundenanforderungen. Ursprünglich war es angedacht, die Ergebnisse immer auf Abfrage neu zu berechnen. Die entsprechende Prozedur, die die Ergebnisse ermittelt und in einen nachgebauten Result-Cache speichert, war ursprünglich eine Funktion, welche SQL-Resultate aufbereitete. Die Funktion lieferte direkt einen Datentyp, der mittels der TABLE()-Funktion in Oracle SQL in eine Tabelle konvertiert werden konnte. Um den Prozess zu optimieren, wurde die Funktion direkt als „Pipelined Function“ erstellt. Das bedeutet, dass nicht erst ein Ergebnisarray erstellt werden musste, welches danach an die SQL-Prozedur übergeben wurde, welche dann wiederum dieses Array in einen SQL-Result konvertierte, sondern es wurden direkt mittels der Oracle-Prozedur PIPE_ROW() die Resultate in den SQL-Result herausgespielt. Der Vorteil davon war nun, dass man sich Arbeitsspeicherplatz ersparte, sowie die Funktion abgebrochen werden konnte, wenn beispielsweise nur die ersten fünf Datensätze verlangt wurden. Hätte man nicht auf die Pipelined-Function zurückgegriffen, hätte immer erst die komplette Ergebnismenge ermittelt werden müssen und danach erst hätte man die entsprechende Anzahl an gewünschten Datensätzen auslesen können. Durch den Aufwand der im vorherigen Kapitel erwähnten überschneidenden Feature-Abdeckungen bei Addons, aber auch der allgemeinen Ermittlung der Lizenzabdeckung und der Kostenberechnung, welche die Berechnung der Ergebnismenge verlangsamten, kam es zu erhöhten Durchlaufzeiten bei der

Ergebnisanalyse. Durch die länger andauernde Berechnung ergab sich nicht nur eine erhöhte Wartezeit beim Aufruf der Ergebnisse, sondern auch beim Filtern oder neu Sortieren dieser. Zur Behebung dieses Umstandes wurde eine zusätzliche Tabelle für die Speicherung der Ergebnisse eingeführt, welche einen Result-Cache darstellen soll. Die Ergebnisse sind einerseits verknüpft mit der entsprechenden Kundenanforderung, andererseits über eine m:n-Tabelle mit allen in Verbindung stehenden Lizenzen. Ändert sich nun eine Lizenz (und deren Subtabellen) oder eine Kundenanforderung (und deren Subtabellen), so werden die Resultateinträge auf invalide gestellt und zur Neuberechnung freigegeben. Solange die Anforderungen und Lizenzabdeckungen gleichbleiben (was den Regelfall entspricht), wird das Ergebnis nicht neu berechnet, sondern direkt aus der Result-Tabelle ermittelt.

Der letzte Punkt, welcher überarbeitet wurde, befindet sich im Bereich der Lizenzabdeckung. Es kann vorkommen, dass Lizenzen auf unterschiedliche Weise eingesetzt werden können, es handelt sich hierbei aber immer noch um die gleichen Lizenzen. Als Beispiel kann man hier die Lizenzabdeckung der „Oracle Database Standard Edition 2“ betrachten. Diese Lizenz kann einerseits als reine Serverlizenz verwendet werden, andererseits aber auch als abgespeckte RAC-Lösung (Real Application Cluster). Es handelt sich hierbei um dieselbe Lizenz mit unterschiedlichen Einsatzmöglichkeiten. Dementsprechend sind aber auch die Anforderungen der Lizenz unterschiedlich. Während die Serverlösung zwei Prozessoren zulässt, darf bei der RAC-Lösung auf jedem Mainboard beispielsweise nur ein Prozessor verwendet werden. Ursprünglich war geplant, gleiche Lizenzen mit unterschiedlichen Lizenzabdeckungen hierfür zu definieren. Dadurch würden aber die Strukturen deutlich unübersichtlicher und komplexer werden, weshalb entschieden wurde, für beide Lizenzeinsätze separate Lizenzen zu erstellen. Im Namen der Lizenz wird dann der entsprechende Einsatz verdeutlicht. Um nun abbilden zu können, dass es sich um dieselbe Lizenz handelt, wurde der neue Beziehungstyp „Konvertierbar“ eingeführt. Dadurch wird ersichtlich, dass die RAC-Lizenz jederzeit in eine Serverlizenz umgewandelt und unter den entsprechend anderen Bedingungen eingesetzt werden kann.

All diese Änderungen wurden in die Modelle integriert, der Prototyp selbst beinhaltet nicht zwingend alle diese Umsetzungen, ist aber darauf aufgebaut, diese auf einfache Weise bei der Weiterentwicklung zu integrieren.

Nach diesem Konzept wurde nun also der Prototyp entwickelt und anschließend getestet. Wie diese Testfälle aussahen und warum der entsprechende Output zustande kam, wird im nächsten Kapitel beschrieben.

7. Funktionaler Test und Diskussion der Ergebnisse

Nachdem nun der Prototyp umgesetzt wurde, soll in weiterer Folge ein funktionaler Test durchgeführt werden, welcher ermittelt, ob die Umsetzung korrekt reagiert und sinnvoll einzusetzen ist. Dafür wurden diverse Infrastrukturen und Lizenzen vorbereitet, welche anschließend in diversen Anforderungen zur Ermittlung von geeigneten Lizenzen herangezogen wurden.

7.1 Lizenzfassung

Wie bereits in der Aufgabenstellung beschrieben, wurden vor allem alle Oracle Datenbank Lizenzen für die Version 12c mit den tatsächlichen Bedingungen und Preisen erfasst, welche aus dem online verfügbaren Oracle Store ermittelt wurden.¹⁵ Darunter fallen folgende Lizenzen und Optionen in diversen Ausprägungen, unter anderem die unterschiedlichen Lizenzanforderungen, welche sich durch die verschiedenen Lizenztypen ergeben, wie auch die diversen verschiedenen Einsatzmöglichkeiten, zum Beispiel die RAC-Option (Real Application Cluster) in der Standard Edition 2, die später noch näher erläutert wird:

- Oracle Database Personal Edition
- Oracle Database Standard Edition 2
- Oracle Database Enterprise Edition
- Oracle Database Enterprise Edition Options - Active Data Guard
- Oracle Database Enterprise Edition Options - Advanced Analytics
- Oracle Database Enterprise Edition Options - Advanced Compression
- Oracle Database Enterprise Edition Options - Advanced Security
- Oracle Database Enterprise Edition Options - Airlines Data Model
- Oracle Database Enterprise Edition Options - Communications Data Model
- Oracle Database Enterprise Edition Options - Database In-Memory
- Oracle Database Enterprise Edition Options - Database Vault
- Oracle Database Enterprise Edition Options - Label Security
- Oracle Database Enterprise Edition Options - Multitenant
- Oracle Database Enterprise Edition Options - OLAP

¹⁵ (Oracle, Oracle Store, 2018)

- Oracle Database Enterprise Edition Options - Partitioning
- Oracle Database Enterprise Edition Options - Real Application Cluster (RAC)
- Oracle Database Enterprise Edition Options - Real Application Testing
- Oracle Database Enterprise Edition Options - Retail Data Model
- Oracle Database Enterprise Edition Options - Spatial and Graph
- Oracle Database Enterprise Edition Options - TimesTen Application-Tier Database Cache
- Oracle Database Enterprise Edition Options - Utilities Data Model
- Oracle Database Enterprise Management - Cloud Management Pack for Oracle Database
- Oracle Database Enterprise Management - Data Masking and Subsetting Pack
- Oracle Database Enterprise Management - Database Lifecycle Management Pack
- Oracle Database Enterprise Management - Diagnostics Pack
- Oracle Database Enterprise Management - Tuning Pack
- Oracle NoSQL Database Enterprise Edition

Um ersichtlich zu machen, worauf bei der Lizenzerstellung zu achten ist, wollen wir zunächst ein paar Beispiele von Anforderungen an bestimmte Lizenzen näher betrachten.

Als gutes Beispiel für eine Lizenz mit komplexeren Anforderungen an die Infrastruktur stellt die Oracle Database Standard Edition 2 dar. Sie ist eine kostengünstigere Variante für nicht überproportional leistungsfähige Server und somit nur mit einigen Einschränkungen zu verwenden. Will man die Datenbank auf einer leistungsfähigeren Infrastruktur betreiben, muss man zur teureren Enterprise Edition greifen.

Die Lizenz ist sowohl als Lizenztyp „pro User“ als auch „pro CPU“ verfügbar und hat folgende Einschränkungen:

- Die maximale Socketanzahl auf dem verwendeten Mainboard darf nicht mehr als zwei sein, unabhängig davon, ob sie bestückt sind.
- Es dürfen maximal 16 CPU-Threads verwendet werden.
- Die minimale Menge an NUPs (Named User Plus – in diesem Fall also die minimale Useranzahl) ist 10 User pro Server.
- Als RAC-Lösung darf nur jeweils ein Socket pro Server verwendet werden.

- Als RAC-Lösung dürfen maximal 8 CPU-Threads pro Server verwendet werden.

Um dies darzustellen, musste die Lizenz erst einmal unterteilt werden, da sich zusätzlich zu den unterschiedlichen Lizenztypen auch unterschiedliche Einsatzzwecke ergeben, nämlich als Server oder RAC, wurden in Summe vier unterschiedliche Lizenzen mit unterschiedlichen Anforderungen erstellt:

- Oracle Database Standard Edition 2 – per User
- Oracle Database Standard Edition 2 – per CPU
- Oracle Database Standard Edition 2 (RAC) – per User
- Oracle Database Standard Edition 2 (RAC) – per CPU

Sehen wir uns hiervon nun die komplexere Variante „Oracle Database Standard Edition 2 (RAC) – per User“ an. Es ergeben sich folgende Anforderungen für diese Lizenz:

Tabelle 45: Beispiel der Lizenzanforderung für Oracle Standard Edition 2 (RAC) - per User

Anforderung	Wert
Lizenztyp	per User
Preis	304
Minimale Usermenge	10
Useranzahl	1
Maximale CPU-Threads in Verwendung	8
Maximale Socket-Menge	2
Maximale Socket-Menge in Verwendung	1
Minimale Geräteanzahl	2
Geräteanzahl	1
Maximale Geräteanzahl	2

Durch den Lizenztyp „per User“ wird bestimmt, dass das Attribut Useranzahl die abgedeckte Menge zum angegebenen Preis darstellt (in diesem Fall kostet die Lizenz pro User 304€). Durch die minimale Usermenge ist gewährleistet, dass auch bei Userzahlen kleiner als 10, die benötigte Stückzahl und damit der Preis für mindestens 10 NUPs veranschlagt wird. Die Geräteanzahl mit dem Wert 1 gibt an, dass die entsprechenden Minimal- und Maximalwerte pro Gerät gelten. Wären die Angaben über alle betroffenen Geräte hinweg gültig, würde das Feld Geräteanzahl leer bleiben und damit keinen Einfluss auf die restlichen Daten haben. Dadurch, dass man nun also bei einem Cluster eine minimale Geräteanzahl von 2 benötigt, wird auch die entsprechende minimale

Usermenge verdoppelt. Als Mindestabnahmemenge sind nun also 20 NUPs gefordert. Ebenso verhält es sich mit den CPU-Sockel und -Thread-Einschränkungen. Beide werden entsprechend auf die summierte Komponentenanzahl aller beteiligten Server (in diesem Fall zwei) geprüft. Zusätzlich wird hier aber auch noch geprüft, ob die einzelnen Komponenten des Clusters nicht die Maximal-Angaben überschreiten, denn schließlich würde es sonst bei der kumulierten Menge möglich sein, eine Teilkomponente überdimensionieren zu können, solange man eine andere unterdimensioniert. Dies soll allerdings nicht möglich sein.

In der Benutzeroberfläche würde die Erfassung also wie in Abbildung 35: Lizenzanforderungserfassung in der GUI ersichtlich aussehen.

The screenshot shows the LMT (License Management Tool) interface. The left sidebar contains navigation links: Zusammenfassung, Kundenverwaltung, Infrastrukturverwaltung, Softwareverwaltung, Lizenzverwaltung, Softwareanalyse, and Anforderungen. The main area is titled 'Erstellen/Bearbeiten' and contains a form for creating or editing a license. The form includes fields for 'Bezeichnung' (Oracle Database Standard Edition 2 (R)), 'Lizenztyp' (pro User), 'Lizenzform' (Kommerzielle), 'Lizenzklasse' (Vollversion), 'Lizenzart' (Einzelplatzlize), 'Startdatum', 'Enddatum', 'Preis' (304), and various resource limits like 'Minimale Usermenge' (10), 'Maximale Usermenge', 'Minimale CPU-Core-Menge', 'Maximale CPU-Core-Menge', etc. A table at the bottom shows the license details for 'Oracle Database 12c'.

Software	Nummer	Features	Startdate	Enddate	Löschen
Oracle Database	12c	Analytic Functions, Datenbank, Pluggable Database, Real Application Cluster (RAC), virtuelle Spalten	-	-	

Abbildung 35: Lizenzanforderungserfassung in der GUI

Zusätzlich zu den Oracle Database Lizenzen wurde noch eine weitere Testlizenz erstellt. Dies hatte den Sinn, die Funktionalität von featureübergreifende Addons zu prüfen. Die Optionen von Oracle, welche der Lizenzklasse Addon entsprechen, sind unter 12c allesamt modular aufgebaut, sodass in der Featureabdeckung keine Überschneidungen auftraten. Um diesen Fall vernünftig testen zu können, wurde in der Testlizenz eine komplexe Kombination aus Addons und deren Featureabdeckung erstellt, welche in der Realität wahrscheinlich nicht diese Auswüchse annehmen würde, aber zum Test der kompletten Abdeckung der Addonermittlung sehr hilfreich war.

Tabelle 46: Lizenz und Featureabdeckung von Testlizenz

	Testlizenz	Testaddon 1	Testaddon 2	Testaddon 3	Testaddon 4
Testfeature 1	X				
Testfeature 2		X	X		X
Testfeature 3			X	X	X
Testfeature 4		X		X	X
Testfeature 5		X			X

Wie in Tabelle 46: Lizenz und Featureabdeckung von Testlizenz ersichtlich, sind die Testaddons pro Spalte dargestellt, während die möglichen Features in den unterschiedlichen Zeilen angeführt werden. Die grün hinterlegten und mit einem X markierten Zellen geben an, dass ein Addon ein bestimmtes Feature abdeckt. So deckt also beispielsweise das Testaddon 1 die Features Testfeature 2, Testfeature 4 und Testfeature 5 ab. Wie in der Tabelle ersichtlich, kommt es zu einigen Überschneidungen bei der Featureabdeckung der angebotenen Addons, welche in den Anforderungstests relevant sein werden, vor allem im Hinblick auf die Performance.

7.2 Infrastrukturerfassung

Zum Testen der Anforderungsanalyse wurden entsprechend einige Infrastrukturelemente und Kombination erstellt, um möglichst viele Konstellationen durchtesten zu können. Wie in Abbildung 36: Infrastrukturaufbau für die Testszenarien ersichtlich, gibt es mehrere Komponenten.

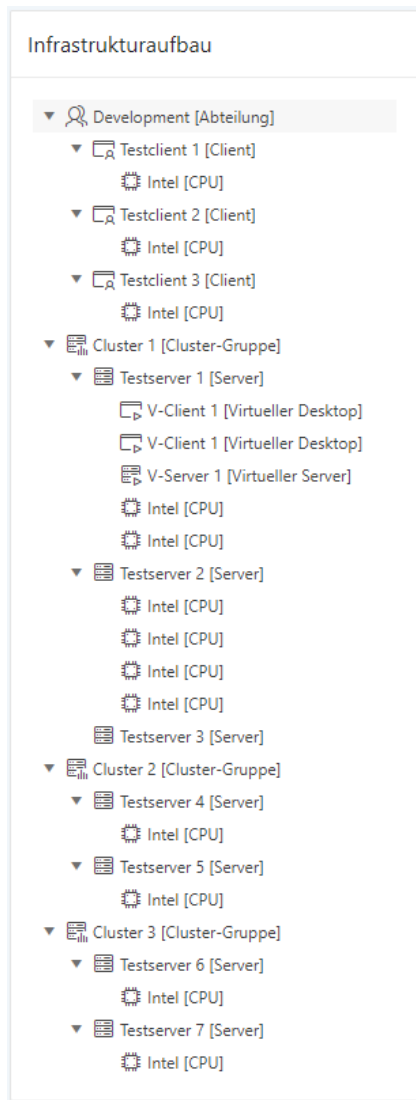


Abbildung 36: Infrastrukturaufbau für die Testszenarien

Um neben der Aufgliederung der einzelnen Komponenten auch deren Details zur Bestückung zu erfahren, folgt mit Tabelle 47: Werte der Infrastrukturkomponenten eine Aufstellung der einzelnen Bestückungen und deren entscheidenden Werte.

Tabelle 47: Werte der Infrastrukturkomponenten

Gruppe	Subkomponente	Sockets	Cores	Threads
Abteilung	Testclient 1	1	4	4
	Testclient 2	1	2	2
	Testclient 3	1	4	8
Cluster 1	Testserver 1	2	2x8	2x8
	Testserver 2	4	4x4	4x4
	Testserver 3	4	k.a.	k.a.
Cluster 2	Testserver 4	2	10	10
	Testserver 5	2	4	4

Cluster 3	Testserver 6	2	8	8
	Testserver 7	2	8	8

Mit diesen Datensätzen kann nun die Kontrolle der Funktionalität der Anforderungsanalyse erfolgen.

7.3 Anforderungstests

Da nun die Grundlage für eine Anforderungsanalyse geschaffen ist, soll mittels diverser Anforderungen ein gewünschtes Ergebnis ermittelt und analysiert werden. Die Tests werden in einzelne Testcases unterteilt und das erhaltene Ergebnis anschließend genauer beschrieben.

7.3.1 Testcase 1 – Datenbank für einen User auf einem Client

Im ersten Testfall soll eine Oracle 12c Lizenz ermittelt werden, die für einen User benötigt wird. Die entsprechend zugewiesene Infrastruktur ist der Testclient 1, deren Eckdaten in Tabelle 47: Werte der Infrastrukturkomponenten eingesehen werden können.

Die entsprechende Anforderungserstellung sieht nun wie in Abbildung 37: Anforderung für Testcase 1 gezeigt aus, das entsprechende Ergebnis ist in Abbildung 38: Ergebnis zu Testcase 1 abgebildet.

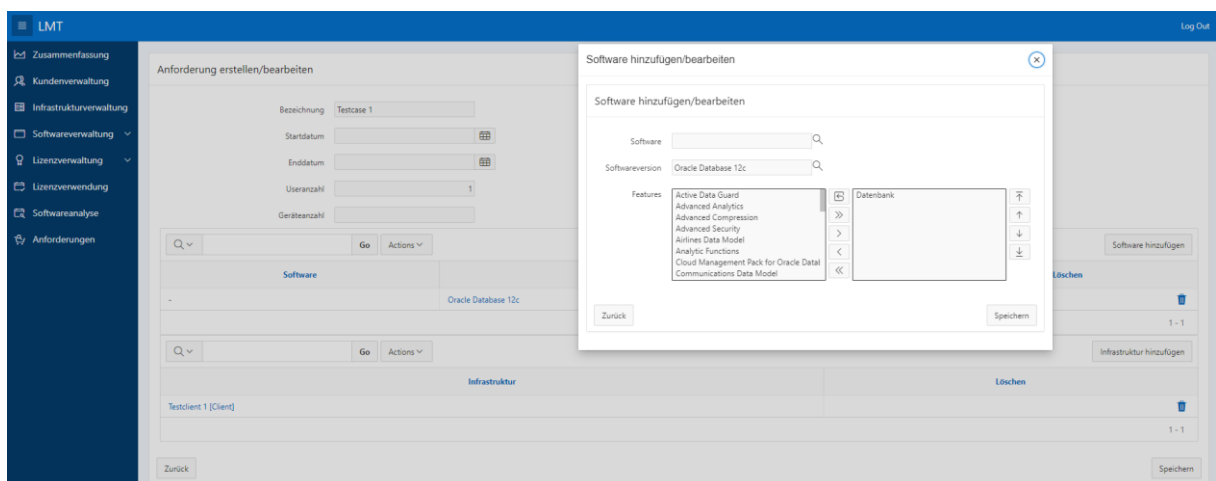


Abbildung 37: Anforderung für Testcase 1

Ergebnisse

Lizenz	Addons	Stück	Preis	Abgedeckte Features	Fehlende Features	Erfüllte Anforderungen	Anforderungen	Un erfüllte Anforderung
Oracle Database Personal Edition (pro User) (399 EUR)		1	399	• Datenbank		• Useranzahl: 1		
Oracle Database Standard Edition 2 (pro User) (304 EUR)		10	3040	• Datenbank		• Geräteanzahl: 1 • CPU-Socketanzahl: 1 • CPU-Socketanzahl in Verwendung: 1 • CPU-Threads in Verwendung: 4	• Minimale Useranzahl: 10	
Oracle Database Standard Edition 2 (RAC) (pro User) (304 EUR)		20	6080	• Datenbank		• CPU-Socketanzahl: 1 • CPU-Socketanzahl in Verwendung: 1 • CPU-Threads in Verwendung: 4	• Minimale Useranzahl: 20 • Minimale Geräteanzahl: 2	
Oracle Database Standard Edition 2 (pro CPU) (15194 EUR)		1	15194	• Datenbank		• CPU-Socketanzahl: 1 • CPU-Socketanzahl in Verwendung: 1 • CPU-Threads in Verwendung: 4		
Oracle Database Standard Edition 2 (RAC) (pro CPU) (15194 EUR)		1	15194	• Datenbank		• CPU-Socketanzahl: 1 • CPU-Socketanzahl in Verwendung: 1 • CPU-Threads in Verwendung: 4	• Minimale Geräteanzahl: 2	
Oracle Database Enterprise Edition (pro User) (825 EUR)		25	20625	• Datenbank			• Minimale Useranzahl: 25	
Oracle Database Enterprise Edition (pro CPU) (41240 EUR)		1	41240	• Datenbank				

Abbildung 38: Ergebnis zu Testcase 1

Das günstigste Ergebnis, welches sowohl alle Kundenanforderungen abdeckt und alle Lizenzbedingungen erfüllt, ist die Oracle Database Personal Edition. Sie ist nach der vorgeschlagenen Sortierreihenfolge der Applikation an erster Stelle und entspricht den erwarteten Ergebnissen, denn die Oracle Database Personal Edition ist eine kostengünstige Datenbanklizenz, die zwar alle Funktionen zulässt, aber auf einen einzigen User beschränkt ist.

Als zweites Ergebnis erhalten wir die Oracle Standard Edition 2 mit dem Lizenztyp „pro User“. Da hier die minimale Useranzahl 10 ist, werden die Kosten auch entsprechend auf 10 User hochgerechnet, weil unsere Anforderung unter der Minimalmarke liegt.

An dritter Stelle folgt die Oracle Standard Edition 2 RAC-Lizenz, ebenfalls mit Lizenztyp „pro User“. Da hier eine minimale Useranzahl von 10 pro Server verlangt wird und im RAC-Verbund mindestens 2 Server verlangt werden, ergibt sich die minimale Useranzahl von 20, welche auch korrekt zur Berechnung des Preises erfolgt.

Danach folgen noch die deutlich teureren CPU-Lizenzen, sowie die Enterprise Lizenzen nach demselben Muster.

7.3.2 Testcase 2 – Datenbank für einen RAC mit zwei Rechnern mit unterschiedlicher CPU-Threadanzahl

In Testcase 2 wollen wir nun etwas Komplexeres suchen und zwar eine Datenbanklizenz für einen RAC (Real Application Cluster, in unserem Fall „Cluster 2“), welcher zwei Server aufweist, wobei einer der Server 10 Threads aufweist, der andere nur 4. Der

aktuelle Prototyp erkennt nicht, dass für einen Cluster bestimmte Features zu verwenden sind. Für diesen ist der Cluster eine Maschine mit den kumulierten Daten von den einzelnen darunterliegenden Servern. Dementsprechend muss in der Featureanforderung explizit mitgegeben werden, dass wir die Funktionalität eines Clusters wünschen, wie in Abbildung 39: Anforderung für Testcase 2 ersichtlich.

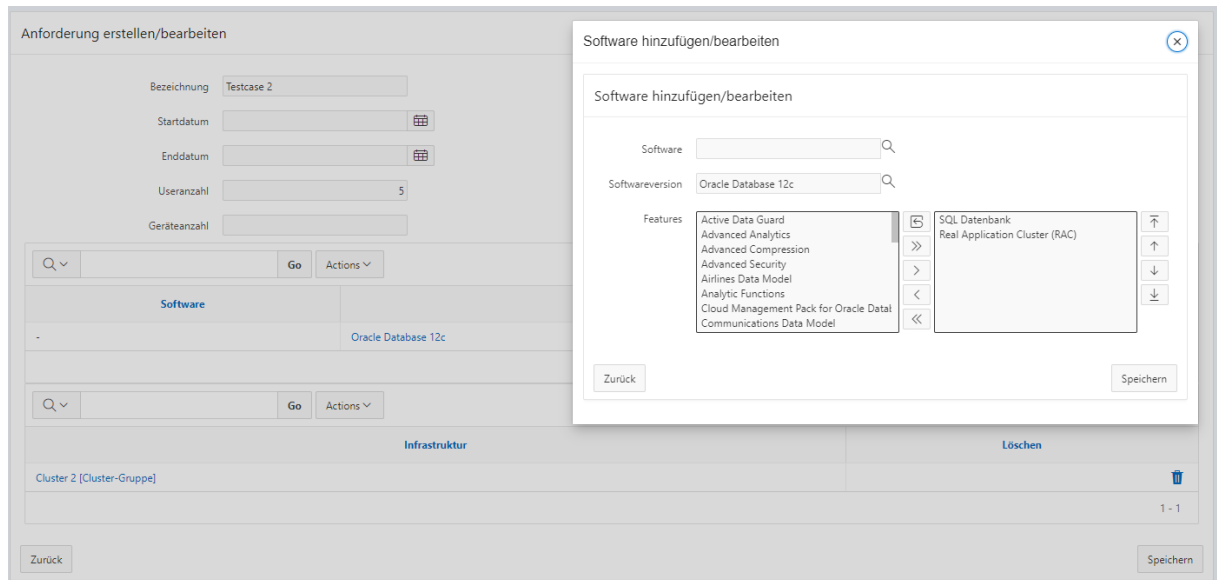


Abbildung 39: Anforderung für Testcase 2

Wie wir in Abbildung 40: Ergebnisse zu Testcase 2 sehen, erhalten wir als einzig gültige Lösung eine Enterprise Edition mit dem RAC Option. Grund dafür ist, dass einerseits für die Personal Edition die Useranzahl zu hoch ist und für die Standard Edition 2 die Teilkomponente Testserver 4 zu viele CPU-Threads vorweist (nämlich 10 Threads, statt den erlaubten 8).

Ergebnisse

Lizenz	Addons	Stück	Preis	Abgedeckte Features	Fehlende Features	Erfüllte Anforderungen	Anforderungen	Unbefüllte Anforderung
Oracle Database Enterprise Edition (pro User) (825 EUR)	• Oracle Database Enterprise Edition Options - Real Application Cluster (RAC) (pro User) (399 EUR)	25	30600	- Real Application Cluster (RAC) - SQL Datenbank			Minimale Useranzahl: 25	
Oracle Database Personal Edition (pro User) (199 EUR)		5	1995	- Real Application Cluster (RAC) - SQL Datenbank				Maximale Useranzahl: 1 (vorhanden: 5)
Oracle Database Standard Edition 2 (RAC) (pro User) (104 EUR)		20	6080	- Real Application Cluster (RAC) - SQL Datenbank		- Geraateanzahl: 2 - CPU-Socketanzahl: 4 - CPU-Socketanzahl in Verwendung: 2 - CPU-Threads in Verwendung: 14	Minimale Useranzahl: 20	Maximale CPU-Threads in Verwendung von Testserver 4: 8 (vorhanden: 10)
Oracle Database Standard Edition 2 (RAC) (pro CPU) (15194 EUR)		2	30388	- Real Application Cluster (RAC) - SQL Datenbank		- Geraateanzahl: 2 - CPU-Socketanzahl: 4 - CPU-Socketanzahl in Verwendung: 2 - CPU-Threads in Verwendung: 14		Maximale CPU-Threads in Verwendung von Testserver 4: 8 (vorhanden: 10)

Abbildung 40: Ergebnisse zu Testcase 2

7.3.3 Testcase 3 – Datenbankanforderung mit fehlenden Informationen.

In diesem Testcase wollen wir uns das Verhalten der Ergebnisanalyse ansehen, wenn uns Teilinformationen fehlen. Hierzu erstellen wir eine Anforderung für eine Datenbanklizenz zu Testserver 3, welcher zwar die Sockets, aber keine zugewiesenen CPUs angegeben hat. Außerdem verzichten wir auf die Angabe der benötigten User. Die entsprechende Anforderung ist in Abbildung 41: Anforderung für Testcase 3 ersichtlich.

Anforderung erstellen/bearbeiten

Bezeichnung: Testcase 3
Startdatum:
Enddatum:
Useranzahl:
Geräteanzahl:

Go Actions

Software

Oracle Database 12c

Go Actions

Infrastruktur

Testserver 3 [Server]

Zurück

Software hinzufügen/bearbeiten

Software:
Softwareversion: Oracle Database 12c

Features:

Active Data Guard
Advanced Analytics
Advanced Compression
Advanced Security
Airlines Data Model
Analytic Functions
Cloud Management Pack for Oracle Data
Communications Data Model

SQL Datenbank

Zurück
Speichern

Speichern

Abbildung 41: Anforderung für Testcase 3

Als Ergebnis erhalten wir nun eine Auswertung, die die fehlenden Daten mit Minimalvoraussetzungen befüllt. Dieses ist in Abbildung 42: Ergebnis zu Testcase 3 ersichtlich.

Ergebnisse

Go Zurück

Ergebnisse annehmen

3 Fehlende Angaben, fehlende Features, unerfüllte Anforderu...

Lizenz	Addons	Stück	Preis	Abgedeckte Features	Fehlende Features	Erfüllte Anforderungen	Anforderungen	Unbefüllte Anforderung
Oracle Database Personal Edition (pro User) (399 EUR)		1	399	• SQL Datenbank			• Minimale Useranzahl: 1	
Oracle Database Enterprise Edition (pro User) (825 EUR)		25	20625	• SQL Datenbank			• Minimale Useranzahl: 25	
Oracle Database Enterprise Edition (pro CPU) (41240 EUR)		1	41240	• SQL Datenbank				
Oracle Database Standard Edition 2 (pro User) (304 EUR)		10	3040	• SQL Datenbank		• Geräteanzahl: 1 • CPU-Socketanzahl in Verwendung: 0	• Minimale Useranzahl: 10 • Minimale CPU-Threads in Verwendung:	• Maximale CPU-Socketanzahl: 2 (vorhanden: 4)
Oracle Database Standard Edition 2 (RAC) (pro User) (304 EUR)		20	6080	• SQL Datenbank		• CPU-Socketanzahl in Verwendung: 0	• Minimale Useranzahl: 20 • Minimale Geräteanzahl: 2 • Minimale CPU-Threads in Verwendung:	• Maximale CPU-Socketanzahl: 2 (vorhanden: 4)
Oracle Database Standard Edition 2 (RAC) (pro CPU) (15194 EUR)		1	15194	• SQL Datenbank		• CPU-Socketanzahl in Verwendung: 0	• Minimale Geräteanzahl: 2 • Minimale CPU-Threads in Verwendung:	• Maximale CPU-Socketanzahl: 2 (vorhanden: 4)
Oracle Database Standard Edition 2 (pro CPU) (15194 EUR)		1	15194	• SQL Datenbank		• CPU-Socketanzahl in Verwendung: 0	• Minimale CPU-Threads in Verwendung:	• Maximale CPU-Socketanzahl: 2 (vorhanden: 4)

Abbildung 42: Ergebnis zu Testcase 3

Die fehlenden Daten führen also nicht automatisch zu einer Ablehnung eines Lizenzvorschlages, es wird allerdings durch die gelbe Einfärbung darauf hingewiesen, dass durch mindestens ein fehlendes Kriterium bei der Kundenanforderung nicht ausreichend Daten vorhanden sind, um eine endgültige Aussage treffen zu können. Als Beispiel wird hier die Oracle Database Personal Edition angeführt, welche nur für einen User zulässig ist. Durch die fehlende Angabe der Useranzahl kann nicht ermittelt werden, ob der Einsatz der Personal Edition hier zielführend ist. Da aber bereits genug Daten vorhanden sind, um den Ausschluss gewisser Lizenzen zu bestätigen (in diesem Fall sind zu viele CPU-Sockets für eine Oracle Database Standard Edition 2 vorhanden), werden diese sehr wohl wie gewohnt rot hinterlegt. Für die Ermittlung der Enterprise Edition auf „pro CPU“-Basis wird die Useranzahl hingegen nicht benötigt, deshalb stellt dieser Eintrag ein bereits gültiges Ergebnis dar und hat dementsprechend einen weißen Hintergrund.

7.3.4 Testcase 4 – Anforderung einer Lizenz mit Addon-übergreifenden Featureabdeckungen

Der nächste Testfall stellt eine Besonderheit dar, die mit aktuellen Oracle-Lizenzen nicht erreicht wird, da dort die Addons (in der Oracle-Welt „Options“ genannt) so modular gehalten sind, dass diese kein Feature eines anderen Addons abdecken. Hierzu wurde

eine spezielle Testlizenz erstellt, welche in Kapitel 7.1 Lizenzfassung näher beschrieben wurde. Man beachte auch, dass keine Featureanforderung auf das Testfeature 1 gestellt wird, welches durch die Vollversion „Testlizenz 1“ abgedeckt wird. Zusätzlich wird eine Useranzahl von 5 mitangegeben.

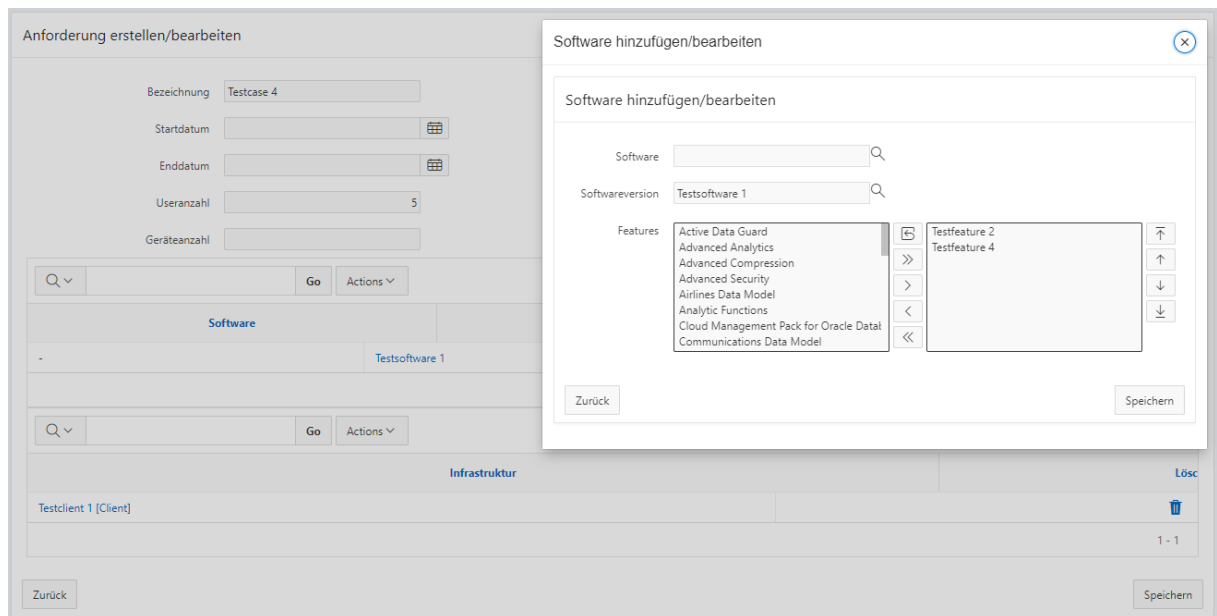


Abbildung 43: Anforderung für Testcase 4

Da diese Testlizenzen alle vom Typ „pro User“ sind, entstehen keine Hardwareanforderungen, weshalb wir der Übersichtlichkeit halber die entsprechenden Spalten in der Abbildung der Ergebnisse ausblenden, uns dafür aber die zusätzlich gebotenen Features anzeigen lassen. Das Ergebnis sieht man in Abbildung 44: Ergebnis zu Testcase 4.

Ergebnisse

Go Actions						
2 Unerfüllte Anforderu..., Fehlende Features						
Lizenz	Addons	Stück	Preis	Abgedeckte Features	Fehlende Features	Zusätzliche Features
Testlizenz [pro User] (10 EUR)	Testaddon 1 [pro User] (5 EUR)	5	75	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 5
Testlizenz [pro User] (10 EUR)	Testaddon 4 [pro User] (7 EUR)	5	85	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 3 - Testfeature 5
Testlizenz [pro User] (10 EUR)	- Testaddon 2 [pro User] (3 EUR) - Testaddon 3 [pro User] (4 EUR)	5	85	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 3
Testlizenz [pro User] (10 EUR)	- Testaddon 1 [pro User] (5 EUR) - Testaddon 2 [pro User] (3 EUR)	5	90	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 3 - Testfeature 5
Testlizenz [pro User] (10 EUR)	- Testaddon 1 [pro User] (5 EUR) - Testaddon 3 [pro User] (4 EUR)	5	95	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 3 - Testfeature 5
Testlizenz [pro User] (10 EUR)	- Testaddon 2 [pro User] (3 EUR) - Testaddon 4 [pro User] (7 EUR)	5	100	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 3 - Testfeature 5
Testlizenz [pro User] (10 EUR)	- Testaddon 3 [pro User] (4 EUR) - Testaddon 4 [pro User] (7 EUR)	5	105	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 3 - Testfeature 5
Testlizenz [pro User] (10 EUR)	- Testaddon 1 [pro User] (5 EUR) - Testaddon 4 [pro User] (7 EUR)	5	110	- Testfeature 2 - Testfeature 4		- Testfeature 1 - Testfeature 3 - Testfeature 5
1 - 8						

Abbildung 44: Ergebnis zu Testcase 4

Es werden also alle sinnvollen Kombinationen der Lizenzdarstellung ermittelt und angeführt. Die günstigste Variante ergibt sich durch die Kombination von der Vollversion „Testlizenz“ in Kombination mit dem Addon „Testaddon 1“. Aber auch die danach folgenden Kombination sind preislich nahe beieinander. Anhand der zusätzlichen Features kann nun also entschieden werden, ob eventuelle zusätzliche, vorher noch nicht eingeplante Features, einen etwas erhöhten Preis rechtfertigen, das günstigste Bundle ist also nicht zwingend die optimalste Lösung, auf jeden Fall aber die kostenschonendste.

Bei der Durchführung zeigt sich deutlich, dass die diversen Abdeckungen von Kundenanforderungen durch unterschiedliche Lizenztöpfe (also die Kombination aus Vollversion und etwaigen Addons) ein deutlicher Mehraufwand zur Ermittlung der

geeigneten Kombinationen entsteht. Dies resultiert in hohe Durchlaufzeiten bei der Ergebnisfindung. Entsprechend war nun die Überlegung, die Ergebnisse nicht jedes Mal neu zu berechnen, sondern in eine entsprechende Result-Cache-Tabelle abzulegen und nur bei Veränderung der Ausgangssituation, sei es durch das Abändern der Kundenanforderungen oder der dazugehörenden Lizenzabdeckung, neu zu berechnen. Ändern sich die beteiligten Daten nicht, so kann das Ergebnis direkt aus der Result-Cache-Tabelle ermittelt werden und muss nicht jedes Mal aufs Neue berechnet werden, was die Anzeige und das Handling mit den Ergebnissen deutlich optimiert.

Generell ist aber auch darauf zu achten, entsprechende Features, die den einzelnen Lizenzen, vor allem aber den Addons zugewiesen werden, nicht zu detailliert zu erstellen, um eine exponentiell wachsende Anzahl an Möglichkeiten der Zusammenstellung entsprechender Lizenztöpfe zu vermeiden.

Die Testfälle sollten verdeutlichen, wie der Ansatz zur Ermittlung von geeigneten Lizenzen auf Basis von Kundenanforderungen mittels dieses Tools ermöglicht werden. Sie zeigen, welche Anforderungen bereits abgedeckt werden können und wo noch Anpassungsbedarf besteht. Außerdem sollten sie einen kleinen Einblick darüber liefern, wie versucht wird, die komplexen Lizenzanforderungen möglichst genau zu analysieren und so dem Kunden eine oder mehrere optimale Lösungen zu präsentieren, anhand derer er nun die Kaufentscheidung tätigen kann, um im nächsten Schritt die entsprechenden Lizenzen in seinen Lizenzpool aufzunehmen. Dies ist zwar nicht mehr Teil dieses Prototyps, soll aber sukzessive der Applikation hinzugefügt werden.

8. Zusammenfassung und Conclusio

Diese Arbeit beschäftigte sich mit der Planung zur Erstellung eines Tools zur Unterstützung des Lizenzmanagements und sollte alle Bereiche der derzeit benötigten Anforderungen an ein Lizenzmanagement ausarbeiten und entsprechend in einer sinnvollen Weise die einzelnen Prozesse mit einem selbst entwickelnden Tool unterstützen. Danach wurde ein Prototyp entwickelt, welcher den Einsatz des Tools demonstriert und die Komplexitäten der Handhabung von Lizenzen aufweist.

Der Prototyp umfasst einerseits die Erfassung von Stammdaten (Software, Lizenz, Kunde, Infrastruktur), sowie andererseits die Abbildung der Beziehung dieser Daten.

Auf Lizenzseite kann definiert werden, welche Bedingungen Lizenzen an eine Infrastruktur oder Useranzahl stellen. Dies kann sowohl eine vorgeschriebene Mindestmenge beim Erwerb sein, als auch eine Limitierung von einzelnen Lizenz-Editionen, die durch Maximalwerte dargestellt werden. Ebenfalls kann erfasst werden, welche Software eine Lizenz abdeckt und vor allem welche Features der Software von der entsprechenden Edition abgedeckt werden.

Durch die Literatur wurden viele Aspekte der Relationen von Lizenzen untereinander im Vorfeld ermittelt und konnten entsprechend bei der Planung umgesetzt werden. So gibt es die Möglichkeit der Abbildung von Addons und Upgrades, sowie eines Verweises zu einem etwaig benötigten Wartungsvertrag über entsprechende Lizenzbeziehungen. Was nachträglich noch ergänzt wurde, war die Abbildung von äquivalenten Lizenzen mit unterschiedlichen Anforderungen, beziehungsweise unterschiedlichen Einsatzgebieten, die jederzeit in die jeweilig andere Anwendungsform umkonvertiert werden dürfen.

Die aus der Literatur analysierte Unterteilung der Lizenzen in Form, Art, Klasse und Typ war nur teilweise hilfreich. Einerseits war die Unterteilung der Lizenztypen (per User, per Gerät, per CPU) und der Lizenzklasse (Vollversion, Addon, Upgrade, ...) ausschlaggebend für die richtige funktionale Verwendung der Lizenz, andererseits waren die Erfassung der Lizenzform (kommerziell, Open Source, ...) und der Lizenzart (Einzelplatz, Mehrplatz) irrelevant für die Ermittlung des Lizenzeinsatzes, da sich diese durch die detaillierteren Angaben der Lizenzattribute automatisch ableiten ließen. Die Erfassung dieser Attribute ist dennoch erhalten geblieben, da die Information an sich

sinnvoll ist. Es bleibt aber zu überlegen, ob diese Attribute zukünftig automatisch hergeleitet werden und nicht mehr direkt angegeben werden müssen.

Auf Kundenseite kann die eingesetzte oder geplante Infrastruktur eines Unternehmens abgebildet werden, um sowohl bei der Ermittlung einer passenden Lizenz entsprechende Kennzahlen zu liefern, als auch beim anschließenden Lizenzeinsatz als entsprechende Zuweisung zu dienen.

Der wichtigste Punkt des Prototyps stellt die darauffolgende Kundenanforderungsanalyse dar. Es handelt sich hierbei um eine Unterstützung, die in den meisten aktuell am Markt befindlichen Lizenzmanagementtools fehlt, nämlich die Ermittlung einer entsprechenden Lizenzabdeckung zu einer gewünschten Software und einem gewünschten Einsatzgebiet (abgebildet durch die Infrastruktur). Dies wird aktuell für eine Software und eine Infrastrukturkomponente ermöglicht, welche aber auch eine Gruppierung, wie eine Abteilung oder ein Cluster, sein kann. Zukünftig sollten hier aber noch weitere Kombinationen ermöglicht werden.

Die aktuelle Anforderungsberechnung deckt vor allem die Oracle Datenbanklizenzen ab. Sie ist zwar bereits für weitere Lizenzdefinitionen entwickelt, aber aufgrund der hohen Anzahl an Kombinationsmöglichkeiten noch nicht komplett durchgetestet. Dies soll in weiterer Folge immer mit dem Hinzufügen neuer Lizenzen geschehen. Zu beachten ist hier auch, dass mit dem Erweitern der möglichen Infrastrukturtypen, Lizenztypen, sowie der Beziehungstypen dieser beiden auch entsprechende Anpassungen beziehungsweise Erweiterungen an der Ermittlung der Lizenzen für die Kundenanforderungen durchgeführt werden müssen.

Die Anforderungsanalyse deckt weiters im aktuellen Prototypen noch keine zeitliche Beschränkung ab, weder in der Laufzeit, noch in der Zeit der Verfügbarkeit. Dies wird einen der nächsten Schritte darstellen, um nicht mehr am Markt erhältliche Lizenzen aus der Ermittlung herausfiltern zu können.

Ausstehend sind auch noch die Bereiche der Softwareanalyse, welche im ersten Schritt dem Kunden helfen soll, mögliche Softwarekandidaten bewerten und einen Pool qualifizierter Software erstellen zu können, sowie der Bereich der eigentlichen Softwareverwendung, also den Aufbau eines Softwarepools inklusive der Zuweisung zu entsprechenden Infrastrukturkomponenten, mit Unterstützung von Clienttools, die

ermöglichen sollen, den aktuellen Softwareeinsatz durch das Auslesen entsprechender Daten auf den jeweiligen Clients zu ermitteln und zentral im Unterstützungstool abzubilden.

Sobald dies implementiert wird, wird auch eine entsprechende Anzeige für die Compliance von in Verwendung befindlichen Lizenzen benötigt, um entsprechende Engpässe ersichtlich zu machen.

Danach gilt es ein letztes Mal die Anforderungsanalyse zu erweitern, um unter Berücksichtigung bereits vorhandener Lizenzen im Unternehmen nur wirklich benötigte fehlende Lizenzen, sowie etwaige Lizenzupgrades ermitteln zu können, anstatt nur den Kauf von Vollversionen zu forcieren.

Wie man bereits bei der Implementierung des Prototyps gesehen hat, ist selbst ein kleiner Ausschnitt aus diesem Bereich eine äußerst komplexe Angelegenheit, bei der es gilt, auch auf die kleinsten Details zu achten, um zu verhindern, dass eine Lizenz nicht ordnungsgemäß eingesetzt wird. Gerade in der heutigen Zeit können Lizenzbedingungen äußerst komplexe Ausmaße annehmen, die man beim erstmaligen Durchlesen der Lizenzbedingungen noch gar nicht alle erfassen kann. Umso wichtiger ist es, ein fehlerfreies Tool anbieten zu können, welches diese Eckpunkte verdeutlicht und aufzeigt, wo eventuelle Lizenzvergehen eintreten können, sei es, da eine Hardware nicht den Anforderungen entspricht oder beispielsweise eine Minimalstückzahl, welche von mehreren Faktoren abhängig sein kann, nicht korrekt ermittelt wurde.

Genau bei diesem Punkt sollte es auch klar sein, dass beim Erfassen neuer Lizenzen mitsamt deren Bedingungen streng darauf geachtet werden muss, dass alle Eckpunkte korrekt erfasst wurden, um sie nachträglich fehlerfrei verarbeiten zu können. Im aktuellen Prototyp ist die korrekte Lizenzerfassung einer der aufwändigsten Punkte. Man muss alle Faktoren korrekt aus der Angabe des Lizenzgebers ableiten und diese dann auch korrekt im System erfassen, um die Anforderungsanalyse korrekt durchlaufen zu lassen. Dies ist einer der Gründe, warum vorläufig nur Systemadministratoren neue Lizenzen erfassen dürfen.

Ein weiterer Punkt ist die Überlegung und Festlegung der Features, welche definieren, welche Kundenanforderungen durch eine Lizenz abgedeckt werden. Erstellt man zu detaillierte Anforderungspunkte (mittels Features), so leidet die Performance bei der

Auswertung. Generiert man sie zu generisch, so sind sie keine vernünftige Unterstützung für einen Kunden, der mithilfe einer Auswahl dieser die geeignetste Lizenz ermitteln möchte. Es gilt einen passenden Mittelweg zu finden, um eine perfekte Analyse zu ermöglichen.

Neben den bereits implementierten und geplanten Teilen ist es wichtig, die Veränderungen am Lizenzmarkt im Auge zu behalten. Neue Lizenzanforderungen könnten geschaffen werden, die neue Komplexitäten mit sich bringen könnten oder aber auch Erleichterungen.

Zukünftig ist zu überlegen, ob die Abbildung von SaaS-Lizenzen eventuell automatisiert erfolgen kann, indem man über eine Schnittstelle oder mithilfe eines Crawlers und einem Zugang zur Verwaltung der Unternehmenslizenzen bei einem entsprechenden SaaS-Anbieter, die benötigten Daten auslesen kann, um diese automatisiert in das unternehmensweite Lizenzmanagementtool zu übernehmen.

Ebenso könnte das System dahingehend erweitert werden, um Lizenzanbietern zu ermöglichen, ihre Lizenzangebote mitsamt den vorgesehenen Anforderungen und der daraus resultierenden Abdeckung selbstständig zu erfassen. Es könnte sich schließlich eine Win-Win-Situation ergeben, da die Unternehmen damit neue Kunden erreichen können, die gerade in Betracht ziehen, neue Software für deren Unternehmen anzuschaffen. Im Umkehrschluss hat der Kunde den größtmöglichen Überblick über die aktuelle Lizenzlandschaft am Markt.

Aber dies sind aktuell noch Visionen, der erste Schritt ist durch den Prototyp getan, weitere werden folgen, um die nötige Anforderung an das ausgearbeitete Unterstützungstool für das Lizenzmanagement Schritt für Schritt umzusetzen.

9. Literaturverzeichnis

- altiris. (2004, Jänner 9). *altiris - Software License Management*. Retrieved Mai 12, 2010, from www.altiris.com
- BMDW. (2018). *Rechtsinformationssystem des Bundes*. Retrieved from Rechtsinformationssystem des Bundes: <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10001848>
- Casares, J. (2016). *EUPL [European Union Public Licence]*. <https://eupl.eu>.
- Creamer, L. (2017, 09 28). *PCMag UK*. Retrieved from PCMag UK: <http://uk.pcmag.com/cloud-services/79270/guide/the-best-asset-management-software-of-2017>
- Dietmar Jähnel, A. S. (2003). *Informatikrecht*. Wien: Springer.
- Duden. (2018). *Lizenz*. <https://www.duden.de/rechtschreibung/Lizenz>.
- Duman, Y. (2008). *Eine kritische Analyse des Lizenzmanagements und von ASP-Lösungen anhand einer empirischen Untersuchung zur Handhabung von Lizenzsoftware in deutschen KMU's*. Mannheim: Grin.
- Fina, a.-P. D. (n.d.). *Der Schutz von Computerprogrammen (Software) im österreichischen Urheberrecht*.
- Gartner Peer Insights. (2018). Retrieved from Gartner Peer Insights: <https://www.gartner.com/reviews/market/software-asset-management-tools>
- Gehring, R., & Lutterbeck, B. (2004). *Urheber- und Lizenzrecht im Bereich von Open-Lehmanns Media*.
- Groll, T. (2016). *1x1 des Lizenzmanagements*. München: Carl Hanser Verlag.
- Gull, D., & Wehrmann, A. (2009). *Optimized Software Licensing – Combining License Types in a License Portfolio*.
- Härtel, E. (2015). *Entwicklung eines Leitfadens für das Software-Asset-Management in mittelständischen und Großunternehmen mit Schwerpunkt auf dem korrekten Umgang mit Softwarelizenzen*. Grin.

- Henn, G. (2003). *Patent- und Know-how-Lizenzvertrag: Handbuch für die Praxis*.
<http://books.google.at/books?hl=de&lr=&id=I8DpqlkjupQC&oi=fnd&pg=PR5&dq=lizenz+rechtlich&ots=lyyREHEUbq&sig=SYm9dJu54xag6R2NnxPgLvuo73w#v=onepage&q=lizenz%20rechtlich&f=false>: C.F.Müller.
- Jahnel, D., Schramm, A., & Staudegger, E. (2002). *Informatikrecht*. Wien: Springer-Verlag.
- Koratamaddi, C. (2015, 04). *oracle.com*. Retrieved from oracle.com:
<http://www.oracle.com/technetwork/developer-tools/apex/learnmore/apex-50-overview-2526922.pdf>
- Mayer, H. (2009). Urheber- und Lizenzrecht. In H. Mayer, *Medienmanagement*. VS Verlag für Sozialwissenschaften.
- Microsoft. (2018). Retrieved from Microsoft: <https://www.microsoft.com/en-us/sam/tools.aspx>
- Oracle. (2005, Juni). *Oracle database License Information 10g R2*. Retrieved Mai 12, 2010, from www.oracle.com
- Oracle. (2018). *Oracle Store*. Retrieved from Oracle Store: <https://shop.oracle.com>
- Oracle. (n.d.). *apex.oracle.com*. Retrieved from apex.oracle.com: <https://apex.oracle.com>
- Raekow, Y., Simmendinger, C., & Krämer-Fuhrmann, O. (2009). *License management in grid and high performance computing*. Springer-Verlag.
- Runge, C. (2004). *IT-Asset-Management - welche Software und Datenverwaltung für IT-Objekte ist zweckmäßig?* Norderstedt, Deutschland: GRIN Verlag.
- Salmen-Fuchs, T. (2003). *Management von Softwarelizenzen*. Norderstedt, Deutschland: GRIN Verlag.
- Saygili, Y. (2015, 02 23). *blogs.oracle.com*. Retrieved from blogs.oracle.com:
<https://blogs.oracle.com/imc/oracle-rest-data-services-on-oracle-database-cloud-application-express>
- Sommerhuber, J. (2005). *Ein Leitfaden durch das Software-Recht in Österreich*. Wien: Institut für Rechtswissenschaften der TU Wien.

Systems, R. (2000). *License Management Guide*.

Terzievski, K. (2004, 03 31). *ZDNet*. Retrieved from ZDNet:

<https://www.zdnet.de/39121069/alle-systeme-unter-kontrolle-fuenf-asset-management-tools>

Tuunanen, T., Koskinen, J., & Kärkkäinen, T. (2008). *Automated software license analysis*. Springer Science+Business Media.

Ventroni, S. (2007). *Copyrights und Lizenzmanagement*.

Windley, P. J. (n.d.). *Managing IT Assets*. utah.gov.

10. Abbildungsverzeichnis

Abbildung 1: Software-Lifecyclemodell nach (Groll, 2016)	24
Abbildung 2: Phasen einer Lizenz	27
Abbildung 3: Use Case der Anforderungsanalyse	46
Abbildung 4: Use Case der Lizenzverwendung	48
Abbildung 5: Use Case des Unterstützungstools zur Lizenzverwaltung	49
Abbildung 6: Systemabbildung Oracle Apex	56
Abbildung 7: Oracle Apex Konfiguration mit Oracle-Komponenten	57
Abbildung 8: Oracle Apex - Schema und Workspace	58
Abbildung 9: Globale Navigation	61
Abbildung 10: Kundenverwaltung	63
Abbildung 11: Infrastrukturverwaltung	64
Abbildung 12: Softwareverwaltung	66
Abbildung 13: Lizenzverwaltung	68
Abbildung 14: Lizenzverwendung	70
Abbildung 15: Softwareanalyse	71
Abbildung 16: Lizenzanforderung	73
Abbildung 17: System-Scan über Webservice	74
Abbildung 18: Lokaler System-Scan mit anschließendem Upload	74
Abbildung 19: Datenmodell LMT	77
Abbildung 20: Datenmodell Kundenbereich	80
Abbildung 21: Datenmodell Infrastrukturbereich	88
Abbildung 22: Datenmodell Softwarebereich	91
Abbildung 23: Datenmodell Lizenzbereich	98
Abbildung 24: Datenmodell dynamische Bedingungen	101
Abbildung 25: Datenmodell für Logtabelle	103
Abbildung 26: Beispiel der Übersichtsseite anhand der Kundenverwaltung	109
Abbildung 27: Darstellung einer Datenerfassungsseite mit einer weiteren Detailübersicht anhand der Kundenbearbeitung	110
Abbildung 28: Darstellung einer modalen Bearbeitungsseite anhand der Userbearbeitung	110
Abbildung 29: Abbildung der Features eines modalen Erfassungsfensters anhand der Userbearbeitung	111

Abbildung 30: Darstellung des Komfortfeatures bei der Infrastrukturübersicht	112
Abbildung 31: Darstellung der Untermenüpunkte.....	112
Abbildung 32: Darstellung der Anforderungsübersicht mit Ergebnisschaltfläche	113
Abbildung 33: Darstellung der ermittelten Ergebnisse zu einer konkreten Anforderung	114
Abbildung 34: Beispiel einer Ergebnisanzeige.....	125
Abbildung 35: Lizenzanforderungserfassung in der GUI.....	133
Abbildung 36: Infrastrukturaufbau für die Testszenarien	135
Abbildung 37: Anforderung für Testcase 1.....	136
Abbildung 38: Ergebnis zu Testcase 1.....	137
Abbildung 39: Anforderung für Testcase 2.....	138
Abbildung 40: Ergebnisse zu Testcase 2.....	139
Abbildung 41: Anforderung für Testcase 3.....	139
Abbildung 42: Ergebnis zu Testcase 3.....	140
Abbildung 43: Anforderung für Testcase 4.....	141
Abbildung 44: Ergebnis zu Testcase 4.....	142

11. Tabellenverzeichnis

Tabelle 1: Spaltendefinition der Tabelle LMT_KUNDEN	78
Tabelle 2: Spaltendefinition der Tabelle LMT_BENUTZER.....	79
Tabelle 3: Spaltendefinition der Tabelle LMT_ROLLEN	81
Tabelle 4: Spaltendefinition der Tabelle LMT_LIZENZPOOLS.....	82
Tabelle 5: Spaltendefinition der Tabelle LMT_LL_P_BEZIEHUNGEN	82
Tabelle 6: Spaltendefinition der Tabelle LMT_LL_P_TYPEN.....	82
Tabelle 7: Spaltendefinition der Tabelle LMT_LIZENZVERWENDUNG	83
Tabelle 8: Spaltendefinition der Tabelle LMT_LIZENZDOKUMENTE.....	83
Tabelle 9: Spaltendefinition der Tabelle LMT_SOFTWAREBEWERTUNGEN	84
Tabelle 10: Spaltendefinition der Tabelle LMT_LSB_STATUS.....	84
Tabelle 11: Spaltendefinition der Tabelle LMT_KUNDENANFORDERUNGEN.....	84
Tabelle 12: Spaltendefinition der Tabelle LMT_KUNDENBEDINGUNGEN	85
Tabelle 13: Spaltendefinition der Tabelle LMT_LKA_RESULTS.....	86
Tabelle 14: Spaltendefinition der Tabelle LMT_LLI_LKAR.....	87
Tabelle 15: Spaltendefinition der Tabelle LMT_INFRASTRUKTUREN.....	89
Tabelle 16: Spaltendefinition der Tabelle LMT_LIS_TYPEN.....	89
Tabelle 17: Spaltendefinition der Tabelle LMT_LIS_BEZIEHUNGEN	90
Tabelle 18: Spaltendefinition der Tabelle LMT_LIB_TYPEN	90
Tabelle 19: Spaltendefinition der Tabelle LMT_LIB_LIS.....	90
Tabelle 20: Spaltendefinition der Tabelle LMT_SOFTWAREPACKAGES.....	91
Tabelle 21: Spaltendefinition der Tabelle LMT_VERSIONEN	92
Tabelle 22: Spaltendefinition der Tabelle LMT_LIZENZ_FEATURES.....	92
Tabelle 23: Spaltendefinition der Tabelle LMT_FEATURES.....	92
Tabelle 24: Spaltendefinition der Tabelle LMT_KUNDENBEDINGUNG_FEATURES.....	93
Tabelle 25: Spaltendefinition der Tabelle LMT_LIZENZFORMEN	94
Tabelle 26: Spaltendefinition der Tabelle LMT_LIZENZARTEN	94
Tabelle 27: Spaltendefinition der Tabelle LMT_LIZENZKLASSEN.....	94
Tabelle 28: Spaltendefinition der Tabelle LMT_LIZENZTYPEN.....	94
Tabelle 29: Spaltendefinition der Tabelle LMT_LIZENZEN.....	95
Tabelle 30: Spaltendefinition der Tabelle LMT_LIZENZEINSAETZE	98
Tabelle 31: Spaltendefinition der Tabelle LMT_PREISTABELLEN	99
Tabelle 32: Spaltendefinition der Tabelle LMT_LIZENZABDECKUNGEN	99

Tabelle 33: Spaltendefinition der Tabelle LMT_LLI_BEZIEHUNGEN.....	100
Tabelle 34: Spaltendefinition der Tabelle LMT_LLB_TYPEN.....	100
Tabelle 35: Spaltendefinition der Tabelle LMT_LIZENZBEDINGUNGEN	101
Tabelle 36: Spaltendefinition der Tabelle LMT_WERTE	102
Tabelle 37: Spaltendefinition der Tabelle LMT_ATTRIBUTE	102
Tabelle 38: Spaltendefinition der Tabelle LMT_DATENTYPEN.....	102
Tabelle 39: Spaltendefinition der Tabelle LMT_DATENTYPENOPERATOREN.....	103
Tabelle 40: Spaltendefinition der Tabelle LMT_OPERATOREN.....	103
Tabelle 41: Spaltendefinition der Tabelle LMT_LOGS	103
Tabelle 42: Logspalten.....	116
Tabelle 43: Addon-Feature-Abdeckung	120
Tabelle 44: Lizenzanforderung für Oracle Database Enterprise Edition.....	122
Tabelle 45: Beispiel der Lizenzanforderung für Oracle Standard Edition 2 (RAC) - per User	132
Tabelle 46: Lizenz und Featureabdeckung von Testlizenz.....	134
Tabelle 47: Werte der Infrastrukturkomponenten	135

12. Anhang: Abstract

In today's business life many corporate processes are supported by IT-applications. In order to keep track of the software in use within a company with an increasing number of employees, this master thesis elaborates an application to support software asset management. The first part of this paper contains an analysis of literature about software asset management and already existing tools. Based on the obtained information, appropriate models are created to describe the implementation of the application, after which the implementation of a first prototype is described. In conclusion, the knowledge gained during practical implementation of the prototype is compared with the information gained from the previous theoretical research. Finally, the achieved goals and the outstanding processes are summarized.

Source code as well as a user manual for installing and using the prototype are attached to this work.

Heutzutage werden viele Prozesse in Unternehmen durch Computeranwendungen unterstützt. Um bei steigender Mitarbeiterzahl nicht den Überblick über die im Unternehmen eingesetzte Software, und die damit verbundenen benötigten Software-Lizenzen zu verlieren, soll in dieser Masterthesis eine Anwendung ausgearbeitet werden, welche das Lizenzmanagement unterstützt. Dazu erfolgt im ersten Teil eine Analyse der Fachliteratur, sowie eine Recherche der bereits am Markt erhältlichen Produkte. Anschließend werden anhand der gewonnenen Informationen entsprechende Modelle erstellt, die die Umsetzung der Anwendung beschreiben sollen, woraufhin die Umsetzung eines ersten Prototyps erläutert wird. Im Abschluss wird die gewonnene Erkenntnis, die bei der praktischen Umsetzung des Prototyps entstand, mit der gewonnenen Information aus der zuvor getätigten theoretischen Recherche verglichen. Außerdem werden die erreichten Ziele und noch ausstehenden Tätigkeiten zur fertigen Umsetzung der Applikation zusammengefasst.

Der Sourcecode, sowie eine Bedienungsanleitung zur Installation und Handhabung des ausgearbeiteten Prototyps liegen der Arbeit bei.