

# Analyse, Systemdesign und Architekturentwurf einer elektronischen Gesundheitsakte für Nutztiere

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieurin**

im Rahmen des Studiums

**Software Engineering and Internet Computing**

eingereicht von

**Andrea Füresz**

Matrikelnummer 0525548

an der  
Fakultät für Informatik der Technischen Universität Wien

Betreuung  
Betreuer: Ao.Univ.-Prof. Mag.rer.nat. Dipl.-Ing. Dr.techn. Rudolf Freund

Wien, 10.05.2012

\_\_\_\_\_  
(Unterschrift Verfasserin)

\_\_\_\_\_  
(Unterschrift Betreuer)

# Eidesstattliche Erklärung

Andrea Füresz  
Bahnstrasse 6, 2540 Bad Vöslau

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

I hereby declare that I am the sole author of this thesis, that I have completely indicated all sources and help used, and that all parts of this work - including tables, maps and figures - if taken from other works or from the internet, whether copied literally or by sense, have been labelled including a citation of the source.

---

(Ort, Datum)

---

(Unterschrift Verfasser/In)

# Danksagung

An dieser Stelle möchte ich den Personen danken, die mir bei der Anfertigung dieser Diplomarbeit zur Seite standen. In erster Linie bedanke ich mich bei meiner Familie, die mich während meines gesamten Studiums auf allen Wegen unterstützt hat, an mich geglaubt hat und mir alles was ich erreicht habe, ermöglicht hat.

Außerdem danke ich den Mitarbeitern des Instituts und allen, die mir bei meiner Arbeit hilfreich zur Seite standen, die mich in einer freundlichen Atmosphäre beraten, meine Fragen beantwortet und meine Arbeit Korrektur gelesen haben.

Weiterer Dank gilt Dipl.Tzt. Berthold Grassauer, der sich freundlicherweise bereit erklärt hat, mir bei einem persönlichen Gespräch Fragen zum Thema dieser Arbeit zu beantworten.

Abschließend möchte ich noch meinem Kollegen Constantin Gruber danken, der mir ebenfalls aufgrund seiner langjährigen Erfahrung in der Softwareentwicklung für die Nutztierbranche mit inhaltlichen Anregungen zur Seite stand.

# Abstract

This Master's thesis deals with a system for storing medical relevant information in the farm animal industry. The aim of this thesis is to develop a software architecture design for such a system. It is about the simple and efficient processing of data and the ability to allow complete documentation as well as economic management within the veterinary clinics and veterinary practices.

It will analyse the current market situation, the legal constraints as well as technology standards before going into system design and architecture. Furthermore, the requirements of this system will be characterized and the stakeholders will be defined, for whom a role concept will be developed. This thesis will create the conceptual design, provide a textual description and a visual representation - by the use of UML - from the information gained. The analytical results will be derived from extensive Internet research and interviews with experts from the livestock industry.

The benefit of this work lies in a system that documents the health condition, the previous medical history and medications administered to an animal completely as well as to make this data centrally available. Such a system would help to improve food quality and contribute to better monitoring of diseases, because of the centrally available data a better control would be possible. The final conclusion of this study provides a solid basis for further developments in the implementation of an electronic health record for farm animals by the detailed analysis and the resulting conceptual design.

## Keywords

Software Architecture, System Design, Farm Animals, Electronic Health Act, Animal Health Record, Analysis,

# Kurzfassung

Diese Diplomarbeit beschäftigt sich mit einem System zur Speicherung von gesundheitsrelevanten Daten aus der Nutztierbranche. Das Ziel dieser Arbeit besteht darin, einen Softwarearchitekturentwurf für ein derartiges System zu entwickeln. Dabei geht es darum, die einfache und effiziente Verarbeitung von Daten, die Möglichkeit zur lückenlosen Dokumentation sowie wirtschaftliches Management innerhalb der Tierkliniken und Tierarztpraxen zu ermöglichen.

Vor dem Systemdesign und dem Entwurf wird die derzeitige Lage am Markt, der gesetzliche Rahmen sowie die Technologiestandards, die dabei zum Einsatz kommen könnten, analysiert. Weiters werden die Anforderungen an dieses System charakterisiert und Stakeholder definiert, für die ein Rollenkonzept entwickelt wird. Aus den gewonnenen Ergebnissen entsteht der konzeptionelle Entwurf, der neben einer textuellen Beschreibung durch den Einsatz von UML auch eine visuelle Darstellung liefert. Die Analyseergebnisse stammen aus umfangreichen Internetrecherchen sowie Gesprächen mit Fachleuten aus der Nutztierbranche.

Der Nutzen dieser Arbeit liegt in einem System, das den gesundheitlichen Zustand, die bisherige Krankengeschichte und die verabreichten Medikamente an ein Tier lückenlos dokumentiert sowie diese Daten zentral gesammelt verfügbar macht. Ein derartiges System trägt sowohl zur Verbesserung der Nahrungsmittelqualität sowie zu einer besseren Überwachung von Krankheiten bei, da durch die zentral verfügbaren Daten eine bessere Kontrolle möglich wäre. Das zentrale Ergebnis dieser Arbeit liefert durch die ausführliche Analyse und dem daraus entstehenden konzeptionellen Entwurf eine solide Grundlage für weitere Entwicklungen im Bereich der Umsetzung einer elektronischen Gesundheitsakte für Nutztiere.

## Schlüsselwörter

Softwarearchitektur, Systemdesign, Nutztiere, elektronische Gesundheitsakte, Animal Health Record, Analyse.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>vi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Problemstellung . . . . .	1
1.2 Nutzen der elektronischen Gesundheitsakte für Nutztiere . . . . .	2
1.3 Zielsetzung der Arbeit . . . . .	2
1.4 Gliederung der Arbeit . . . . .	3
<b>2 Grundlagen aus der medizinischen Informatik</b>	<b>4</b>
2.1 State of the Art . . . . .	4
2.1.1 Elektronische Gesundheitsakte . . . . .	5
2.1.2 Derzeit am Markt vorhandene Software . . . . .	7
2.1.2.1 easyVET . . . . .	7
2.1.2.2 Vetinf . . . . .	10
2.1.2.3 ANIMAL-office . . . . .	12
2.1.2.4 Rinderdatenbank der Agrarmarkt Austria . . . . .	14
2.1.2.5 Rinderdatenverbund für Mitglieder - RDV4M . . . . .	15
2.1.2.6 Gesundheitsmonitoring Rind . . . . .	16
2.1.3 Die Gesundheitsakte in der Nutztierbranche . . . . .	16
2.2 Gesetzliche Lage . . . . .	17
2.3 Standards . . . . .	19
2.3.1 Health Level Seven (HL7) . . . . .	19
2.3.2 Logical Observation Identifier Names and Codes (LOINC) . . . . .	23
2.3.3 Digital Imaging and Communications in Medicine (DICOM) . . . . .	26
2.3.4 Systematized Nomenclature of Medicine and Veterinary Medicine (SNOMED) . . . . .	28
2.3.5 Nomina Anatomica Veterinaria . . . . .	30
<b>3 Grundlagen zur Softwarearchitektur</b>	<b>32</b>
3.1 Einführung . . . . .	32
3.2 Qualitätskriterien . . . . .	33
3.3 Architekturentwurfsmuster . . . . .	36
3.3.1 MVC (Model-View-Controller) . . . . .	36
3.3.2 N-Tier-Architektur . . . . .	37
3.4 Datenmodellierung . . . . .	39
3.5 ORM (Object-Relational-Mapping) . . . . .	41
3.6 Unified Modeling Language (UML) . . . . .	41
3.6.1 Anwendungsfalldiagramm . . . . .	42
3.6.2 Aktivitätsdiagramm . . . . .	44
3.6.3 Komponentendiagramm . . . . .	45
3.6.4 Verteilungsdiagramm . . . . .	46
3.7 Webservices . . . . .	46

3.7.1	SOAP . . . . .	47
3.7.2	WSDL . . . . .	48
3.7.3	UDDI . . . . .	49
3.8	Zugriffskontrolle . . . . .	49
3.8.1	Discretionary Access Control (DAC) . . . . .	50
3.8.2	Mandatory Access Control (MAC) . . . . .	50
3.8.3	Role Based Access Control (RBAC) . . . . .	50
<b>4</b>	<b>Analyse</b>	<b>52</b>
4.1	Stakeholder . . . . .	52
4.2	Anforderungsanalyse . . . . .	54
4.2.1	Erhebung der Anforderungen . . . . .	54
4.2.2	Nichtfunktionale Anforderungen . . . . .	55
4.2.3	Funktionale Anforderungen . . . . .	57
4.2.4	Beschreibung der Anwendungsfälle . . . . .	58
4.2.5	Use Case Modellierung . . . . .	79
4.3	Zugriffs- und Rollenkonzept . . . . .	82
4.3.1	Funktionale Rollen . . . . .	82
4.3.2	Regeln für die Zugriffskontrolle . . . . .	83
<b>5</b>	<b>Architekturentwurf</b>	<b>87</b>
5.1	MVC-Modell . . . . .	87
5.2	3-Tier Architektur . . . . .	88
5.3	Datenmodell . . . . .	90
5.4	UML Darstellung . . . . .	91
5.4.1	Aktivitätsdiagramm . . . . .	92
5.4.2	Komponentendiagramm . . . . .	93
5.4.3	Verteilungsdiagramm . . . . .	97
5.5	Systemarchitektur . . . . .	98
5.6	Standards in der Konzeption . . . . .	101
5.7	Sicherheit . . . . .	102
<b>6</b>	<b>Zusammenfassung</b>	<b>105</b>
<b>7</b>	<b>Ausblick</b>	<b>107</b>
	<b>Abbildungsverzeichnis</b>	<b>109</b>
	<b>Tabellenverzeichnis</b>	<b>110</b>
	<b>Abkürzungsverzeichnis</b>	<b>111</b>
	<b>Web Referenzen</b>	<b>113</b>
	<b>Literatur</b>	<b>114</b>

# 1 Einleitung

Für die Tierhaltung und Versorgung ist eine angemessene Dokumentation tiergesundheitsrelevanter Daten unerlässlich. Dabei geht es nicht nur um die einfache und effiziente Verarbeitung von Daten, sondern auch um lückenlose Dokumentation sowie wirtschaftliches Management innerhalb der Tierkliniken und Tierarztpraxen.

In der Nutztierhaltung sind auch Tierhalter zu einer umfassenden Dokumentation verpflichtet, welche im heutigen Zeitalter am Besten digital verwaltet werden und so fälschungssicher und verschlüsselt über das Internet übertragen werden kann.

Ziel ist es, durch die Analyse und den Architekturentwurf die Grundlage zur Entwicklung einer Plattform zu erarbeiten, um damit die Speicherung und Weiterverarbeitung der wichtigen Daten für Tierhalter und Tierärzte mit einer gemeinsamen Schnittstelle zu ermöglichen, die den Dokumentationsaufwand vereinfacht.

## 1.1 Motivation und Problemstellung

Motivation dieser Arbeit ist es, einen Architekturentwurf einer elektronischen Gesundheitsakte für Nutztiere auf der Basis einer umfassenden Analysephase zu entwickeln, da solche integrierten IT-Lösungen derzeit noch nicht existieren.

Weiters kann dadurch eine fachspezifische Nutzung der Dokumentation stattfinden, was bislang nicht der Fall ist. Fachspezifisch bedeutet, dass Rückverfolgbarkeit, Qualitätssicherung und Verbraucherschutz stattfinden kann. Mit einer derartigen Lösung könnte man im Hinblick auf die genannten Nutzungsmöglichkeiten sowie den Entwicklungspotentialen im Bereich der Tierzucht den stetig wachsenden Anforderungen gerecht werden.

Die Mannigfaltigkeit in der tiergesundheitsrelevante Daten gespeichert werden führt dazu, dass weniger umfangreiche Analysen über das Vorkommen von Tierkrankheiten gemacht werden können.

Für die Entwicklung des Architekturentwurfs zur Erfassung einer elektronischen Patientenakte werden durch Befragungen und der gemeinsamen Ausarbeitung mit qualifizierten Personen aus der Nutztierhaltung die Anforderungen und Wünsche an einen elektronischen Gesundheitsakt evaluiert.

Während der Entwicklung müssen verschiedene Bereiche erforscht werden wie z.B. der derzeitige Stand im Bereich elektronischer Tiergesundheitsakten, die Verwendung einheitlicher Formate, die Struktur der Daten die gespeichert und zwischen Systemen übertragen werden, die eindeutige Identifikation von Tieren, Kunden und anderen tiergesundheitsrelevanten Institutionen, um quer übergreifenden Austausch medizinischer Daten zu ermöglichen sowie verschiedener Mechanismen, durch die unauthorisierte Zugriffe auf sensible Informationen verhindert werden können. Daher muss auch großes Augenmerk auf die Analyse von gängigen Standards zur Datenübertragung gelegt werden.

Weiters trägt auch die gesetzliche Lage, die mit einem Schwerpunkt zu berücksichtigen ist, zum Entwurf bei.

## 1.2 Nutzen der elektronischen Gesundheitsakte für Nutztiere

Derzeit ist es so, dass sämtliche Diagnosen und gesundheitsrelevanten Informationen, die Tierärzte während Behandlungen sammeln, in deren internen Praxismanagementsystemen gespeichert werden. Diese Daten sind nur innerhalb der Praxen und Kliniken zugänglich. Wenn ein Tierarzt gewechselt wird, besteht für den neuen Arzt kein Zugang zu einem Akt der bisherigen Krankengeschichte des Tieres. Die Daten sind verstreut und so sind Auswertungen zum Tierbestand schwierig zu erstellen. Daher muss an einer Lösung gearbeitet werden, um den gesundheitlichen Zustand, die bisherige Krankengeschichte und die verabreichten Medikamente an ein Tier lückenlos zu dokumentieren sowie diese Daten zentral gesammelt in einem System verfügbar zu machen. Ein derartiges System würde sowohl zur Verbesserung der Nahrungsmittelqualität sowie zur besseren Überwachung von Krankheiten beitragen, da durch die zentral verfügbaren Daten eine bessere Kontrolle möglich wäre.

## 1.3 Zielsetzung der Arbeit

In der Humanmedizin stellt die elektronische Patientenakte ein wichtiges Thema dar und soll als Hilfsmittel für eine lückenlose Dokumentation der individuellen Krankengeschichte eines Patienten dienen. So können die Informationen wirksam zur medizinischen Versorgung beitragen. Dieses Konzept der elektronischen Akte soll nun auch in der Nutztierhaltung eingeführt werden.

Mit dieser Arbeit wird das Fundament für die Entwicklung erstellt, d.h. explorative Recherche, Analyse und daraus folgender Architekturentwurf werden am Ende das Resultat darstellen.

Inhaltlich werden zwei große Bereiche erarbeitet werden: ein Analyseteil und ein Architekturentwurf, der aufbauend auf den Analyseteil dessen Ergebnisse in Umsetzung bringt.

Die Grundlagenanalyse gliedert sich wiederum in Teilbereiche, die ausführlich Themen und Fragen hinsichtlich der Entwicklung des Entwurfs einer elektronischen Tiergesundheitsakte behandeln. Dabei wird als erstes mittels Analyse die derzeitige Situation in Österreich recherchiert, wobei aufgezeigt wird, ob und welche Umsetzungen in die Richtung einer elektronischen Akte derzeit vorhanden sind.

Die gesetzliche Lage wird ebenfalls analysiert und wichtige Gesetze die z.B. auf Tierarzneimittel und die Tierkennzeichnung eingehen, werden zusammengefasst. Da auch der Datenaustausch zwischen Systemen ein wichtiger Punkt ist, soll durch Analyse der gängigen Standards eine Möglichkeit zur Datenhaltung, dem Datenaustausch und in weiterer Folge das Zusammenfügen der Informationen aus verschiedenen Quellen erarbeitet werden. Standards für z.B. Datenhaltung könnten in veterinärmedizinischer Software in vielen Bereichen Vorteile mit sich bringen. Dadurch wäre der Austausch, der Inhalt sowie die Konsistenz der Daten verbessert, redundante Erfassungen wären stark verringert und es wäre möglich, Verbesserungen in der Patientenversorgung messbar zu machen, um nur einige Vorteile zu erwähnen.

Im praktischen Teil der Arbeit wird der Architekturentwurf der Gesundheitsakte aufbauend auf den bisher gewonnenen Ergebnissen behandelt. Die Anforderungen an einen

Tiergesundheitsakt werden mithilfe qualifizierter Personen aus der Nutztierbranche erarbeitet. Hierbei werden technische Anwendungsfälle der Anforderungen sowie der Subsysteme erarbeitet, die die Lösung aus technischer Sicht darstellen (Datenstrukturen, Datenflüsse, etc).

Da sensitive Daten nicht von jeder Person eingesehen werden dürfen, wird auch ein Zugriffs- und Rollenkonzept im Zusammenhang mit Datensicherheit entwickelt. Das im Rahmen und als Ziel dieser Diplomarbeit entstehende Designdokument soll als Grundlage für eine weiterführende Implementierung dienen.

## **1.4 Gliederung der Arbeit**

Nach der Einleitung werden in Kapitel 2 die theoretischen Grundlagen behandelt insbesondere der aktuelle Stand über derzeit am Markt verfügbare veterinärmedizinische Software sowie eine Beschreibung der elektronischen Gesundheitsakte für Menschen. Weiters folgt ein Überblick über die gesetzliche Lage und eine Vorstellung diverser vorhandener Standards, die im medizinischen Bereich Anwendung finden.

In Kapitel 3 werden dem Leser die theoretischen Grundlagen der Softwarearchitektur vermittelt, die für das Verständnis dieser Arbeit vonnöten sind. Weiters werden Architekturmuster wie das Model-View-Controller-Pattern oder objektrelationale Abbildungen in Datenbanken beleuchtet. Da UML derzeit die dominierende Sprache für Softwaresystem-Modellierung ist, wird auf die Sprache selbst, sowie überblicksweise auf die verschiedenen Diagrammart eingegangen. Den Abschluss dieses Kapitels bildet eine Einführung in Webservices sowie ein theoretischer Einblick in die unterschiedlichen Zugriffskonzepte.

In Kapitel 4 werden Anforderungen analytisch dargestellt und ein Rollenkonzept entwickelt.

Abschnitt 5 behandelt den praktischen Teil dieser Arbeit. Dabei wird die Konzeption des Systems betrachtet und die im vorigen Kapitel vorgestellten Methoden der Softwarearchitektur finden Anwendung. Den Abschluss bildet eine Diskussion über die Sicherheit dieses Systems.

In den letzten beiden Kapiteln wird eine Zusammenfassung mit Ausblick geliefert.

# 2 Grundlagen aus der medizinischen Informatik

In diesem Kapitel werden derzeit existierende Lösungen auf dem Gebiet der elektronischen Gesundheitsakte für Nutztiere analysiert, die gesetzlichen Grundlagen beleuchtet, sowie Standards, die bei der Umsetzung zur Verwendung kommen könnten, untersucht. Unter Verwendung der gewonnenen Erkenntnisse und Ergebnisse wird die erforderliche Funktionalität der elektronischen Gesundheitsakte (ELGA) für Nutztiere abgeleitet.

## 2.1 State of the Art

In der Praxis gibt es noch keine Realisierung der elektronischen Gesundheitsakte für Tiere. Es existieren verschiedene Praxismanagement Software Lösungen, die für den veterinärmedizinischen Bereich konzipiert wurden und auch von Ärzten und Kliniken verwendet werden. Diese Lösungen spezialisieren sich nicht nur mehr oder weniger auf die Erfassung gesundheitsrelevanter Daten sowie Möglichkeiten zum Datenaustausch, sondern auch auf die wirtschaftlichen Bereiche im Praxisalltag, wie z.B. Praxisverwaltung und -organisation, Rechnungswesen und Buchhaltung.

Wie in der Arbeit von C. Egger-Danner [20] beschrieben, sind vorhandene Lösungen vorwiegend Insellösungen und integrierte IT-Lösungen für Tierhalter und Tierärzte existieren derzeit noch nicht.

In der Arbeit von R.M. Bastos und D.D.A. Ruiz [8] heißt es, dass in den meisten Fällen Daten in sehr spezifischer Form gesammelt werden, somit ist die Form dieser Daten mit aus anderen Quellen stammenden Daten inkompatibel. Weiters wird beschrieben, dass in vielen Instanzen Daten auch noch immer in Papierform abgelegt werden, was den Zugriff auf solche Dokumente erschwert. Der Bedarf an einer IT-Lösung, die standardisiert die Speicherung, den Zugriff und den Austausch von Daten ermöglicht, ist sehr groß.

In der IBM Machbarkeitsstudie zum Elektronische Gesundheitsakte (ELGA) Projekt [30] heißt es, dass gerade durch die Digitalisierung der Daten verschiedener Gesundheitseinrichtungen elektronische Information die Papierform ersetzt. Da jedoch die Datenformate der Institutionen oft unterschiedlich sind, gestaltet sich der Austausch als schwierig. Die Verwendung von internationalen Standards wie Health Level 7 (HL7) erscheint daher als die beste Lösung. Die Viehwirtschaft kann ungemein von Systemen profitieren, die lückenlos die gesundheitsrelevanten Daten des Tierbestandes verwalten. Dadurch kann eine hohe Fleischqualität garantiert werden und sich ausbreitende Krankheiten können rasch erkannt werden [65].

### 2.1.1 Elektronische Gesundheitsakte

ELGA steht für den österreichischen Projektnamen der elektronischen Gesundheitsakte und ist international unter Electronic Health Record (EHR) bekannt. Bei einem EHR hat der Arzt oder das Krankenhaus die Kontrolle über die Informationen sowie die Möglichkeit, diese einzusehen. Diese Daten sind daher auch beim Arzt oder beim Krankenhaus elektronisch gespeichert. Eine weitere spezielle Form eines Health Record wäre der Personal Health Record (PHR), bei der die Informationen in der Hand vom Patienten liegen. Diese Informationen können entweder nur auf Papier vorhanden oder elektronisch gespeichert sein. Bei der elektronischen Speicherung sind die Daten meistens über eine Webseite zugänglich und Ärzte können über diese Seite neue Daten hinzufügen.

Seit einigen Jahren wird hierzulande intensiv an der Einführung einer ELGA gearbeitet, die sämtliche medizinisch relevanten Daten und Befunde eines Patienten enthält und für Patienten und Ärzte mit Berechtigung orts- und zeitunabhängig am Ort der Behandlung abrufbereit zur Verfügung steht [57].

Dazu wurde im November 2009 die ELGA GmbH gegründet [23]. Unternehmensgegenstand der ELGA GmbH ist:

„die nicht auf Gewinn gerichtete Erbringung von im Allgemeininteresse liegenden Serviceleistungen auf dem Gebiet der Daseinsvorsorge im Bereich von e-Health zur Einführung und Implementierung der elektronischen Gesundheitsakte (ELGA)“. [17]

Da es sich bei der Umsetzung der ELGA um personenbezogene Daten handelt, muss dem Datenschutz hohe Priorität eingeräumt werden.

Zu den Aufgaben der ELGA GmbH gehört [23]:

- Die Planung, Überwachung und Evaluierung der technischen und organisatorischen Maßnahmen.
- Die Festlegung und Überwachung der Einhaltung von Sicherheitsanforderungen.
- Die Erarbeitung der Qualitätsanforderungen für alle Daten, Dokumente, Prozesse und Softwareprodukte, die im Rahmen der ELGA Anwendung finden.
- Sicherstellung der reibungslosen Interoperabilität zwischen den Systemen.
- Interessenausgleich bei der Umsetzung und Problemlösungsstrategien.

Im Jahr 2006 wurde von IBM Österreich im Auftrag der Bundesgesundheitsagentur eine Machbarkeitsstudie [24] durchgeführt. Aus dieser Studie wurden die Basiskomponenten sowie die Kernanwendungen für die Umsetzung als Ergebnisse gewonnen. Zu den Basiskomponenten gehören:

- **Patientenidentifikation und Patientenindex:**  
Die eindeutige Zuordnung von Dokumenten zu Patienten erfolgt über die eindeutige Kennung einer Person, die durch den Patientenindex dargestellt wird.
- **Gesundheitsdiensteanbieterindex:**  
Die Gesundheitsdiensteanbieter werden ebenfalls durch eine eindeutige Kennung identifiziert.

- **Berechtigungssystem**  
Dadurch wird der unerlaubte Zugriff von unbefugten und der erlaubte Zugriff von befugten Personen auf die Inhalte der ELGA geregelt.
- **Dokumentenregister und Dokumentenspeicher:**  
Das Dokumentenregister beinhaltet ein Verzeichnis der vorhandenen Dokumente, wobei nur die Metadaten gespeichert sind. Durch einen Link gelangt man zum tatsächlichen Dokument. Diese dezentrale Speicherung ermöglicht es, dass effizient und performant nach Dokumenten gesucht werden kann. Die Speicherung der Daten erfolgt in den dezentralen Dokumentenspeichern.
- **Portal:**  
Das Portal stellt den Patienten und Gesundheitsdiensteanbietern den zentralen Zugang zu den gesundheitsrelevanten Informationen zur Verfügung.

Die Kernanwendungen gliedern sich folgendermaßen:

- **e-Medikation:**  
Bei dieser Anwendung werden Abläufe von der Verordnung bis zur Medikamentenausgabe unterstützt um dadurch in weiterer Folge medizinisch unerwünschte Arzneimittel-Wechselwirkungen und Mehrfachverschreibungen zu vermeiden sowie die korrekte Umsetzung der Medikationstherapie zu unterstützen.
- **e-Befund:**  
Der e-Laborbefund stellt elektronisch alle Ergebnisse aus Laboruntersuchungen dar. Der e-Radiologiebefund unterscheidet sich vom e-Laborbefund dadurch, dass hier Daten aus Radiologischen Untersuchungen bereitgestellt werden.
- **e-Patientenbrief:**  
Der e-Patientenbrief ist die elektronische Abbildung des Patientenbriefes.

Aus der Arbeit von A. Ströher „Die elektronische Gesundheitsakte (ELGA) in Österreich - Eine Evaluierung in Bezug auf funktionale Benutzeranforderungen“ [57] geht hervor, dass aus Patientensicht der Wunsch und die Notwendigkeit einer elektronischen Gesundheitsakte besteht. Weiters müssen über ELGA beziehbare Informationen valide sein. Jedoch sind Bedenken, ob die Einführung von ELGA zu einem „gläsernen Patienten“ führt, auszuräumen. Zum Thema Datenschutz sei an dieser Stelle auf die Arbeit von [66] verwiesen, wo der Autor viele Fragen im Zusammenhang mit der Modernisierung und Automation des Gesundheitswesens durch die Diskussionen mit Ärzten und Patienten thematisiert.

**Technische Grundlagen** Bei der Entwicklung der ELGA werden Standards zur Umsetzung der Information und Kommunikation eingesetzt [17]. Auf die verschiedenen Standards wird in Abschnitt 2.3 näher eingegangen. Verwendet wird unter anderem das IHE Framework [31], welches spezifische Implementierungen von etablierten Standards definiert, um dadurch eine effektive Systemintegration zu erreichen. Weiters ermöglicht es eine angemessene Verteilung der medizinischen Informationen und unterstützt eine optimale Patientenversorgung.

Als Dokumentenstandard kommt HL7 zum Einsatz, welcher in Abschnitt 2.3.1 vorgestellt wird. Es ergibt sich ein rasches und einfaches Lesen der Dokumente, die auf Basis dieses Standards gestaltet wurden.

Der verwendete Standard für Begrifflichkeiten ist Logical Observation Identifiers Names and Codes (LOINC), siehe Abschnitt 2.3.2. Dabei handelt es sich um eine Sammlung von allbekannten Identifikatoren zur Bezeichnung von Untersuchungs- und Testergebnissen aus dem medizinischen Bereich, um eine Erleichterung des elektronischen Datenaustausches zu bieten.

Für die Speicherung, den Austausch und das Bilddatenmanagement kommt Digital Imaging and Communications in Medicine (DICOM) zum Einsatz, näheres dazu in Abschnitt 2.3.3. All diese Funktionen, also sowohl das Format zur Datenspeicherung wie auch das Kommunikationsprotokoll, sind von DICOM standardisiert.

## 2.1.2 Derzeit am Markt vorhandene Software

Im Folgenden werden drei der am Markt verfügbaren Praxismanagement Software Lösungen aus dem veterinärmedizinischen Bereich vorgestellt. Diese drei wurden ausgewählt, da sie sich bei einer Befragung im Rahmen der Diplomarbeit „Erhebung und Analyse der Anforderungen an einen Animal Health Record (AHR) für Kleintiere“ von Viktoria Willner als die drei am häufigsten in Tierkliniken und Tierarztpraxen zum Einsatz kommenden Systeme herausgestellt haben. Ob und welche Schnittstellen diese Systeme haben und wie die genaue technische Umsetzung aussieht, konnte aufgrund der vorliegenden Informationen nicht beurteilt werden. Daher findet sich nachfolgend nur eine Beschreibung der Funktionalitäten zusammen mit einigen visuellen Eindrücken. Im Falle von VetInf 2.1.2.2 kann jedoch davon ausgegangen werden, dass es eine Webservice Schnittstelle gibt, um Daten des Systems abzugreifen.

Nach diesen Systemen folgt ein Eindruck über spezielle Systeme aus der Rinderzucht.

### 2.1.2.1 easyVET

easyVet [15] wurde von der VetZ GmbH entwickelt, die ihren Sitz in Deutschland hat und zu einem der führenden Anbieter von Software in der Tiermedizin gehört. Das Unternehmen entwickelte verschiedene Lösungen abgesehen von easyVET wie z.B.

- **easyIMAGE:**  
Hierbei handelt es sich um eine Bildverarbeitungssoftware, ebenfalls für den tiermedizinischen Bereich und ermöglicht zusammen mit easyVET eine effiziente Bildverarbeitung für alle Bereiche der bildgebenden Diagnostik.
- **XDR1:**  
Ein System für digitales Röntgen, welches durch integrierte Flachdetektoren von Canon sowohl stationär als auch mobil funktioniert.
- **easyPAGE:**  
Mit easyPAGE werden Leistungen wie individuell zugeschnittenes Design, Programmierung und Wartung angeboten, um Tierarztpraxen und Tierkliniken einen professionellen Internetauftritt zu ermöglichen.

- vetsXL.com:  
Ein veterinärmedizinisches Webportal, welches Ärzten, Universitäten und Unternehmen aus diesem Bereich weltweit zur Verfügung steht und dadurch Kommunikation mit Kollegen, Auswertungen und Analysen sowie Laborergebnisse online und Bildupload gestattet.

easyVet wurde nicht nur für den Kleintier- und Pferdebereich entwickelt, sondern kommt auch in Nutztierpraxen zum Einsatz. Mit diesem System können verschiedene Aufgaben die im Klinik- und Praxisalltag anfallen erledigt werden. Zu diesen Aufgaben gehören:

- die Verwaltung diverser Daten (Kunden, Tiere, Behandlungen):  
Dies erfolgt durch den elektronischen Karteikasten, welcher übersichtlich - angelehnt an den üblichen Karteikasten - Informationen darstellt. Die Darstellung im System erfolgt in einem dreigeteilten Fenster, wobei in den Karteien im oberen Drittel die Kunden erfasst sind. Sobald man einen auswählt, werden in den Karteien des zweiten Drittels die Tiere des obigen Kunden angezeigt. Wird in weiterer Folge ein Tier ausgewählt, werden im unteren Karteikasten alle dazugehörigen Behandlungen angezeigt. Zusammen mit easyIMAGE kann dazugehöriges Bild- und Filmmaterial direkt in der Karteikarte verwaltet werden.
- Behandlung und Befundung:  
Mit Hilfe von Standardbehandlungen, die der Benutzer definiert, können sich wiederholende, gleichbleibende Behandlungen inhaltlich festgelegt werden, d.h. es werden Leistungen und zu verabreichende Medikamente während und nach der Behandlung erfasst. Standardisierte Befundbögen dienen zur Vereinfachung der Dokumentation.
- Praxismanagement mit Übersicht über die Patienten im Wartezimmer, den Terminkalender und interne Kommunikation
- Buchhaltung und Abrechnung:  
Durch eine eingebaute Finanzzentrale können Zahlungseingänge, Behandlungsumsätze und Rechnungen erfasst, dargestellt und analysiert werden. Durch die integrierte Buchhaltung wird keine zusätzliche Software benötigt.
- Medikamentenverwaltung und Medikamentenverzeichnis:  
Für jedes Arzneimittel können die Lagerbestände verwaltet und Abgabe- und Anwendungsbelege angelegt werden.
- Funktionen zur Erstellung von Analysen und Auswertungen uvm.:  
Vordefinierte Statistiken und Filter ermöglichen eine Vielzahl an Möglichkeiten um den Erfolg der Praxis respektive der Klinik grafisch darzustellen.

Die folgenden zwei Abbildungen zeigen zwei Screenshots der Software, die der Homepage der VetZ GmbH entnommen wurden, ebenso wie die Informationen über die Software selbst [15]. Abbildung 2.1 zeigt den sogenannten „Karteikasten“ der Software. In der oberen Kartei wird der Kunde ausgewählt, die mittlere Kartei wählt das Tier des Kunden aus und der untere Bereich die Behandlung, die ausgeführt wurde.

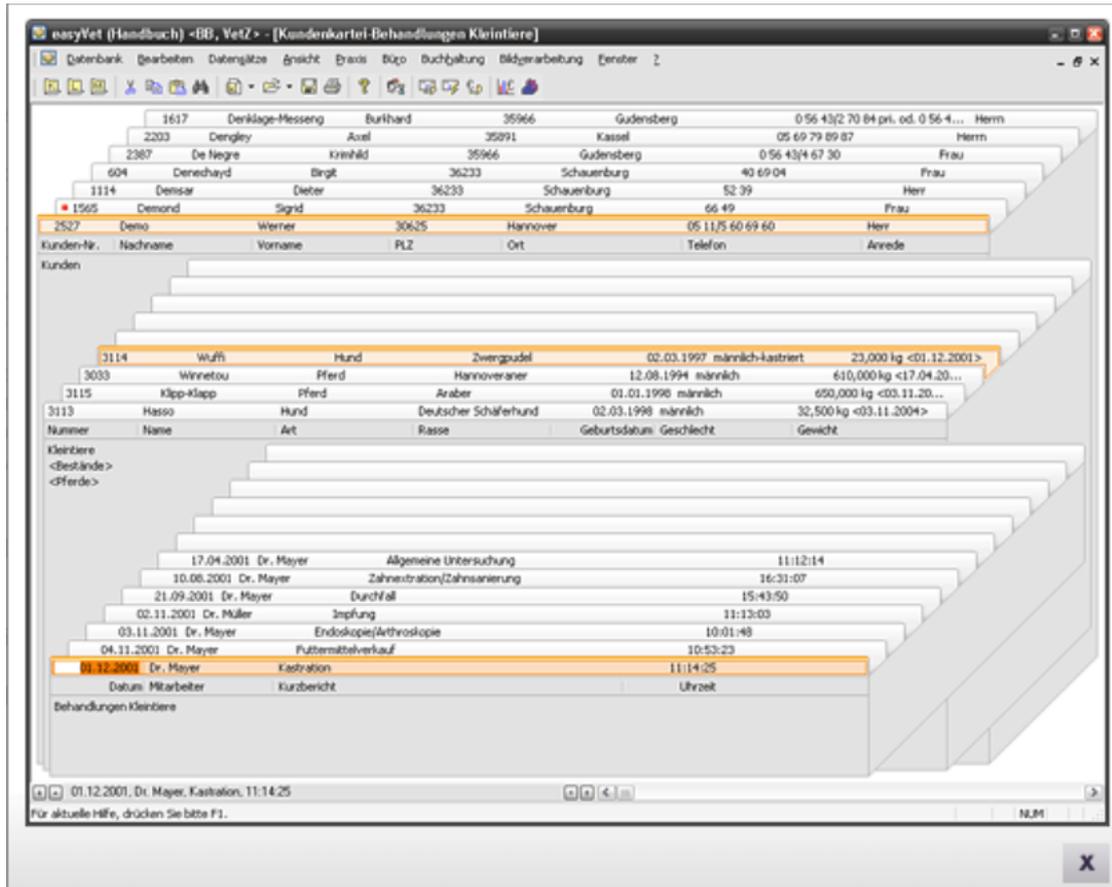


Abbildung 2.1: easyVet Karteikasten Ansicht

Abbildung 2.2 zeigt die Maske einer Behandlung für den Kunden und das Tier, die über die Karteikastenmaske ausgewählt wurden.

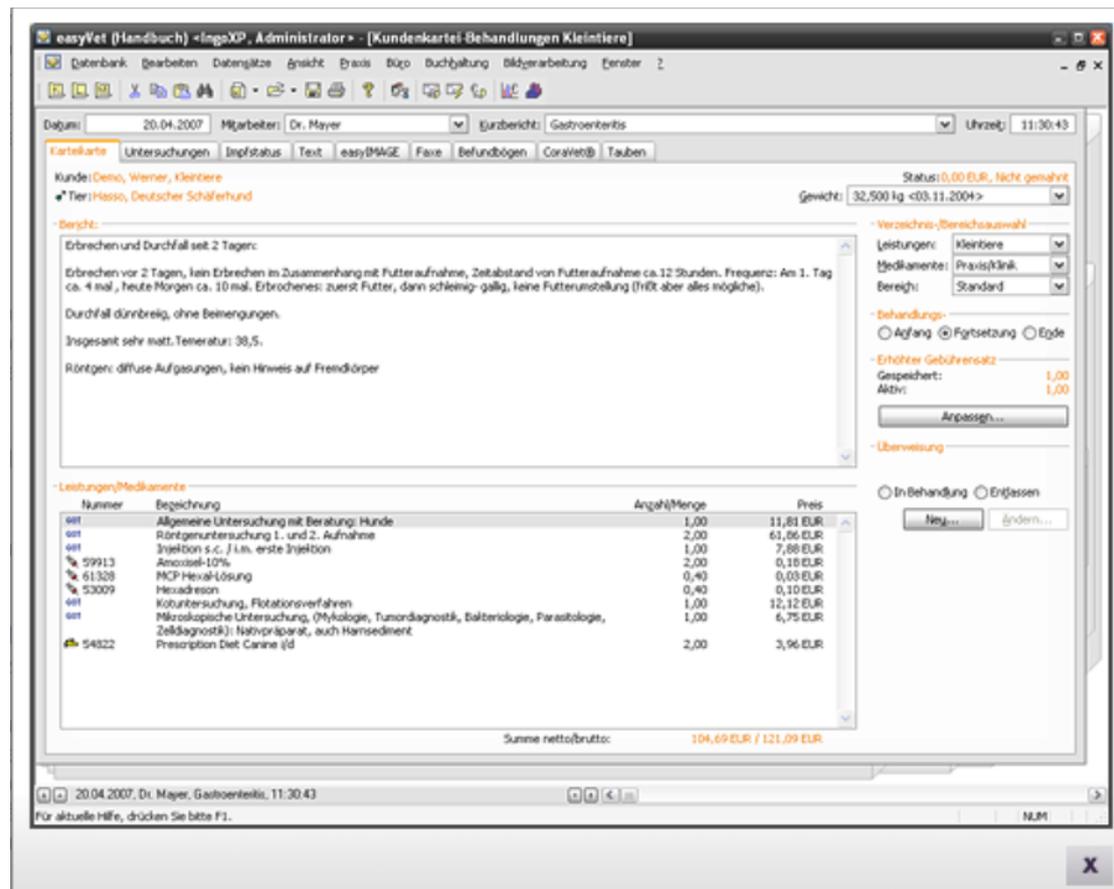


Abbildung 2.2: easyVet Ansicht einer Behandlung

Eine Demo von easyVET steht dem Leser unter [6] zur Verfügung.

### 2.1.2.2 Vetinf

Vetinf [63] kommt als Praxisverwaltungssoftware in Tierpraxen und -kliniken in Österreich, Deutschland, Spanien und Luxemburg zum Einsatz und wurde von der Vetinf GmbH - einem Tierarztsoftware spezialisierten Unternehmen - in Deutschland entwickelt und zählt dort zu den Marktführern.

Vetinf umfasst folgende, in veterinärmedizinischen Kliniken und Praxen benötigte Bereiche:

- Datenverwaltung von Patient und Tier,
- Behandlung und Befundung,
- Medikamentenverzeichnis,
- Buchhaltung und Abrechnung,
- Funktionen zur Erstellung von Analysen.

Bei Vetinf kann je nach Anforderungen der Praxis aus drei Programmvarianten und zwei Zusatzmodulen gewählt werden.

**Vetinf Basis** bietet ein Grundpaket an Funktionen für Standardanforderungen an, wie z.B. die Verwaltung der Patientendaten, Dokumentation der Behandlung und das Rechnungswesen.

**Vetinf Standard** enthält zusätzliche Funktionen wie z.B. eine Apothekenverwaltung, Schnittstellen zu diversen Laborgeräten, die Einbindung von Laborbefunden und Analyse- und Auswertungsmöglichkeiten.

**Vetinf Komplett** ermöglicht die mobile Verwendung der Software am Laptop und beinhaltet weiters eine umfassende Buchhaltungslösung.

**Zusatzmodul Vetinf Mobil** kann einzeln erworben und zusammen mit Vetinf Basis oder Vetinf Standard betrieben werden. Es ermöglicht die Eingabe von Behandlungsdaten von unterwegs sowie die Anbindung digitaler Bildsoftware.

**Zusatzmodul Vetinf FiBu** kann ebenfalls einzeln erworben und zusammen mit Vetinf Basis oder Vetinf Standard betrieben werden und integriert ein Finanzbuchhaltungs-Modul in das System.

Weiters werden ergänzend Schnittstellen zu digitalen Bildverarbeitungsprogrammen für Röntgen- oder Ultraschallaufnahmen angeboten. Abbildung 2.3 zeigt einen Screenshot der Software, in dem ein Kunde zusammen mit dem zu behandelnden Tier und seinen erfassten Behandlungen zu sehen ist. Die Abbildung sowie die Informationen über die Software wurden aufgrund der Informationen auf der Homepage der VetInf GmbH [63] zusammengetragen.

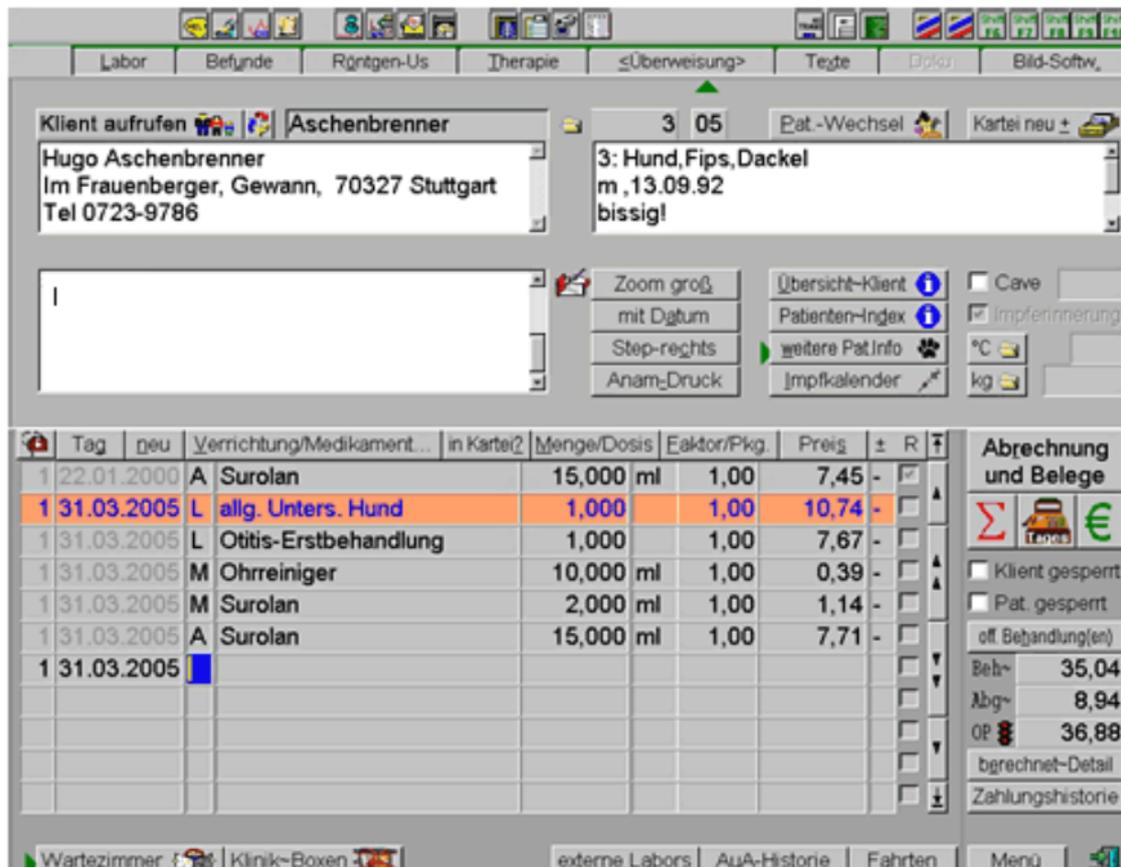


Abbildung 2.3: VetInf Ansicht einer Behandlung

### 2.1.2.3 ANIMAL-office

Als letzte zu erwähnende Softwarelösung wird an dieser Stelle ANIMAL-office [2] vorgestellt. Hierbei handelt es sich um ein Produkt der SEG Informationstechnik GmbH mit Sitz in Österreich.

ANIMAL-office ist eine Praxismanagementsoftware, die Funktionen für den Betrieb in einer auf Kleintiere spezialisierten Praxis bereitstellt. Weiters steht ein Großtiermodul bereit, welches bei Bedarf in das System integriert werden kann.

**ANIMAL-office Kleintier** Mithilfe von Barcodescannern werden die Tiere bzw. die Besitzer identifiziert und das Stammdatenblatt im System aufgerufen. Die Software hilft dem Benutzer bei der Erstellung der Formulare in höchster Qualität und erlaubt weiters eine Automation der weiteren Datenverarbeitung.

Administrative und koordinative Funktionen helfen Praxisabläufe einfach und übersichtlich zu gestalten. Tägliche Routinearbeiten wie Terminvereinbarungen oder Einteilung von Ressourcen können vom Benutzer erledigt werden. Behandlungskosten und entsprechende Rechnungen werden mit der integrierten Abrechnungsfunktion bearbeitet. In weiterer Folge besteht die Möglichkeit, erforderliche Dokumente im Zusammenhang mit der Behandlung auszustellen.

Abbildung 2.4 zeigt die Maske, in der die Definition der Stammdaten eines Tieres erfolgt.

The screenshot shows the 'ANIMAL Office' software interface for a patient named Cindy. The window title is 'Patienten'. The interface is divided into several sections:

- Klient:** Musterkunde Gerhard, Hk., 1419, 4820 Bad Ischl, Lindestraße 3
- Patient:** Cindy, Kleiner Münsterländer, 2437
- Kollegen:** Mustermaier, Hilfszoo Chary, 9.090,02 U, 28,50 O

Navigation tabs include: Allgemein, Aktionen, LAB, CHECK, DOKU, Info, Termine, Stall, Bilder, Futter.

**General Information:**

- Name: Cindy
- Rasse: Kleiner Münsterländer HUNDE
- Fellfarbe: Grau
- Geschlecht: weiblich
- Gewicht: 25,000 kg,  kastriert
- geboren: 4.10.1990, 15 Jahre, 1 Monat
- verstorben: 74 KB

**Identification and Breeding Data:**

- ChipNr./Ohrenmarke: 7F7F2F0D1E
- Heimtierausweis: 0400001244
- Datum Chipping: 01.05.1991
- Sohle Chipping: Inke Halbsitz
- Zwingername: [empty]
- Züchter: Hannes Leitner
- Abzeichen: [empty]

**Actions and Media:**

- Nächster Klient: [empty]
- Bestitzerwechsel: [empty]
- SMS: [empty]
- Email: [empty]
- Foto: [Image of a dog's head]
- Foto laden: [button]
- Foto-Detaile: [button]
- Tier bei: [dropdown menu]
- registrieren: [button]

Abbildung 2.4: Animal Office Kleintier Stammdatenblatt

**ANIMAL-office Großtier** Das Großtiermodul von ANIMAL-office stellt dem Tierarzt eine Fülle wichtiger Informationen zur Verfügung, um eine optimale Bestandsbetreuung zu gewährleisten.

Die Anpaarungshilfe wird dem Tierarzt durch die integrierte Zuchtwertschätzung in Kombination mit verschiedenen Auswertungen der Leistungsdaten zur Verfügung gestellt. Mithilfe der Daten, die während einer Behandlung gewonnen werden, kann zusammen mit den Zuchtwerten die Besamung mit entsprechenden Aktionen in der Bestandsbetreuung aktiviert werden.

Abbildung 2.5 zeigt die Anpaarungshilfe des Großtierpraxen.

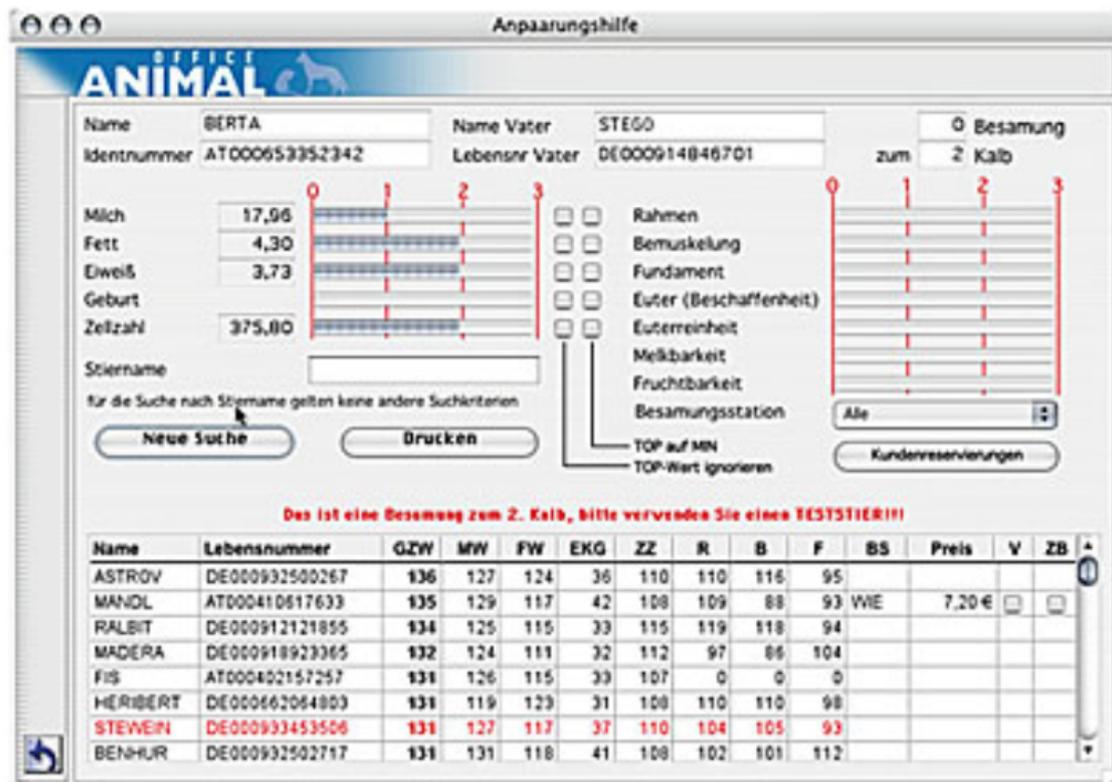


Abbildung 2.5: Animal Office Großtier Anpaarungshilfe

ANIMAL-office optimiert die Durchführung verschiedener Aufgaben für Tierärzte, die im Rahmen des Tiergesundheitsdienst (TGD) [59], auf den in Abschnitt 4.1 noch näher eingegangen wird, erledigt werden müssen. Dazu gehört z.B. eine regelmäßige Betriebserhebung bei den Landwirten oder die Datenübertragung zum TGD über das Internet. Weiters kümmert sich die Software automatisch um Abgabebelege, die im Anschluss an eine Behandlung durchgeführt werden sowie um die vom TGD verlangte Leergutverwaltung für abgegebene Medikamente.

Die Screenshots sowie Informationen über das System wurden aufgrund der Informationen, die auf der Homepage von ANIMAL-office [2] zur Verfügung stehen, zusammengetragen.

#### 2.1.2.4 Rinderdatenbank der Agrarmarkt Austria

Seit 1. Oktober 1999 existiert in Österreich eine zentrale Rinderdatenbank, die von der Europäischen Union beschlossen und von der Agrarmarkt Austria (AMA) umgesetzt wurde. Das Ziel dieses Systems ist es, eine bessere Rückverfolgbarkeit des Rindfleischs zu gewährleisten und zur Seuchenbekämpfung beizutragen. Diese Datenbank steht den Veterinären online über das Internetserviceportal eAMA zur Verfügung [14]. Die Benutzer haben somit die Möglichkeit, direkt diverse Verwaltungsabläufe wie Anträge, Meldungen und Abfragen durchzuführen.

In Österreich ist jeder Rinderhalter dazu verpflichtet, innerhalb einer Frist von sieben Tagen jede Geburt, jeden An- und Verkauf, jede Schlachtung sowie jeden Abgang an

die zentrale Rinderdatenbank der AMA zu melden. Die Datenbank dient weiters als Datengrundlage für Viehbestandserhebungen sowie für die Rinderzucht. [1]

### 2.1.2.5 Rinderdatenverbund für Mitglieder - RDV4M

Im Jahr 2000 wurde das Online-System des Rinderdatenverbundes realisiert, welches mehr als zehn Millionen Rinder umfasst und als Schnittstelle für die Landwirte dient. Die Zentrale Arbeitsgemeinschaft österreichischer Rinderzüchter (ZAR) gründete am 11. April 2001 die 100% Tochter ZuchtData EDV Dienstleistungen Ges.m.b.H., die die Verwaltung des Systems übernimmt. Der Rinderdatenverbund für Mitglieder (RDV4M) ermöglicht den Mitgliedern der Landeskontrollverbände (siehe Abschnitt 4.1) online selbstständig alle Daten der Herde abzufragen. Ein Teil der Daten, die im System erfasst sind, kommt von der AMA über eine Datenbankschnittstelle wie z.B. Geburtsmeldungen, Ohrmarkenmeldungen usw. Ein weiterer Teil kommt über die Organe der Kontrollverbände wie Probemelkungen und andere ausgewertete Daten. Landwirte können selber im System keine Daten eingeben. Seit kurzem können sie jedoch Aktionen und Beobachtungen vom eigenen Herdebestand erfassen. Wenn beispielsweise eine Geburt beobachtet wird, wird das zwar in der Datenbank gespeichert, jedoch in separaten Strukturen, die nicht zu den „Echtdaten“ gehören. Es handelt sich dabei um reine Beobachtungen, die als solche abgelegt und ausgewertet werden.

Die Landwirte erhalten auf Grund der erhobenen Daten und Kontrollwerte vom Landeskontrollverband (LKV) Berichte über die Herde zugeschickt (z.B. täglich oder jährlich...).

Das System besteht aus folgenden Programmteilen [52]:

- RDV4M

Dieser Programmteil setzt sich aus folgenden Modulen zusammen:

- Modul Betrieb  
Dieses Modul liefert verschiedene Informationen über die Herde, bisher erbrachte Leistungen, Verendungen sowie Zuchtwerte der Tiere.
- Modul Tier  
Das Tiermodul stellt Informationen über Abstammung, Laktationen, Kalbungen, Belegungen und Fleischleistungen jedes Tieres dar.
- Modul Grafik  
Ergebnisse der Probemelkungen der Kontrolljahre werden durch Grafiken in diesem Modul visualisiert.
- Modul Admin  
Dieses Modul ermöglicht die Einstellung verschiedener Grenzen und Einschränkungen, um die Anzeige einiger Ansichten in anderen Modulen zu verändern.

- Fütterung

Dieses Service bietet den Landwirten die Möglichkeit, Rationsberechnungen auf Basis der Leistungsdaten der Herde durchzuführen und dadurch ihre Fütterung zu optimieren.

- Anpaarung  
Aus einem Stierpool werden optimale Besamungsvorschläge für die Kühe des Betriebes ermittelt, wobei wichtige Merkmale wie etwa Milch, Fleisch, Fitness oder Exterieur beachtet werden.

Das System integriert derzeit neben Österreich auch Bayern, Schleswig-Holstein und Baden-Württemberg und verwaltet ca. 65.000 Betriebe mit über 2 Millionen Tieren.

#### **2.1.2.6 Gesundheitsmonitoring Rind**

Da in Skandinavien bereits seit 1975 Tiergesundheitsdaten erfasst werden und dadurch auch Rückgänge bei verschiedenen Krankheiten wie z.B. Mastitis verzeichnet werden konnten, entschied man sich auch in Österreich das Forschungsprojekt Gesundheitsmonitoring Rind umzusetzen.

Durch die Zusammenarbeit verschiedener Organisationen wie Rinderzucht, Veterinärmedizin, Leistungsprüfung, Wissenschaft, Interessensvertretern und der Ministerien wurde dieses Projekt in den Jahren 2006 bis 2010 umgesetzt. Die vom LKV erfassten Tiergesundheitsdaten werden in der Zucht und im Herdenmanagement berücksichtigt [16].

Es werden Diagnosedaten, Tierdaten, Betriebsnummer und die Nummer des behandelnden Tierarztes im Verwaltungsprogramm des Rinderdatenverbundes vom Landeskontrollverband erfasst. Die Diagnosedaten von Arzneimittelbelegen werden durch die Angabe eines zweistelligen Codes der Diagnose, der vom Tierarzt eingetragen wird, erfasst. Dafür wird ein landesweit gültiger Diagnoseschlüssel verwendet.

Diagnosedaten zusammen mit Daten aus der Leistungsprüfung helfen dabei, Probleme im Tierbestand rechtzeitig zu lokalisieren. Sofern diese Daten den Tierärzten zur Verfügung stehen, können krankheitsvorbeugende sowie krankheitsbekämpfende Maßnahmen durchgeführt werden. Für die Übermittlung dieser Daten ist die Zustimmung des Landwirtes notwendig [51].

#### **2.1.3 Die Gesundheitsakte in der Nutztierbranche**

Die Entwicklung einer zentral abrufbaren elektronischen Gesundheitsakte für Nutztiere, die die gesamte Krankengeschichte eines Tieres dokumentiert, existiert derzeit noch nicht. Es gibt bereits in der Rinderbranche in diese Richtung gehende Projekte, wie z.B. das Gesundheitsmonitoring (Abschnitt 2.1.2.6), jedoch werden dort nur Diagnosedaten erfasst, ohne dazugehörige Arzneimitteldaten.

Bei anderen Nutztierarten, wie z.B. Schweinen, Schafen oder Ziegen existieren auf diesem Gebiet noch keine Umsetzungen. Je nach Art der Tiere müsste bei der Umsetzung der Akte berücksichtigt werden, ob die Identifikation und Dokumentation gesundheitsrelevanter Daten eines einzelnen Tieres möglich ist, oder ob nach Box, Stallnummer, Teich oder Stock unterschieden werden muss.

Es wäre denkbar, die elektronische Gesundheitsakte für die Nutztiere Rind, Geflügel, Schwein, Schafe, Ziegen, Bienen und Fische zu entwickeln.

## 2.2 Gesetzliche Lage

In diesem Kapitel werden einige Auszüge aus verschiedenen Gesetzen zitiert, die für die Umsetzung der elektronischen Gesundheitsakte für Nutztiere relevant sind. Dabei geht es um Vorgaben und Pflichten, die von Tierärzten und Landwirten erfüllt werden müssen. Es wurden die wichtigen greifenden Gesetze und Verordnungen analysiert, wobei die zitierten Paragraphen einen Eindruck über die Rahmenbedingungen schaffen sollen. Im Zuge weiterer Bestrebungen bei der Umsetzung der elektronischen Gesundheitsakte besteht legislativer Klärungs- und Handlungsbedarf bei der Prüfung bzw. den Voraussetzungen hinsichtlich der Themen in Bezug auf verpflichtender Einträge und der Verfügbarkeit der gespeicherten Daten.

### **Tierarzneimittelkontrollgesetz 2002**

§ 4a. (1) „Der Tierarzt hat über das Datum der Untersuchung der Tiere, Name und Anschrift der Tierhalter, die Angaben zur Identität und Anzahl der behandelten Tiere, die Diagnose, die verschriebenen Tierarzneimittel, Anwendungsart, die verabreichte Dosis, die Behandlungsdauer und die einzuhaltenden Wartezeiten in geeigneter Weise Buch zu führen. Die betreffenden Unterlagen sind mindestens fünf Jahre lang aufzubewahren und auf Verlangen der Behörde oder des Aufsichtsorgans (§ 9 Abs. 2) zur Kontrolle vorzulegen.“

(2) „Der Tierarzt hat alle an den Tierhalter abgegebenen Arzneimittel mit einer Signatur auf dem Behältnis zu versehen, auf der Name und Anschrift des Tierarztes sowie das Abgabedatum vermerkt sein müssen. Außerdem hat der Tierarzt für alle an den Tierhalter abgegebenen Tierarzneimittel einen Abgabeschein auszustellen, auf dem Art und Menge des Tierarzneimittels, Name und Anschrift des Tierarztes sowie das Abgabedatum vermerkt sind...“

§ 7. (2) „... Im Rahmen dieser Tiergesundheitsdienste darf der Tierarzt den Tierhalter in Hilfeleistungen, welche über die für die übliche Tierhaltung und Tierpflege notwendigen Tätigkeiten hinausgehen, sowie in die Anwendung von Arzneimitteln bei landwirtschaftlichen Nutztieren einbinden, wenn dies unter genauer Anleitung, Aufsicht und schriftlicher Dokumentation von Art, Menge und Anwendungsweise erfolgt. ... Die Dokumentation ist vom Tierarzt mindestens fünf Jahre lang aufzubewahren und der Behörde auf Verlangen vorzulegen...“

§ 8. (1) „...Tierärzte gemäß 3 4a des Tierärztegesetzes sowie öffentliche Apotheken sind verpflichtet, über den Verkehr mit Tierarzneimitteln genaue Aufzeichnungen zu führen. Zu jedem Wareneingang und Warenausgang sind folgende Angaben festzuhalten:

1. Datum,
2. Bezeichnung des Tierarzneimittels,
3. Chargennummer,
4. eingegangene oder ausgelieferte Menge,
5. Name und Anschrift des Lieferanten oder Empfängers,

6. gegebenenfalls Name und Anschrift des verschreibenden Tierarztes sowie eine Kopie des Rezeptes.“

(4) „Hersteller, Zulassungsinhaber (Depositeure), Arzneimittel-Großhändler, Tierärzte, die zur Führung einer tierärztlichen Hausapotheke berechtigt sind, sowie öffentliche Apotheken haben auf Verlangen des Landeshauptmannes oder des Bundesministers für Gesundheit und Frauen diesem alle verfügbaren Daten gemäß Abs. 1 über den Verkehr mit Tierarzneimitteln geordnet nach Art, Menge und Bezieher schriftlich mitzuteilen.“

### **Rückstandskontrollverordnung 2006**

Die Rückstandskontrollverordnung befasst sich mit Kontrollmaßnahmen bezüglich der Rückstände von Stoffen in Tieren und Lebensmitteln.

§ 7. „Die vom Landeshauptmann betrauten Personen haben Aufzeichnungen über die entnommenen Proben zu führen. Nach Vorliegen des Untersuchungsbefundes ist dieser Befund den Aufzeichnungen anzuschließen. Der amtliche Tierarzt hat Angaben über die im Rahmen der Schlachtung entnommenen Proben sowie die Ergebnisse der Untersuchungen den Aufzeichnungen gemäß § 8 Fleischuntersuchungsverordnung 2006 (FIUVO), BGBl. II Nr. 109/2006, anzufügen.“

§ 12. (1) „Der behandelnde Tierarzt ist verpflichtet, im Rahmen seiner Tätigkeit in Tierhaltungsbetrieben die Bestimmungen dieser Verordnung einzuhalten. Dabei ist im Bestandsregister fortlaufend Zeitpunkt und Art der verordneten oder durchgeführten Behandlungen, die genauen Angaben zur Identität der behandelten Tiere und bei Erzeugnissen der Aquakultur die genaue Kennzeichnung der Teiche sowie die jeweiligen Wartezeiten noch am Tage der Behandlung einzutragen. Bei Rindern ist zur Feststellung der Identität die Rinderkennzeichnungs-Verordnung 2008, BGBl. II Nr. 201/2008, bei Schweinen, Schafen und Ziegen ist die Tierkennzeichnungs- und Registrierungsverordnung 2007, BGBl. II Nr. 166/2007, zu beachten.“

### **Tierkennzeichnungs- und Registrierungsverordnung 2009**

§ 19. (1) „Tierhalter von Schweinen, Schafen oder Ziegen ausgenommen Transporteure haben ein Bestandsregister zu führen. Alle Eintragungen sind innerhalb von sieben Tagen zu tätigen und im Falle von Schweinen mindestens drei Jahre lang, im Falle von Schafen und Ziegen mindestens sieben Jahre, aufzubewahren und auf Verlangen der Behörde zur Einsicht vorzulegen. Sofern das Bestandsregister automationsgestützt geführt wird, ist der Tierhalter verpflichtet, die für die Behörde vorgesehenen Ausdrucke auf seine Kosten zu erstellen.“

### **Tiergesundheitsdienst-Verordnung 2009**

§ 8. (5) „TGD-Betreuungstierärzte haben folgende Vorgaben zu erfüllen beziehungsweise nachstehende Bestimmungen einzuhalten:

1. Sie sind verpflichtet, Betriebserhebungen gemäß Anhang 3 durchzuführen und zu dokumentieren; ...“

§ 9. (1) „TGD-Tierhalter haben folgende Anforderungen zu erfüllen: ...

4. Das Bestandsregister einschließlich des Behandlungsregisters ist in geordneter und leicht überprüfbarer Form (Dokumentation der TAM-Abgabe, TAM-Anwendung und

Rückgabe) zu führen und am Betrieb mindestens fünf Jahre lang - auch nach dem Ausscheiden aus dem TGD - aufzubewahren und den behördlichen Kontrollorganen auf Verlangen zur Einsicht vorzulegen...

6. Der Tierhalter hat in seinem Betrieb eine rückverfolgbare Dokumentation des Viehverkehrs zu gewährleisten...“

(2) „Bei Zusammenarbeit mit dem TGD-Betreuungstierarzt hat der TGD-Tierhalter folgende Bestimmungen einzuhalten:

1. TGD-Tierhalter, die für mehrere Tierarten Betreuungsverhältnisse eingehen, haben die Aufzeichnungen nach Tierarten getrennt im Bestandsregister einschließlich Behandlungsregister zu führen...“

(3) „Bei Einbindung von TGD-Tierhaltern in die Anwendung von Arzneimitteln gemäß des 4. Abschnittes dieser Verordnung bestehen für diese, unbeschadet weiterer Regelungen dieser Verordnung oder anderer Rechtsvorschriften, folgende Pflichten: ...

7. Sie haben dafür Sorge zu tragen, dass die dem TGD-Arzneimittelanwender vom TGD-Betreuungstierarzt überlassenen Tierarzneimittel nur gemäß den Anleitungen des Tierarztes am zugehörigen Betrieb angewendet und diese Anwendung schriftlich im Behandlungsregister dokumentiert wird...“

§ 12. (7) „Die Abgabe, Anwendung und Rückgabe der Tierarzneimittel ist gemäß Anhang 5 zu dokumentieren.“

## 2.3 Standards

In diesem Abschnitt werden gängige Standards vorgestellt, von denen einige bereits bei der Umsetzung der ELGA Verwendung finden. Drei der fünf im Folgenden Standards kamen bei der elektronischen Gesundheitsakte für Menschen zur Anwendung. Diese drei sind Health Level 7 für den Datenaustausch im Gesundheitswesen zwischen den unterschiedlichen Computersystemen der verschiedenen Organisationen, DICOM für den Austausch von Bilddaten im Gesundheitswesen sowie LOINC zur Beschreibung von Untersuchungs- und Testergebnissen mit allgemeingültigen Bezeichnern. Weiters werden zwei spezielle Standards aus dem veterinärmedizinischen Bereich betrachtet, SNOMED als Nomenklatur sowie Nomina Anatomica Veterinaria als Terminologie.

### 2.3.1 Health Level Seven (HL7)

Die Organisation HL7 entwickelt Standards für den elektronischen Austausch von Daten im Gesundheitswesen. HL7 existiert derzeit in den Versionen 2.x sowie 3.x und ist frei verfügbar, d.h. der Erwerb von Lizenzen ist nicht notwendig.

In der Version 2 besteht laut [68] eine Nachricht aus mehreren Segmenten, deren Struktur durch den Standard definiert ist. Jedes Segment besteht weiter aus verschiedenen Feldern, die wiederum weiter in Komponenten und Subkomponenten aufgespalten werden können. Verschiedenen Ereignisse können eine Nachricht auslösen, wie z.B. die Aufnahme oder die Entlassung eines Patienten oder das Bereitstellen von Untersuchungsergebnissen.

In [68] wird folgendes Beispiel einer HL7 Nachricht angeführt:

```

MSH|^~\&|LIS|1021|HIS|1034|201010151727||ORU^R01| MSG00001|P|2.4
PID||0001|||Mary||19810401|F||XINSONGJIANG Rd.^Shanghai^^201620^ CHN
PV1||NER||||GSU|||||E|||||20101015174900|201 01015
OBR||1||09527539461000270|1000270^BILIRUBIN,TOTAL^^TBILI||2010101518050
0|||||20101015182500||||1793569|| 0952753946||2010101518530000||100|F||^RT
OBX||NM|1000270^BILIRUBIN,TOTAL^^TBILI|0001|0. 7|mg/dL|0.2-1.2|||
F||20101015185300||579

```

Im ersten Segment steht der Nachrichtentyp, ORU R01, bei dem es sich um ein Behandlungsergebnis aus den Leberfunktionstests der Patientin Mary handelt, zu sehen in Segment zwei. Der Doktor beauftragte das Labor am 15. Oktober 2010 um 17:49 Patientin Mary auf Bilirubin zu testen und übermittelte den Bericht um 18:53 des selben Tages. Sobald die Untersuchung im Laborinformationssystem beendet war, wurde eine ORU Nachricht erstellt und zurück an das Krankenhausinformationssystem gesendet.

Aus [13] geht hervor, dass Version 3 auf einem generischen Datenmodell basiert, dem Reference Information Model (RIM) und die Implementierung der zu übertragenden Nachrichten im XML Format festlegt. Die Clinical Document Architecture (CDA) von HL7 Version 3.x ist ein Standard zur Spezifikation der Struktur und der Semantik von medizinisch relevanten Dokumenten mit der Absicht zum Datenaustausch. Ein derartiges Dokument ist ein festgelegtes und vollständiges Informationsobjekt, das aus Text, Bildern und anderen multimedialen Inhalten bestehen kann.

CDA Dokumente werden durch folgende Aspekte charakterisiert:

- Sie liegen im Extensible Markup Language (XML) Format vor.
- Ihr Inhalt wird durch das HL7 RIM festgelegt und sie verwenden die Datentypen der HL7 Version 3.

Einige der Ziele und Anforderungen an den Standard sind den nachstehenden Punkten zu entnehmen [46]:

- Die Steigerung der Qualität in der Gesundheitsversorgung.
- CDA soll auch für Personen ohne technischen Hintergrund lesbar sein.
- Die Dokumente der CDA sollen über Systemgrenzen hinweg unkompliziert verarbeitet werden können.
- Der Standard muss abgeleitet vom RIM von HL7 sein.

Eine beispielhafte Darstellung eines CDA Dokumentes mit seinen Hauptkomponenten sieht wie folgt aus [13]:

---

```

1 <ClinicalDocument >
2 ...CDA Header ...
3 <StructuredBody >

```

---

```

4     <section >
5         <text >... </ text >
6         <Observation >... </ Observation >
7         <Observation >
8             <reference >
9                 <ExternalObservation >... </ ExternalObservation >
10            </reference >
11        </Observation >
12    </section >
13    <section >
14        <section >... </ section >
15    </section >
16 </StructuredBody >
17 </ClinicalDocument >

```

Ein CDA Dokument ist ein definiertes und vollständiges Informationsobjekt, welches außerhalb des Datentransfer Kontextes existieren kann oder aber als MIME - verschlüsselter Anhang innerhalb einer HL7 Nachricht existiert.

Zum näheren Verständnis werden nachfolgend einige der technischen Artefakte, die von CDA verwendet werden, erläutert [13]:

**HL7 RIM** ist ein statisches objektorientiertes Informationsmodell für Klassen- und Attributdefinitionen [54]. Basierend auf UML besteht das Reference Information Model aus einer generischen Menge von Klassen, aus denen weiter gesundheitspezifische Klassen abgeleitet werden.

RIM besteht aus drei Bereichen - semantischer Kontrolle, strukturierten Dokumenten und Nachrichtenkontrolle. Die Datenübertragung wird über vier Primärklassen und zwei Linkklassen gebildet. Die vier Primärklassen sind:

- **Act:**  
Ein Act ist als bewusste gesundheitsbezogene Aktion in der Businessdomäne von HL7 definiert. Eine Instanz ist ein Datensatz eines Acts.
- **Entity**  
Ein Entity ist ein physischer Gegenstand oder eine Gruppierung von mehreren physischen Dingen. Dazu gehören keine elektronischen gesundheitsrelevanten Daten, Nachrichten, Datenstrukturen o.ä.
- **Role:**  
Bei Personen ist eine Rolle für gewöhnlich eine Position oder ein Beruf und für Orte oder Dinge wird die gewöhnliche Bedeutung dieser Gegenstände gebraucht. Die Rolle eines Patienten wird daher von einer Person gespielt.
- **Participation:**  
Diese Klasse existiert nur innerhalb des Wirkungsbereiches eines Acts. Acts haben mehrere Participants und jeder ist ein Entity in einer Role. Eine Role bezeichnet die Kompetenz während Participation die Durchführung bezeichnet.

Die zwei Linkklassen werden gebildet von:

- **Act Relationship:**  
Hierbei handelt es sich um einen Verband zwischen einem Source-Act und einem Target-Act.
- **Role Link:**  
Diese Art von Link bildet eine Beziehung zwischen zwei Entity Roles ab, z.B. die Beziehung zwischen einem Krankenhausarzt und der Klinik oder die Verbindung zwischen Patient und Klinik und drückt dadurch die Beziehung zwischen Arzt und Patient aus.

Abbildung 2.6 zeigt beispielhaft ein UML Klassendiagramm des RIM [54].

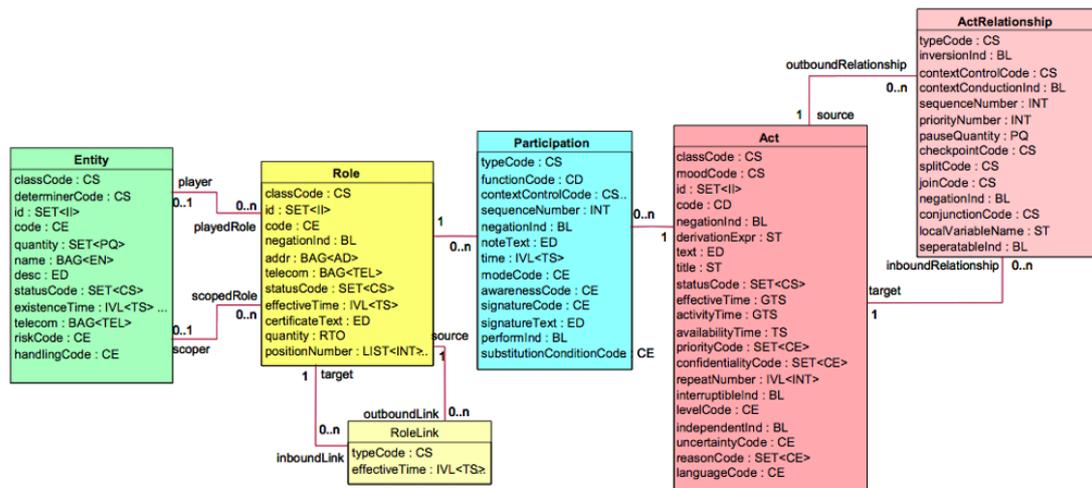


Abbildung 2.6: UML Klassendiagramm des Reference Information Models

**HL7 Datentypen der Version 3** definieren laut [13] das strukturelle Format der zu übertragenden Daten innerhalb eines RIM Attributes und beeinflussen dadurch die Menge der zulässigen Werte. Einige Datentypen haben nur sehr wenig intrinsischen semantischen Inhalt. Allerdings definiert HL7 auch komplexere Datentypen. Jedes Attribut im RIM ist mit genau einem Datentypen verknüpft.

**HL7 Lexikalische Domänen** repräsentieren Werte für kodierte CDA Komponenten. Bei diesen Domänen wird zwischen zwei Typen unterschieden: "Coded, No Extensions (CNE)", wobei die einzigen erlaubten Werte jene aus der Domäne sind; oder "Coded, WITH Extensions (CWE)", wobei auch Werte die nicht in der Domäne vorkommen verwendet werden dürfen [13]. Wenn eine kodierte CDA Komponente einen zugehörigen CNE Wert hat, sind die erlaubten Werte durch den Standard fixiert und wie in dem folgenden Beispiel spezifiziert (cf. Tabelle 2.1):

Code	Definition
APND (append)	Das aktuelle Dokument ist eine Ergänzung zum Übergeordneten Dokument.
RPLC (replace)	Das aktuelle Dokument ist ein Ersatz für das Übergeordnete Dokument.
XFRM (transform)	Das aktuelle Dokument ist eine Transformation des Übergeordneten Dokumentes.

Tabelle 2.1: Werte für relatedDocument.typeCode (CNE)

Bei der Umsetzung des Standards in der Praxis werden die Anwendungsdaten die in verschiedenen Formaten vorliegen in das HL7 Nachrichtenformat übertragen. Diese Nachrichten werden zum Empfängersystem über das Netzwerkprotokoll übertragen und auf der Empfängerseite wieder in das dort erforderliche Format geparkt. [35]

Dieser Standard kann also zur Übertragung verschiedener gesundheitsrelevanter Informationen zwischen unterschiedlichen Systemen verwendet werden. Tierärzte können Diagnose-, Medikamenten- oder allgemeine Tierdaten die sie digital erfassen, in diesem einheitlichen Format an das zentrale System der elektronischen Gesundheitsakte für Nutztiere übertragen oder von dort abrufen.

### 2.3.2 Logical Observation Identifier Names and Codes (LOINC)

LOINC dient dazu, allgemeingültige Identifikatoren (Namen und Codes) im Zusammenhang mit existierenden Standards wie HL7 (Abschnitt 2.3.1) oder DICOM (Abschnitt 2.3.3) bereitzustellen, um Untersuchungsergebnisse aus dem klinischen Umfeld zu beschreiben. LOINC soll den Austausch von Klinik- und Labordaten zwischen verschiedenartigen Computeranlagen erleichtern.

[41] beschreibt, dass um diesen Prozess zu ermöglichen, ein vollqualifizierter Name auf standardisiertem Weg erstellt wird, sodass Benutzer lange Namen für ihre Untersuchungen erstellen und diese mit eindeutigen Identifikatoren verlinken können.

Jedem Namen wurde ein eindeutiger, dauerhafter Code zugewiesen, der LOINC Code genannt wird. Dieser Code soll von Systemen verwendet werden, um Test- und Untersuchungsergebnisse in elektronischen Akten zu identifizieren.

Der vollqualifizierte Name eines Test- oder Beobachtungsergebnisses besteht aus fünf oder sechs Hauptbestandteilen [41]:

- Der Name der Komponente respektive der Analyt, der gemessen wurde (z.B. Glukose, Propranolol).
- Das beobachtete Merkmal (z.B. Konzentration einer Substanz, Masse, Volumen).
- Der Beobachtungszeitraum (z.B. über einen gewissen Zeitraum, Momentaufnahme).
- Der Typ der Probe (z.B. Urin, Serum).

- Die Art der Messung (z.B. qualitativ vs. quantitativ).
- und wenn sinnvoll, die Methode der Messung (z.B. Radioimmunassay - eine Labormethode zur quantitativen Bestimmung kleinster Substanzmengen).

Die formale Syntax setzt sich wie folgt zusammen [41]:

<Analyte/Komponente>:<Art des Merkmals>:<Zeitaspekt>:<Probentyp>:<Art der Messung>:<Methode>

Der Doppelpunkt ist Teil des Namens und wird zur Teilung der Hauptteile des Namens benötigt.

Der erste Teil des Namens (Komponente/Analyte) kann weiter in drei Unterbereiche aufgespalten werden, die durch das Auslassungszeichen (^) getrennt werden. Der erste Teilbereich kann multiple Ebenen von taxonomischen Spezifikationen beinhalten, die durch Punkte (.) getrennt werden.

Der dritte und vierte Teil des Namens (Zeitaspekt und Probentyp) können ebenfalls durch einen weiteren Unterbereich modifiziert werden, der vom ersten durch ein (^) getrennt wird. Beim Zeitaspekt kann der Modifizierer andeuten, dass die Beobachtung auf Basis einer Kriterie gewählt wurde (Maximum, Minimum, Durchschnitt, usw.); Beim Probentyp identifiziert der Modifikator die Herkunft der Probe, wenn sie nicht vom Patienten stammt (z.B. Blutspende, Fötus).

Beispiele für vollqualifizierte LOINC Namen sind [41]:

- Sodium:SCnc:Pt:Ser/Plas:Qn,
- Creatinine renal clearance:VRat:24H:Ur+Ser/Plas:Qn,
- Sodium:SCnc:Pt:Urine:Qn,
- Glucose^2H post 100 g glucose PO:MCnc:Pt:Ser/Plas:Qn,
- Sodium:SRat:24H:Urine:Qn,
- ABO group:Type:Pt:Bld^donor:Nom,
- Binocular distance:Len:Pt:Head^fetus:Qn:US.measured.

**Namenskonventionen** LOINC unterliegt verschiedenen Namenskonventionen. Ein kleiner Auszug kann der folgenden Aufzählung entnommen werden [41]:

#### 1. Abkürzungen in Komponenten/Analyte Namen

Abkürzung	Voller Name
Ab	Antibody
Ag	Antigen
HIV	Human Immunodeficiency Virus
HLA	Human Histocompatibility Complex Derived Antigens
NOS	Not otherwise specified
RNA	Ribonucleic Acid
rRNA	Ribosomal Ribonucleic Acid

Tabelle 2.2: Beispiele für Komponenten Abkürzungen

2. Generelle Regeln für Namen von Komponenten die Teil eines vollqualifizierten Namen sind.
  - a) Der Identifikator der Substanz die gemessen wurde, steht vorne, d.h. „Hepatitis A Antikörper (Ab)“ nicht „Antikörper, Hepatitis A“.
  - b) Verwendet werden soll der generische Name eines Medikamentes, nicht sein Markenname.
  - c) Weiters soll der volle taxonomische Name von Erregern oder Viren (keine Krankheit) bei der Beschreibung von Test- oder Diagnosedaten verwendet werden, d.h. z.B. „herpes simplex virus Ab“ nicht „HSV Ab“.
  - d) Wenn ein Test den Namen eines Bakteriums beinhaltet, wird für den LOINC Namen der volle Bakterien Name vom International Journal of Systematic and Evolutionary Microbiology verwendet. Im Fall von Virennamen wird der Name aus dem Index Virum verwendet.
  - e) Vitamine werden durch ihren chemischen Namen dargestellt, z.B. Thiamin statt Vitamin B1
  - f) Antikörper werden immer als Nomen beschrieben, z.B. Myocardium Ab statt Myocardial Ab.
  - g) Alkoholnamen bestehen nur aus einem Wort, z.B. Methanol statt Methyl Alkohol oder Ethanol statt Ethyl Alkohol usw.
  - h) Griechische Buchstaben wie Alpha, Beta, Gamma, etc. werden immer ausgesprochen, z.B. Alpha Tocopherol statt A-Tocopherol
  - i) usw...

Die LOINC Datenbank beinhaltet neben den qualifizierten Namen noch zwei weitere Felder, den *Short Convenient Name* und den *Long Common Name*. Der Short Convenient Name ist ein kurzer Ausdruck eines Namens von einem LOINC Konzept. Die Namen sind nicht länger als 30 Zeichen wodurch sie für Labor- und Auswertungssysteme geeignet sind.

Von Benutzern kamen regelmäßig Anfragen nach einem schönen, anzeigbaren Namen, der in User Interfaces für die Anzeige verwendet werden kann. Während die systematisch erzeugten LOINC Standard Kurznamen garantiert eindeutig sind, sind sie meistens nicht sehr benutzerfreundlich. Im Gegensatz dazu sind jedoch die anwenderfreundlichen Namen oft mehrdeutig. [41]

Datentransferstandards wie HL7 (Abschnitt 2.3.1) verwenden meistens ein Triple <Identifikator Code>^<beschreibender Text>^<Kodierung> für Felder, die verschlüsselte Einträge enthalten. LOINC produziert mindestens drei Namen für jeden Term, den sechsteiligen vollqualifizierten Namen, den Kurznamen und den langen benutzerfreundlichen Namen.

In [41] kann mehr zu diesem Thema nachgelesen werden.

Die LOINC Terminologien werden in Datenbanken publiziert und können ebenfalls wie Systematized Nomenclature of Medicine and Veterinary Medicine (SNOMED) zur einheitlichen Darstellung von Untersuchungsergebnissen in der elektronischen Gesundheitsakte für Nutztiere herangezogen werden. Der Standard könnte innerhalb von CDA Dokumenten zum Einsatz kommen und den Datenaustausch zwischen Systemen der Veterinärmediziner, Landeskontrollverbände oder Ministerien erheblich erleichtern. Die LOINC Tabellen stehen auf der Webseite [36] in verschiedenen Formaten wie z.B. Microsoft Access oder ASCII zur Verfügung.

### 2.3.3 Digital Imaging and Communications in Medicine (DICOM)

DICOM Funktionen werden verwendet, um gespeicherte Bilder aufzufinden und Attribute von spezifischen Objektinstanzen zu aktualisieren und aufzurufen. Dieser Standard unterstützt die Kompatibilität der Übertragung von Bildmaterial aus der Medizin zwischen verschiedenen Anbietern. DICOM spezifiziert zusätzlich zur Speicherung auch Abfragedienste für Bilder, stellt jedoch selber keine Datenbank dar. Die Informationsobjekte und Informationsmodelle des Standards bilden einen Kontext zur Kommunikation [49].

Der Standard besteht aus verschiedenen Ebenen in Bezug auf das ISO OSI Netzwerk Modell. Dieses Netzwerkmodell besteht aus sieben Schichten. Zur Erklärung:

Ganz oben auf Ebene sieben befindet sich die Anwendungsschicht, die der Applikation Zugriffe auf verschiedene Services liefert. Darunter folgt die Präsentationsschicht, welche die Information in eine unabhängige Form konvertiert um den Datenaustausch zwischen Systemen zu ermöglichen. Auf Ebene fünf liegt die Sessionschicht, die für die Prozesskommunikation zuständig ist. Ebene vier stellt die Transportschicht dar, welche der End-zu-End-Kommunikationskontrolle dient. Darunter folgt die Netzwerkschicht, die für das Routing der Information im Netzwerk verantwortlich ist. Layer zwei bildet die Datensicherungsschicht, die der fehlerfreien Übertragung dient. Die unterste Schicht wird vom Physikalischen Layer gebildet, über den Bits über einen Kommunikationskanal übertragen werden. [43]

[44] beschreibt, dass der Standard unabhängig vom physikalischen Layer ist, weil er keine physikalische Verbindung definiert. Innerhalb ist das Upper Layer Protocol (ULP) definiert, das ein abstraktes Protokoll ist, welches die Datenkapselung festlegt. Dieses Protokoll ist auf einer höheren Ebene als der Layer fünf des ISO OSI Modells.

Auf der Applikationsschicht existieren fünf primäre Funktionsbereiche [44]:

- Übertragung und Speicherung von vollständigen Objekten (wie Bilder, Frequenzen und Dokumente),

- Abfrage dieser Objekte,
- Effizienz der spezifischen Aktionen (wie z.B. das Ausdrucken von Bildern),
- Arbeitsablauforganisation,
- Qualität und Konsistenz der Bilder (Darstellung und Druck).

Neben dem eigentlichen Entwicklungszweck von DICOM, nämlich den Datenaustausch von medizinischen Bildern zu standardisieren und ihn dadurch zu vereinfachen, werden auch netzwerkorientierte Dienste für die Übertragung oder das Drucken von Bildern, Medienformaten für den Datenaustausch, Arbeitsablauforganisation, Konsistenz und Qualität der Präsentation sowie Anforderungen an die Konformität der Geräte und Programme definiert [44].

[44] beschreibt weiters, dass im Standard die Information Object Definition (IOD) eingeführt wurde, um Attribute darzustellen, die eine bestimmte Eigenschaft des Bildes beschreiben. IODs haben eine wohldefinierte Bedeutung und ihre Attribute beschreiben präzise den Typ des Objektes, Daten des Patienten, ausgeführte Behandlungsabläufe oder Berichte genauso wie die technische Information über das medizinische Bildgerät, welches bei der Untersuchung verwendet wurde. Technische Information beinhaltet den Namen des Bildgeräteherstellers, die Seriennummer des Gerätes und andere Details über das Gerät. Hier könnte man z.B. die Daten eines Mammographiegerätes erwähnen, wie Aufnahmezeit, X-Ray, Anodenmaterial, Kompressionsdruck, Temperatur, etc. Diese Attribute variieren, wenn Geräte von verschiedenen Herstellern mit verschiedenen Modalitäten (CT, MRT, Mammographie) verglichen werden.

Der Standard definiert weiters auch Netzwerkdienste, die für den Transfer von Information verwendet werden. Um einen Transfer durchzuführen, müssen beide Standorte die selben Dienste und Objekte unterstützen. Zwei Standorte die dieselben Dienste und Objekte unterstützen werden auch Service-Objekt Paar Klassen genannt (SOP Klasse). Es existieren viele weitere Dienste die es erlauben Daten und Objekte mit verschiedenen Ausprägungen abzufragen. Ein wichtiger Dienst ist DICOM Print, der den Datentransfer zum Drucken über das Netzwerk ermöglicht. [44]

Für die Übertragung und die Kommunikation spezifiziert DICOM ein Netzwerkprotokoll, welches TCP/IP nutzt. Das Protokoll definiert die Bedienung von Service-Klassen außerhalb von simplen Datenübertragungen und bietet einen Mechanismus um Objekte bei der Übertragung über das Netzwerk eindeutig zu identifizieren. Informationsobjekte sind nicht nur für Bilder definiert, sondern auch für Patienten, Studien, Berichte und andere Datenbündel. An dieser Stelle ist es wichtig zu erwähnen, dass der DICOM Standard keine Hardware-Schnittstellen zur Verbindung von Geräten definiert. Stattdessen stellt er eine Unterstützung bereit um Informationen basierend auf dem Upper Layer Protocol (ULP) des ISO OSI Netzwerk Modells, die unabhängig vom physikalischen Netzwerk sind, auszutauschen. [44]

DICOM ist ein umfangreicher und komplexer Standard, weil er sich für verschiedene medizinische Fachgebiete eignen muss.

Es ist sehr wichtig einen Standard zu haben, der in allen Krankenhäusern für den selben oder ähnlichen Auswertungsprozess verwendet werden kann. Ein eindeutiger Standard hilft dabei Probleme zu vermeiden die auftreten, wenn Patienten von einer Klinik in eine andere verlegt werden. DICOM bietet auch eine Verbindungsflexibilität zwischen

unterschiedlichen medizinischen Systemen. Der große Vorteil des Standards liegt in der Unterstützung aller medizinischer Fachbereiche und das macht ihn so reichhaltig. Ein weiterer Vorteil liegt auch in der Speicherung von analogen Bildern und Daten, weil sehr wenig Speicherplatz verwendet wird und so die Übertragung sehr einfach ist.

Der Standard wird in den meisten bildverarbeitenden Systemen implementiert, sodass die Bilder bereits standardkonform erstellt werden. Es existieren viele Betrachtungsprogramme für DICOM Bilder, kommerzielle sowie Lösungen, die frei zum Download verfügbar sind. Als populärste freie Programme können DicomWorks [12] und Irfan-View [32] genannt werden.

Genauso wie in der Humanmedizin soll der DICOM Standard in der Veterinärmedizin, respektive speziell bei der elektronischen Gesundheitsakte für Nutztiere Verwendung finden und dabei den Austausch sowie die Archivierung digitaler Aufnahmen und Röntgenbilder zwischen den Tierkliniken, -praxen und der zentralen Schnittstelle der ELGA für Nutztiere erleichtern.

### **2.3.4 Systematized Nomenclature of Medicine and Veterinary Medicine (SNOMED)**

Bei SNOMED handelt es sich um einen Fachwortschatz, der von Medizinern und artverwandten Berufsgruppen verwendet wird. Die Definition einer Nomenklatur laut [45] ist wie folgt:

„wissenschaftl. (systematisierende) Namengebung u. -zusammenstellung innerhalb eines naturwissenschaftl. Fachgebietes.“

Es umfasst ein geregeltes Vokabular mit einer umfangreichen Abdeckung von Krankheiten, klinischen Befunden, Therapien, Behandlungsmethoden und daraus resultierenden Ergebnissen.

In [67] ist zu lesen, dass SNOMED für die elektronische Datenspeicherung und Datenabfrage in klinischen Informationssystemen optimiert wurde. Es existieren über 110.000 Ausdrücke, die mit einer klinischen Datenbasis verlinkt sind und mehr als 180.000 Terme mit eindeutigen, computerlesbaren Codes, die zeitgemäße medizinische Methoden widerspiegeln. Darüber hinaus sind über 260.000 explizite Beziehungen zwischen Termen vorhanden, die fallspezifische Abfragen basierend auf der Krankheitshierarchie, Anatomie, Ätiologie, etc. ermöglichen.

SNOMED wird vom SNOMED Editorial Board unterstützt, welches von Terminologieexperten, Domänenexperten, Vereinigungen mit professionellen Gesellschaften und staatlichen Organisationen gebildet wird.

Die logische Syntax besteht aus einer modifizierten Knowledge Representation System Specification (KRSS) Syntax, die vereinfacht wurde. Näheres zur KRSS Syntax kann in [48] nachgelesen werden.

Die Modellierung des Standards erfolgte auf Basis des Convergent Medical Terminology (CMT) und entstand in einem iterativen Prozess.

[67] beschreibt weiters, dass SNOMED in elektronischen Gesundheitsakten dort verwendet wird, wo fehlerfreie und genaue Daten betreffend klinischen Informationen grundlegend sind, wie z.B. Diagnosedaten oder Übertragung von Bildinformationen. Dabei sollen aber keine Daten ersetzt werden, die bereits in strukturierter Form vor-

handen sind, wie z.B. Verschreibungen. Vielmehr soll für die Benutzer eine Referenz dargestellt werden, die strukturierte Terminologien bereitstellt. Bezogen werden können die Daten aus relationalen Tabellen die die Konzepte (Ausdrücke), Terme und deren Beziehungen beinhalten.

Beispiele für Synonyme (Terme) [67]:

- Jeder Ausdruck hat einen eindeutigen Code:
  - D3-1600 = Herzfehler
  
- Jedes Synonym hat einen eindeutigen Code:
  - D3-1600.1 = Herzfehler
  - D3-1600.6 = Herzmuskelschwäche
  - D3-1600.7 = Herzschwäche
  - D3-1600.6 = Herzrhythmusstörung

Laut [53] wird bei SNOMED jeder Term einer Gruppe bzw. Klassifikation zugeteilt, die bestimmte Charakteristiken aufweist. So werden alle anatomischen Terme in eine Klasse, die biochemischen und physiologischen in eine andere und alle lebenden Organismen und Tumore in ihre jeweilige Gruppe zusammengefasst. Innerhalb dieser Klassen existiert eine weitere Ordnung der Terme, z.B. werden die anatomischen Terme weiter unterteilt in Organsysteme des Körpers (Atemwege, Herz-Kreislauf, Magen-Darm,...) und die biochemischen Terme sind nach ihrer Struktur und Funktion gruppiert (chemische Elemente, Verbindungen, Hormone,...).

In SNOMED Version 3.3 existieren elf Hauptklassen, die ein separates Modul darstellen und Tabelle 2.3 entnommen werden können:

<b>Module</b>
T = Topographie
M = Morphologie
F = Funktion
L = Lebende Organismen
C = Chemikalien, Arzneimittel und biologische Produkte
A = Naturkräfte und Aktivitäten
J = Tätigkeiten
S = Sozialer Kontext
D = Krankheit / Diagnose
P = Prozeduren
G = Generelle Bindungen, Modifikatoren

Tabelle 2.3: SNOMED Module

Jeder Termcode beinhaltet implizit Informationen, die über die alphanumerische Repräsentation dargestellt werden.

T-30000 Herzkreislaufsystem

T-32000 Herz

T-32020 Herzmuskel

Die implizite hierarchische Information die im Termcode enthalten ist, sieht folgendermaßen aus:

Der Herzmuskel „ist Teil vom“ Herzen und das Herz „ist Teil vom“ Herzkreislaufsystem.

Um bei der Erfassung von Diagnosen und Krankheiten eines Tieres in der elektronischen Gesundheitsakte einheitliche Terme zu verwenden, wäre es sinnvoll, sich eines dafür vorgesehenen, umfangreichen Standards wie SNOMED zu bedienen. Es existieren verschiedene freie und kommerzielle Browser, mit deren Hilfe die SNOMED Daten abgerufen werden können. [53]

### 2.3.5 Nomina Anatomica Veterinaria

Bis 1895 gab es kein generelles Abkommen über die Fachbezeichnungen von menschlicher oder veterinärer Anatomie. Jede Nation hatte seine eigene Terminologie. Viele Strukturen hatten verschiedene Namen in verschiedenen Ländern und viele waren nach der Person benannt, die sie zuerst verwendete. In vielen Fällen wurde das selbe Organ mit den Namen mehrerer Anatomen verschiedener Staaten assoziiert. Die ersten Bemühungen eine vereinheitlichte anatomische Terminologie zu entwickeln wurden von der *Basel Nomina Anatomica* (B.N.A.) unternommen. Diese Fachbezeichnungen waren jedoch nicht auf Tiere anwendbar, da die Terme der Ausrichtung auf dem aufrechten menschlichen Körper basierten. Deswegen wurde im selben Jahr, nämlich 1895, ein Komitee für veterinärmedizinische anatomische Fachbezeichnungen gebildet. [62] Die folgenden Prinzipien von Nomina Anatomica Veterinaria (N.A.V.) dienten als Richtlinien bei der Arbeit des Komitees:

- Abseits von einigen Ausnahmen soll jedes anatomische Konzept durch einen alleinstehenden Term bezeichnet werden.
- Jeder Term soll in der offiziellen Liste in Latein angeführt werden, die Anatomen der unterschiedlichen Länder dürfen den lateinischen Ausdruck in die jeweilige Sprache übersetzen.
- Jeder Term soll so kurz und einfach wie möglich sein.
- Die Terme sollen einfach zu merken sein und vor allem einen lehrreichen und beschreibenden Wert haben.
- Strukturen, die topographisch sehr ähnlich sind, sollen auch ähnliche Namen erhalten; z.B. Arteria femoralis, Vena femoralis, Nervus femoralis.
- Werden Adjektive differenziert, sollten immer gegensätzliche verwendet werden, wie größtes und kleinstes.
- Terme die von Eponymen (Namensgebern) abgeleitet sind, sollten nicht verwendet werden.

[62]

Bei der ELGA für Nutztiere könnte dieser Standard eventuell zum Einsatz kommen um anatomische Fachbezeichnungen standardisiert zu speichern. Ob er jedoch im konkreten Fall von Nutzen ist und ob nicht bereits der Einsatz von SNOMED und LOINC ausreichend sind, wird sich in der fortschreitenden Analyse der Gesundheitsakte erst zeigen.

## 3 Grundlagen zur Softwarearchitektur

Die Grundlage eines guten Softwaresystems ist eine gute Softwarearchitektur mit einer Reihe von Designentscheidungen, die während der Entwicklung getroffen werden. Softwarearchitektur beschreibt wie Softwaresysteme konzipiert und gebaut sind. In der Softwareentwicklung ist der primäre Fokus die Architektur, da die Entwicklung von qualitativ hochwertigen Produkten abhängig von den beim Architekturentwurf getroffenen Entscheidungen ist. Ein architekturzentrierter Ansatz hat seinen Schwerpunkt auf dem Design des Systems.

Am Ende entspricht die Designqualität der Produktqualität, da es extrem ungewöhnlich wäre, wenn ein Softwaresystem sehr hohe Qualität hätte und gleichzeitig ein dürftiges Systemdesign. Softwarearchitektur muss also bereits von Grund auf beim Systemdesign und bei der Entwicklung des Systems stattfinden. Es muss im Vordergrund stehen, mehr als der Entwicklungsprozess, mehr als die Analyse und auch mehr als die Programmierung. [42]

### 3.1 Einführung

Die Softwarearchitektur ist die Beschreibung der gesamten Struktur des Systems. Sie ist definiert als eine Sammlung von Komponenten zusammen mit einer Beschreibung, wie diese Komponenten zusammenspielen [4].

Bei der Architektur werden keine Details spezifiziert, sondern die tragenden Komponenten eines Systems. Der Architekturentwurf reicht von der Analyse des Problems bis zur Realisierung der Lösung und zielt darauf ab, Komplexität handhabbar zu machen. Weiters beschäftigt sich die Architektur mit Entscheidungen, die systemweite Auswirkungen haben. Es werden nicht nur strukturelle Beschreibungen des Systems gemacht. Viel mehr geht es auch um die Arbeit des Architekten selber, dessen Aufgabenbereich auch die Zusammenarbeit mit den Stakeholdern des Systems umfasst.

Die Architektur als Oberbegriff lässt sich in die Architektur der Domäne, die technische Architektur sowie die Architektur der Plattform gliedern.

**Architektur der Domäne** spezifiziert den Problembereich, für den das System entwickelt wird. Sie wird getrieben von der Art der Domäne sowie den funktionalen Anforderungen (siehe Abschnitt 4.2.3).

**Technische Architektur** ist unabhängig von der Domäne und behandelt die Umsetzung der nichtfunktionalen Anforderungen (siehe Abschnitt 4.2.2).

**Architektur der Plattform** beschäftigt sich mit Hardware- und Softwarekomponenten.

Abgesehen von der Umsetzung der funktionalen sowie der nichtfunktionalen Anforderungen, sollten bei einem Architekturentwurf die Prinzipien der *losen Kopplung* sowie der *hohen Kohäsion* umgesetzt werden.

Kopplung steht für die Abhängigkeit der Komponenten eines Systems. Komponenten können Klassen, Module, Hardwareelemente o.ä. sein. Zwei Komponenten sind gekoppelt, wenn im Fall, dass Komponente 1 geändert wird, auch Komponente 2 geändert werden muss, da sonst das System nicht korrekt funktioniert. Dieses Negativbeispiel macht deutlich, dass eine lose Kopplung angestrebt werden sollte, um eine leichtere Verständlichkeit der Komponenten für sich allein zu ermöglichen.

Innerhalb einer Komponente sollte ein großer Zusammenhalt angestrebt werden. Das ermöglicht die einfache Verständlichkeit und Veränderbarkeit ohne dabei auf andere Komponenten einzuwirken.

Je höher die Kohäsion einzelner Komponenten ist, desto loser die Kopplung untereinander.

Weitere zu beachtende Prinzipien für einen guten Softwarearchitekturentwurf sind Information Hiding (Geheimnisprinzip), Separation of Concerns (Trennung der Zuständigkeiten) sowie das Open-Closed Principle.

Beim Information Hiding geht es darum, die Schnittstelle von der Implementierung zu trennen, damit Zugriffe nur über die Schnittstelle erfolgen und die Implementierung gekapselt ist. So ist die Logik leicht austauschbar.

Bei der Separation of Concerns ist die Idee, dass Komponenten nicht mehr als eine Aufgabe haben, die sie ausführen.

Das Open-Closed-Principle besagt, dass Komponenten offen sein sollen, um Erweiterungen leicht zu ermöglichen und gleichzeitig geschlossen sein sollen, sodass die Schnittstelle stabil ist.

## 3.2 Qualitätskriterien

Qualitätskriterien einer Architektur sind die Bezugspunkte, die das Verhalten des Systems innerhalb der Umgebung für die sie entwickelt wurde, beschreiben. Anhand dieser Kriterien kann die Eignung sowie die Tauglichkeit einer Architektur gemessen werden. Dieser Abschnitt soll einen Überblick über gängige Qualitätskriterien geben.

Francisca Losavio et al [37] beschreiben in ihrer Arbeit Qualitätskriterien für Softwarearchitekturen und gehen dabei näher auf den ISO/IEC 9126 Standard ein. Gemäß der ISO/IEC 9126 Norm wird Qualität als eine Reihe von Merkmalen und Eigenschaften eines Produktes definiert, die sich auf der Eignung zur Erfüllung festgelegter oder vorausgesetzter Erfordernisse beziehen. Hierbei können verschiedene Perspektiven von Qualität in Betracht gezogen werden: z.B. die Benutzersicht als die Qualität des Endproduktes; die Entwicklersicht, als die Qualität des Zwischenproduktes, die durch die verschiedenen Stakeholder während des Entwicklungsprozesses generiert wurde; und die Endbenutzerverwaltungssicht als Marketinganforderung.

Die Gesamtqualität eines Produktes kann durch die Kombination der verschiedenen Sichten ausgedrückt werden. In diesem Kontext wird die Benutzer- und Architektensicht behandelt. Das ISO 9126 Qualitätsmodell ist heute allgemein als die State-of-the-Art Produktqualitätsspezifikation akzeptiert. Es schlägt eine Menge aus sechs indivi-

duellen Qualitätsmerkmalen auf hohem Niveau vor, die als eine Gruppe von Eigenschaften eines Softwareproduktes definiert sind, durch die dessen Qualität beschrieben und gewertet wird. Die Qualitätsmerkmale werden als Ziele für die Validierung, die eine externe Qualitätskontrolle darstellt, sowie der Verifikation, die für internes Qualitätsmanagement steht, in den verschiedenen Entwicklungsphasen eingesetzt. Sie sind weiter in Untereigenschaften aufgeteilt, um die messbaren Eigenschaften zu erhalten.

Die sechs Hauptqualitätsmerkmale des ISO 9126 Standards werden nachfolgend beschrieben [37]:

**Funktionalität** beschreibt die Fähigkeit eines Softwareproduktes Funktionen anzubieten, die festgelegte und vorausgesetzte Erfordernisse erfüllen, wenn die Software unter bestimmten Bedingungen verwendet wird.

Sub-Eigenschaften der Funktionalität sind:

- **Tauglichkeit**  
Das Vorhandensein von geeigneten Funktionen für die erforderlichen Aufgaben.
- **Genauigkeit**  
Das Liefern der richtigen oder vereinbarten Ergebnisse oder Effekte mit dem erforderlichen Grad an Präzision.
- **Interoperabilität**  
Die Fähigkeit, mit einem oder mehreren spezifizierten System zu kommunizieren.
- **Sicherheit**  
Die Fähigkeit, unauthorisierte Zugriffe auf Programme oder Daten zu verhindern.
- **Konformität**  
Die Fähigkeit, Standards, Konventionen oder Bestimmungen zu befolgen.

**Zuverlässigkeit** beschreibt die Fähigkeit eines Softwareproduktes, sein Leistungs-niveau unter angegebenen Bedingungen und für einen festgelegten Zeitraum zu erhalten.

Sub-Eigenschaften der Zuverlässigkeit sind:

- **Reife**  
Die Fähigkeit, Ausfälle zu vermeiden, die aufgrund von Fehlern in der Software entstehen würden.
- **Fehlertoleranz**  
Die Fähigkeit, ein gewisses Level an Leistung im Fall eines Software Fehlers aufrecht zu erhalten.
- **Wiederherstellbarkeit**  
Wird ausgedrückt durch die Fähigkeit, das Level an Leistung sowie Daten wiederherzustellen und der Zeit und dem Aufwand, der dafür benötigt wird.
- **Konformität**  
Die Fähigkeit, Standards, Konventionen oder Bestimmungen zu befolgen.

**Benutzbarkeit** beschreibt die Fähigkeit eines Softwareproduktes verständlich, erlernbar, benutzbar und für den Benutzer ansprechend zu sein, wenn die Software unter bestimmten Bedingungen verwendet wird.

Sub-Eigenschaften der Benutzbarkeit sind:

- **Verständlichkeit**  
Die Fähigkeit ermöglicht, dass ein Benutzer einfach versteht ob die Software geeignet ist und wie sie für bestimmte Aufgaben eingesetzt werden kann.
- **Erlernbarkeit**  
Die Fähigkeit ermöglicht, dass ein Benutzer die Anwendung erlernen kann.
- **Bedienbarkeit**  
Die Fähigkeit ermöglicht, dass ein Benutzer mit dem Produkt arbeiten und es steuern kann.
- **Konformität**  
Die Fähigkeit Standards, Konventionen oder Bestimmungen zu befolgen.

**Effizienz** beschreibt die Fähigkeit eines Softwareproduktes, angemessene Leistung zu erbringen, bezogen auf die Menge der verwendeten Ressourcen unter bestimmten Bedingungen.

Sub-Eigenschaften der Effizienz sind:

- **Zeitverhalten**  
Die Fähigkeit, angemessene Antwortzeiten, Verarbeitungszeiten und Durchsatzraten zu bieten, wenn die Aufgabe unter den gegebenen Bedingungen durchgeführt wird.
- **Ressourcenverhalten**  
Die Anzahl und die Art der Ressourcen die benötigt werden und die Dauer der Verwendung während die Funktion durchgeführt wird.
- **Konformität**  
Die Fähigkeit, Standards, Konventionen oder Bestimmungen zu befolgen.

**Wartbarkeit** beschreibt die Fähigkeit eines Softwareproduktes zur Modifikation. Modifikationen beinhalten Korrekturen, Verbesserungen oder Anpassungen der Software an Veränderungen der Umgebung sowie der Anforderungen und funktionalen Spezifikationen.

Sub-Eigenschaften der Wartbarkeit sind:

- **Analysierbarkeit**  
Die Fähigkeit, Mängel oder die Ursache für Ausfälle zu diagnostizieren bzw. die Teile die zu korrigieren sind zu identifizieren.
- **Veränderbarkeit**  
Die Fähigkeit, zu ermöglichen, dass eine spezielle Korrektur oder Veränderung implementiert werden kann.

- **Stabilität**  
Die Fähigkeit, unerwartete Effekte einer Modifikation an der Software zu vermeiden.
- **Testbarkeit**  
Die Fähigkeit, zur Validierung, d.h. zu überprüfen, ob das System die Anforderungen erfüllt.
- **Konformität**  
Die Fähigkeit, Standards, Konventionen oder Bestimmungen zu befolgen.

**Portabilität** beschreibt die Fähigkeit eines Softwareproduktes, von einer Umgebung in eine andere übertragen zu werden.

Sub-Eigenschaften der Portabilität sind:

- **Anpassbarkeit**  
Die Fähigkeit, die Funktionalität an verschiedene Umgebungen anzupassen, z.B. durch Generizität oder Parametrisierung.
- **Installierbarkeit**  
Die Fähigkeit, in einer spezifizierten Umgebung installiert werden zu können bzw. einem Mechanismus, der dies ermöglicht.
- **Koexistenz**  
Die Fähigkeit, mit anderen unabhängigen Softwareprodukten in einer gemeinsamen Umgebung zu existieren und gemeinsame Ressourcen zu verwenden und auszutauschen.
- **Austauschbarkeit**  
Die Fähigkeit, statt eines anderen Softwareproduktes für den selben Zweck in der selben Umgebung verwendet zu werden.
- **Konformität**  
Die Fähigkeit, Standards, Konventionen oder Bestimmungen zu befolgen.

[37]

## 3.3 Architekturmuster

Architekturmuster beschreiben grundlegende Organisationsstrukturen von Softwaresystemen sowie systemweite Eigenschaften. Bei Softwarearchitekturentwürfen können sie als eine Art Schablone angewendet werden. In diesem Abschnitt werden zwei der Muster vorgestellt, die bei der Umsetzung der elektronischen Gesundheitsakte für Nutztiere konkret zur Anwendung kommen.

### 3.3.1 MVC (Model-View-Controller)

Das Management von User Interfaces ist eigentlich eine schwierige Aufgabe für Entwickler. Es besteht eine erhebliche Komplexität in der Sicherstellung, dass Informationen dem Benutzer exakt angezeigt werden, dass weiters Benutzer die Anwendung

angemessen kontrollieren können und dass die Sichten und Domänenobjekte synchronisiert werden. Um diese Komplexität zu bewältigen, wurden Architekturen wie Model View Controller (MVC) entwickelt.

Dieser Architekturstil trennt die Software in drei Kategorien:

**Modelle**, die die Grundfunktionalität des Systems repräsentieren (z.B. ein Tier Objekt).

**Sichten**, den sogenannten Views, die diese Modelle dem Benutzer anzeigen, z.B. eine Tabelle, die die Attribute des Tieres speichert und dem Benutzer auf geeignete Weise darstellt. Es können von einem Modell mehrere Sichten vorhanden sein.

**Steuerungen**, den sogenannten Controllern, die bestimmen, wie der Benutzer mit der Sicht interagiert und die Daten der Modelle oder Sichten manipuliert. [39]

Abbildung 3.1 zeigt die Trennung zwischen Modell, Sicht und dem Controller.

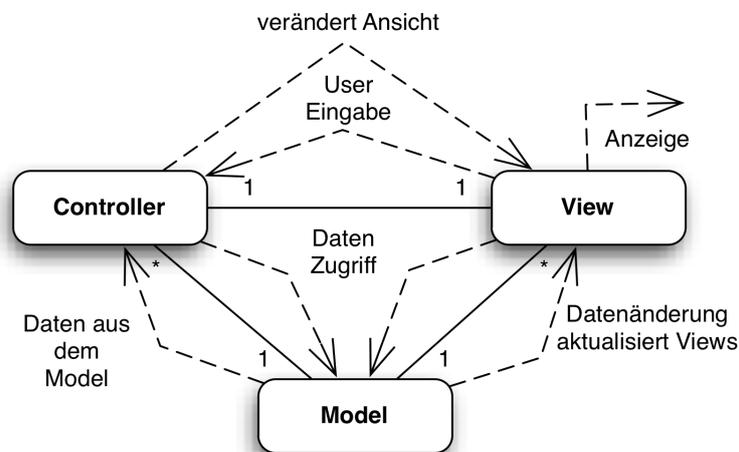


Abbildung 3.1: Multiplizitäten und Nachrichtenfluss in einem MVC Modell [61]

### 3.3.2 N-Tier-Architektur

Die Anzahl der Schichten einer Architektur beschreibt, wie der Client mit dem Server kommuniziert. Der Term Tier ist hier die Bezeichnung für eine Schicht bzw. wird im Folgenden auch Layer genannt. Die verschiedenen Formen der N-Tier-Architektur werden von Jasni Mohamad Zain et al. [69] beschrieben:

**1-Tier-Architektur** bedeutet, dass sämtliche Logik und Präsentation der Daten auf einer Maschine stattfindet. Zugriffe auf Dateien finden lokal oder über freigegebene Laufwerke statt. Diese Form der Architektur ist einfach aber unsicher. Daten können leicht verloren gehen, da sie durch den direkten Zugriff leicht verschoben oder gar gelöscht werden könnten. Weiters ist die gleichzeitige Bearbeitung durch mehrere Benutzer nicht möglich.

**2-Tier-Architektur** bedeutet, dass es eine getrennte Client- und Serverschicht gibt. Auf dem Client läuft die gesamte Applikation inklusive Logik und Datenverarbeitung und auf dem Server liegt die Datenbank. Beim Start des Programms auf der Clientseite wird eine Verbindung zum Server aufgebaut, wodurch die Kommunikation stattfinden

kann. Die Daten auf dem Server sind dadurch sicher und das Bearbeiten oder Löschen ist nur mit entsprechenden Benutzerrechten möglich. Der Zugriff auf die Datenbank kann von mehreren Benutzern gleichzeitig erfolgen. Der Client wird Fat-Client oder auch Thick-Client genannt, da er die Präsentation der Daten zusammen mit der Applikationslogik ausführt. Dadurch entsteht eine Menge Traffic über das Netzwerk, was mit langsamen Internetverbindungen ein Problem darstellen kann.

**3-Tier-Architektur** beinhaltet im Vergleich zu den beiden zuvor genannten Architekturen einen zusätzlichen Layer zwischen dem Client- und der Datenbankschicht. Durch diesen Zwischenlayer beschränkt sich die Aufgabe des Clients nur noch auf die Anzeige der Daten, genannt Thin-Client, wie z.B. bei einem Webbrowser. Die 3-Tier-Architektur ist besser skalierbar als die anderen Lösungen, da hier einfach mehrere Businesslogikschichten auf jeweils eigenen Servern laufen können, um eine gute Performance zu gewährleisten.

**4-Tier-Architektur** basiert auf der 3-Tier-Architektur und hat eine zusätzliche Schicht, die Datenzugriffsschicht, zwischen der Businesslogikschicht und der Datenbankschicht eingeführt. Diese zusätzliche Schicht ist verantwortlich für die Verwaltung und die Kontrolle der Datenbankschicht.

**5-Tier-Architektur** besitzt sämtliche Schichten der 4-Tier-Architektur jedoch mit einer zusätzlichen Schicht auf oberster Ebene, also über der Präsentationsschicht. Dieser Layer ist die Visualisierungsschicht und kümmert sich um die Formatierung des Layouts der Applikation. Da heutzutage Applikationen nicht nur über Standrechner sondern auch über mobile Geräte wie Tablet PC's oder Smartphones bedient werden, kümmert sich der Visualisierungslayer um die Anzeige auf den verschiedenen Geräten.

[69]

In der Arbeit von Paul D. Manuel und Jarallah AlGhamdi [40] wurden die besonderen Eigenschaften einer N-Tier-Architektur erarbeitet und werden hier nachfolgend aufgezählt:

- **Autonom**  
Ein Server muss so autonom sein, dass er seine eigenen kritischen Ressourcen schützen kann und in kritischen Momenten eigene Entscheidungen trifft.
- **Bedienbarkeit**  
Die Architektur soll den Benutzer bei der Ausführung seiner Aufgaben effizient und effektiv unterstützen.
- **Verteilbarkeit**  
Die Clients und Komponenten der Architektur sollen effizient Daten verarbeiten, wenn Hardware Plattformen über das Netzwerk verteilt sind.
- **Interoperabilität**  
Die Anwendungen innerhalb der Architektur sollen konsistent zusammenarbeiten, um die Aufgaben der Benutzer des Systems auszuführen.

- **Standardisierung**  
Die Komponenten der Architektur sollen auf weit verbreiteten Softwarestandards basieren.
- **Erweiterbarkeit**  
Die Architektur soll einfach anpassbar sein, um neuen und sich ändernden Anforderungen gerecht zu werden.
- **Internationalisierbarkeit**  
Die Applikation muss die Informationen in allen Sprachen und Formaten der Länder und Kulturen anzeigen, in denen sie verwendet wird.
- **Handhabbarkeit**  
Die Systemadministratoren sollen in der Lage sein, die Ressourcen der Rechenumgebung wirtschaftlich zu konfigurieren, zu diagnostizieren, zu überwachen, zu warten und zu kontrollieren.
- **Portabilität**  
Das System muss einfach von einem System auf ein anderes portiert werden können.
- **Skalierbarkeit**  
Die Applikation muss effizient mit wachsenden Benutzerzahlen und wachsenden unternehmerischen Anforderungen umgehen können.
- **Sicherheit**  
Die Architektur muss die Informationen und die Rechnerressourcen vor unauthorisierten Zugriffen schützen.
- **Wiederverwendbarkeit**  
Die Wiederverwendung von existierenden Softwarekomponenten ist der Schlüssel zur effektiven Nutzung von wertvoller Software Engineering Erfahrung.
- **Zuverlässigkeit**  
Qualitatives Design von Objektklassen und Frameworks mit Fokus auf maximaler Wiederverwendbarkeit erhöht die Zuverlässigkeit des resultierenden Softwaresystems.

### 3.4 Datenmodellierung

Ein Datenmodell ist eine konzeptionelle Darstellung der Datenstrukturen einer Datenbank. Die Datenstrukturen bestehen aus Datenobjekten, den Beziehungen zwischen den Datenobjekten und den Regeln die deren Verwendung bestimmen. Das Datenmodell konzentriert sich auf die benötigten Daten und wie diese organisiert werden sollten und ist unabhängig von Hardware oder Software Einschränkungen. Die Daten werden nicht so ausspezifiziert angezeigt, wie eine Datenbank sie tatsächlich benötigt, sondern repräsentiert sie als Konzept für den Benutzer.

Grechenig et al [27] schreibt dazu folgendes: „Die Datenmodellierung beginnt bereits bei der Analyse und erstreckt sich bis hin zum konzeptionellen Entwurf eines implementierungsfähigen Datenbankmodells und bildet dabei die Grundlage für die Entwicklung

von persistierenden Informationssystemen. Somit bestimmt ein Datenmodell auch langfristig die Qualität eines Softwareproduktes.“

Bei der Datenmodellierung kommt gängigerweise das Entity-Relationship-Modell (ER-Modell) zum Einsatz. Das ER-Modell definiert die Methode, die während der Analyse und dem Design von informationsbasierten Systemen verwendet wird. Datenbankentwickler verwenden diese Methode oft um Anforderungen zu gewinnen und die Architektur eines Datenbanksystems zu entwerfen. Das Ergebnis dieser Methode ist eine Liste von Entitäten, Relationen und Bedingungen.

Die Modellierung basiert auf Artefakten, die entweder physische Artefakte repräsentieren wie Produkte oder Angestellte oder die die Vorgänge zwischen den Artefakten repräsentieren, wie eine Bestellung oder eine Lieferung. Weiters fokussiert sich die ER-Modellierung auf die Beziehungen zwischen den Artefakten. Diese Beziehungen können entweder binär sein, also zwei Artefakte verbinden oder ternär und zwischen mehreren Artefakten sein. [26]

Die Hauptelemente eines ER-Modells sind:

- Entitäten  
Beispiele für Entitäten wären Landwirte oder Tiere. Eine spezielle Ausprägung von einem Landwirten wäre z.B. die Instanz Max Mustermann.
- Attribute  
Die Struktur einer Entität wird durch Attribute definiert. Ein Attribut kann als Eigenschaft einer Entität gesehen werden, wie bei einem Landwirten z.B. Name, Titel, Adresse, Betriebsnummer, Geburtsdatum oder zugehöriger Kontrollverband. Da die Unterscheidung bei Entitäten durch die verschiedenen Werte ihrer Attribute gemacht wird muss sichergestellt werden, dass die Werte der Attribute einer Entität immer eindeutig sind, d.h. ein Betrieb hat immer eine eindeutige Namens- und Betriebsnummern Kombination.
- Beziehungen  
Mit Beziehungen werden aussagekräftige Verbindungen zwischen den Entitäten definiert, es existieren aber nicht zwischen allen Paaren von Entitäten Verbindungen. Als Beispiel wäre eine Beziehung zwischen Landwirt und Tier „gehört“, d.h. ein Tier gehört einem Landwirten.

[26] Spezialisierungen und Generalisierungen können komplexere Datenstrukturen in technischen Anwendungen darstellen. Mittels Spezialisierung können Entitäten definiert werden, die ein spezifisches Segment einer darüber geordneten Entität definieren. Die spezialisierte Entität erbt die Struktur und das Verhalten von der darüber geordneten Entität, erweitert diese und ist damit nie weniger als die Oberklasse. Eine Entität beschreibt somit ein allgemeineres Konzept als eine andere Entität.

Generalisierungen stehen dazu im Gegensatz und stellt die umgekehrte Beziehung der Generalisierung dar. Die generalisierte Entität (die Oberklasse) repräsentiert eine allgemeine Struktur und ein Verhalten für alle Untertypen und beinhaltet alle der allgemeinen Attribute der Unterklasse. Die Darstellung im ER-Diagramm erfolgt mittels einer „ist-ein“ Beziehung zwischen den Entitäten.

Beispiele für eine Generalisierung / Spezialisierung wären z.B. Kunde - ist-eine - Person oder Tier - ist-ein - Lebewesen. [26]

Die Beziehung zwischen Entitäten können mit Kardinalitäten weiter spezifiziert werden. Die Kardinalität gibt an, wieviele Entitäten eines bestimmten Typs mit wievielen Kardinalitäten eines anderen Typs in Verbindung stehen können. Es können drei verschiedene Fälle auftreten:

- 1:1  
Eine Entität steht mit genau einer anderen Entität in Verbindung.
- 1:N  
Eine Entität steht mit keiner oder mehreren anderen Entitäten in Verbindung. Die Entität auf der anderen Seite steht mit genau einer Entität in Verbindung.
- N:M  
Es können auf beiden Seiten keine oder mehrere Entitäten in Verbindung stehen.

Das Datenmodell der elektronischen Gesundheitsakte für Nutztiere findet sich in Kapitel 5.3.

### 3.5 ORM (Object-Relational-Mapping)

Eine große Zahl von Anwendungen die heutzutage erstellt werden, vor allem Webanwendungen, beruhen auf zwei weit verbreiteten Technologien: Objektorientierter Programmierung (OOP) und relationalen Datenbanksystemen. Diese Art von Datenbanksystemtechnologien speichern Daten in tabellarischen Strukturen, erfassen die Beziehungen zwischen diesen Tabellen und unterstützen Abfragen auf den gespeicherten Daten.

In [3] steht dazu, dass die Technologien von OOP und relationalen Datenbanken voneinander verschieden aber trotzdem auf zwei Arten verwandt sind. Sie ergänzen sich gegenseitig, da ein

Relationales Datenbankmanagementsystem (RDBMS) ein Framework bietet, damit objektorientierte Applikationen ihre Daten speichern können und umgekehrt die objektorientierte Anwendung ein Interface bereithält zwischen den Daten und dem Benutzer und so die Bearbeitung und Manipulation der Daten ermöglicht.

Weiters existiert eine deutliche Übereinstimmung zwischen Datenbanktabellen auf der einen und objektorientierten Klassen auf der anderen Seite. Um als Beispiel das System der elektronischen Gesundheitsakte für Nutztiere heranzuziehen, würde die Datenbank eine „Betrieb“ Tabelle beinhalten mit beispielsweise den Spalten Betriebsnummer, Name, Adresse und einer korrespondierenden „Betrieb“ Klasse im Applikationscode mit den Variablen Betriebsnummer, Name, Adresse.

Objekt-Relationales Mapping (ORM) ist eine Technologie, die es anstrebt, die Brücke zwischen der objektorientierten Welt und der relationalen Welt zu automatisieren, und die Duplizierung von Daten, die damit verbundenen Wartungskosten sowie die Fehleranfälligkeit zu beseitigen. [3]

### 3.6 UML

UML ist eine standardisierte allgemeine Modellierungssprache im Bereich der Informatik sowie des Software Engineerings. Der Standard wurde von der Object Management

Group [47] entwickelt. Er repräsentiert zwei verschiedene Sichten eines Systemmodells [55]:

1. Statische oder strukturelle Sicht  
Diese Sicht hebt die statische Struktur eines Systems unter Verwendung von Objekten, Attributen, Abläufen und Beziehung hervor. Die strukturelle Sicht beinhaltet Anwendungsfalldiagramme, Klassendiagramme und Kompositionsstrukturdiagramme.
2. Dynamische oder Verhaltenssicht  
Sie hebt das dynamische Verhalten des Systems durch die Modellierung der Zusammenarbeit von Objekten und Veränderungen des internen Zustandes von Objekten hervor. Diese Sicht beinhaltet Sequenzdiagramme, Aktivitätsdiagramme und Zustandsdiagramme.
3. Softwareorganisatorische Sicht  
Diese Sicht stellt die Untergliederung der Modelle selbst dar. Modelliert wird sie mittels Paketdiagrammen, Komponentendiagrammen und Verteilungsdiagrammen.

UML dient dem Zweck der Modellierung von nebenläufigen oder verteilten Anwendungen. [55]

UML ist eine ausdrucksstarke Modellierungssprache, mit der verschiedene Arten von Modellen erstellt werden können, um ein Software-System zu spezifizieren, zu konstruieren, zu visualisieren sowie zu dokumentieren. Nachfolgend werden die Diagrammartentypen näher beschrieben, die für den Entwurf der Architektur der ELGA für Nutztiere in Kapitel 4 sowie 5 verwendet werden.

### 3.6.1 Anwendungsfalldiagramm

Anwendungsfalldiagramme bestehen aus Akteuren, Anwendungsfällen sowie den Beziehungen zwischen den Anwendungsfällen, wobei eine funktionale Zerlegung des zu entwickelnden Systems erfolgt.

**Akteure** sind Subjekte, die mit dem System interagieren, indem sie Anwendungsfälle ausführen. Die Darstellung der Kommunikation eines Akteurs mit dem System wird im Diagramm durch eine Assoziation mit den Anwendungsfällen modelliert, wobei ein Akteur mit mindestens einem aber auch beliebig vielen Anwendungsfällen assoziiert werden kann. Ein Akteur ist nicht unbedingt ein menschlicher Benutzer, es kann sich hierbei auch z.B. um das E-Mail System der Anwendung handeln, die einen Anwendungsfall anstößt.

**Anwendungsfälle** repräsentieren die funktionalen Anforderungen des zu entwickelnden Systems und beschreiben somit das erwartete Verhalten. Sie lassen sich in der Analysephase durch die Problemstellung der Domäne sowie durch die Wünsche des Kunden feststellen.

**Beziehungen** zwischen Anwendungsfällen können als „include“, „extend“, oder als Generalisierung modelliert werden.

Eine „include“ Beziehung bedeutet, dass ein Anwendungsfall A einen anderen Anwendungsfall B inkludiert, d.h. A benötigt B unbedingt um korrekt zu funktionieren, B kann allerdings auch separat ausgeführt werden.

Eine „extend“ Beziehung bedeutet, dass ein Anwendungsfall B einen anderen Anwendungsfall A erweitert. Beide Anwendungsfälle können separat ausgeführt werden. Der Anwendungsfall B, der A erweitert, kann - muss aber nicht - von A zur Ausführung angestoßen werden.

In Abbildung 3.2 sind eine „include“ sowie eine „extend“ Beziehung zu sehen.

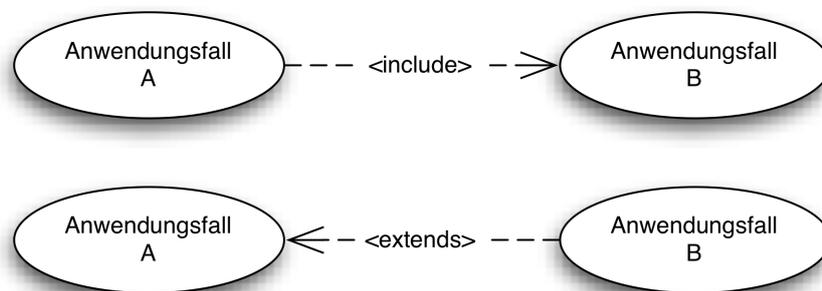


Abbildung 3.2: Include und Extend Beziehung in einem Anwendungsfalldiagramm

Die Generalisierung von Anwendungsfällen ermöglicht es, abstrakte Anwendungsfälle in spezielle zu unterteilen. Der spezielle Anwendungsfall erbt dabei das Verhalten sowie alle Beziehungen des abstrakten Anwendungsfalles.

In der Arbeit von A. Dedeker und B. Liebermann [11] geht es um den Use Case Ansatz, der aufgrund der breit gefächerten Interessen der Stakeholder doch eine deutliche Herausforderung darstellt. Zuerst muss diese Technik Informationen für eine verschiedenartige Gruppe von Leuten bieten, die in unterschiedlichen Projektphasen an das Projekt herantreten. Zum Beispiel müssen die Use Cases zu Projektbeginn eine vollständige Beschreibung des Systemverhaltens darstellen, jedoch dabei gleichzeitig für Kunden leicht verständlich sein. Weiters müssen sie die Problem Domäne, die Systemanforderungen und die Kundenerwartungen korrekt und vollständig beschreiben. Derzeit wird mit Use Case Diagrammen das zukünftige System auf einer relativ hohen, konzeptionellen Ebene beschrieben.

Während der Design- und Analysephase wird das Use Case Modell zur Basis für die Identifikation von Analyse- und Designklassen sowie von Subsystemen. Diese werden aufgrund des Verhaltens erstellt, welches aus den Objekten des Use Case Modells abgeleitet wird. Während der Implementierungsphase werden die Designklassen und Subsysteme als Basis für die Entwicklung des Codes, der Daten und der rechnerischen Interaktionen von den Entwicklern verwendet. Weiters haben A. Dedeker und B. Liebermann Vorteile von Use Case Diagrammen identifiziert, die wie folgt beschrieben werden können [11]:

- Architekten, Entwickler und Analysten gewinnen bereits früh im Projekt ein besseres Verständnis der Verbindung zwischen den Objekten der Domäne und den funktionalen Anforderungen.
- Die Darstellung von potentiellen Systemobjekten basierend auf der Sicht des Analysten der Problemdomäne können das objektorientierte Denken in die funktionale dekompositionsorientierte Use Case Diagrammtechnik einführen.
- Systemanalysten erhalten Hilfe bei der Überwindung der Kritik, dass Anforderungen eine unklare semantische Bedeutung haben.
- Architekten können die Wahrscheinlichkeit reduzieren, dass sie ein Systemprojekt auf eine Weise an Teams übergeben, die die negativen Auswirkungen der funktionalen Dekomposition verstärkt.

Die Zuweisung von einem Projekt zu Entwicklungsteams, die ausschließlich auf der Zerlegung der funktionalen Anforderungen - also deren Dekomposition - basiert, verursacht mindestens zwei ernsthafte Probleme. Erstens sind die Softwareteams gefährdet, Arbeitsaufwände dupliziert durchzuführen, weil gemeinsame Objekte aus den Anforderungen aufgrund der Zerlegung mehrfach entworfen werden. Zweitens werden solche Projekte bei der Integration vor einer Herausforderung stehen. Mit einem geeigneten Use Case Diagramm können Projekt Manager die Aufgaben den Teams so zuweisen, dass dabei die internen Objekte sowie die zugehörigen funktionalen Anforderungen berücksichtigt werden. [11]

### 3.6.2 Aktivitätsdiagramm

Aktivitätsdiagramme präsentieren den Kontroll- und Datenfluss zwischen Aktionen auf prozedurale Weise. Laut [5] können Aktivitätsdiagramme dazu verwendet werden, um Anwendungsfälle zu modellieren. Üblicherweise ist es für eine Anwendungsfallbeschreibung notwendig, die beteiligten Akteure zu identifizieren und den Aktivitätsfluss aufzuzeigen.

In dieser Art von Diagrammen werden die Aktivitäten identifiziert, die benötigt werden, um einen Use Case auszuführen und den dazugehörigen Beziehungen. Außerdem können die Objekte bestimmt werden, bei welchem Anwendungsfall sie beteiligt sind und wie sich ihre Rolle, ihr Zustand und ihre Attribute ändern.

Aktivitätsdiagramme bestehen aus:

#### **Aktivität**

Eine Aktivität spezifiziert eine Abfolge von Aktionen, Kontroll- und Datenflüssen.

#### **Aktion**

Aktionen beschreiben ein Verhalten.

#### **Objektknoten**

Objektknoten repräsentieren die Daten, die während eines Ablaufes innerhalb der Aktivität fließen.

### Kanten

Kanten dienen der Verbindung zwischen zwei Knoten und bilden dadurch einen Übergang. Es gibt zwei Arten von Kanten.

Kontrollflusskanten bilden einen Fluss zwischen einer Vorgängeraktionen und einer Nachfolgeraktion.

Objektflusskanten transportieren beim Fluss von einem Knoten zum anderen Daten.

### Kontrollelemente

Der Aktivitätsablauf wird durch einen Initialknoten gestartet und durch einen Endknoten beendet. Weiters wird das Ende eines Kontrollflusses durch einen eigenen Endknoten markiert.

Alternativentscheidungen bei Abläufen werden durch Verzweigungs- und Verbindungsknoten dargestellt. Die Entscheidung welche Kante gewählt wird, kann durch Bedingungen festgelegt werden.

Nebenläufige Abläufe sind dazu da, um gleichzeitig eingehende oder ausgehende Abläufe zu vereinigen bzw. aufzuteilen.

In Abbildung 3.3 sind die vorgestellten Konzepte dargestellt.

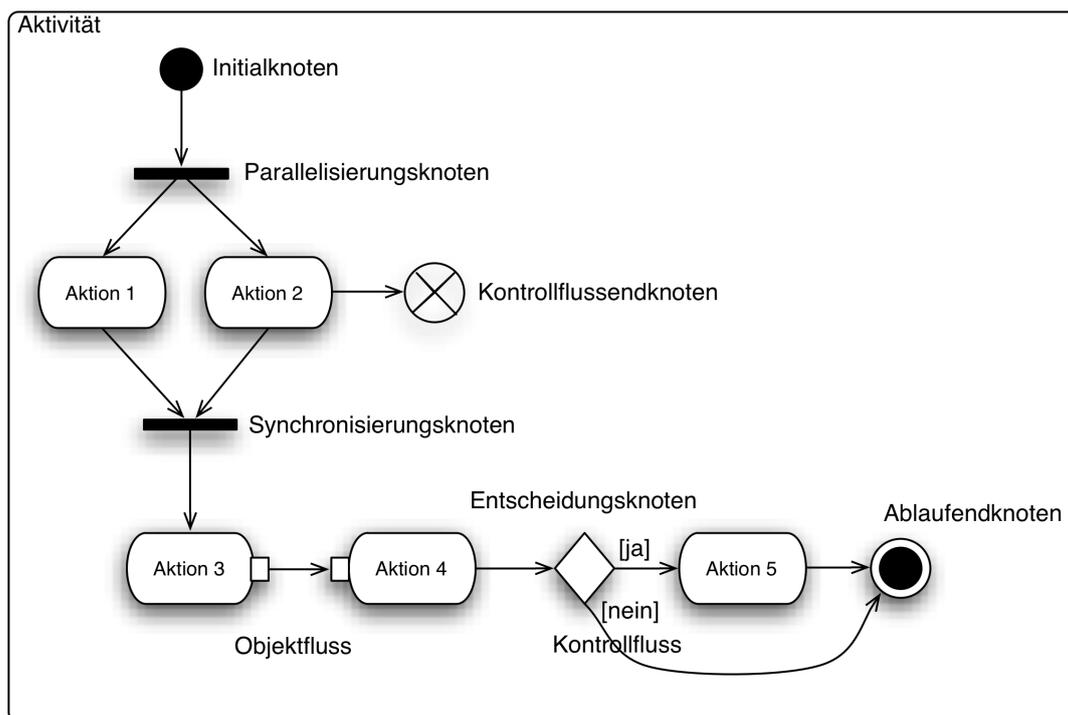


Abbildung 3.3: Konzepte in einem Aktivitätsdiagramm

### 3.6.3 Komponentendiagramm

Das Komponentendiagramm dient zur Darstellung der Struktur eines Systems und beschreibt die Komponenten, die die Anwendung bilden. Aber nicht nur die Komponenten werden modelliert, sondern auch deren Beziehungen und Interfaces, über die sie miteinander kommunizieren.

Wenn Applikationsentwickler eine Anwendung aus Komponenten entwickeln benötigen sie eine Übersicht über die Architektur der Applikation, sodass Designentscheidungen darauf aufbauen können. [38]

Die drei Elemente dieser Diagrammart sind Komponenten, Interfaces und Abhängigkeiten, die eine Beziehung der zwei zuvor genannten Elemente darstellen. Ein Beispiel hierfür ist in Abbildung 3.4 dargestellt. Die Komponenten K1 und K2 sind Abhängig voneinander über Interface I.

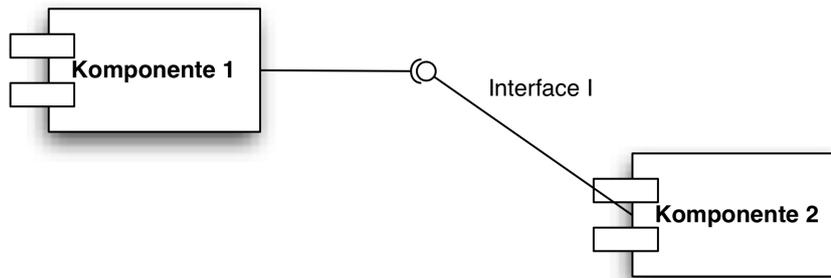


Abbildung 3.4: Typisches UML Komponentendiagramm [38]

### 3.6.4 Verteilungsdiagramm

Verteilungsdiagramme werden verwendet, um die Konfiguration der Elemente zur Laufzeit zu modellieren sowie die Softwarekomponenten die darin existieren. Es werden die physikalischen Knoten dargestellt und deren Assoziationen untereinander. Jeder Knoten kann Instanzen von Komponenten zur Laufzeit beinhalten und dadurch anzeigen, dass diese Komponente in diesem Knoten „lebt“. Optional können auch die Objekte innerhalb der Komponenten modelliert werden. Sie stellen also dar, wie die Software auf der Hardware verteilt ist.

## 3.7 Webservices

Webservices erlauben unterschiedlichen Applikationen auf unterschiedlichen Plattformen die Kommunikation miteinander. Es handelt sich um gekapselte, lose gekoppelte, vertraglich geregelte Funktionen, die über Standard Protokolle definiert sind, wobei [28]:

- gekapselt bedeutet, dass die Implementierung der Funktionen von außen nicht sichtbar ist;
- lose gekoppelt bedeutet, dass die Veränderung der Implementierung der einen Funktion keine Veränderung der aufrufenden Funktion verlangt;
- vertraglich geregelt bedeutet, dass öffentlich verfügbare Beschreibungen des Funktionsverhaltens vorliegen, aus denen hervorgeht, wie die Funktion gebunden werden kann sowie welche Input und Output Parameter sie hat.

Webservices verwenden XML über Hypertext Transfer Protocol (HTTP) zur Kommunikation. XML ist ein weit verbreitetes Format für den Datenaustausch. Es bildet die fundamentale Basis für beinahe jeden anderen Layer der für Webservices verwendet wird. Die folgende Aufzählung zeigt von oben nach unten die Layer, die den so genannten Webservice-Stack bilden [28]:

- Universal Discovery Description Integration (UDDI)
- Web Services Definition Language (WSDL)
- Simple Object Access Protocol (SOAP)
- XML

Webservices werden bei der Umsetzung der elektronischen Gesundheitsakte für Nutztiere verwendet, um die Kommunikation zwischen dem System und Systemen von externen Anbietern von Gesundheitsdaten bzw. Praxismanagementsoftware von Tierärzten zu ermöglichen.

In den folgenden Abschnitten werden die Layer, die auf der XML Schicht liegen beschrieben. [28]

### 3.7.1 SOAP

Da laut [28] Webservices in einer heterogenen Umgebung laufen, müssen die Protokolle, die für den Datentransfer zwischen den Funktionen verantwortlich sind, unabhängig von jeglicher Laufzeitumgebung sein. SOAP ist ein Protokoll mit eben dieser Eigenschaft. Es definiert einen einfachen Mechanismus, um Applikationssemantik durch ein modular verpacktes Modell darzustellen und einen Verschlüsselungsmechanismus, um Daten innerhalb der Module zu verschlüsseln.

SOAP Nachrichten bestehen aus einem „Envelope“ Tag als Wurzelement, welches zwei weitere Elemente enthält: einen optionalen Header und einen Body. Der Header kann einige Headereinträge beinhalten, um Informationen zur Authentifizierung oder zur Verschlüsselung der Daten bereitzustellen. Der Body besteht aus der eigentlichen Nachricht, die einen Remote Procedure Call (RPC) darstellt, d.h. den Aufruf einer Funktion in einem anderen Adressraum. Für einen Methodenaufruf werden folgenden Informationen benötigt [28]:

- Die Uniform Resource Identifier (URI) der Resource.
- Der Methodenname.
- Eine optionale Methodensignatur.
- Die Parameter der Methode.
- Optionale Headerdaten.

Die URI der Ressource ist nicht innerhalb des Envelopes der Nachricht definiert, dies wird von der HTTP Request URI übernommen. Jede SOAP Nachricht beinhaltet genau einen Methodenaufruf. Somit wird also für einen Request ein Methodenaufruf in einer SOAP Nachricht verschickt und das Ergebnis der Funktion, der Response, wiederum über eine eigene SOAP Nachricht empfangen. [28]

### 3.7.2 WSDL

WSDL - Web Service Definition Language - beschreibt Netzwerkservices durch Verwendung einer XML Grammatik. Es stellt eine Dokumentation für verteilte Systeme bereit und hat das Ziel, die Kommunikation zwischen Applikationen zu automatisieren. Während SOAP die Kommunikation zwischen einem Requester und einem Provider definiert, beschreibt WSDL die Services, die von einem Provider angeboten werden und kann als Rezept verwendet werden, um geeignete SOAP Nachrichten zu erstellen, mit denen auf diese Services zugegriffen wird.

Die Dokumentstruktur eines WSDL Dokumentes besteht aus dem „Definitions“ Tag der die Wurzel darstellt und der sechs Elemente beinhaltet. Diese sechs Elemente können unterteilt werden in die abstrakte Definition sowie in die konkrete Definition. [28]

#### **Import Element**

Durch die Verwendung von Import Statements können WSDL Dokumente in mehrere unabhängige Dokumente aufgeteilt werden, wodurch die Struktur klarer und einfacher wartbar wird.

#### **Types Element**

Alle Definitionen der Datentypen, die für das Senden und Empfangen der Nachrichten notwendig sind, befinden sich innerhalb dieses Elementes. Laut Spezifikation soll dies mittels XML Schema Definitionen erfolgen.

#### **Message Element**

Innerhalb dieses Elementes werden die Daten definiert, die zwischen dem Service Requester und dem Service Provider ausgetauscht werden. Dieses Element beinhaltet das „Part“ Element, das auf das bereits definierte „Types“ Element referenziert.

#### **Port Type Element**

Dieses Element definiert eine Reihe von verwandten Operationen. Dafür werden innerhalb eines „operation“ Tags der Name der Funktion sowie Input und etwaige Output Parameter definiert.

#### **Binding Element**

Die bisher definierten Elemente beschreiben die Operationen auf eine generische Art. Es wurden keine Aussagen über die konkrete Implementierung gemacht. Im nächsten Schritt wird die Operation an das SOAP Protokoll gebunden, was innerhalb des Binding Elementes geschieht.

#### **Service Element**

Ein Service ist eine Reihe von verwandten Ports. Verschiedene Ports werden verwendet, wenn mehrere Implementierungen des selben Bindings existieren.

[28]

Beispiele für WSDL Dokumente sind auf der Homepage des World Wide Web Consortium (W3C) [64] zu finden.

### 3.7.3 UDDI

UDDI - Universal Description Discovery and Integration - ist ein Standard der entworfen wurde, um einen Verzeichnisdienst für Webservices zu liefern. UDDI ist ähnlich wie ein Telefonbuch aufgebaut und besteht aus folgender Aufteilung:

- **Yellow Pages**  
Hier können Anbieter von Services aufgefunden werden. Anbieter werden entsprechend bestimmter Informationen in Klassen eingeordnet, um den passenden Service schnell zu finden.
- **White Pages**  
Hier sind Informationen über den Serviceanbieter zu finden, wie z.B. Adresse, Kontaktdaten und bekannte Unternehmenskennzahlen.
- **Green Pages**  
Dieser Bereich beinhaltet technische Informationen über den Webservice, die vom Anbieter freigegeben wurden.

[28]

Abbildung 3.5 illustriert das Zusammenspiel zwischen einem Webservice Requester, einem Provider sowie dem UDDI Dienst.

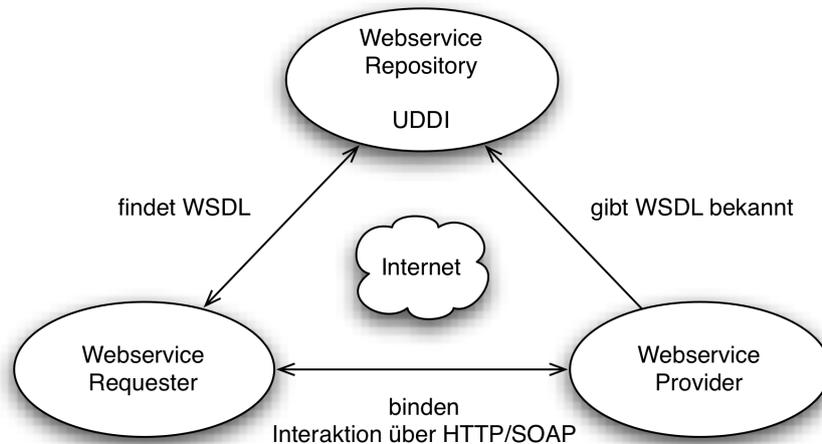


Abbildung 3.5: Zusammenspiel zwischen UDDI, Webservice Requester und Provider

## 3.8 Zugriffskontrolle

Im Kapitel Analyse 4 in Abschnitt 4.3 wird für das System der elektronischen Gesundheitsakte für Nutztiere ein Rollenkonzept erstellt. Das Rollenkonzept sorgt dafür, dass es ein System dahinter gibt, welcher Benutzer welche Teile der Applikation bzw. welche Daten und Informationen einsehen kann. Die theoretischen Grundlagen zur Zugriffskontrolle sowie welche Arten es davon gibt, sind im Folgenden nachzulesen.

Die Hauptaufgabe der Zugriffskontrolle ist die Verwaltung der Zugriffe und Rechte der Benutzer eines Systems um sicherzustellen, dass Daten, Informationen und Ressourcen

nicht unerlaubt verwendet werden. Zugriffskontrolle regelt auch die Einschränkung des Zugriffs eines Subjekts auf ein Objekt und kontrolliert Anfragen auf Daten gemäß der Identität des Benutzers [50].

Es wird zwischen Lese- und Schreibrechten unterschieden. Benutzer mit Leserecht auf bestimmte Systemteile können auf die dort zur Verfügung gestellten Informationen zugreifen, sie jedoch nicht verändern. Ein Benutzer mit Schreibrechten kann Dateien hinzufügen, verändern oder löschen. Im folgenden werden drei Arten der Zugriffskontrolle vorgestellt:

### 3.8.1 Discretionary Access Control (DAC)

Discretionary Access Control (DAC) ist eine benutzerbestimmbare Zugriffskontrolle und ermöglicht den Zugriff auf Objekte auf Basis der Identität des Benutzers oder der Benutzergruppe. Benutzer haben die komplette Kontrolle über alle Programme die sie besitzen und ausführen und können weiters ihre eigenen Rechte auf andere Benutzer übertragen. Betriebssysteme wie Linux, UNIX und Windows NT/SERVER haben diese Art der Rechteverwaltung. Dieses Modell ist nur für kleine, geschlossene Anwendungsumgebungen geeignet. [50]

### 3.8.2 Mandatory Access Control (MAC)

Mandatory Access Control (MAC) ist die zwingend erforderliche Zugriffskontrolle und ist strenger als DAC. Hierbei wird jeder Benutzer und jedes File ausgestattet mit einer bestimmten Sicherheitsklasse und Benutzer können ihre eigenen Sicherheitsklassen oder die von anderen nicht verändern. Nur der Administrator des Systems kann die Zugriffsrechte an Benutzer oder Benutzergruppen vergeben. Durch Vergleiche mit den Sicherheitsklassen eines Benutzers oder denen der Dateien auf die zugegriffen werden soll entscheidet das System, ob der Zugriff erlaubt ist.

Zu den Sicherheitsklassen von MAC gehören streng geheim (TS = top secret), geheim (S = secret), vertrauenswürdig (C = confidential), eingeschränkt (R = restricted) und uneingeschränkt (U = unrestricted). Dabei gilt  $TS > S > C > R > U$ .

MAC wird dort verwendet, wo eine einfache Sicherheitsstrategie ausreichend ist, die Sicherheitsregeln eindeutig sind und die Richtung des Informationsflusses relativ fixiert ist, wie bei Domänen des Militärs, jedoch nicht in großen, verteilten und komplexen Umgebungen. [50]

### 3.8.3 Role Based Access Control (RBAC)

Der Kerninhalt von Role Based Access Control (RBAC), also der rollenbasierten Zugriffskontrolle ist es, Zugriffsberechtigungen an bestimmte Rollen zu vergeben. Benutzer können mehr als eine Rolle besitzen. Jedes Subjekt kann nur seine eigenen geregelten Zugriffe ausführen. Jede Operation im System die durchgeführt werden soll, muss mit der Rolle übereinstimmen, d.h. ein Benutzer kann die Funktionen ausführen, die die Rolle anbietet. RBAC ist eine sehr effektive Zugriffskontrolle mit vielen Charakteristiken wie Flexibilität, Einfachheit, Sicherheit und ist weit verbreitet in der Rechteverwaltung in großen Datenbanksystemen. [50]

Abbildung 3.6 liefert eine bildliche Darstellung der Beziehungen zwischen Benutzern,

Rollen/Gruppen, Abbildungen und System-Objekten [18]. RBAC besteht somit aus einer Menge von Subjekten (Benutzern), einer Menge von Objekten (Dateien, Ressourcen), einer Menge von Rollen und einer Menge von Zugriffsrechten (lesend, schreibend, kein Zugriff). Mehrere Benutzer können die gleichen Rollen zugewiesen haben, es kann aber auch ein Benutzer mehrere Rollen besitzen, abhängig von den zu erfüllenden Aufgaben.

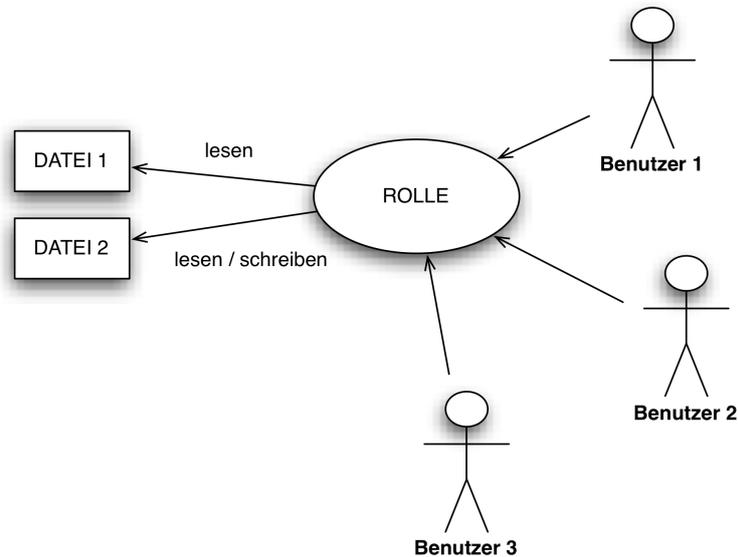


Abbildung 3.6: Rollenmodell der rollenbasierten Zugriffskontrolle

# 4 Analyse

Dieses Kapitel wird eine strukturierte Analyse der Domäne durchführen. Dabei wird das Problemumfeld untersucht und durch Benutzerbefragungen sowie Anforderungsanalysen ein Analysemodell entwickelt. Dieses Modell dient als Übergang zur Designphase und soll beschreiben, wie das Problem gelöst werden kann.

## 4.1 Stakeholder

Zu den Stakeholdern eines Projektes gehören Sponsoren, Auftraggeber, Entwickler sowie Leute die spezielles Know-How zur Verfügung stellen damit das Endprodukt entstehen kann, wie z.B. Netzwerkspezialisten, Veterinärmediziner oder Grafik Designer. Weitere Beteiligte können dem Wartungs- oder Schulungspersonal angehören oder ein Käufer, Systembetreiber, eine Marketingabteilung oder ein Projektgegner sein [27]. Im folgenden Abschnitt wird jedoch nur auf die Endbenutzer der elektronischen Gesundheitsakte für Nutztiere eingegangen sowie auf Akteure, die die gesetzlichen Rahmenbedingungen definieren und die Einhaltung dieser kontrollieren.

### **Tierärzte und Tierkliniken**

Nach § 12. (1)

„Folgende Tätigkeiten dürfen unbeschadet der anderen Personen gemäß § 1 Abs. 3 zustehenden Befugnisse nur von Tierärzten ausgeübt werden (vorbehaltene Tätigkeiten):

1. Untersuchung und Behandlung von Tieren;
2. Vorbeugungsmaßnahmen medizinischer Art gegen Erkrankungen von Tieren;
3. operative Eingriffe an Tieren;
4. Impfung, Injektion, Transfusion, Infusion, Instillation und Blutabnahme bei Tieren;
5. Verordnung und Verschreibung von, Arzneimitteln für Tiere;
6. Schlachttier- und Fleischuntersuchung;
7. Ausstellung von tierärztlichen Zeugnissen und Gutachten;
8. künstliche Besamung von Haustieren."

Tierärzte sind für Betriebe zuständig, behandeln Tiere, erstellen Diagnosen, verschreiben Arzneimittel und Behandlungen. Diese Abläufe könnten im System der ELGA für Nutztiere erfasst werden um so die lebenslange Krankengeschichte eines Tieres verfügbar zu machen.

**Landwirte**

Landwirte, die Nutztiere halten und dadurch zu deren Aufgaben die Erzeugung von tierischen Nahrungsmitteln wie Milch, Eier oder Fleisch zählen, sind besonders an der Gesundheit ihrer eigenen Tiere interessiert.

**Weiterverarbeitende Industrie**

Zu diesen Stakeholdern gehören Schlachtbetriebe, Molkereien, Fischereien usw. Jene, die die tierischen Produkte bevor sie zum Endverbraucher gelangen be- und weiterverarbeiten.

**Konsumenten**

Zu den Konsumenten zählen nicht nur die Kunden, die die Ware im Geschäft kaufen, sondern auch schon Handelsbetriebe, die sie zum Verkauf anbieten.

**Agrarmarkt Austria - AMA**

Die Hauptaufgabe der Agrarmarkt Austria Marketing GmbH ist die Förderung des Absatzes verschiedener landwirtschaftlicher Produkte wie Fleisch, Eier, Milchprodukte usw. Weiters ist sie für die Registrierung der Tiere verantwortlich, für Ohrmarkenvergabe, Geburtsmeldungen usw. Die AMA ist in diesem Projekt das Exekutivorgan und stellt erforderliche Daten über den Tierbestand zur Verfügung.

**Verbände und Organisationen**

In Österreich existiert eine Vielzahl verschiedener Verbände, Organisationen, Arbeitsgemeinschaften usw. für landwirtschaftliche Nutztiere deren Ziele die Leistungskontrolle, Verbesserung der Tiergesundheit oder die Zucht und Optimierung bestimmter Rassen sein können. Ein Beispiel für einen Verband wird im Folgenden vorgestellt:

- **Landeskontrollverbände - LKV**

Zu den Aufgaben der acht Landeskontrollverbände die in Österreich existieren, gehört die Erfassung diverser Daten sowie die Überprüfung der Milch, die von den Milchkühen der Landwirte geliefert wird. Die Mitgliedschaft in einem LKV ist freiwillig und nicht gesetzlich vorgeschrieben, es sind jedoch trotzdem die meisten Milchbauern Mitglied in einem Kontrollbetrieb. Die Berichte des LKV, die aufgrund einer monatlichen Melkung entstehen und Aussagen über die Qualität und Inhaltsstoffe der Milch enthalten, dienen den Landwirten als Entscheidungshilfe im Herdenmanagement bzw. um gesundheitsfördernde Maßnahmen zu treffen. Weiters sind diese Daten die Grundlage für eine gelungene Zuchtwertschätzung.

**Ministerien**

Die Ministerien gehören zu der rahmengebenden Kraft bei der Umsetzung der elektronischen Gesundheitsakte für Nutztiere. Sie definieren, welche Daten elektronisch erfasst werden dürfen oder müssen. Sie schaffen die gesetzliche Grundlage und kontrollieren die Einhaltung derselbigen.

- **Bundesministerium für Land- und Forstwirtschaft, Umwelt und Wasserwirtschaft**  
Dieses Ministerium [34] befasst sich unter anderem mit der Lebensmittelqualität

tierischer Produkte. Diese Qualität unterliegt verschiedenen Faktoren, die im Stall bei der Haltung und Gesundheit der Tiere sowie der verabreichten Futter- und Arzneimittel zu finden sind.

- **Bundesministerium für Gesundheit**  
Zu einem der Schwerpunkte dieses Ministeriums [22] zählt die Tiergesundheit mit Fokus auf der Überwachung und der Bekämpfung von Tierkrankheiten, Tierarzneimittelverabreichungen usw...

### **Tiergesundheitsdienst**

Ziele des TGD, der österreichweit in jedem Bundesland außer Wien eingerichtet wurde, sind die Tiergesundheit zu verbessern, bei Tierbeständen den Einsatz von Tierarzneimitteln zu minimieren, die Qualität tierischer Produkte zu erhöhen, die Betreuung einer geeigneten Seuchenprophylaxe und -bekämpfung, sowie die Beratung landwirtschaftlicher Tierhalter. Dem TGD können Tierärzte und tierhaltende Landwirte beitreten.

Es existieren verschiedene Programme, die Vorgehensweisen zur Bekämpfung von Krankheiten und zur Verbesserung und Erhaltung der Tiergesundheit [60] beschreiben. Diese Programme wurden von den verschiedenen Arbeitsgruppen erstellt. Tierhalter können an diesen Programmen teilnehmen und sind in diesem Fall vom Tiergesundheitsdienst zu registrieren.

Registrierte Besitzer von Nutztieren werden von Tierärzten, die ebenfalls teilnehmende des TGD sind beraten und unterstützt. Mehrmals im Jahr werden die Betriebe auf Einhaltung der gesetzlichen Vorschriften überprüft. Dabei wird kontrolliert, ob verabreichte Arzneimittel dokumentiert sowie Tierschutzbestimmungen und ausreichende Hygiene und Fütterung eingehalten werden. Diese Kontrollen werden vom Tierarzt durchgeführt, es können jedoch auch zusätzlich Kontrollen durch den Landeshauptmann oder der Bezirksverwaltungsbehörde erfolgen.

## **4.2 Anforderungsanalyse**

In diesem Abschnitt werden die Anforderungen an das System analysiert. Dabei ergeben sich eine Reihe nichtfunktionaler sowie funktionaler Anforderungen, die vom System erfüllt werden müssen.

### **4.2.1 Erhebung der Anforderungen**

Die Anforderungsanalyse ist ein iterativer Prozess. Zuerst werden die Anforderungen an ein System informell über Gesprächsführung analysiert und vereinbart. Im nächsten Schritt werden sie dokumentiert und danach validiert. Am Schluss der Erhebung wird entschieden, ob die Anforderungen bereits ausgereift genug sind, oder ob eine weitere Iteration notwendig ist um sie zu verfeinern und zu überarbeiten.

In dieser Arbeit wurde eine Iteration durchgeführt, weitere werden notwendig sein.

Ausgangspunkt der Anforderungsermittlung war ein Teil der im vorigen Abschnitt definierten beteiligten und betroffenen Stakeholder. Darunter befanden sich ein Tierarzt, zwei Domänenexperten und sechs Entwickler. Die Entwickler beschäftigen sich mit unterschiedlichen Schwerpunkten in dem Themengebiet der elektronischen Gesundheits-

akte. Zu den Schwerpunkten gehören Analyse der gesetzlichen Grundlagen, Stakeholderanalyse, Datenbankentwurf und Implementierung. Der Tierarzt und die Domänenexperten wurden in Interviews befragt, welche Anforderungen sie an die elektronische Patientenakte für Nutztiere haben. Aus den Ergebnissen der Interviews wurde zusammen mit den Entwicklern eine Menge von Anforderungen an das System erarbeitet, die mindestens erfüllt sein müssen. Die analysierten Softwarelösungen aus Abschnitt 2.1.2 die derzeit am Markt vorhanden sind, wirkten ebenfalls bei der Definition der Anforderungen mit ein.

Alle Stakeholder wurden in persönlichen Treffen interviewt. Weiters konnten Anforderungen auch mittels Literaturrecherche sowie über die in bestehenden Systemen umgesetzten Anforderungen bestimmt werden. Die definierten Anforderungen wurden anhand von User Interface Mockups zusammen mit den anderen Entwicklern geprüft.

Bei der Erhebung der Anforderungen wurde der Arbeitsablauf jedes Stakeholders mit dem System analysiert. Den befragten Stakeholder wurde weiters die Frage gestellt, wie der erforderliche Arbeitsablauf aus der Sicht des Tierarztes, des Betriebes, der Konsumenten, usw. aussehen könnte. Weiters wurden mittels Brainstorming Funktionen an das System ermittelt. Nach den Interviews wurden die Ziele der Personen definiert und nach Stakeholdergruppe zusammengefasst. Die aus analysierten Softwarelösungen erarbeiteten Anforderungen wurden durch Aussagen der befragten Personen verifiziert, erweitert und teilweise modifiziert, so dass sich die Ziele des Systems durch unterschiedliche Quellen entwickelten. In internen Diskussionen mit den Entwicklern wurde die Umsetzbarkeit in Bezug auf Technik und Ressourcen geklärt.

Zusammenfassend wurden folgende Techniken bei der Anforderungsanalyse angewandt:

- Interviews
- Diskussionen
- Brainstorming
- Ergebnisse aus der Domänenanalyse
- Vergleich mit am Markt vorhandener Software

### 4.2.2 Nichtfunktionale Anforderungen

In der traditionellen Softwareentwicklung sind nichtfunktionale Anforderungen meist Qualitätsattribute, die eine gewisse Herausforderung im Entwicklungsprozess darstellen. Sie beschreiben keine spezifischen Funktionalitäten, die das System haben muss. Die Wichtigkeit, sie zu definieren und beim Architekturentwurf sowie bei der Entwicklung zu berücksichtigen, wurde in der Softwareentwicklung bereits bewiesen. Es ist jedoch schwierig festzustellen, wann eine nichtfunktionale Anforderung vollständig umgesetzt wurde.

Weiters widersprechen sich nichtfunktionale Anforderungen in manchen Fällen. Diese Art von Konflikt ist schwer zu lösen und beeinflusst in weiterer Folge die funktionalen Anforderungen. Eine weitere Herausforderung wäre, dass sie sich auch auf Entitäten beziehen können, die im Stadium der Anforderungsanalyse noch nicht bekannt sind. [21]

Die nichtfunktionalen Anforderungen der elektronischen Gesundheitsakte für Nutztiere wurden als eine Mischung aus der in der Machbarkeitsstudie [25] der elektronischen Gesundheitsakte für Menschen definierten speziellen für dieses System anwendbaren Anforderungen sowie aus verschiedenen generischen Anforderungen an Systeme definiert. Daraus ergaben sich folgende zwölf Punkte, die nachstehender Aufzählung entnommen werden können. Diese Liste stellt eine erste Fassung der nichtfunktionalen Anforderungen dar, die in weiterer Folge noch erweitert und adaptiert werden kann.

### **Nichtfunktionale Anforderungen an die elektronische Gesundheitsakte für Nutztiere**

- Die ELGA für Nutztiere darf keine Kosten für Ärzte verursachen.
- Die Datenerfassung und die Datenabfrage darf zu keinem erhöhten Aufwand für Ärzte führen.
- Die Bedienung muss einfach und zeitsparend sein.
- Die Anwendung soll über das Internet in jedem gängigen Browser gestartet werden können.
- Für die Kommunikation der Systeme untereinander ist die Verschlüsselung mit SSL („secure socket layer“) bzw. TLS („transport layer security“) und gegenseitige Authentifizierung mit „client authentication“ zu sichern. Mehr dazu im Kapitel Security 5.7
- Antwortzeiten bei Suchabfragen sollen generell unter 5 Sekunden liegen. Bei der Anforderung von einfachen Textdokumenten können 5 Sekunden als Richtwert gelten. Für Bilddaten weniger als 1 Minute.
- Das System muss 24 h am Tag 7 Tage die Woche verfügbar sein.
- Die Komponenten des Systems müssen redundant z.B. auf mehreren Standorten verfügbar sein um die Ausfallsicherheit sowie die Lastverteilung als Vorteile daraus zu ziehen.
- Das System muss mit verschiedenen Nutzer-Kapazitäten zurecht kommen, da zu gewissen Tageszeiten eine maximale Anzahl an Benutzern darauf zugreifen wird.
- Das System muss gut skalierbar sein, d.h. mit steigender Last und Benutzeranzahl dürfen die erforderlichen Ressourcen nur im gleichen Verhältnis steigen.
- Das System muss einfach erweiterbar sein, z.B. Integration weiterer Nutztierarten oder die Implementierung zusätzlicher Funktionen sollen nahtlos in das System eingefügt werden können.
- Das System muss leicht anpassbar sein, wenn sich Anforderungen mit der Zeit verändern.
- Wenn gegebenenfalls Änderungen gemacht werden müssen, muss durch gewisse Kriterien, die bei der Entwicklung erfüllt werden müssen, die gute Wartbarkeit sichergestellt werden. Solche Kriterien wären beispielsweise eine gute Dokumentation des Codes, ein modular gegliederter Aufbau, eine große Anzahl automatisiert ausführbarer Tests oder die Verwendung von Softwareentwicklungsmustern.

### 4.2.3 Funktionale Anforderungen

Die funktionalen Anforderungen wurden in Zusammenarbeit mit Fachleuten aus der Branche sowie mit Autoren anderer Arbeiten auf diesem Gebiet definiert. Erfahrungen aus der Umsetzung von anderen Systemen flossen ebenfalls mit ein. Im Folgenden ist eine Kurzfassung der Anforderungen angeführt, eine ausführlichere Beschreibung folgt danach.

**Login / Logout** Alle Benutzer müssen sich zuerst anmelden, bevor sie die Funktionalität des Systems nutzen können. Die Sitzung kann durch einen Logout beendet werden.

**Protokollierung aller Zugriffe** Jeder Zugriff auf das System, jede Änderung, jede Datenabfrage muss in einem Logfile protokolliert werden.

**Einpflügen neuer Betriebsdaten** Sobald neue Betriebe dazukommen, können die Daten über eine Datenbankschnittstelle in das System integriert werden.

**Bearbeiten bestehender Betriebsdaten** Betriebe haben die Möglichkeit, bestimmte Daten die ihren Betrieb betreffen zu ändern.

**Einpflügen neuer Tierdaten** Sobald neue Tiere geboren werden, können die Daten über eine Datenbankschnittstelle in das System integriert werden.

**Suchfunktion** Allen Benutzern des Systems steht eine Suchfunktion zur Verfügung, um auf schnellem Weg die gewünschte Information zu finden. Gesucht werden kann dabei nach Betrieben, Tieren, Tierärzten und Medikamenten.

**Betriebsliste ansehen** Abgesehen von den Betrieben steht den Benutzern des Systems eine Ansicht zur Verfügung, die eine Liste der Betriebe darstellt. Diese Liste kann nach bestimmten Kriterien gefiltert und sortiert werden.

**Detailansicht eines Betriebes ansehen** Aus der Betriebsliste kann ein Betrieb ausgewählt werden, um Detailinformationen anzuzeigen.

**Auswahl einer Tierart** Da Betriebe in manchen Fällen mehr als nur eine Tierart halten, kann in diesem Fall ausgewählt werden, welche Art angezeigt werden soll.

**Tierliste ansehen** In der Tierliste werden die Tiere einer Art dargestellt. Diese Liste kann nach bestimmten Kriterien gefiltert und sortiert werden.

**Detailansicht eines Tieres ansehen** Aus der Tierliste kann ein Tier ausgewählt werden, um Detailinformationen anzuzeigen.

**Erfassung von Gesundheitsdaten für ein Tier** In der Detailansicht besteht für Tierärzte und Verbände die Möglichkeit, Gesundheitsdaten für ein Tier zu erfassen.

**Hochladen von Dokumenten** Benutzer wie Tierärzte oder Verbände, die Daten erfassen können, haben die Möglichkeit Dokumente und Bilder (Röntgenbilder, usw...) hochzuladen und einem Tier zuzuordnen.

**Tierarztindex ansehen** Betrieben steht ein Tierarztindex zur Verfügung, der alle eingetragenen Tierärzte beinhaltet. Diese Liste kann nach bestimmten Kriterien gefiltert und sortiert werden.

**Tierarzt für einen Betrieb freischalten** Aus dem Tierarztindex kann ein Tierarzt ausgewählt werden um ihn für einen Betrieb freizuschalten. Dieser Tierarzt hat dann die Möglichkeit, die Betriebsdaten einzusehen und Gesundheitsdaten zu den Tieren zu erfassen.

**Generieren von Berichten** Den Benutzern des Systems steht ein Berichtsgenerator zur Verfügung, um Auswertungen und Reports zu erstellen.

**Arzneimittelkatalog ansehen** Tierärzte können die verschriebenen Medikamente aus einem Arzneimittelkatalog auswählen, und einer Diagnose bzw. Behandlung zuordnen. Dieser Katalog kann nach bestimmten Kriterien gefiltert und sortiert werden.

**Meldungen erfassen** Tierärzte und Verbände können meldepflichtige Umstände und Beobachtungen erfassen

**Meldungen senden** Tierärzte und Verbände können meldepflichtige Umstände und Beobachtungen die sie erfasst haben an zuständige Behörden senden.

**Berechtigungen anlegen** Administratoren können für Benutzergruppen Berechtigungen vergeben um Zugriffe auf bestimmte Daten und Ansichten zu regeln.

**Abfrage der Zugriffsprotokollierung** Administratoren können die Zugriffe auf das System sowie die dabei geänderten und angezeigten Daten abfragen.

**Schnittstelle zu gängigen Praxissoftwarelösungen** Für Tierärzte steht eine Webservice-Schnittstelle zu gängigen Praxissoftwarelösungen zur Verfügung, um Daten die der Tierarzt mit seiner Software erfasst hat in das System zu übertragen und vice versa.

#### 4.2.4 Beschreibung der Anwendungsfälle

Grechenig et. al. [27] beschreibt einen Ansatz aus der Praxis, um Anwendungsfälle (Use Cases) detailliert darzustellen. Diese Darstellung wird im Folgenden verwendet und ist den Tabellen 4.1 bis 4.23 zu entnehmen. Vorab wird eine Erklärung der Elemente der Tabelle geliefert, die ebenfalls bei Grechenig et. al. [27] nachzulesen ist.

- Name  
Eine kurze Benennung des Anwendungsfalles

- **Beschreibung**  
Eine detaillierte Beschreibung des Anwendungsfalles
- **Beteiligte Akteure**  
Das sind diejenigen Personen, die bei der Ausführung des Anwendungsfalles beteiligt sind. Akteure sind verschiedene Benutzer, die mit dem System interagieren, wie im Fall der elektronischen Gesundheitsakte Betriebe, Tierärzte, Konsumenten, usw.
- **Status**  
Der Status beschreibt den Fortschritt der Arbeit an dem Anwendungsfall. In dieser Arbeit wurden alle auf Status Abgenommen gesetzt, wobei sie in weiterer Folge bei der Umsetzung noch verfeinert werden können. Weitere Beispiele wären in Arbeit, im Review oder abgelehnt.
- **Verwendete Anwendungsfälle**  
Wenn zwei Anwendungsfälle ineinandergreifen, bzw. wenn ein Anwendungsfall einen anderen benötigt um ausgeführt zu werden, werden diese Anwendungsfälle an dieser Stelle bekannt gegeben.
- **Auslöser**  
Ein Auslöser ist eine Bedingung, die für die Ausführung eines Anwendungsfalles erfüllt sein muss.
- **Invarianten**  
Bei Invarianten handelt es sich um Bedingungen, die vor, während und nach der Ausführung des Anwendungsfalles sowie im Fehlerfall erfüllt sein müssen.
- **Ergebnis**  
Der eintretende Zustand im Fall eines erfolgreich ausgeführten Anwendungsfalles.
- **Standardablauf**  
Beschreibt das am häufigsten eintretende Szenario leicht verständlich in einzelnen Schritten aufgeführt. Am Ende steht das Ziel, das mit der Ausführung des Anwendungsfalles erreicht wird.
- **Alternative Ablaufschritte**  
Beschreiben alternative Szenarien, die neben dem Standardablauf eintreten können. Hier können auch Fehler- oder Misserfolgswfälle beschrieben werden.
- **Hinweise**  
Alle Informationen, die nicht mit den bisher aufgeführten Punkten abgedeckt werden kann, können hier erfasst werden.
- **Änderungsgeschichte**  
Versionsnummer des Anwendungsfalles.

[27]

<b>Name</b>	<b>Login</b>
Beschreibung	Alle Benutzer müssen sich zuerst einloggen, bevor sie die Funktionalität des Systems nutzen können.
Beteiligte Akteure	Alle
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Benutzer möchte die Funktionen des Systems nutzen.
Vorbedingungen	Benutzer ist nicht eingeloggt.
Invarianten	Der Benutzer ist dem System bekannt und hat gültige Logindaten.
Ergebnis	Der Benutzer wurde in das System eingeloggt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Benutzer ruft über einen Browser die URL des Systems auf</li> <li>2. Benutzer gibt seine Logindaten ein</li> <li>3. Benutzer drückt auf Login</li> <li>4. Benutzer ist eingeloggt.</li> </ol>
Alternative Ablaufschritte	<ol style="list-style-type: none"> <li>1. Benutzer ruft über einen Browser die URL des Systems auf</li> <li>2. Benutzer gibt seine Logindaten ein</li> <li>3. Benutzer drückt auf Login</li> <li>4. Validierung der Logindaten schlägt fehl</li> <li>5. Benutzer wird nicht eingeloggt.</li> </ol>
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.1: Use Case Login

<b>Name</b>	<b>Logout</b>
Beschreibung	Die Benutzer können die Sitzung durch einen Logout beenden.
Beteiligte Akteure	Alle
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Benutzer möchte die Sitzung beenden.
Vorbedingungen	Benutzer ist eingeloggt.
Invarianten	Der Benutzer ist dem System bekannt und hat gültige Logindaten.
Ergebnis	Der Benutzer wurde aus dem System ausgeloggt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer beendet die Sitzung durch Klicken des Logout Buttons.</li> </ol>
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.2: Use Case Logout

Name	Protokollierung aller Zugriffe
Beschreibung	Jeder Zugriff auf das System, jede Änderung, jede Datenabfrage muss in einem Logfile protokolliert werden.
Beteiligte Akteure	System
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Ein Benutzer nutzt die Funktionen des Systems.
Vorbedingungen	Es sind Benutzeraccounts angelegt, die das System verwenden können.
Invarianten	-
Ergebnis	Jeder Zugriff wurde in einem Logfile protokolliert.
Standardablauf	<ol style="list-style-type: none"> <li>1. Ein Benutzer nutzt die Funktionalität des Systems</li> <li>2. Das System speichert einen Logeintrag in ein File mit Datum, welcher Benutzer, Funktion die aufgerufen wurde und Daten die dabei verändert wurden.</li> </ol>
Alternative Ablaufschritte	-
Hinweise	Jeder Zugriff auf eine Sicht sowie jede Veränderung eines Datensatzes stellt einen eigenen Eintrag im Logfile dar.
Änderungsgeschichte	Version 1.0

Tabelle 4.3: Use Case Protokollierung aller Zugriffe

Name	Einpflegen neuer Betriebsdaten
Beschreibung	Sobald neue Betriebe dazukommen, können die Daten über eine Datenbankschnittstelle in das System integriert werden.
Beteiligte Akteure	Administrator, System
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Betriebe die noch nicht im System erfasst sind, sollen eingepflegt werden.
Vorbedingungen	Es gibt Betriebe, die noch nicht im System erfasst sind.
Invarianten	-
Ergebnis	Die Betriebsdaten wurden eingepflegt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Die Betriebsdaten liegen digital vor</li> <li>2. Das System importiert die Daten in die Datenbank</li> <li>3. Dem neuen Betrieb werden Rollen für die Zugriffskontrolle zugewiesen</li> <li>4. Für den Betrieb werden Zugangsdaten angelegt</li> </ol>
Alternative Ablaufschritte	<ol style="list-style-type: none"> <li>1. Die Betriebsdaten liegen nicht digital vor</li> <li>2. Der Administrator legt den Betrieb in der Datenbank an</li> <li>3. Dem neuen Betrieb werden Rollen für die Zugriffskontrolle zugewiesen</li> <li>4. Für den Betrieb werden Zugangsdaten angelegt</li> </ol>
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.4: Use Case Einpflegen neuer Betriebsdaten

Name	Bearbeiten bestehender Betriebsdaten
Beschreibung	Betriebe haben die Möglichkeit, bestimmte Daten die ihren Betrieb betreffen, zu ändern.
Beteiligte Akteure	Administrator, Betrieb
Status	Abgenommen
Verwendete Anwendungsfälle	Anzeigen der Betriebsdetails
Auslöser	Betriebsdaten haben sich geändert.
Vorbedingungen	Der Betrieb ist bereits im System erfasst.
Invarianten	Der Betrieb ist im System vorhanden
Ergebnis	Die Betriebsdaten wurden aktualisiert.
Standardablauf	1. Der Betrieb bearbeitet die zu ändernden Daten 2. Der Betrieb bestätigt die Änderungen durch abspeichern.
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.5: Use Case Bearbeiten bestehender Betriebsdaten

Name	Einpflegen neuer Tierdaten
Beschreibung	Sobald neue Tiere geboren werden, können die Daten über eine Datenbankschnittstelle in das System integriert werden.
Beteiligte Akteure	System
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Neue Tiere sind in den Bestand eines Betriebes dazugekommen (Geburt, Ankauf, ...).
Vorbedingungen	Der Betrieb ist im System vorhanden.
Invarianten	Der Betrieb ist im System vorhanden
Ergebnis	Die Betriebsdaten wurden aktualisiert.
Standardablauf	1. Die Tierdaten liegen digital vor 2. Das System importiert die Daten in die Datenbank
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.6: Use Case Einpflegen neuer Tierdaten

Name	Suchfunktion
Beschreibung	Allen Benutzern des Systems steht eine Suchfunktion zur Verfügung, um auf schnellem Weg die gewünschte Information zu finden. Gesucht werden kann dabei nach Betrieben, Tieren, Tierärzten und Medikamenten.
Beteiligte Akteure	Alle
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte schnell Informationen abrufen.
Vorbedingungen	Der Benutzer kennt bestimmte Merkmale, nach denen gesucht werden soll.
Invarianten	Die Informationen werden nicht verändert
Ergebnis	Die Suche liefert ein Ergebnis.
Standardablauf	1. Der Benutzer gibt in der Suchmaske die Suchkriterien ein 2. Das System liefert ein Ergebnis
Alternative Ablaufschritte	1. Der Benutzer gibt in der Suchmaske die Suchkriterien ein 2. Das System liefert kein Ergebnis
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.7: Use Case Suchfunktion

Name	Betriebsliste ansehen
Beschreibung	Abgesehen von den Betrieben steht den Benutzern des Systems eine Ansicht zur Verfügung, die eine Liste der Betriebe darstellt. Diese Liste kann nach bestimmten Kriterien gefiltert und sortiert werden.
Beteiligte Akteure	Administrator, Tierarzt, Verbände, Konsumenten, Ministerien / AMA
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte die Betriebsliste ansehen.
Vorbedingungen	Der Benutzer navigiert zur Betriebslistenansicht.
Invarianten	Die Betriebsliste wird nicht verändert
Ergebnis	Die Betriebsliste wird angezeigt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Betriebslistenansicht</li> <li>2. Das System zeigt die Betriebsliste an</li> <li>3. Der Benutzer kann die Betriebsliste ansehen</li> </ol>
Alternative Ablaufschritte	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Betriebslistenansicht</li> <li>2. Das System zeigt die Betriebsliste an</li> <li>3a. Der Benutzer filtert die Betriebsliste nach Spaltenkriterien</li> <li>3b. Der Benutzer sortiert die Betriebsliste spaltenweise</li> <li>4. Der Benutzer kann die Betriebsliste ansehen</li> </ol>
Hinweise	Je nach Akteur und Zugriffsrecht auf bestimmte Informationen können Spalten in der Betriebsliste für die Rolle sichtbar oder unsichtbar sein.
Änderungsgeschichte	Version 1.0

Tabelle 4.8: Use Case Betriebsliste ansehen

<b>Name</b>	<b>Detailansicht eines Betriebes ansehen</b>
Beschreibung	Aus der Betriebsliste kann ein Betrieb ausgewählt werden, um Detailinformationen anzuzeigen.
Beteiligte Akteure	Administrator, Tierarzt, Verbände, Konsumenten, Ministerien / AMA
Status	Abgenommen
Verwendete Anwendungsfälle	Betriebsliste ansehen
Auslöser	Der Benutzer möchte einen Betrieb im Detail sehen.
Vorbedingungen	Der Benutzer wählt einen Betrieb aus der Betriebslistenansicht.
Invarianten	Die Betriebsliste wird nicht verändert
Ergebnis	Die Betriebsdetails werden angezeigt.
Standardablauf	1. Der Benutzer wählt aus der Betriebslistenansicht einen Betrieb aus 2. Das System zeigt den ausgewählten Betrieb im Detail
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.9: Use Case Detailansicht eines Betriebes ansehen

<b>Name</b>	<b>Auswahl einer Tierart</b>
Beschreibung	Da Betriebe in manchen Fällen mehr als nur eine Tierart halten, kann in diesem Fall ausgewählt werden, welche Art angezeigt werden soll.
Beteiligte Akteure	Administrator, Betrieb, Tierarzt, Verbände, Konsumenten, Ministerien / AMA
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte eine Tierart des Betriebes genauer betrachten.
Vorbedingungen	Der Benutzer hat einen Betrieb aus der Betriebslistenansicht ausgewählt.
Invarianten	-
Ergebnis	Die Tierliste einer Tierart wird angezeigt.
Standardablauf	1. Der Benutzer wählt in der Detailansicht des Betriebes eine Tierart aus 2. Das System zeigt die Tierliste der gewählten Tierart
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.10: Use Case Auswahl einer Tierart

<b>Name</b>	<b>Tierliste ansehen</b>
Beschreibung	In der Tierliste werden die Tiere einer Art dargestellt. Diese Liste kann nach bestimmten Kriterien gefiltert und sortiert werden.
Beteiligte Akteure	Administrator, Tierarzt, Verbände, Konsumenten, Ministerien / AMA
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte die Tierliste ansehen.
Vorbedingungen	Der Benutzer navigiert zur Tierlistenansicht.
Invarianten	Die Tierliste wird nicht verändert
Ergebnis	Die Tierliste wird angezeigt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Tierlistenansicht</li> <li>2. Das System zeigt die Tierliste an</li> <li>3. Der Benutzer kann die Tierliste ansehen</li> </ol>
Alternative Ablaufschritte	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Tierlistenansicht</li> <li>2. Das System zeigt die Tierliste an</li> <li>3a. Der Benutzer filtert die Tierliste nach Spaltenkriterien</li> <li>3b. Der Benutzer sortiert die Tierliste spaltenweise</li> <li>4. Der Benutzer kann die Tierliste ansehen</li> </ol>
Hinweise	Je nach Akteur und Zugriffsrecht auf bestimmte Informationen können Spalten in der Tierliste für die Rolle sichtbar oder unsichtbar sein.
Änderungsgeschichte	Version 1.0

Tabelle 4.11: Use Case Tierliste ansehen

Name	Detailansicht eines Tieres ansehen
Beschreibung	Aus der Tierliste kann ein Tier ausgewählt werden, um Detailinformationen anzuzeigen.
Beteiligte Akteure	Administrator, Betrieb, Tierarzt, Verbände, Konsumenten, Ministerien / AMA
Status	Abgenommen
Verwendete Anwendungsfälle	Tierliste ansehen
Auslöser	Der Benutzer möchte ein Tier im Detail sehen.
Vorbedingungen	Der Benutzer wählt ein Tier aus der Tierlistenansicht.
Invarianten	Die Tierliste wird nicht verändert
Ergebnis	Die Tierdetails werden angezeigt.
Standardablauf	1. Der Benutzer wählt aus der Tierlistenansicht ein Tier aus 2. Das System zeigt das ausgewählte Tier im Detail
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.12: Use Case Detailansicht eines Tieres ansehen

Name	Erfassung von Gesundheitsdaten für ein Tier
Beschreibung	In der Detailansicht besteht für Tierärzte und Verbände die Möglichkeit, Gesundheitsdaten für ein Tier zu erfassen.
Beteiligte Akteure	Tierarzt, Verbände
Status	Abgenommen
Verwendete Anwendungsfälle	Detailansicht eines Tieres ansehen
Auslöser	Der Benutzer möchte Gesundheitsdaten zu einem Tier erfassen.
Vorbedingungen	Ein Tier aus der Tierlistenansicht wurde ausgewählt.
Invarianten	-
Ergebnis	Die Gesundheitsdaten für ein Tier wurden erfasst.
Standardablauf	1. Der Benutzer gibt die Daten in das System ein 2. Die Eingabe wird durch Speichern bestätigt
Alternative Ablaufschritte	-
Hinweise	Je nach Rolle können unterschiedliche Gesundheitsdaten erfasst werden.
Änderungsgeschichte	Version 1.0

Tabelle 4.13: Use Case Erfassung von Gesundheitsdaten für ein Tier

<b>Name</b>	<b>Hochladen von Dokumenten</b>
Beschreibung	Benutzer wie Tierärzte oder Verbände, die Daten erfassen können, haben die Möglichkeit Dokumente und Bilder (Röntgenbilder, usw...) hochzuladen und einem Tier zuzuordnen.
Beteiligte Akteure	Tierarzt, Verbände
Status	Abgenommen
Verwendete Anwendungsfälle	Detailansicht eines Tieres ansehen
Auslöser	Der Benutzer möchte Dokumente oder Bilder zu einem Tier hochladen.
Vorbedingungen	Ein Tier aus der Tierlistenansicht wurde ausgewählt.
Invarianten	-
Ergebnis	Die Dokumente bzw. Bilder für ein Tier wurden hochgeladen.
Standardablauf	1. Der Benutzer wählt über eine Upload-Maske die Dokumente bzw. Bilder aus 2. Die Dokumente bzw. Bilder werden hochgeladen
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.14: Use Case Hochladen von Dokumenten

<b>Name</b>	<b>Tierarztindex ansehen</b>
Beschreibung	Betrieben steht ein Tierarztindex zur Verfügung, der alle eingetragenen Tierärzte beinhaltet. Diese Liste kann nach bestimmten Kriterien gefiltert und sortiert werden.
Beteiligte Akteure	Betrieb
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte den Tierarztindex ansehen.
Vorbedingungen	-
Invarianten	-
Ergebnis	Der Tierarzt wurde für den Betrieb freigeschaltet und kann nun die Betriebs- und Tierinformationen einsehen. Weiters kann er nun Gesundheitsdaten zu einem Tier erfassen.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zum Tierarztindex</li> <li>2. Das System zeigt den Tierarztindex an</li> <li>3. Der Benutzer kann den Tierarztindex ansehen</li> </ol>
Alternative Ablaufschritte	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zum Tierarztindex</li> <li>2. Das System zeigt den Tierarztindex an</li> <li>3a. Der Benutzer filtert den Tierarztindex nach Spaltenkriterien</li> <li>3b. Der Benutzer sortiert den Tierarztindex spaltenweise</li> <li>4. Der Benutzer kann den Tierarztindex ansehen</li> </ol>
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.15: Use Case Tierarztindex ansehen

Name	<b>Tierarzt für einen Betrieb freischalten</b>
Beschreibung	Aus dem Tierarztindex kann ein Tierarzt ausgewählt werden um ihn für einen Betrieb freizuschalten. Dieser Tierarzt hat dann die Möglichkeit, die Betriebsdaten einzusehen und Gesundheitsdaten zu den Tieren zu erfassen.
Beteiligte Akteure	Betrieb
Status	Abgenommen
Verwendete Anwendungsfälle	Tierarztindex ansehen
Auslöser	Der Benutzer möchte aus dem Tierarztindex einen Tierarzt auswählen, um ihn seinem Betrieb zuzuweisen.
Vorbedingungen	Der Tierarzt ist für den Betrieb noch nicht freigeschaltet.
Invarianten	-
Ergebnis	Der Tierarzt wurde für den Betrieb freigeschaltet und kann nun die Betriebs- und Tierinformationen einsehen. Weiters kann er Gesundheitsdaten zu einem Tier erfassen.
Standardablauf	1. Der Benutzer wählt mithilfe der Filter- und Sortiermöglichkeit einen Tierarzt aus der Liste aus 2. Der ausgewählte Tierarzt wird dem Betrieb zugewiesen.
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.16: Use Case Tierarzt für einen Betrieb freischalten

<b>Name</b>	<b>Generieren von Berichten</b>
Beschreibung	Den Benutzern des Systems steht ein Berichtsgenerator zur Verfügung, um Auswertungen und Reports zu erstellen.
Beteiligte Akteure	Alle
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte Auswertungen oder Berichte erstellen.
Vorbedingungen	Der Benutzer befindet sich in einer Ansicht, die die Reporterstellung erlaubt.
Invarianten	-
Ergebnis	Der Bericht wurde generiert.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zu einer Ansicht, aus der er einen Bericht generieren will</li> <li>2. Der Benutzer löst die Generierung durch Betätigung eines Buttons aus.</li> <li>3. Der Bericht wird generiert.</li> </ol>
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.17: Use Case Generieren von Berichten

Name	Arzneimittelkatalog ansehen
Beschreibung	Tierärzte können die verschriebenen Medikamente aus einem Arzneimittelkatalog auswählen, und einer Diagnose bzw. Behandlung zuordnen. Dieser Katalog kann nach bestimmten Kriterien gefiltert und sortiert werden.
Beteiligte Akteure	Tierarzt
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte den Arzneimittelkatalog ansehen.
Vorbedingungen	Der Benutzer navigiert zum Arzneimittelkatalog.
Invarianten	Der Arzneimittelkatalog wird nicht verändert.
Ergebnis	Der Arzneimittelkatalog wird angezeigt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zum Arzneimittelkatalog</li> <li>2. Das System zeigt den Arzneimittelkatalog an</li> <li>3. Der Benutzer kann den Arzneimittelkatalog ansehen</li> </ol>
Alternative Ablaufschritte	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zum Arzneimittelkatalog</li> <li>2. Das System zeigt den Arzneimittelkatalog an</li> <li>3a. Der Benutzer filtert den Arzneimittelkatalog nach Spaltenkriterien</li> <li>3b. Der Benutzer sortiert den Arzneimittelkatalog spaltenweise</li> <li>4. Der Benutzer kann den Arzneimittelkatalog ansehen</li> </ol>
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.18: Use Case Arzneimittelkatalog ansehen

<b>Name</b>	<b>Meldungen erfassen</b>
Beschreibung	Tierärzte und Verbände können meldepflichtige Umstände und Beobachtungen erfassen.
Beteiligte Akteure	Tierarzt, Verband
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer hat einen meldepflichtigen Umstand beobachtet.
Vorbedingungen	Der Benutzer navigiert zur Detailansicht des Betriebes.
Invarianten	Betriebsdaten werden nicht verändert.
Ergebnis	Der meldepflichtige Umstand wurde erfasst.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Detailansicht des Betriebes</li> <li>2. Der Benutzer erstellt einen neuen Eintrag in der dafür vorgesehenen Maske</li> <li>3. Der meldepflichtige Eintrag wurde angelegt und ist dem Betrieb zugeordnet.</li> </ol>
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.19: Use Case Meldungen erfassen

<b>Name</b>	<b>Meldungen senden</b>
Beschreibung	Tierärzte und Verbände können meldepflichtige Umstände und Beobachtungen die sie erfasst haben an zuständige Behörden senden.
Beteiligte Akteure	Tierarzt, Verband
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer hat einen meldepflichtigen Umstand erfasst und möchte diesen an die zuständige Behörde senden.
Vorbedingungen	Der Benutzer wählt den meldepflichtigen Eintrag aus, den er senden möchte.
Invarianten	Betriebsdaten werden nicht verändert
Ergebnis	Der meldepflichtige Umstand wurde an die zuständige Behörde gesendet.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Maske mit den meldepflichtigen Einträgen</li> <li>2. Der Benutzer wählt den meldepflichtigen Eintrag aus, den er senden möchte</li> <li>3. Der Benutzer löst den Sendeprozess durch Klick auf den dafür vorgesehenen Button aus.</li> </ol>
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.20: Use Case Meldungen senden

<b>Name</b>	<b>Berechtigungen anlegen</b>
Beschreibung	Administratoren können für Benutzergruppen Berechtigungen vergeben um Zugriffe auf bestimmte Daten und Ansichten zu regeln.
Beteiligte Akteure	Administrator
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte für eine Benutzergruppe Berechtigungen festlegen.
Vorbedingungen	Der Benutzer navigiert zur Rollenübersicht.
Invarianten	Es werden keine Daten verändert
Ergebnis	Die Berechtigungen wurden angelegt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Rollenübersicht</li> <li>2. Der Benutzer legt einen neuen Eintrag an</li> <li>3. Der Benutzer wählt für diesen Eintrag die gewünschten Berechtigungen und ermöglicht dadurch Zugriffe auf bestimmte Funktionen und Sichten im System</li> </ol>
Alternative Ablaufschritte	-
Hinweise	-
Änderungsgeschichte	Version 1.0

Tabelle 4.21: Use Case Berechtigungen anlegen

Name	Abfrage der Zugriffsprotokollierung
Beschreibung	Administratoren können die Zugriffe auf das System sowie die dabei geänderten und angezeigten Daten abfragen.
Beteiligte Akteure	Administrator
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Benutzer möchte Informationen über Zugriffe erhalten.
Vorbedingungen	Der Benutzer navigiert zur Zugriffsdatenabfrage.
Invarianten	Es werden keine Daten verändert
Ergebnis	Die Zugriffsdaten werden abgefragt.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zur Zugriffsdatenabfrage</li> <li>2. Der Benutzer schränkt die Abfrage nach bestimmten Kriterien ein</li> <li>3. Die Abfrage wird durchgeführt</li> <li>4. Der Benutzer erhält das Ergebnis der Abfrage zur Ansicht</li> </ol>
Alternative Ablaufschritte	-
Hinweise	Die Kriterien für die Abfrage wären z.B. nach Benutzer, für einen Zeitraum, Zugriff auf Ansichten, Erfassung von Datensätzen usw...
Änderungsgeschichte	Version 1.0

Tabelle 4.22: Use Case Abfrage der Zugriffsprotokollierung

Name	Schnittstelle zu gängigen Praxissoftwarelösungen
Beschreibung	Für Tierärzte steht eine Webservice-Schnittstelle zu gängigen Praxissoftwarelösungen zur Verfügung, um Daten die der Tierarzt mit seiner Software erfasst hat in das System zu übertragen und vice versa.
Beteiligte Akteure	Tierarzt
Status	Abgenommen
Verwendete Anwendungsfälle	-
Auslöser	Der Tierarzt möchte Daten zwischen den Systemen austauschen.
Vorbedingungen	Der Benutzer navigiert zum Modul für den Datenaustausch.
Invarianten	Es werden keine Daten verändert
Ergebnis	Die Daten wurden übertragen.
Standardablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zum Modul für den Datenaustausch</li> <li>2. Der Benutzer wählt aus, dass er Daten senden möchte</li> <li>3. Der Benutzer wählt aus, welche Daten er übertragen möchte</li> <li>4. Die Daten werden zur Praxissoftware übertragen.</li> </ol>
Alternative Ablaufschritte	<ol style="list-style-type: none"> <li>1. Der Benutzer navigiert zum Modul für den Datenaustausch</li> <li>2. Der Benutzer wählt aus, dass er Daten empfangen möchte</li> <li>3. Der Benutzer wählt aus, welche Daten er empfangen möchte</li> <li>4. Die Daten werden in das System der elektronischen Gesundheitsakte für Nutztiere übertragen.</li> </ol>
Hinweise	Die Einstellungen die zur Übertragung zum System des Tierarztes notwendig sind, müssen vor dem ersten Sende- bzw. Empfangsvorgang vorgenommen werden.
Änderungsgeschichte	Version 1.0

Tabelle 4.23: Use Case Schnittstelle zu gängigen Praxissoftwarelösungen

### 4.2.5 Use Case Modellierung

Neben einer textuellen Beschreibung - wie der im vorherigen Abschnitt 4.2.4 - besteht die Möglichkeit, Anforderungen mittels Use Case Diagrammen in UML zu modellieren. Im Folgenden sollen die oben beschriebenen Anforderungen auf diese Art dargestellt werden.

Die folgenden beiden Abbildungen zeigen Use Cases aus der Sicht des Betriebes respek-

tive des Landwirtes sowie aus der Sicht des Tierarztes und des Administrators. Weitere vorkommende Akteure wie z.B. die Datenbankschnittstelle oder ein externes System stellen keine Personen dar. Die Akteure sind durch Assoziationen mit den Use Cases verbunden, die sie ausführen können.

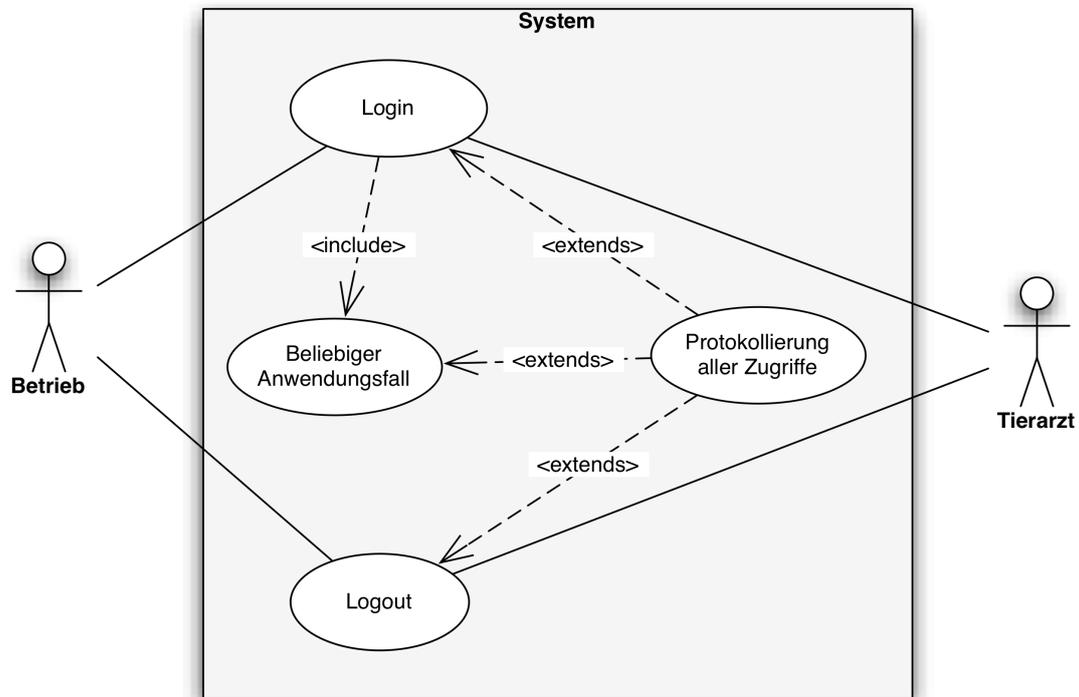


Abbildung 4.1: Anwendungsfalldiagramm Login, Logout und Zugriffsprotokollierung

Abbildung 4.1 zeigt, dass den Akteuren jede beliebige Funktion respektive jeder definierte Anwendungsfall des Systems erst ab dem Login zur Verfügung steht. Weiters ist dem Modell zu entnehmen, dass jeder Zugriff von jedem Benutzer protokolliert wird.

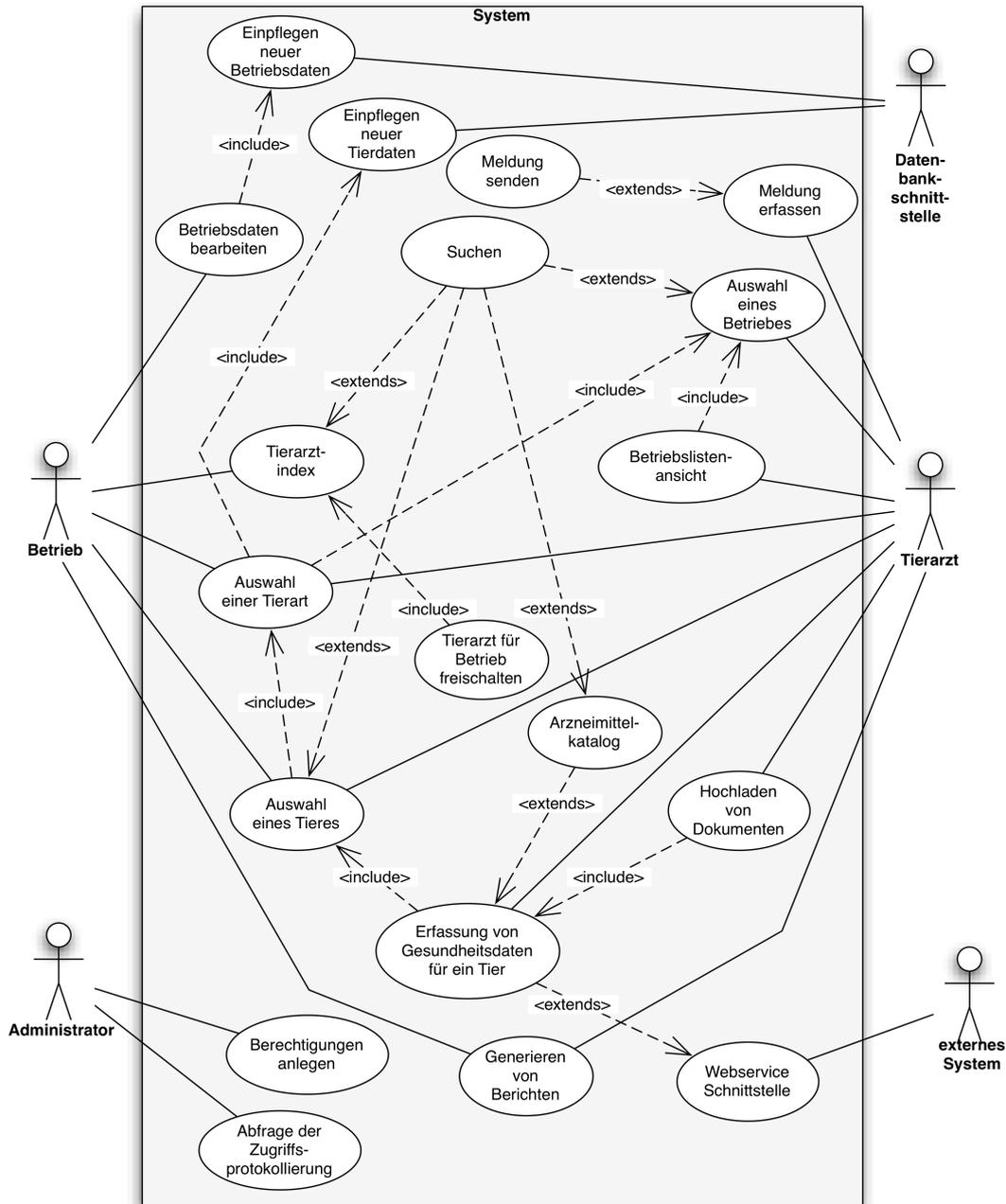


Abbildung 4.2: Anwendungsfalldiagramm für die Erfassung von Gesundheitsdaten

In Abbildung 4.2 ist zu sehen, dass ein Betrieb z.B. seine Betriebsdaten bearbeiten kann (Tabelle 4.5), wobei diese Funktion dem Tierarzt nicht zur Verfügung steht. Betriebsdaten können allerdings nur bearbeitet werden, wenn sie vorher über eine Datenbankschnittstelle in das System eingepflegt wurden.

Die Suchfunktion (Tabelle 4.7) steht beiden Akteuren zur Verfügung und ist im System ab dem Login ausführbar. So kann schnell ein bestimmtes Objekt gefunden werden. Die Suche steht auch auf verschiedenen Masken zur Verfügung, wie z.B. Suchen im Tierarztindex, Suchen innerhalb der Tierliste, Suchen im Arzneimittelkatalog oder Suchen

innerhalb der Betriebsliste. Die Suche kann aber auch ausgeführt werden, ohne zu einer dieser Masken zu navigieren.

Gesundheitsdaten für ein Tier zu erfassen (Tabelle 4.13) ist eine Funktion, die dem Tierarzt zur Verfügung steht, dem Betrieb aber nicht. Um diesen Anwendungsfall auszuführen, muss der Tierarzt nach dem Login entweder über die Suche zu einem bestimmten Tier, für das er die Daten erfassen will, gelangen, oder über die Betriebsauswahl (Tabelle 4.9) weiter zur Tierartenauswahl (Tabelle 4.10) weiter zur Auswahl eines bestimmten Tieres (Tabelle 4.12), um anschließend zur Erfassungsmaske für die Gesundheitsdaten zu gelangen. Bei der Erfassung von Medikationen in dieser Maske steht dem Tierarzt ein Arzneimittelkatalog zur Verfügung, über den das zu verabreichende Medikament ausgewählt werden kann. Die Erfassung von Gesundheitsdaten für ein Tier bietet dem Tierarzt auch die Möglichkeit, Dokumente wie Röntgenbilder oder Befunde hochzuladen. Erfasste Daten können über einen Webservice (Tabelle 4.23) an andere Systeme gesendet werden.

Der Betrieb kann bis zur Detailansicht eines Tieres navigieren und sich dort die Gesundheitsakte seines Tieres mit allen vergangenen Diagnosen, Medikationen und Befunden ansehen. Jedoch kann er selber an dieser Stelle keine Daten erfassen. Tierärzte können weiters Meldungen erfassen (Tabelle 4.19), um diese dann an eine Behörde zu senden (Tabelle 4.20).

Den Akteuren steht ein Berichtsgenerator (Tabelle 4.17) zur Verfügung, der Auswertungen zu den erfassten Daten erstellt.

Die Funktion Berechtigungen anzulegen (Tabelle 4.21) und die Zugriffsprotokollierung (Tabelle 4.22) abzufragen, steht nur Administratoren zur Verfügung.

## 4.3 Zugriffs- und Rollenkonzept

Anhand der vorgestellten Zugriffskontrollkonzepte aus Abschnitt 3.8 stellt sich heraus, dass bei der Umsetzung der ELGA für Nutztiere nur das RBAC Modell für den Einsatz in Frage kommt, da es genau für diese Art von großen Datenbanksystemen in komplexen Domänen ausgerichtet ist.

### 4.3.1 Funktionale Rollen

In Anlehnung an der in [19] verwendeten Darstellung der funktionalen Rollen des Systems sind diese sowie eine zugehörige Beschreibung Tabelle 4.24 zu entnehmen.

<b>Funktionale Rolle</b>	<b>Beschreibung</b>
A. Administrative Rolle	Diese Rolle ist für die Entwickler des Systems vorgesehen. Sie können auf alle Daten zugreifen, anderen Benutzern Rechte vergeben und weitere Rollen anlegen.
B. Landwirt	Der Landwirt sieht seinen eigenen Betrieb. Er darf Behandlungen erfassen oder auch vom Tierarzt hinterlassene Diagnosebelege. Er bestimmt, wer auf seinen Betrieb lesenden respektive schreibenden Zugriff hat.
C. Tierarzt	Der Tierarzt sieht die Betriebe die er behandelt bzw. für die er freigeschaltet wurde. Hier stellt sich die wichtige Frage, ob der Tierarzt alle Diagnosen die jemals auf einem Betrieb gemacht wurde sehen darf (also auch von den vor ihm zuständigen Tierärzten) oder nur seine eigenen Diagnosen. Tierärzte die als Vertretung auf einen Betrieb kommen, müssen für diesen freigeschaltet werden. Hier ist zu diskutieren, welche Daten sie von diesem Betrieb in diesem Fall sehen können dürfen.
D. Verbände / Organisationen	Sehen die Daten der Betriebe, für die sie zuständig sind. Sie können Kontrolldaten erfassen und anonymisierte Daten über Tierbestände, diverse Auswertungen über die Regionen zu Tiergesundheit oder Seuchenebefall ansehen.
E. Ministerien / AMA	Diese Rollen sollen im Anlassfall Betriebsdaten einsehen können.
F. Weiterverarbeitende Industrie	Diese Rollen sollen eine eingeschränkte Sicht auf Betriebsdaten und ausgewählte Tierdaten haben.
G. Konsumenten	Diese Rollen sollen eine sehr eingeschränkte Sicht haben mit einigen wenigen Informationen über den Betrieb und Schlachtdaten des Tieres.

Tabelle 4.24: Funktionale Rollen des Systems

### 4.3.2 Regeln für die Zugriffskontrolle

In [19] werden Regeln für die Zugriffskontrolle aufgestellt, die auf dem RBAC Access Control Framework basieren. Diese Art der Darstellung wird übernommen, um eine Liste der Regeln aufzustellen. Die Aufbereitung einer Regel sieht wie folgt aus:

Erlaube [**Benutzer/Rolle/Subjekt**] die Ausführung [**Zugriffsart**] auf [**Objekt**] für [**Zweck**] versehen mit [**Bedingung**]  
Durchführung von [**Pflicht**].

Die resultierende Liste besteht aus 28 Regeln die die Zugriffsrechte auf die Informationen und Daten der elektronischen Gesundheitsakte bestimmen.

1. Erlaube [**Administrator**] die Ausführung [**lesen / schreiben**] auf [**alle Daten und Sichten der elektronischen Gesundheitsakte**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
2. Erlaube [**Landwirt**] die Ausführung [**lesen / schreiben**] auf [**seine eigene Betriebsicht**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
3. Erlaube [**Landwirt**] die Ausführung [**lesen**] auf [**Ansicht seiner eigenen Tierliste**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
4. Erlaube [**Landwirt**] die Ausführung [**lesen / schreiben**] auf [**Detailansicht eines seiner Tiere**] für [**Erfassung von Tiergesundheitsdaten**] versehen mit [**eingeschränkten Schreibrechten**] Durchführung von [**Eintrag im Log-File**].
5. Erlaube [**Landwirt**] die Ausführung [**lesen**] auf [**Ansicht Tierarztindex**] für [**Freischaltung eines Tierarztes für den Betrieb**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
6. Erlaube [**Landwirt**] die Ausführung [**lesen**] auf [**Berichtsgenerator**] für [**Erstellung von Berichten und Auswertungen**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
7. Erlaube [**Tierarzt**] die Ausführung [**lesen**] auf [**Ansicht seiner betreuten Betriebe**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
8. Erlaube [**Tierarzt**] die Ausführung [**lesen**] auf [**Ansicht der Tierliste eines seiner betreuten Betriebe**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
9. Erlaube [**Tierarzt**] die Ausführung [**lesen / schreiben**] auf [**Detailansicht eines Tieres eines seiner betreuten Betriebe**] für [**Erfassung von Tiergesundheitsdaten, Hochladen von Dokumenten (Röntgenbilder usw.)**] versehen mit [(**Ausnahmen bei Bedarf**)] Durchführung von [**Eintrag im Log-File**].
10. Erlaube [**Tierarzt**] die Ausführung [**lesen**] auf [**Ansicht Arzneimittelkatalog**]

- für **[Einbindung in Diagnoseerfassung]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
11. Erlaube **[Tierarzt]** die Ausführung **[lesen / schreiben]**  
auf **[Erfassung von besonderen Umständen]**  
für **[Meldepflichtige Krankheitsfälle]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  12. Erlaube **[Tierarzt]** die Ausführung **[lesen]**  
auf **[Berichtsgenerator]**  
für **[Erstellung von Berichten und Auswertungen]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  13. Erlaube **[Verbände / Organisationen]** die Ausführung **[lesen]**  
auf **[Ansicht der betreuten Betriebe]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  14. Erlaube **[Verbände / Organisationen]** die Ausführung **[lesen]**  
auf **[Detailansicht eines betreuten Betriebes]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  15. Erlaube **[Verbände / Organisationen]** die Ausführung **[lesen]**  
auf **[Ansicht der Tierliste eines betreuten Betriebes]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  16. Erlaube **[Verbände / Organisationen]** die Ausführung **[lesen / schreiben]**  
auf **[Detailansicht eines betreuten Betriebes]**  
für **[Erfassung von verbandsrelevanten Betriebsdaten]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  17. Erlaube **[Verbände / Organisationen]** die Ausführung **[lesen / schreiben]**  
auf **[Detailansicht eines Tieres eines betreuten Betriebes]**  
für **[Erfassung von verbandsrelevanten Tiergesundheitsdaten]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  18. Erlaube **[Verbände / Organisationen]** die Ausführung **[lesen]**  
auf **[Berichtsgenerator]**  
für **[Erstellung von Berichten und Auswertungen]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
  19. Erlaube **[Ministerien / AMA]** die Ausführung **[lesen]**  
auf **[Betriebsansicht eines zu überprüfenden Betriebes]**

- versehen mit **[Freischaltung im Anlassfall]**  
Durchführung von **[Eintrag im Log-File]**.
20. Erlaube **[Ministerien / AMA]** die Ausführung **[lesen]**  
auf **[Ansicht aller Tiere eines zu überprüfenden Betriebes]**  
versehen mit **[Freischaltung im Anlassfall]**  
Durchführung von **[Eintrag im Log-File]**.
21. Erlaube **[Ministerien / AMA]** die Ausführung **[lesen]**  
auf **[Detailansicht Tier eines zu überprüfenden Betriebes]**  
für **[Überprüfung der Tiergesundheit]**  
versehen mit **[Freischaltung im Anlassfall]**  
Durchführung von **[Eintrag im Log-File]**.
22. Erlaube **[Ministerien / AMA]** die Ausführung **[lesen]**  
auf **[Berichtsgenerator]**  
für **[Erstellung von Berichten und Auswertungen über Regionen, Betriebe, Tiere]**  
versehen mit **[(Ausnahmen bei Bedarf)]**  
Durchführung von **[Eintrag im Log-File]**.
23. Erlaube **[Konsumenten]** die Ausführung **[lesen]**  
auf **[Ansicht aller Betriebe]**  
versehen mit **[Besondere Einschränkung welche Betriebsdaten angezeigt werden]**  
Durchführung von **[Eintrag im Log-File]**.
24. Erlaube **[Konsumenten]** die Ausführung **[lesen]**  
auf **[Ansicht der Tierartenliste eines Betriebes]**  
versehen mit **[Besondere Einschränkung welche Daten angezeigt werden]**  
Durchführung von **[Eintrag im Log-File]**.
25. Erlaube **[Konsumenten]** die Ausführung **[lesen]**  
auf **[Ausgewählte Sicht auf Tierdetails eines Betriebes]**  
versehen mit **[Besondere Einschränkung welche Daten angezeigt werden (Schlachtdatum, usw...)]**  
Durchführung von **[Eintrag im Log-File]**.
26. Erlaube **[weiterverarbeitende Industrie]** die Ausführung **[lesen]**  
auf **[Ansicht aller Betriebe]**  
versehen mit **[Besondere Einschränkung welche Daten angezeigt werden]**  
Durchführung von **[Eintrag im Log-File]**.
27. Erlaube **[weiterverarbeitende Industrie]** die Ausführung **[lesen]**  
auf **[Ansicht der Tierartenliste eines Betriebes]**  
versehen mit **[Besondere Einschränkung welche Daten angezeigt werden]**  
Durchführung von **[Eintrag im Log-File]**.
28. Erlaube **[weiterverarbeitende Industrie]** die Ausführung **[lesen]**  
auf **[Ausgewählte Sicht auf Tierdetails eines Betriebes]**  
versehen mit **[Besondere Einschränkung welche Daten angezeigt werden]**  
Durchführung von **[Eintrag im Log-File]**.

# 5 Architekturf Entwurf

In diesem Kapitel wird ausgehend von den Grundlagen aus dem Kapitel 2, dem theoretischen Hintergrund der verwendeten Architekturmuster und -techniken aus dem Kapitel 3 sowie der Analyse aus Kapitel 4 der Softwarearchitekturf Entwurf beschrieben.

Softwarearchitektur betrifft das Design der Struktur eines Software Systems, einschließlich seinem gesamten Verhalten und seiner Zerlegung in einfache rechnerische Elemente [29].

## 5.1 MVC-Modell

In Abbildung 5.1 ist die MVC-Architektur aus [61] adaptiert für die elektronische Gesundheitsakte für Nutztiere abgebildet.

Die funktionalen Rollen des Systems, die in Abschnitt 4.3.1 definiert wurden, interagieren über die View mit dem System. Sie dient der aufbereiteten Anzeige der Daten für die Benutzer. Die Views nehmen weiters von Rollen mit Schreibrechten die Dateneingabe entgegen. Für jede Ansicht im System gibt es eine View, wie z.B. für die Login-Seite, für die Ansicht der Betriebsliste, für die Ansicht der Tierliste, für die Erfassung von Gesundheitsdaten für ein Tier, für den Tierarztindex, für die Medikamentenliste usw. Je nach Rolle sind diese Sichten zugänglich oder nicht.

Jede View ist mit einem Controller verbunden. Der Controller nimmt Benutzereingaben entgegen und verarbeitet diese, d.h. er sendet die Eingaben an das Modell. Sobald das Modell verändert wurde, werden die dazugehörigen Views aktualisiert und dem Benutzer dadurch sichtbar gemacht.

Das Modell besteht aus den Tabellen der Datenbank, die durch objektrelationales Mapping im System als Klassen existieren und von dort aus manipuliert werden können. Die Zusammenhänge der Tabellen können dem Abschnitt 5.3 entnommen werden, wo ein Datenmodell erstellt wurde.

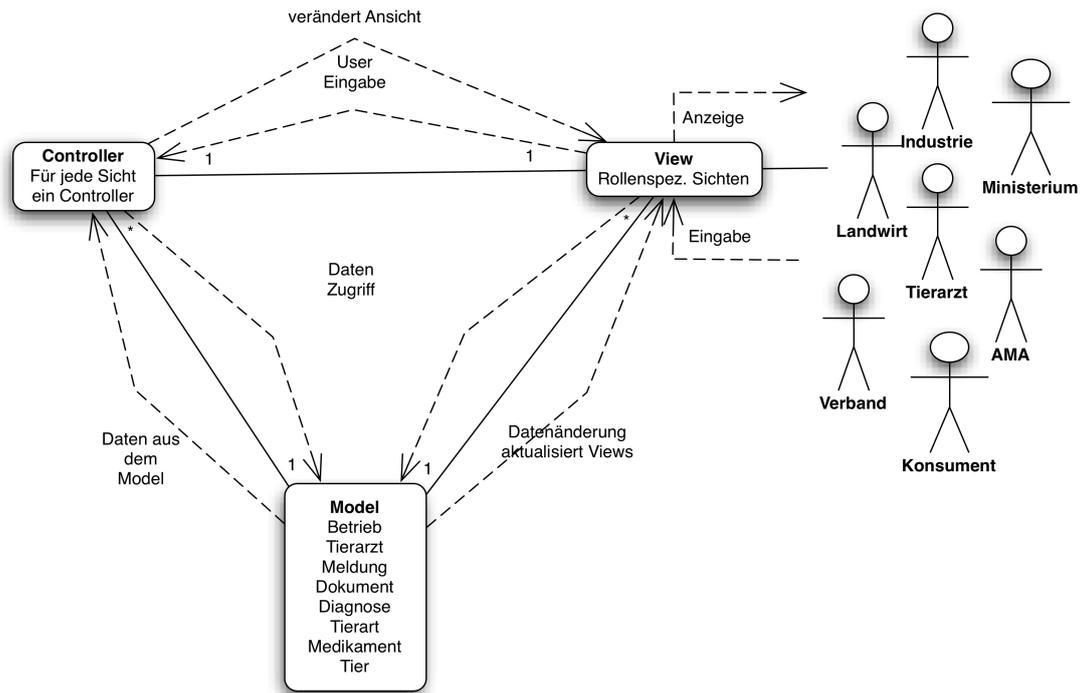


Abbildung 5.1: MVC Architektur der elektronischen Gesundheitsakte für Nutztiere

## 5.2 3-Tier Architektur

Abbildung 5.2 zeigt die 3-Tier-Architektur des Systems, die sehr populär ist und in vielen Softwaresystemen umgesetzt wird [58].

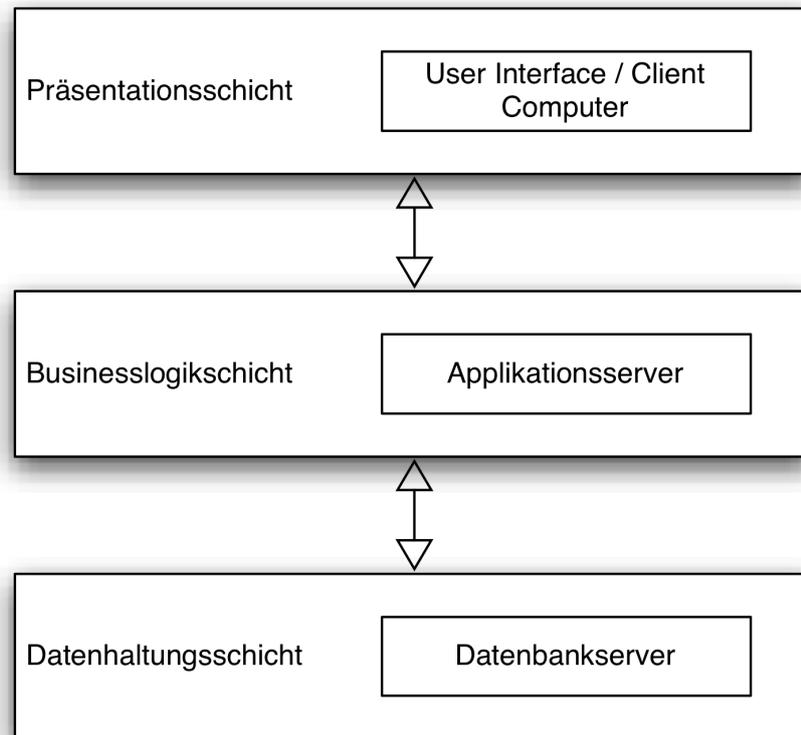


Abbildung 5.2: 3-Tier-Architektur der elektronischen Gesundheitsakte für Nutztiere

Die Architektur ist unterteilt in drei Ebenen der Datenverarbeitung.

Die Präsentationsschicht stellt das User Interface der Applikation dar und fungiert als Schnittstelle zwischen der Applikation und den Benutzern der elektronischen Gesundheitsakte. Sie bietet einen benutzerfreundlichen Zugriff auf das System. Da über das User Interface alle Aufgaben und Zugriffe durchgeführt werden, ist es eine viel verwendete Schicht.

Die Businesslogikschicht ist der Kern der Applikation, da hier die Logik des Systems realisiert wird, welche für die gesamte Funktionalität verantwortlich ist. Diese Schicht läuft auf einem eigenen Applikationsserver oder Webserver.

Die Datenhaltungsschicht ist verantwortlich für die Datenspeicherung, den Datenzugriff und die Optimierung. Sie muss die Aktualisierung und die Abfrage großer Datenmengen umgehend ausführen. Diese Schicht ist auf einem eigenen Datenbankserver angesiedelt.

Die Businesslogik und das User Interface sind auf unterschiedlichen Plattformen angesiedelt, mit einem geeigneten Kommunikationsprotokoll dazwischen. Die Kommunikation und der Datenaustausch in den unterschiedlichen Schichten erfolgt über die Businesslogik. Wenn die Businesslogik der Datenbank oder am Applikationsserver verändert werden muss, ändert sich für die Clients nichts. Dadurch wird die Wiederverwendbarkeit des Systems verbessert, der Entwicklungszyklus verkürzt und die Wartungskosten reduziert.

Die Modularisierung ermöglicht es auf eine einfache Art das System horizontal oder

vertikal zu erweitern. Auf der einen Seite, kann das System auf eine größere und leistungsfähigere Plattform übertragen werden; auf der anderen Seite kann die größere Dimension die Netzwerkkaplikation des Systems verstärken. Als Resultat der Kapselung der Businesslogikschicht ist die Datenverteilung und die Datenbank selber für die Präsentationsschicht und somit den Client nicht sichtbar.

Die 3-Tier-Architektur hat ein sichereres Securitymanagement als z.B. die 1-Tier- oder die 2-Tier-Architektur. Selbst wenn Angreifer von außen die Sicherheitsabwehr der Präsentationsschicht durchbrechen, kann sich das System durch weitere Sicherheitsmechanismen in den darunter liegenden Tiers schützen.

Das Ziel dieser 3-Tier-Architektur ist es, die Berechnung und die Daten zu verteilen. Wenn sich der Ablaufprozess bei der Erfassung und Einsicht in die Gesundheitsdaten der Nutztiere ändert, muss die Mittelschicht bzw. die Geschäftslogik dementsprechend angepasst werden. Wenn sich lediglich nur einige Felder ändern oder Werte durch Kalkulationen aktualisiert werden müssen, kann dies an die Datenhaltungsschicht weitergegeben werden in Form von Speicherprozeduren oder Triggern. Wenn sich allerdings die Struktur der Datenbank ändert, muss jede Schicht geändert werden.

## 5.3 Datenmodell

Die Darstellung des Datenmodells erfolgt mittels der Chen-Notation [9] und ist in Abbildung 5.3 zu sehen. Es liefert eine statische Sicht auf das System.

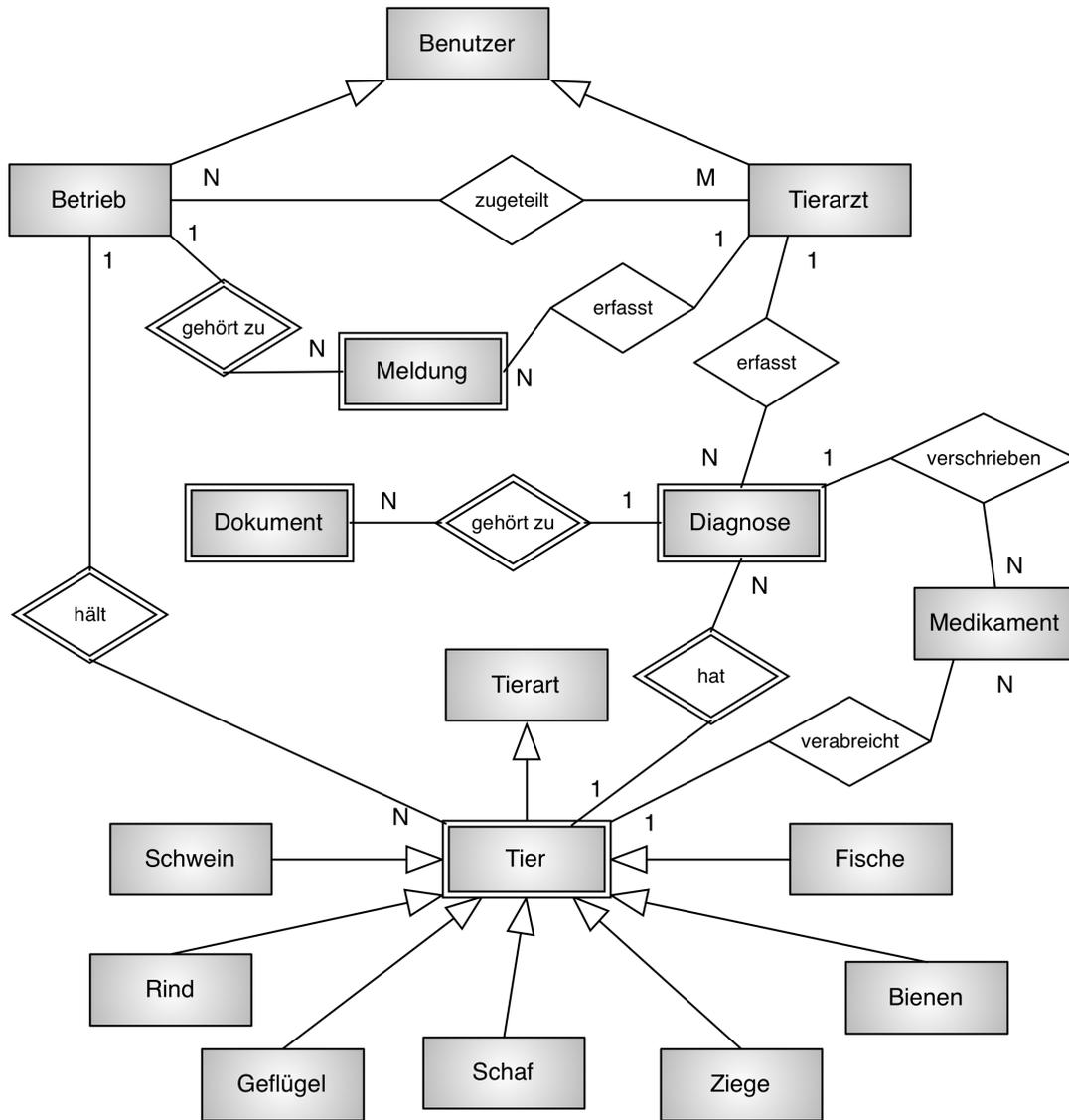


Abbildung 5.3: Datenmodell der Gesundheitsakte für Nutztiere

Dieser konzeptionelle Entwurf enthält die Informationseinheiten, die in der späteren Implementierung des Datenbankmanagementsystems zum Einsatz kommen. Die genauen Attribute, die zu den Entitäten gehören und später in den Tabellen gespeichert werden, sind Teil einer eigenen wissenschaftlichen Arbeit zum Thema Datenbankentwurf der elektronischen Gesundheitsakte für Nutztiere.

## 5.4 UML Darstellung

Nachfolgend werden die in Kapitel 3.6 vorgestellten Diagrammartentypen praktisch angewendet.

### 5.4.1 Aktivitätsdiagramm

Abbildung 5.4 stellt das Aktivitätsdiagramm für die Aktivität „Gesundheitsdaten für ein Nutztier erfassen“ in einer dynamischen Sichtweise dar. Diese Aktivität wird mittels Partitionen aus der Sicht des Betriebes sowie des Tierarztes modelliert und stellt den Aktionsfluss für beide Akteure dar. Die Aktionen im linken Rechteck können vom Betrieb ausgeführt werden, die Aktionen im rechten Rechteck vom Tierarzt und das mittlere Rechteck beinhaltet alle Aktionen, die beide Akteure ausführen können.

Nach dem Login befindet sich der Betrieb auf seiner Betriebsstartseite, von wo aus er über die Auswahl einer Tierart, weiter über die Auswahl eines Tieres die Gesundheitsdaten des gewählten Tieres ansehen kann. Wie in Abschnitt 4.3.2 nachzulesen ist, hat der Betrieb nur Leserechte und darf daher die Diagnosen seiner Herde nur einsehen. Die Datenerfassung ist ihm nicht gestattet und somit endet für ihn an dieser Stelle der Ablauf der Aktivität.

Tierärzte wählen nach dem Login einen Betrieb aus, den sie einsehen und dessen Tiergesundheitsdaten sie bearbeiten wollen. Danach folgt der gleiche Ablauf wie für einen Betrieb, nämlich die Auswahl der Tierart, danach die Auswahl eines Tieres, um so zur Einsicht der Gesundheitsdaten zu gelangen. Tierärzte können nun Gesundheitsdaten erfassen, also Diagnosen in das System eingeben. Wenn dem Tier Medikamente verabreicht werden sollen, werden diese bei der Erfassung aus der Medikamentenliste ausgewählt und so mit der Diagnose verknüpft. Falls zusätzliche Dokumente wie Laborbefunde oder Röntgenbilder vorhanden sind, besteht die Möglichkeit, diese hochzuladen. Wenn alle Daten erfasst sind, endet für den Tierarzt an dieser Stelle der Ablauf der Aktivität.

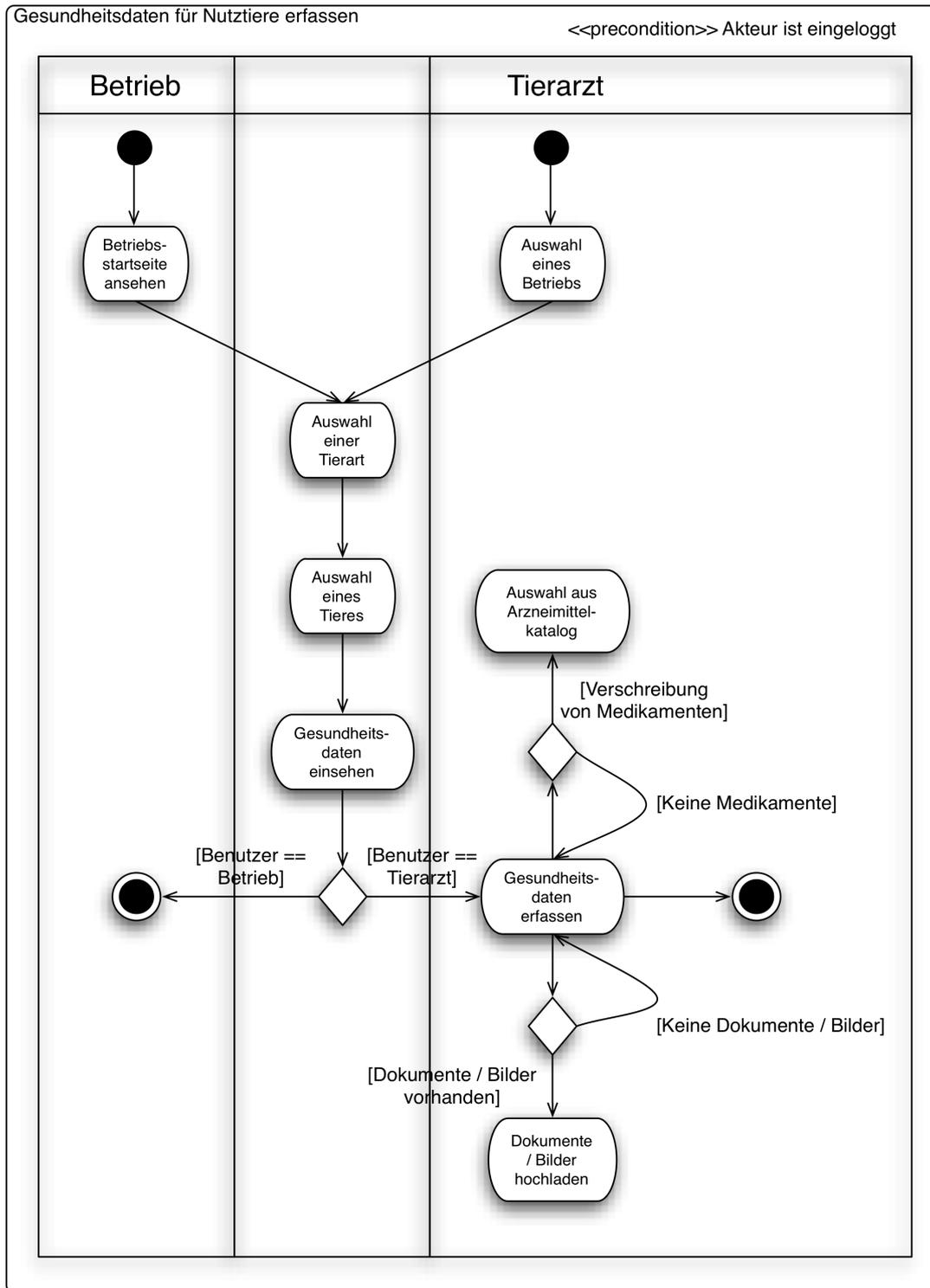


Abbildung 5.4: Aktivitätsdiagramm

### 5.4.2 Komponentendiagramm

Abbildung 5.5 zeigt das Komponentendiagramm für das System, welches die logische Struktur verdeutlichen soll.

Die Beziehungen zwischen den Komponenten sind über Interfaces definiert. Das Diagramm zeigt die Organisation und Abhängigkeiten einzelner technischer Systemkomponenten, die im Folgenden ein wenig näher beschrieben werden sollen:

Der **Präsentationslayer** bildet die Sicht des Benutzers auf das System, sozusagen das User Interface auf der Clientseite, über das die Interaktion mit dem System abläuft.

Der **Dokumenten- und MessagingService** regelt den Datenaustausch und die Nachrichtenübertragung zwischen den Systemen durch Regeln, die vom HL7-Standard festgelegt sind und ermöglicht dadurch Interoperabilität. Die CDA des HL7-Standard dient in dieser Komponente der Definition und dem gemeinsamen Gebrauch von klinischen Dokumenten. Näheres zu HL7 und CDA findet sich in Abschnitt 2.3.1.

Der **TransaktionsService** wird verwendet, um Nachrichten zwischen Systemen auszutauschen und um Diagnosedaten zu erfassen. Im Fehlerfall wird der User informiert. Alle anderen Aktionen werden nur von der Applikation geprüft, um die Skalierbarkeit auf Datenbankebene zu erleichtern. Im Fehlerfall wird auch hier der Benutzer benachrichtigt.

Der **BilddatenService** implementiert den DICOM-Standard (Abschnitt 2.3.3), um die Speicherung im festgelegten Format und den Datenaustausch der medizinischen Bilddaten über ein geeignetes Kommunikationsprotokoll zu ermöglichen.

Der **BenutzerAccountService** regelt die Benutzerverwaltung sowie das Rollenkonzept mit den dazugehörigen Zugriffsrechten für die verschiedenen Rollen und Benutzer.

Der **SecurityService** regelt die Security Anforderungen an das System. Die gespeicherten Information müssen sicher sein und dürfen nicht von unauthorisierten Benutzern eingesehen werden. Weiters muss auch die Durchführung von Transaktionen sicher ablaufen.

Der **LoggingService** protokolliert alle Aktionen die im System durchgeführt werden, in einer LoggingTabelle. Dabei wird erfasst, wer, wann, welche Aktion getätigt hat. Dabei hat jede Aktion, jeder Benutzer und jede Benutzergruppe einen eigenen Code, um falls gewünscht, verschiedene Abfragen mit Hilfe dieses Codes auf die Loggingtabelle durchzuführen und dadurch bestimmte Informationen zu gewinnen.

Der **WebService** stellt eine Schnittstelle zur Verfügung, um Tierärzten den Datenaustausch zwischen ihrer in der Praxis verwendeten Software und dem System der ELGA für Nutztiere Daten auszutauschen. Die gängigsten Lösungen wie Vetinf, easyVET und ANIMAL-office wurden bereits im Kapitel Grundlagen in Abschnitt 2.1.2 vorgestellt.

Der **ReportService** ermöglicht den Benutzern des Systems die Erstellung von Berichten im PDF und CSV Format.

Der **SuchService** implementiert eine optimierte Suche, damit Betriebe, Tiere und sonstige Daten schnell gefunden werden können.

Der **ImportService** bietet eine Schnittstelle, um Tier- und Betriebsdaten aus anderen Systemen zu importieren, die bereits dort erfasst wurden, wie z.B. aus der Rinderdatenbank der AMA.

Der **PersistenzService** implementiert die Speicherung in die Datenbank mittels ORM (Abschnitt 3.5).

Die **Datenbank** speichert alle Daten des Systems in Tabellen.

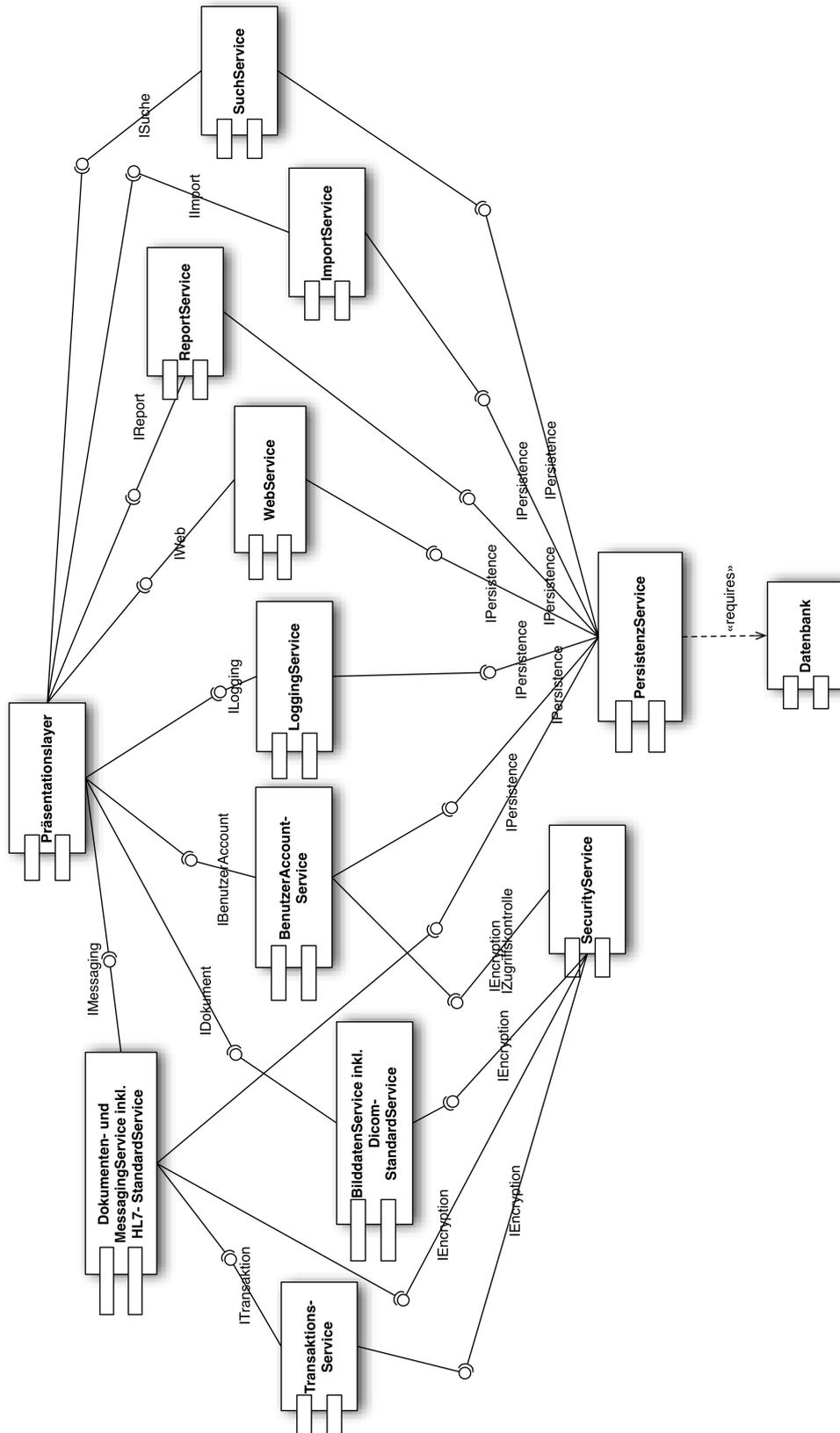


Abbildung 5.5: Komponentendiagramm

### 5.4.3 Verteilungsdiagramm

Abbildung 5.6 zeigt das Verteilungsdiagramm für das System und zeigt damit die physische Struktur.

Das Diagramm zeigt, dass Benutzer auf das System über einen Webbrowser mit Internetverbindung zugreifen können. Die Präsentationsschicht mit dem Benutzerinterface läuft über einen Webserver, die Datenbank läuft auf einem eigenen Datenbankserver und die Implementierung der Komponenten befindet sich auf einem Anwendungsserver. Die Server sind alle vierfach repliziert um eine hohe Ausfallsicherheit mit einer Verfügbarkeit von 99,99% zu garantieren. Für die Berechnung der Verfügbarkeit wird mit der bedingten Wahrscheinlichkeit für Parallelschaltungen gerechnet. Die Formel für die Berechnung mit vierfach replizierten Knoten sieht folgendermaßen aus:

$$1 - (1 - P(\text{Nodefunktioniert})^4) = n$$

Mit der Annahme, dass jeder Knoten eine Fehlerwahrscheinlichkeit von mindestens 10% aufweist, ergibt dies eingesetzt in die Formel:

$$1 - (1 - 0,9) * (1 - 0,9) * (1 - 0,9) * (1 - 0,9) = n$$

$$1 - (0,1) * (0,1) * (0,1) * (0,1) = 1 - 0,0001 = n = 0,9999 = \mathbf{99,99\%}$$

Zur Kontrolle kann man den umgekehrten Rechenweg heranziehen:

$$0,9999 = 1 - (1 - P(\text{Nodefunktioniert}))^n$$

$$0,9999 = 1 - (1 - (0,9))^n$$

$$0,9999 = 1 - (0,1)^n$$

$$(0,1)^n = 0,0001$$

$$n * \ln(0,1) = \ln(0,0001)$$

$$n = \frac{\ln(0,0001)}{\ln(0,1)} = \mathbf{4}$$

Es sind also 4 Knoten notwendig, um eine Verfügbarkeit von 99,99% zu gewährleisten.

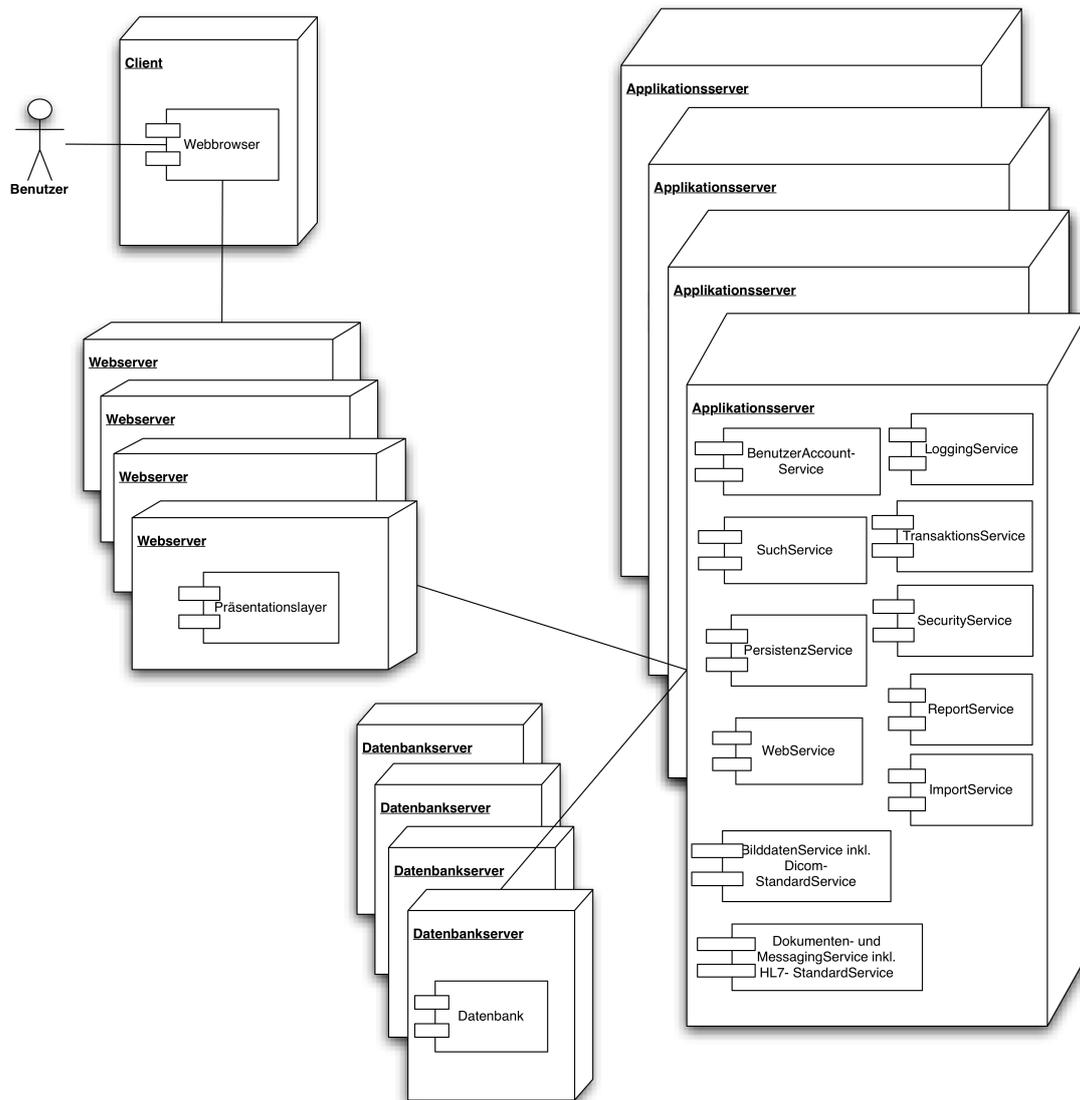


Abbildung 5.6: Verteilungsdiagramm

## 5.5 Systemarchitektur

In Abbildung 5.7 ist die Systemarchitektur der elektronischen Gesundheitsakte für Nutztiere dargestellt.

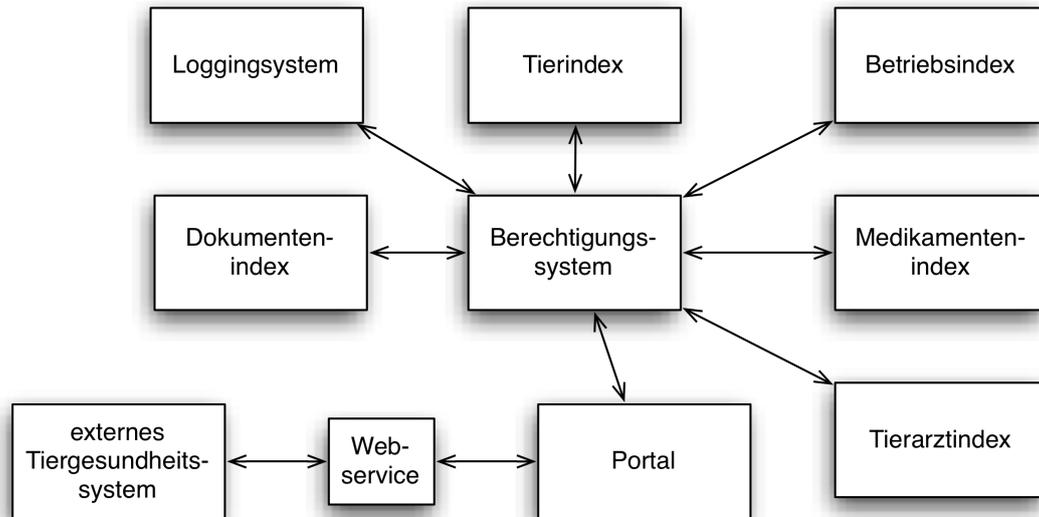


Abbildung 5.7: Systemarchitektur

### Portal

Der Zugriff auf die Informationen der elektronischen Gesundheitsakte erfolgt für die Benutzer des Systems über das Portal, wo die Authentifizierung erfolgt. Es stellt den zentralen Zugang zu den gesundheitsrelevanten Informationen zur Verfügung. Für jeden Benutzer, egal ob Tierarzt, Betrieb oder Konsument, erfolgt der Einstieg über diese Komponente. Konsumenten sind die einzige Benutzergruppe, die sich nicht über einen Benutzernamen und einem Passwort identifizieren müssen, erhalten jedoch auch nur eine minimale Sicht auf das System, mit einigen wenigen Informationen über den Betrieb und dessen Tierbestand. Alle anderen Benutzergruppen, die über den Login Zugriff auf das System (je nach Benutzergruppe mit unterschiedlicher Sicht und Berechtigung) erhalten, können hier Tiergesundheitsdaten einsehen und editieren. Die integrierte Berechtigungsverwaltung ermöglicht es Betrieben, über das Portal anderen Benutzergruppen zu den betriebsinternen Daten Zugang zu gewähren. Es muss jedoch sichergestellt werden, dass keine unbeabsichtigten Freigaben oder Einschränkungen stattfinden. Weiters dient dieses System dazu, in den Daten und Informationen der Gesundheitsakte zu suchen bzw. sie für den Benutzer geeignet darzustellen und Daten zu erfassen bzw. Dokumente und Bilder zu speichern.

### Berechtigungssystem

Das Berechtigungssystem ist verantwortlich für erlaubte und unerlaubte Zugriffe und regelt den Datenschutz. Es ermöglicht nur die Zugriffe, die laut dem Rollenkonzept berechtigt sind, die gewünschte Aktion auszuführen. Hier sind die Rollen im System definiert sowie die Berechtigungen bezogen auf die jeweilige Rolle. Jede Anfrage von außen wird überprüft und nur berechtigte Benutzer können die jeweilige Anfragen ausführen. Das Berechtigungssystem entlastet alle anderen Komponenten der elektronischen Gesundheitsakte für Nutztiere von Berechtigungsprüfungen und bildet neben dem Portal den zentralen Kern des Systems.

**Loggingsystem**

Das Loggingsystem protokolliert sämtliche Zugriffe von Benutzern auf das System. Dies ermöglicht es zu einem späteren Zeitpunkt einzusehen, wer und wann welche Aktion ausgeführt hat. Daraus können auch Auswertungen erstellt werden, um verschiedene Informationen daraus zu gewinnen. Das Loggingsystem ist unabhängig von anderen Systemkomponenten, nimmt zu protokollierende Einträge von allen anderen Komponenten entgegen und legt dafür einen Audit Trail an. Audit Trails erfassen, wer auf das System zugegriffen hat und welche Operationen während einer bestimmten Zeit ausgeführt wurden. Sie sind sehr nützlich zur Sicherstellung der Security des Systems und zur Wiederherstellung verloren gegangener Transaktionen. Die meisten Datenbankmanagementsysteme inkludieren Audit Trail Komponenten, es existieren jedoch auch eigene Softwareprodukte, die den Netzwerkadministratoren die Überwachung der Netzwerkreisourcen ermöglichen. Die geloggt Daten werden gespeichert und können für Auswertungen herangezogen werden. Mit dieser Überwachung des Systems können auch Hackversuche erkannt werden.

**Tierarztindex**

Der Tierarztindex verwaltet die registrierten Tierärzte im System. Hier können die Betriebe einen bestimmten Tierarzt durch gezielte Suche ausfindig machen und für den Datenzugriff auf dem eigenen Betrieb freischalten. Die Registrierung der Tierärzte erfolgt durch die Tierärztekammer. Betriebe wählen aus dem Index den behandelnden Tierarzt und schalten ihn für den eigenen Betrieb frei, womit sie ihm eine Berechtigung zum Lesen und Schreiben von Betriebs- und gesundheitsrelevanten Tierdaten ermöglichen.

**Betriebsindex**

Im Betriebsindex sind alle Betriebe, die im System der elektronischen Gesundheitsakte für Nutztiere registriert sind, gespeichert. Jeder Betrieb wird durch seine Daten und eindeutige Betriebsnummer identifiziert. Der Index stellt eine Filter- und Suchmöglichkeit zur Verfügung, um den richtigen Betrieb aufzufinden. Die Erstbefüllung erfolgt über eine Datenbankschnittstelle, da bereits viele Betriebe in anderen Systemen erfasst sind und wenn möglich von dort importiert werden können. Nach der Initialbefüllung besteht die Möglichkeit, die Daten manuell zu pflegen und aktuell zu halten. Im Zusammenhang mit der Erstbefüllung aus mehreren Quellen kann es zu Dubletten kommen, die bereinigt werden müssen.

**Tierindex**

Der Tierindex verwaltet die Herde eines Betriebes. Durch gezielte Such- und Filtermöglichkeiten können bestimmte Tiere aufgefunden werden. Je nach Tierrasse und gesetzlichen Bestimmungen für die jeweilige Rasse erfolgt die Identifikation der Tiere unterschiedlich. Bei Rindern ist eine Einzeltierkennzeichnung gesetzlich vorgeschrieben, bei anderen in der elektronischen Gesundheitsakte vertretenen Tierarten könnte die Identifikation über Boxnummern oder ähnliches erfolgen.

**Dokumentenindex**

Im Dokumentenindex werden alle Dokumente der elektronischen Gesundheitsakte eines Tieres verwaltet. Berechtigte Personen können gezielt auf das erforderliche Dokument zugreifen. Auch hier besteht die Möglichkeit einer Suche bzw. Filterung der Daten. Zu

jedem Dokument bzw. Bild werden zusätzlich Metadaten gespeichert, um ein schnelles Auffinden zu ermöglichen.

### **Medikamentenindex**

Der Medikamentenindex dient als Schnittstelle für den Tierarzt, um Medikamente nicht händisch eintragen zu müssen, sondern sie aus diesem Katalog auswählen zu können. Die Daten kommen von einem externen Anbieter von Medikamentendaten, auf die über eine Schnittstelle zugegriffen werden kann.

### **Externes Tiergesundheitssystem**

Externe Anbieter von Tiergesundheitssystemen wie z.B. Praxissoftwarelösungen o.ä. können über eine Webservice Schnittstelle mit dem System kommunizieren. Über gesicherte Verbindungen wie Secure Socket Layer (SSL) oder Transport Layer Security (TLS) mit Client Authentication werden Verbindungen zu diesen externen Systemen hergestellt. Anfragen von Systemen die nicht die selben Standards verwenden werden transformiert und so in den Komponenten weiterverarbeitet. Umgekehrt werden die Antworten auf das proprietäre Format zurückgewandelt.

## **5.6 Standards in der Konzeption**

Die folgende Abbildung 5.8 zeigt, wie die Standards mit dem System zusammenwirken können.

Der DICOM Standard wird für die explizite Betrachtung von Bildmaterial in Form von Bildbetrachtungsprogrammen herangezogen. Das Bildmaterial wird, wie in Abschnitt 2.3.3 beschrieben, bereits von den bildverarbeitenden Systemen standardkonform erzeugt.

Bei der Erfassung von Gesundheitsdaten für Tiere stehen die SNOMED Daten über Browser zur Verfügung, über die sie abgerufen werden können. Ähnlich erfolgt es für N.A.V., hier wird eine Datenbasis mit den Fachausdrücken der Terminologie bereitgestellt, über die auf die Daten zugegriffen werden kann.

Die LOINC-Tabellen sind ebenfalls bei der Erfassung von Gesundheitsdaten für ein Tier abrufbar, um die allgemeingültigen Identifikatoren des Standards verwenden zu können. Der letzte zu erwähnende Standard HL7 speichert die erfassten Daten und Informationen standardkonform, um die effiziente und einheitliche Übertragung zu externen Systemen zu ermöglichen.

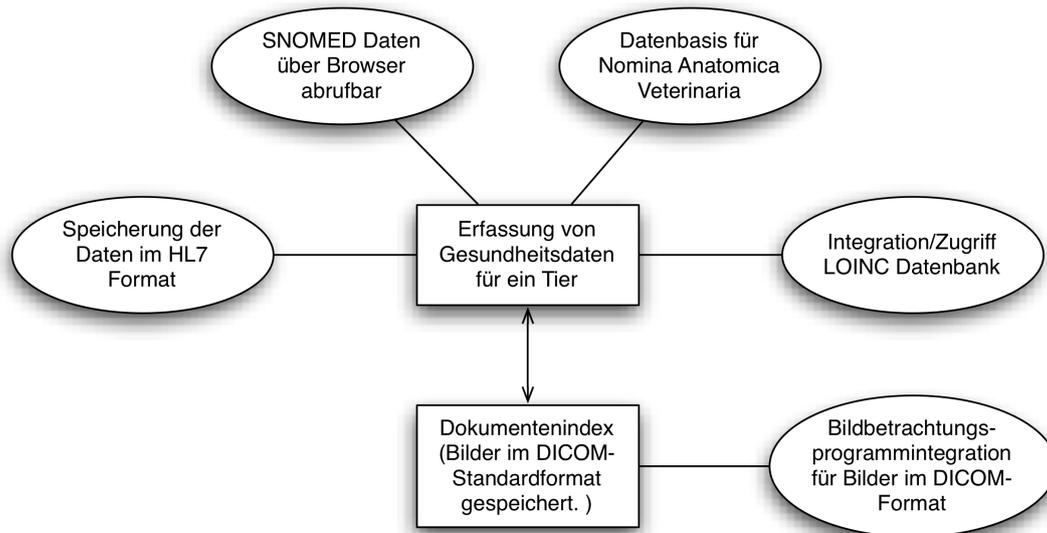


Abbildung 5.8: Zusammenspiel zwischen Standards und Konzeption

## 5.7 Sicherheit

Abschließend wird in diesem Abschnitt noch ein wenig auf das Thema Sicherheit eingegangen. Da in der elektronischen Gesundheitsakte für Nutztiere einige sowohl für Betriebe als auch Tierärzte sicherheitstechnisch sensible Informationen gespeichert werden, müssen die Verbindungen und Zugriffe auf das System verschlüsselt erfolgen. Dazu wird nachfolgend ein gängiges Sicherheitsprotokoll vorgestellt, welches bei der Umsetzung zum Einsatz kommen könnte.

Die Computer- und Netzwerktechnologie dient in vielfacher Weise den Menschen, jedoch sind gleichzeitig Sicherheitsprobleme immer vordergründiger und müssen ernsthaft betrachtet werden.

Mit der wachsenden Popularität von Applikationen, im Besonderen dem raschen Wachstum von Netzwerktechnologien, sind immer mehr Sicherheitsbedrohungen aufgetaucht. Deshalb ist die Informationssicherheit ein sehr wichtiges und dringendes Problem geworden, welches es zu bewältigen gilt. [33]

In den letzten Jahren wurden viele Forschungen und technische Studien im Bereich der Informationssicherheit betrieben. Derzeit gehen diese Forschungen in zwei Richtungen, nämlich in die der Datensicherheit und weiters in die der Netzwerksicherheit. Die Theorie der Datensicherheit basiert auf der Kryptographietheorie, forscht im Bereich der Vertraulichkeit, der Integrität und der Verfügbarkeit der Daten, Datensicherheitsabwehrstrategien usw.

Die Forschungen der Netzwerksecurity laufen in den Bereichen diverser Sicherheitsprotokolle, Sicherheitsmechanismen und Sicherheitservices, wobei diese drei Einheiten die theoretische Basis für ein sicheres Netzwerk bieten. [33]

Bei der Umsetzung der elektronischen Gesundheitsakte muss Authentizität, Vertraulichkeit, Verfügbarkeit und Integrität sichergestellt werden.

Authentizität bedeutet, dass Daten, Dokumente und andere Informationen eindeutig dem Sender zugeordnet werden können und weiters sichergestellt werden kann, dass diese Informationen nach dem Senden nicht mehr verändert werden können. Der Urheber der Daten muss also korrekt authentifizierbar sein.

Die Vertraulichkeit kann bedroht werden, wenn Personen auf Daten Lese- oder Schreibzugriffe haben, die ihnen nicht zustehen. Es ist daher sicherzustellen, dass unbefugte Zugriffe nicht stattfinden können. Das Rollenkonzept aus Abschnitt 4.3 dient dazu, die Vertraulichkeit zu bewahren.

Verfügbarkeit bedeutet, dass alle Funktionen des Systems einem Benutzer zur Verfügung stehen, wenn er darauf zugreifen will und für die er berechtigt ist. Zu den Funktionen gehören auch Daten, Programme und die Hardware des Systems, die für das korrekte Zusammenspiel notwendig sind.

Um Integrität sicherzustellen, müssen die Daten und die Hardware des Systems vor Verfälschungen bewahrt werden, damit immer korrekte Daten verarbeitet werden und Funktionen nicht fehlerhaft ausgeführt werden.

Nachfolgend werden die Konzepte von SSL, TLS und Client Authentication beschrieben, die für die Kommunikation der Systeme untereinander die Verschlüsselung und Authentifizierung sicherstellen.

### **SSL und TLS**

Die erste Version von TLS Version 1.0 basiert auf der letzten Version von SSL 3.0 und ist somit eine Weiterentwicklung von SSL. Diese zwei Protokolle sind nicht vollständig kompatibel, sodass ein TLS Server ein Downgrade auf SSL 3.0 machen muss, um mit SSL 3.0 Clients interoperabel zu sein.

In [7] ist nachzulesen, dass SSL aus zwei Phasen besteht: dem Handshake und der Datenübertragung. Während der Handshake-Phase verwenden Client und Server einen Public-Key Verschlüsselungsalgorithmus um Secret-Key Parameter zu bestimmen. Bei der Secret-Key Verschlüsselung verwenden der Sender und der Empfänger den selben Schlüssel und Daten zu ver- und zu entschlüsseln. Bekannte Secret-Key Verschlüsselungsalgorithmen wären DES (Data Encryption Standard), Triple DES, AES (Advanced Encryption Standard) und RC4. Bei der Public-Key Verschlüsselung werden zwei unterschiedliche Schlüssel verwendet, ein öffentlicher und ein privater Schlüssel. Normalerweise verwendet der Sender den öffentlichen Schlüssel um Nachrichten zu verschlüsseln und der Empfänger verwendet den privaten Schlüssel zur Entschlüsselung. Leider ist die Entschlüsselung von Nachrichten, die mit einem öffentlichen Schlüssel verschlüsselt wurden sehr CPU intensiv. Der populärste Public-Key Algorithmus ist der Rivest-Shamir-Adelman (RSA) Algorithmus.

Während der Übertragungsphase bei SSL verwenden beide Seiten den Secret-Key um Daten sukzessive zu ver-/entschlüsseln. Es wird an dieser Stelle nicht näher auf das SSL Handshake Protokoll eingegangen, für einen tiefergehenden Einblick wird auf die Arbeit von Wesley Chou verwiesen [10].

Mit SSL können drei der oben genannten Sicherheitskategorien sichergestellt werden. Vertraulichkeit, sodass die Person oder die Benutzergruppe gewählt werden kann, die die geschützten Daten lesen können.

Nachrichtenintegrität, sodass die Nachricht die gesendet wird, die Nachricht ist, die der

Empfänger erhält, ohne dass sie bei der Übertragung durch eine Störung oder einen Angreifer verändert wurde.

Endpunkt Authentifikation, d.h. dass der Peer mit dem Kommunikation stattfindet auch tatsächlich der ist, für den er sich ausgibt. [7]

### **Client Authentifikation**

In Szenarien, in denen der Server den Client authentifizieren muss, beinhaltet SSL ein Feature, das als Client Authentication bekannt ist. Auf kryptographischem Level arbeitet die Client Authentication viel mit Zertifikatsignierung und Authentifikation am Server - ein Client Zertifikat wird mittels einem kryptographischen Algorithmus wie z.B. RSA signiert. [7]

Client Authentication ist ein elegantes Feature, um Benutzer, die auf Serverdaten zugreifen wollen, zu authentifizieren, indem ein Client Zertifikat ausgetauscht wird. Das bedeutet, dass keine Zugriffe mehr von anonymen Benutzern im Activity Log der Datenbank gespeichert werden, wenn Internetbenutzer auf das System zugreifen. Dieses Zertifikat wird entweder von einer externen Stelle wie z.B. VeriSign ausgestellt oder intern, sodass sichergestellt werden kann, dass die Person die durch das Zertifikat repräsentiert wird, auch die Person ist, die erwartet wird.

Die Client Authentifikation passiert dann, wenn der Server das Zertifikat während dem SSL Handshake über das Netzwerk anfordert. Der Server kontrolliert, ob und wann die Authentifikation stattgefunden hat respektive stattfindet. Ein Client kann nicht anfragen um authentifiziert zu werden. [56]

Mit dieser Methode kann spezieller Zugriff auf die Datenbank basierend auf dem Benutzernamen sichergestellt werden, anstatt Zugriffe für alle anonymen Benutzer einzurichten. Wenn man zum Beispiel Max Mustermann Schreibrechte und allen anderen Internetbenutzern nur Leserechte einräumen möchte, könnte man für Max's Namen auf einer bestimmten Tabelle die Schreibrechte erlauben und für anonyme Benutzer einen Eintrag mit nur Leserechten anlegen. Ohne der Client Authentifikation könnten alle anonymen Benutzer nur auf dem selben Level Zugriffe haben, mit Client Authentifikation wird diese Beschränkung aufgehoben und eine individuelle Festlegung der Zugriffe ermöglicht. [56]

## 6 Zusammenfassung

In dieser Diplomarbeit ist die Analyse und der Systementwurf einer elektronischen Gesundheitsakte für Nutztiere vorgestellt worden.

Im einleitenden Kapitel Grundlagenanalyse 2 wurden derzeit am Markt vorhandene Umsetzungen der elektronischen Gesundheitsakte untersucht. Dabei wurde der ELGA für Menschen ein Abschnitt gewidmet und ein weiterer Abschnitt beschäftigte sich mit den von Tierärzten und Tierkliniken derzeit am häufigsten verwendeten Softwarelösungen.

Die Grundlagenanalyse gab auch einen Überblick über die gesetzliche Lage und lieferte einige Auszüge aus dem Tierarzneimittelkontrollgesetz, der Rückstandskontrollverordnung, der Tierkennzeichnungs- und Registrierungsverordnung sowie der Tiergesundheitsdienst-Verordnung. Es folgte eine Einführung in Standards, die im medizinischen-technischen Bereich angewendet werden und speziell bei der Umsetzung der elektronischen Gesundheitsakte für Nutztiere geeignet erscheinen. Einige dieser Standards wurden auch bei der Entwicklung der ELGA für den Menschen eingesetzt. Die Standards wurden in ihrer Funktion beschrieben und am Schluss wurden Einsatzmöglichkeiten für dieses Projekt aufgezeigt.

In Kapitel Softwarearchitektur 3 wurde allgemein der Begriff Softwarearchitektur erklärt und welche Qualitätskriterien diese ausmachen. Als nächstes wurden zwei weit verbreitete Muster in der Webapplikationsentwicklung vorgestellt, das Model-View-Controller Muster sowie die N-Tier-Architektur. Beide Muster stellen die grundlegende Architektur eines Systems dar, auf der alles aufbaut. Die Abschnitte Datenmodellierung und UML beschreiben grafische Notationen zur Darstellung von konzeptionellen Entwürfen. Datenmodelle beschreiben die Domäne, ohne aber konkret auf die Umsetzung in einer relationalen Datenbank einzugehen. Die Diagrammarten Aktivitätsdiagramm, Komponentendiagramm und Verteilungsdiagramm wurden erklärt, sodass der Leser in weiterer Folge den Architekturentwurf in Form dieser Diagramme versteht. Der Leser dieser Arbeit wurde in das Thema Webservices eingeführt, da diese die Schnittstelle zu externen Systemen bilden sollen.

Das nächste große Kapitel Analyse 4 beschäftigte sich mit den relevanten Stakeholdern des Systems. Diese wurden analysiert und aufbauend auf deren Interessen wurden bei einer Anforderungsanalyse die nichtfunktionalen sowie die funktionalen Anforderungen definiert. Die funktionalen Anforderungen wurden detailliert beschrieben und für einen Auszug davon wurde mit UML ein Anwendungsfalldiagramm erstellt. Dieses Kapitel wurde mit einer Einführung in unterschiedliche Arten von Zugriffskontrollen und daraus resultierend der Erarbeitung der funktionalen Rollen des Systems und einem Regelkonzept für die Zugriffskontrolle abgeschlossen.

Das letzte große Kapitel bildet das Systemdesign und der Architekturentwurf der elektronischen Gesundheitsakte für Nutztiere. Die Architekturkonzepte, die im Grundlagenkapitel vorgestellt wurden, sind hier praktisch zum Einsatz gekommen. Die Systemarchitektur folgt dem Model-View-Controller Pattern und hat eine 3-Schicht Architektur. Diese Konzepte wurden grafisch veranschaulicht, um für ein besseres Verständnis zu sorgen. Das Datenmodell wurde stark vereinfacht dargestellt, die Attribute der einzel-

nen Tabellen der Domäne wurden nicht definiert, dies soll Teil einer eigenen Arbeit auf diesem Gebiet werden. Durch das Datenmodell wird eine statische Sicht auf das System gewährt. Die Darstellung der Hauptaktivität, nämlich Gesundheitsdaten zu erfassen, in einem Aktivitätsdiagramm liefert eine dynamische Sichtweise und stellt den Ablauf dar. Das Komponentendiagramm dient der Darstellung der logischen Struktur des Systems. Die Komponenten haben sich aus den Ergebnissen der Analyse ergeben und wurden hier definiert und beschrieben. Das Verteilungsdiagramm dient der physischen Darstellung und zeigt auf, welche Knoten und wieviele es davon geben muss, um eine geeignete Ausfallsicherheit zu garantieren. Abschließend wurde die Systemarchitektur mit ihren groben Bausteinen veranschaulicht und die einzelnen Teile beschrieben. Da der Sicherheitsaspekt bei der Entwicklung der ELGA für Nutztiere ein sehr wichtiger ist, müssen geeignete Maßnahmen getroffen werden, um Daten vor unerlaubten Zugriffen zu schützen. Daher bildet eine Einführung in für die Umsetzung relevante Sicherheitsprotokolle den Abschluss dieser Arbeit.

## 7 Ausblick

Das Ziel dieser Arbeit war die Analyse und darauf aufbauend die Entwicklung eines konzeptionellen Architekturentwurfes der elektronischen Gesundheitsakte für Nutztiere. In folgenden Arbeiten müssen die hier erarbeiteten Grundlagen weiter verfeinert werden. Die hier gewonnenen Ergebnisse bieten dazu die nötige Grundlage, um darauf aufbauend die konkrete Entwicklung durchzuführen.

Die gesetzlichen Rahmenbedingungen sind ein komplexer Bereich, der noch genauere Untersuchung erfordert. Die für dieses Thema greifenden Gesetze und Verordnungen wurden auszugsweise vorgestellt und einige Paragraphen zitiert. Sie können als Basis für eine tiefere Auseinandersetzung dienen.

Im Zuge einer umfangreichen Stakeholderanalyse kann ein Fragenkatalog entwickelt werden, um von Personen des Projektumfeldes Aufschluss über wichtige Details, Wünsche und Anforderungen an die elektronische Gesundheitsakte sowie Informationen aus dem Arbeitsalltag zu erlangen. Weiters wird sich dann wahrscheinlich auch immer mehr herausstellen, wer zu den Unterstützern der Projektidee gehört und ob Widerstände vorhanden sind. Ziel soll es sein, vorhandene Unterstützungspotentiale zu nutzen und die Widerstände der Projektgegner auf geeignete Weise abzuschwächen. Je nach Art der Analyse können die Fragen in Form eines Fragebogens an einen ausgewählten Pool von Stakeholdern gesendet werden oder aber in Form von persönlichen Interviews durchgeführt werden. Die Stakeholderanalyse wird viele Aufschlüsse über wichtige Anforderungen geben, d.h. die in dieser Arbeit definierten Anforderungen werden möglicherweise noch erweitert bzw. verfeinert werden müssen.

Aus den Ergebnissen einer tiefgehenden Erforschung der gesetzlichen Lage sowie den Stakeholderanalysen müssen die zu erfassenden Daten konkretisiert werden. Dabei ergibt sich eine Reihe von gesetzlich erforderlichen Daten, die je nach Nutztierart erfasst werden müssen und einer Reihe von Daten, die von den Stakeholdern und Benutzern als notwendig erachtet werden.

Die Entwicklung eines geeigneten Datenbankmodells für die elektronische Gesundheitsakte bietet ebenfalls genug Stoff für eine eigene Arbeit. Es müssen sämtliche Tabellen konkret mit allen Spalten, Schlüsseln und Beziehungen als Ergebnis entstehen. In dieser Arbeit wurde ein konzeptionelles Datenmodell erstellt, welches als Grundlage für die Planung und Implementierung der Datenbank dienen soll.

In weiterer Folge können Entwürfe für das User Interface Design erstellt werden. Die Benutzerfreundlichkeit wird maßgeblich am Erfolg dieses Projektes beteiligt sein, da die meisten Tierärzte bereits gute Praxismanagementsoftware verwenden und ein Umstieg auf ein anderes System bzw. die parallele Verwendung soll nicht durch schlechte Bedienbarkeit verhindert werden.

Die Benutzeroberfläche kann zusammen mit der Implementierung in eine Arbeit gefasst werden. Bevor jedoch implementiert werden kann, müssen die Technologien für die Umsetzung dieser Webplattform ausgewählt werden. Welches Datenbankmanagementsystem soll zum Einsatz kommen, welcher Applikationsserver, welche Programmiersprache oder welches Framework für die clientseitige Benutzeroberfläche verwendet wird, dies sind alles Fragen, die in weiterer Folge eine genauere Betrachtung erfor-

dern. Sobald die Technologieentscheidungen getroffen sind, kann aufbauend auf dem in dieser Arbeit erstellten Architekturentwurf und den Ergebnissen aus weiteren Arbeiten über die hier vorgestellten Themenbereiche die Implementierung erfolgen.

Weitere Ideen wären zum Beispiel, die elektronische Gesundheitsakte für Nutztiere sowohl als mobile Version des Systems als auch als native Applikation für zum Beispiel Smartphones und Tablets mit Android Betriebssystem umzusetzen. So können Ärzte, Landwirte oder andere Benutzergruppen vor Ort ihre Daten erfassen, was den Dokumentationsaufwand verringern würde.

Natürlich wird man sich im Laufe der Entwicklung auch mit der Markteinführung beschäftigen müssen. Wer sollen die Betreiber der Seite sein bzw. welche Kooperationen können hier eingegangen werden.

Die Ergebnisse dieser Analyse und des Architekturentwurfes können eine gute Basis zur Realisierung der elektronischen Gesundheitsakte für Nutztiere bieten.

# Abbildungsverzeichnis

2.1	easyVet Karteikasten Ansicht . . . . .	9
2.2	easyVet Ansicht einer Behandlung . . . . .	10
2.3	VetInf Ansicht einer Behandlung . . . . .	12
2.4	Animal Office Kleintier Stammdatenblatt . . . . .	13
2.5	Animal Office Großtier Anpaarungshilfe . . . . .	14
2.6	UML Klassendiagramm des Reference Information Models . . . . .	22
3.1	Multiplizitäten und Nachrichtenfluss in einem MVC Modell [61] . . . . .	37
3.2	Include und Extend Beziehung in einem Anwendungsfalldiagramm . . . . .	43
3.3	Konzepte in einem Aktivitätsdiagramm . . . . .	45
3.4	Typisches UML Komponentendiagramm [38] . . . . .	46
3.5	Zusammenspiel zwischen UDDI, Webservice Requester und Provider . . . . .	49
3.6	Rollenmodell der rollenbasierten Zugriffskontrolle . . . . .	51
4.1	Anwendungsfalldiagramm Login, Logout und Zugriffsprotokollierung . . . . .	80
4.2	Anwendungsfalldiagramm für die Erfassung von Gesundheitsdaten . . . . .	81
5.1	MVC Architektur der elektronischen Gesundheitsakte für Nutztiere . . . . .	88
5.2	3-Tier-Architektur der elektronischen Gesundheitsakte für Nutztiere . . . . .	89
5.3	Datenmodell der Gesundheitsakte für Nutztiere . . . . .	91
5.4	Aktivitätsdiagramm . . . . .	93
5.5	Komponentendiagramm . . . . .	96
5.6	Verteilungsdiagramm . . . . .	98
5.7	Systemarchitektur . . . . .	99
5.8	Zusammenspiel zwischen Standards und Konzeption . . . . .	102

# Tabellenverzeichnis

2.1	Werte für relatedDocument.typeCode (CNE) . . . . .	23
2.2	Beispiele für Komponenten Abkürzungen . . . . .	25
2.3	SNOMED Module . . . . .	29
4.1	Use Case Login . . . . .	60
4.2	Use Case Logout . . . . .	60
4.3	Use Case Protokollierung aller Zugriffe . . . . .	61
4.4	Use Case Einpflegen neuer Betriebsdaten . . . . .	62
4.5	Use Case Bearbeiten bestehender Betriebsdaten . . . . .	63
4.6	Use Case Einpflegen neuer Tierdaten . . . . .	63
4.7	Use Case Suchfunktion . . . . .	64
4.8	Use Case Betriebsliste ansehen . . . . .	65
4.9	Use Case Detailansicht eines Betriebes ansehen . . . . .	66
4.10	Use Case Auswahl einer Tierart . . . . .	67
4.11	Use Case Tierliste ansehen . . . . .	68
4.12	Use Case Detailansicht eines Tieres ansehen . . . . .	69
4.13	Use Case Erfassung von Gesundheitsdaten für ein Tier . . . . .	69
4.14	Use Case Hochladen von Dokumenten . . . . .	70
4.15	Use Case Tierarztindex ansehen . . . . .	71
4.16	Use Case Tierarzt für einen Betrieb freischalten . . . . .	72
4.17	Use Case Generieren von Berichten . . . . .	73
4.18	Use Case Arzneimittelkatalog ansehen . . . . .	74
4.19	Use Case Meldungen erfassen . . . . .	75
4.20	Use Case Meldungen senden . . . . .	76
4.21	Use Case Berechtigungen anlegen . . . . .	77
4.22	Use Case Abfrage der Zugriffsprotokollierung . . . . .	78
4.23	Use Case Schnittstelle zu gängigen Praxissoftwarelösungen . . . . .	79
4.24	Funktionale Rollen des Systems . . . . .	83

# Abkürzungsverzeichnis

- AMA** Agrarmarkt Austria
- CDA** Clinical Document Architecture
- CMT** Convergent Medical Terminology
- CNE** Coded, No Extensions
- CWE** Coded, WITH Extensions
- DAC** Discretionary Access Control
- DICOM** Digital Imaging and Communications in Medicine
- EHR** Electronic Health Record
- ELGA** Elektronische Gesundheitsakte
- ER-Modell** Entity-Relationship-Modell
- HL7** Health Level 7
- HTTP** Hypertext Transfer Protocol
- IOD** Information Object Definition
- LKV** Landeskontrollverband
- LOINC** Logical Observation Identifiers Names and Codes
- MAC** Mandatory Access Control
- MVC** Model View Controller
- N.A.V.** Nomina Anatomica Veterinaria
- ORM** Objekt-Relationales Mapping
- PHR** Personal Health Record
- RBAC** Role Based Access Control
- RDBMS** Relationales Datenbankmanagementsystem
- RDV4M** Rinderdatenverbund für Mitglieder
- RIM** Reference Information Model
- RPC** Remote Procedure Call
- RSA** Rivest-Shamir-Adelman

**SNOMED** Systematized Nomenclature of Medicine and Veterinary Medicine

**SOAP** Simple Object Access Protocol

**SSL** Secure Socket Layer

**TGD** Tiergesundheitsdienst

**TLS** Transport Layer Security

**UDDI** Universal Discovery Description Integration

**ULP** Upper Layer Protocol

**UML** Unified Modeling Language

**URI** Uniform Resource Identifier

**WSDL** Web Services Definition Language

**XML** Extensible Markup Language

**ZAR** Zentrale Arbeitsgemeinschaft österreichischer Rinderzüchter

## Web Referenzen

- [1] *AgrarMarkt Austria Lebendrinderkennzeichnung und -registrierung*. 2. März 2012. URL: <http://www.ama.at/Portal.Node/public?genetics.am=PCP&p.contentid=10007.19455>.
- [2] *ANIMALoffice*. 8. Jan. 2012. URL: <http://www.animal-office.at/>.
- [6] R. Berger. *easyVET Software Demo*. 8. Jan. 2012. URL: <http://easyvet.eu/demo.html>.
- [12] *DICOM*. 6. März 2012. URL: <http://www.dicomworks.com/>.
- [14] *eAMA - Internetserviceportal der Agrarmarkt Austria*. 2. März 2012. URL: <https://services.ama.at/servlet/>.
- [15] *easyVET*. 8. Jan. 2012. URL: <http://www.vetsxl.com/cm/vetz/Products/easyVET/default.aspx>.
- [17] *ELGA GmbH*. 9. Jan. 2012. URL: <http://www.elga.gv.at/>.
- [22] *Gesundheitsministerium*. 7. März 2012. URL: <http://bmg.gv.at>.
- [23] *Gesundheitsportal*. 9. Jan. 2012. URL: <https://www.gesundheit.gv.at/Portal.Node/ghp/public/content/elga-gmbh-institutionen-gesundheitssystem.html>.
- [24] IBM Österreich GmbH. *IBM Machbarkeitsstudie ELGA*. 9. Jan. 2012. URL: [http://www.elga.gv.at/fileadmin/user\\_upload/uploads/download\\_Papers/Arge\\_Papers/Machbarkeitsstudie\\_ELGA\\_Endbericht\\_21112006.pdf](http://www.elga.gv.at/fileadmin/user_upload/uploads/download_Papers/Arge_Papers/Machbarkeitsstudie_ELGA_Endbericht_21112006.pdf).
- [31] *IHE Framework*. 9. Jan. 2012. URL: [http://www.ihe.net/Technical\\_Framework/](http://www.ihe.net/Technical_Framework/).
- [32] *IrfanView*. 6. März 2012. URL: <http://www.irfanview.com/>.
- [34] *Lebensministerium*. 7. März 2012. URL: <http://www.lebensministerium.at>.
- [36] *Logical Observation Identifiers Names and Codes (LOINC)*. 6. März 2012. URL: <http://loinc.org>.
- [45] *Nomenklatur Definiton*. 2. Mai 2012. URL: <https://www.tk.de/rochelexikon/ro25000/r26976.000.html>.
- [47] *Object Management Group*. 26. März 2012. URL: <http://www.omg.org>.
- [52] *Rinderzucht Austria*. 3. März 2012. URL: <http://www.zar.at/>.
- [56] J. Spera. *SSL client authentication: It's a matter of trust*.
- [59] *Tiergesundheitsdienst*. 9. Jan. 2012. URL: <http://www.tgd.at/>.
- [60] *Tiergesundheitsdienst-Programme*. 9. Jan. 2012. URL: [http://www.bmg.gv.at/home/Schwerpunkte/Tiergesundheit/Tiergesundheitsdienst/Tiergesundheitsdienst\\_Programme\\_TGD\\_Programme\\_](http://www.bmg.gv.at/home/Schwerpunkte/Tiergesundheit/Tiergesundheitsdienst/Tiergesundheitsdienst_Programme_TGD_Programme_).
- [63] *Vetinf*. 8. Jan. 2012. URL: <http://www.vetinf.de/>.
- [64] *W3C*. 13. Apr. 2012. URL: <http://www.w3.org/>.

# Literatur

- [3] J. M. Barnes. “Object-Relational Mapping as a Persistence Mechanism for Object-Oriented Applications”. In: *Honors Projects. Paper 6* (2007).
- [4] M. Barrio und P. de la Fuente. “Software architecture: Object vs. Process Approach”. In: *XVII International Conference of the Chilean Computer Science Society* (1997), S. 9–15.
- [5] R.M. Bastos und D.D.A. Ruiz. “Extending UML activity diagram for workflow modeling in production systems”. In: *HICSS. Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (2002), S. 3786–3795.
- [7] L. D. Bisel. “The Role of SSL in Cybersecutiry”. In: *IT Professional* 9 (2007), S. 22–25.
- [8] J.T. Case, J. Wilcke und A. Hahn. “Status of Animal Health Information Standards in the United States”. In: (2000).
- [9] P. P.-S. Chen. “The Entity-Relationship Model - Toward a Unified View of Data”. In: *ACM Transactions on Database Systems* 1.1 (1976), S. 9–36.
- [10] W. Chou. “Inside SSL: The Secure Sockets Layer Protocol”. In: *IT Professional* (2002), S. 47–52.
- [11] A. Dedeke und B. Lieberman. “Qualifying use case diagram associations”. In: *Computer* (2006), S. 23–29.
- [13] R.H. Dolin u. a. “The HL7 Clinical Document Architecture”. In: (2004).
- [16] C. Egger-Danner u. a. *Gesundheitsmonitoring Rind - Projektbeschreibung*. Techn. Ber. Zentrale Arbeitsgemeinschaft österreichischer Rinderzüchter, 2007.
- [18] D. F. Ferraiolo und D. R. Kuhn. “Role Based Access Controls”. In: *15th National Computer Security Conference* (1992), S. 554–563.
- [19] A. Ferreira, R. Correia und L. Antunes M. Brito. “Usable Access Control Policy and Model for Healthcare”. In: *24th International Symposium on Computer-Based Medical Systems (CBMS)* (2011), S. 1–6.
- [20] J. Fick und R. Doluschitz. “Vernetzung tiergesundheitsrelevanter Daten zu einem integrierten Tiergesundheitssystem”. In: *Züchtungskunde* (2007), S. 11–19.
- [21] M. Galster und E. Bucherer. “A Taxonomy for Identifying and Specifying Non-Functional Requirements in Service-Oriented Development”. In: *IEEE Congress on Services - Part I* (2008), S. 345–352.
- [25] IBM Österreich GmbH. *Machbarkeitsstudie betreffend Einführung der elektronischen Gesundheitsakte (ELGA) im österreichischen Gesundheitswesen*. Endbericht erstellt von IBM Österreich GmbH im Auftrag der Bundesgesundheitsagentur. 2006.
- [26] D. Gornik. *Entity relationship modeling with UML*. Introductory. IBM, 2005.

- [27] T. Grechenig u. a. *Softwaretechnik: Mit Fallbeispielen aus realen Entwicklungsprojekten*. 1. Aufl. München: Pearson Studium, 2009.
- [28] H. Gunzer. *Introduction to Web Services*. Borland, 2002.
- [29] J. Guo. “An Approach for Modeling and Designing Software Architecture”. In: *10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems* (2003), S. 89–97.
- [30] E.-W. Huang u. a. “Generating Standardized Clinical Documents for Medical Information Exchanges”. In: *IT Professional* 12 (2010), S. 26–32.
- [33] W. Kehe. “Security Model Based on Network Business Security”. In: *International Conference on Computer Technology and Development, 2009. ICCTD '09*. 1 (2009), S. 577–580.
- [35] B. Liu u. a. “Design and implementation of information exchange between HIS and PACS based on HL7 standard”. In: *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on* (2008), S. 552–555.
- [37] F. Losavio u. a. “Quality Characteristics for Software Architecture”. In: *Journal of Object Technology* 2.2 (2003), S. 133–150.
- [38] C. Lüer und D. S. Rosenblum. “UML Component Diagrams and Software Architecture - Experiences from the WREN Project”. In: *1st ICSE Workshop on Describing Software Architecture with UML*. 2001, S. 79–82.
- [39] M. J. Mahemoff und L. J. Johnston. “Handling Multiple Domain Objects with Model-View-Controller”. In: *TOOLS '99 Proceedings of the 32nd International Conference on Technology of Object-Oriented Languages* (1999), S. 28–39.
- [40] P. D. Manuel und J. AlGhamdi. “A data-centric design for n-tierarchitecture”. In: *Information Sciences* 150 (2003), S. 195–206.
- [41] C. McDonald u. a. “Logical Observation Identifiers Names and Codes (LOINC) - User’s Guide”. In: (2011).
- [42] N. Medvidovic und R.N. Taylor. “Software architecture: foundations, theory, and practice”. In: *32nd International Conference on Software Engineering, ACM/IEEE 2* (2010), S. 471–472.
- [43] R. L. Miller. *The OSI Model: An Overview*. GSEC Practical Assignment Version 1.2e SANS Institute.
- [44] M. Mustra, K. Delac und M. Grgic. “Overview of the DICOM Standard”. In: *50th International Symposium ELMAR-2008* (2008).
- [46] F. Oberacher. “Die XML basierte „Clinical Document Architecture“(CDA) für medizinische Befundberichte und Arztbriefe”. Magisterarb. Private Universität für Gesundheitswissenschaften Medizinische Informatik und Technik Tirol (UM-IT), 2006.
- [48] P. F. Patel-Schneider und B. Swartout. “Description-Logic Knowledge Representation System Specification from the KRSS Group of the ARPA Knowledge Sharing Effort”. In: (1993).
- [49] F. Prior. “Database Access Methods for Medical Imaging: DICOM, SQL and HTML”. In: *Proceedings of IMAC* (1996).

- [50] B. Qing-hai und Z. Ying. “Study on the access control model in information security”. In: *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC)* (2011), S. 830–834.
- [51] Gmon Rind. *Gesundheitsmonitoring Rind: Entwicklung einer Zuchtwertschätzung für Gesundheitsmerkmale*. Zentrale Arbeitsgemeinschaft österreichischer Rinderzüchter. Dresdnerstraße 89/19, 1200 Wien, 2010.
- [53] D. J. Rothwell und R.A. Cote. “Managing Information with SNOMED: Understanding the Model”. In: *Proc AMIA Annu Fall Symp.* (1996), S. 80–83.
- [54] G. Schadow, C.N. Mead und D.M. Walker. “The HL7 Reference Information Model Under Scrutiny”. In: *MIE*. Hrsg. von Arie Hasman u. a. Bd. 124. *Studies in Health Technology and Informatics*. IOS Press, 2006, S. 151–156.
- [55] Q. Siddique. “Unified Modeling Language to Object Oriented Software Development”. In: *International Journal of Innovation, Management and Technology* 1.3 (2010), S. 264 –268.
- [57] A. Ströher. “Die elektronische Gesundheitsakte (ELGA) in Österreich - Eine Evaluierung in Bezug auf funktionale Benutzeranforderungen”. In: *German Journal for Young Researchers* (2010).
- [58] J. Tie. “Study on application model of three-tiered architecture”. In: *Second International Conference on Mechanic Automation and Control Engineering (MACE)* (2011), S. 7715 –7718.
- [61] M. Veit und S. Herrmann. “Model-View-Controller and Object Teams: A Perfect Match of Paradigms”. In: *2nd Int’ Conf. on Aspect-Oriented Software Development*. ACM Press, 2003, S. 140 –149.
- [62] Prepared by the International Committee on Veterinary Gross Anatomical Nomenclature (I.C.V.G.A.N.) *Nomina Anatomica Veterinaria Fifth Edition*. Published by the Editorial Committee Hannover, Columbia, Gent, Sapporo. 2005.
- [65] S. Warren u. a. “A distributed infrastructure for veterinary telemedicine”. In: *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE* (2003), 1394 –1397 Vol.2.
- [66] T. Weichert. *Die elektronische Gesundheitskarte*. DuD - Datenschutz und Datensicherheit. 2004.
- [67] JR. Wilcke. *SNOMED, What’s in it? Who’s Involved? What’s it for?* AVMA Vendors Meeting Convention - Presentation. 1999.
- [68] L. Xiaoqi u. a. “Research and implementation of medical information format conversion based on HL7 Version 2.x”. In: *Computer Science and Service System (CSSS), 2011 International Conference on* (2011), S. 2440 –2443.
- [69] *Software Engineering and Computer Systems, Part III*. Second International Conference, ICSECS 2011.