Ein Sicherheitskonzept für elektronische Prüfungen an Hochschulen auf Basis eines virtuellen, ticketbasierten Dateisystems

Vom Fachbereich 12 Elektrotechnik und Informatik der Universität Siegen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

Genehmigte Dissertation

von

Dipl.-Ing. Andreas Hoffmann

1. Gutachter: Prof. Dr.rer.nat Roland Wismüller
2. Gutachter: Prof. Dr.rer.nat Dogan Kesdogan

Tag der Einreichung: 07.07.2010
Tag der mündlichen Prüfung: 30.09.2010

gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

Kurzfassung

Aufgrund des Bologna-Prozesses und der damit verbundenen Umstellung auf die BA/MA-Studiengänge spielen elektronische Prüfungen an Hochschulen eine immer größer werdende Rolle. Denn durch die Umstellung erhöhte sich die Anzahl der Prüfungen, was wiederum einen erheblichen Mehraufwand für die Lehrenden bedeutete. Der Einsatz von elektronischen Prüfungen kann dabei aber nicht nur den Mehraufwand deutlich minimieren, sondern kann auch für einen Kulturwandel im Bereich der Prüfungen an Hochschulen sorgen.

Jedoch können die traditionellen papierbasierten Prüfungen nicht so ohne weiteres durch die elektronische Form ersetzt werden. Denn zum einen unterliegen die elektronischen Prüfungen den gleichen formalen Ansprüchen wie die papierbasierten Prüfungen und zum anderen sind die technischen, administrativen und datenschutzrechtlichen Anforderungen zu beachten. Ein Spezialfall ist dabei die Nichtabstreitbarkeit der Lösungen der Prüfungsteilnehmer. Was bei den papierbasierten Prüfungen im Streitfall mittels handschriftlichen Gutachtens bewiesen werden kann, muss bei den elektronischen Prüfungen durch eine vom Prüfungsteilnehmer abgegebene Unterschrift, sei es auf Papier oder elektronisch durch digitale Signaturen, erfolgen. Die derzeitigen Prüfungssysteme stellen z.B. die Rechtssicherheit nur durch Medienbrüche und hohen administrativen Aufwand sicher, was weder praktikabel noch nachhaltig ist.

Das in dieser Arbeit dargestellte Sicherheitskonzept berücksichtigt sowohl die technischen, administrativen als auch die formalen Anforderungen, sowie den Datenschutz und Datensicherheit von elektronischen Prüfungen an Hochschulen. Dabei wird die Notwendigkeit der qualifizierenden digitalen Signaturen begründet und wie diese zusammen mit einem virtuellen, ticketbasierten Dateisystem verwendet werden können. Das virtuelle, ticketbasierte Dateisystem wurde dabei vom Lösungskonzept der elektronischen Gesundheitskarte adaptiert, u.a. deshalb, weil damit die Anonymität trotz Authentizität möglich ist und sowohl die Studierenden als auch die Lehrenden „Herr Ihrer Daten“ bleiben. Das Sicherheitskonzept kann für bestehende Prüfungssystem verwendet werden, was durch eine prototypische Umsetzung an einem webbasierten Prüfungssystem gezeigt werden konnte. Des Weiteren bietet vor allem das virtuelle, ticketbasierte Dateisystem einen multifunktionalen Nutzen für weitere Hochschulanwendungen wie z.B. Evaluierungen und elektronische Übungssysteme.

Danksagungen

Diese Arbeit wäre ohne die Unterstützung verschiedener Personen sicherlich nicht möglich gewesen. Diesen Personen möchte ich an dieser Stelle danken.

Zuerst möchte ich mich bei Herrn Prof. Dr. Roland Wismüller für die intensive Betreuung bedanken. In vielen Gesprächen hat er mir immer wieder neue Impulse und Anregungen zu meiner Arbeit gegeben und die kontinuierliche Entwicklung meiner wissenschaftlichen Arbeit gefördert.

Meinen Kollegen des Lehrstuhls Betriebssysteme und verteilte Systeme und den Mitarbeitern des Zentrums für ökonomische Bildung in Siegen (ZöBiS) möchte ich für ihre konstruktive Kritik und ihre Unterstützung danken.
Außerdem danke ich den Studenten Jens Brennscheidt, Markus Bode, Malte Gronau, Alexander Daraban, Klaus Bernshausen, Michael Garbas und Christoph Hellweg, weil sie mir durch ihre studentischen Arbeiten immer neue Impulse geliefert haben und zum Gelingen dieser Arbeit beigetragen haben.

Ein Dank geht auch an Herr Prof. Dr. Dogan Kesdogan, der sich bereit erklärt hat das Zweitgutachten für diese Arbeit zu erstellen.

Fürs Korrekturlesen geht ein großer Dank an meine Schwägerin Martina Hoffmann und an meinen guten Freund Adrian Greipel.

Ein ganz besonderer Dank geht an meine Freundin Katharina Jung, für ihre liebevolle und unermüdliche Unterstützung während der Promotion.

Meiner Familie und ganz besonders meinen Eltern möchte ich mich für ihre emotionale Unterstützung bedanken. Ohne meine Eltern wäre ein Studium und eine Doktorarbeit niemals möglich gewesen.

Siegen, im Oktober 2010
A. Hoffmann

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Die Hochschulrektorenkonferenz (HRK) hat bereits 2003 in ihrer Empfehlung „Qualitätssicherung im Zuge des Bologna-Prozesses“ gefordert, die Anerkennung von computergestützten Lehrangeboten zu gewährleisten und geeignete Prüfungsformen zu entwickeln [Hop03]. Gleichzeitig führte die Umstellung auf die Bachelor- und Masterstudiengänge dazu, dass die Lehrenden die Prüfungen zeitnah zum Studium anbieten mussten. Des Weiteren sollten den Studierenden während der Lernphasen entsprechende Feedbackmaßnahmen zur Verfügung gestellt werden. Aber auch in den Studienfächern, in denen eine Bachelor-/ Master-Umstellung nicht eingeführt wurde (z.B. Medizin), führten Änderungen in der Approbationsordnung zu einem erhöhten Prüfungsaufkommen [GSS+05]. So waren es ab dem Jahr 2000 vor allem die medizinischen Studiengänge, die computergestützte Prüfungen einsetzten. Dies führte dann gleichzeitig auch zu Zielvereinbarungen zwischen den Bundesländern und deren Hochschulen, in denen sich diese verpflichten, die elektronischen Prüfungen als neue Prüfungsform anzubieten. Auszug aus der Zielvereinbarung zwischen dem Bayerischen Staatsministerium Wissenschaft, Forschung und Kunst und der Universität Augsburg [BSfW08]:

> „*Durch die Einführung der BA/MA-Studiengänge und der Modularisierung der Lehramtsstudiengänge kommen gewaltige Prüfungsbelastungen insbesondere auf die Philosophische Fakultät II zu, zum einen, weil stark nachgefragte Studiengänge dort etabliert sind, zum anderen, weil alle Lehramtsstudierenden die obligatorischen erziehungswissenschaftlichen Anteile ihres Studiums absolvieren müssen. Aus Gründen der Qualitätssicherung kann dies nur durch eine umfassende Ausweitung von eLearning und ePrüfungen bewältigt werden.*“ [BSfW08]

Die Ausweitung von eLearning und ePrüfungen an Hochschulen bedarf einer sicherheitstechnischen Betrachtung der damit verbundenen Softwaresysteme. Denn die Systeme arbeiten mit sehr persönlichen Daten und ein Missbrauch der Daten kann schwerwiegende Folgen für die berufliche Zukunft der Anwender haben. Diese Arbeit konzentriert sich aber nur auf die elektronischen Prüfungssysteme.
Die Sicherheit von eLearning-Systemen ist nicht Bestandteil dieser Arbeit, weil dies bereits ausführlich in [Wei05] und vor allem in [Eib10] betrachtet wurde.

1.1 Problemstellung

Der Einsatz von elektronischen Prüfungen birgt zum einen viele Vorteile gegenüber den traditionellen papierbasierten Prüfungen. Dies sind vor allem die automatisierte Auswertung der Lösungen und das Anlegen von so genannten Fragenpools, die die Wiederverwendung von Fragen vereinfachen.
Dennoch bieten die elektronischen Prüfungsformen auch Risiken: Denn die Anforderungen, die an eine papierbasierte Prüfung gestellt werden, müssen auch für elektronische Prüfungen gelten. Dazu zählen alle prüfungsrechtlichen und datenschutzrechtlichen Anforderungen und ganz besonders die Nichtabstreitbarkeit der Lösungen der Prüfungsteilnehmer. Was bei den papierbasierten Prüfungen im Streitfall mittels handschriftlichen Gutachtens bewiesen werden kann, muss bei den elektronischen Prüfungen durch eine vom Prüfungsteilnehmer abgegebene Unterschrift, sei es auf Papier oder elektronisch durch digitale Signaturen, erfolgen. Des Weiteren ist zu gewährleisten, dass die Prüfungsfragen nicht vor der Freigabe der Prüfung einsehbar sind und dass nur diejenigen die Prüfungen durchführen können, die dazu berechtigt sind.
Der Datenschutz spielt neben der Datensicherheit und der Rechtssicherheit eine entscheidende Rolle. Jedoch widersprechen sich die Anforderungen an den Datenschutz und an die Datensicherheit teilweise komplett[1]. So stehen z.B. die Anonymität als Datenschutzanforderung der Authentizität der Prüfungsteilnehmer gegenüber.
Die derzeitigen Realisierungen bei den etablierten Prüfungssystemen lösen diesen Konflikt nur durch Medienbrüche und hohen administrativen Aufwand, der weder praktikabel noch nachhaltig ist (siehe Abschnitt 4.1).
Das liegt u.a. darin begründet, dass die Umsetzung aller Maßnahmen in Form eines Sicherheitskonzeptes einen hohen Aufwand für die gesamte Hochschule

[1] https://www.bsi.bund.de/cae/servlet/contentblob/561570/publicationFile/30423/15JahreITGrundschutz_Schaar_BfDI_pdf.pdf, aufgerufen am 30.06.2010

bedeuten und somit nur durch einen multifunktionalen Einsatz gerechtfertigt werden kann. Denn gerade die Anforderungen an die Prüfungen finden sich auch bei anderen Anwendungen wieder, so z.B. bei elektronischen Übungen, Evaluationen, Hochschulinformationssystemen etc.
Ein Sicherheitskonzept für die elektronischen Prüfungen muss daher einen multifunktionalen Charakter besitzen und in eine bestehende Hochschullandschaft integrierbar sein.

1.2 Ziele und wissenschaftlicher Gewinn

Das Ziel dieser Arbeit ist, die Anforderungen an die Datensicherheit, an den Datenschutz sowie die prüfungsrechtlichen Anforderungen zu definieren und die Umsetzungen der Anforderungen in einem Sicherheitskonzept trotz der Widersprüche einiger Anforderungen zu realisieren und dennoch unabhängig vom verwendeten Prüfungssystem zu bleiben.
In dieser Arbeit wird ein Sicherheitskonzept entwickelt, das nicht nur die technischen, sondern auch die formalen und administrativen Anforderungen berücksichtigt. Das Sicherheitskonzept ist multifunktional verwendbar und lässt sich in eine bestehende Hochschullandschaft integrieren.
Dabei wird die Notwendigkeit der qualifizierenden digitalen Signaturen begründet, die zusammen mit einem virtuellen, ticketbasierten Dateisystem (vtD) verwendet werden. Das vtD kann nahezu vollständig vom Lösungskonzept der elektronischen Gesundheitskarte (eGK) übernommen werden. Zum einen werden dadurch nur standardisierte Verfahren verwendet und zum anderen löst das vtD die Konflikte zwischen den Anforderungen des Datenschutzes und der Datensicherheit auf. Mit dem vtD ist u.a. die Anonymität trotz Authentizität möglich.
Die formalen Anforderungen werden in dieser Arbeit durch das Zusammenfassen der prüfungsrechtlichen Anforderungen aufgezeigt und wie diese teilweise durch die technischen Maßnahmen (und vor allem das vtD) umgesetzt werden können.
Die administrativen Anforderungen und deren Umsetzung werden ebenfalls beschrieben. So entsteht ein Sicherheitskonzept, das alle Aspekte bzgl. der Sicherheit von elektronischen Prüfungen berücksichtigt.

1.3 Aufbau der Arbeit

Ausgehend von einem Überblick über die verschiedenen Prüfungsformen an Hochschulen werden in Kapitel 2 existierende Organisationsmodelle der Uni-

versitäten Zürich, Bremen, Duisburg-Essen und Münster für elektronische Prüfungen betrachtet. Alle Modelle berücksichtigen den gesamten Prüfungsprozess von der Prüfungsvorbereitung über die Durchführung bis hin zur Auswertung, Einsicht und Archivierung. Allerdings orientieren sich die Modelle überwiegend an den Gegebenheiten der jeweiligen Hochschule und sind nur bis zu einem gewissen Grad allgemeingültig. Aus diesem Grund wird dann ein gemeinschaftliches Organisationsmodell abgeleitet, um die beteiligten Akteure und Prozesse genau zu definieren.

In Kapitel 3 werden die Anforderungen bzgl. der Sicherheit und des Datenschutzes der elektronischen Prüfungen analysiert. Die Sicherheitsanforderungen sind dabei unterteilt in die allgemeinen Sicherheitsanforderungen sowie in prüfungsrechtliche bzw. urheberrechtliche Anforderungen. Vor allem die Fragen der Formvorschrift, Betrugssicherheit und Nichtabstreitbarkeit sind für eine rechtssichere Durchführung entscheidend und werden deshalb besonders ausführlich betrachtet.
Die Sicherheitsanalyse ergibt, dass der Einsatz von qualifizierenden, digitalen Signaturen Voraussetzung für eine rechtssichere Prüfungsdurchführung ist. Als Ergebnis der Analyse werden insgesamt 25 Einzelanforderungen an die Sicherheit und den Datenschutz definiert. Anschließend werden die Maßnahmen beschrieben, wie diese formal, technisch oder administrativ umzusetzen sind.

Die Sicherheitskonzepte der verbreitetsten Prüfungssysteme im deutschsprachigen Raum werden in Kapitel 4 anhand der definierten Anforderungen aus Kapitel 3 betrachtet. Das Ergebnis ist, dass die Systeme die allgemeinen Anforderungen umsetzen, aber nahezu keines der Systeme die digitalen Signaturen einsetzt. Die Umsetzung einer der wichtigsten Anforderungen - der Nichtabstreitbarkeit der Prüfungsangaben und -lösungen - wurde (wenn überhaupt) nur durch Medienbrüche realisiert. Ein Sicherheitskonzept, das alle Anforderungen umsetzt und dabei unabhängig vom verwendeten Prüfungssystem bleibt, existiert nicht.
Aufgrund der Notwendigkeit der qualifizierenden Signaturen, werden anschließend existierende Signaturkartenkonzepte betrachtet, die sich auf die elektronischen Prüfungen anpassen lassen und sich auch in die bestehende Systemlandschaft der Hochschule integrieren lassen. Die Anforderungen an die Spezifikation zur Lösungsarchitektur der elektronischen Gesundheitskarte (eGK) decken sich sehr gut mit den Anforderungen an die elektronischen Prüfungen. Vor allem der Einsatz der qualifizierenden Signaturen und das virtuelle ticketbasierte Dateisystem lassen sich nahezu vollständig auf die elektronischen Prüfungen transferieren.

Aus diesem Grund wird dann in Kapitel 5 das virtuelle, ticketbasierte Dateisystem (vtD) analysiert und auf die elektronischen Prüfungen angepasst. Das vtD ermöglicht u.a. die Anonymität trotz Authentizität und der Benutzer bleibt trotzdem „Herr seiner Daten“. Denn durch das Ticketkonzept bestimmt der Benutzer wer, was, auf welche Art und wie lange von den Daten zu sehen bekommt.

Mit Hilfe des vtD lassen sich die meisten technische Anforderungen umsetzen. Durch eine Erweiterung des Konzeptes in Kapitel 6 für die Protokollierung und Archivierung, sowie eines Mechanismus zur Ausfallsicherheit, können auch die restlichen Anforderungen umgesetzt werden.

In Kapitel 7 werden dann die Realisierungen der einzelnen Konzeptbausteine beschrieben und wie die Integration in ein bestehendes Prüfungssystem erfolgen kann.

Kapitel 8 fasst die Erkenntnisse der Arbeit zusammen und bewertet die Umsetzungen aus Kapitel 7. Die Arbeit schließt mit einem Ausblick auf weiterführende Arbeiten.

Kapitel 2

Elektronische Prüfungen an Hochschulen

Für Sprachtests wie den TOEFL-Test des *Educational Testing Service*[1] und Zertifizierungen in der IT-Branche werden computergestützte Prüfungen bereits länger eingesetzt. Der Softwarehersteller Microsoft zertifiziert bspw. seit 1994 mit Hilfe von computergestützten Prüfungen. Durch die Einführung von Learning Management Systemen (LMS) konnten dann die ersten Erfahrungen mit einer computerunterstützten Lehre gemacht werden [Sch04]. Durch die Einführung von Learning Management- und E-Learning Systemen wurden Teilbereiche der Lehre computergestützt realisiert, was dann dazu führte, den eingeschlagenen Weg weiter zu verfolgen und die E-Learning Umgebungen um Verfahren zur Prüfungsvorbereitung und -durchführung zu ergänzen [Ste06]. Diese Aufbruchstimmung konnte auch in der wachsenden Anzahl an Konferenzen und Veröffentlichungen beobachtet werden.

Dieses Kapitel analysiert zuerst die verschiedenen Begrifflichkeiten rund um das elektronische Prüfen. Anschließend werden die verschiedenen Prüfungsformen an Hochschulen beschrieben, die unabhängig von der verwendete Prüfungsart (papierbasiert oder elektronisch) sind.
Danach werden die speziellen Organisationsformen für elektronische Prüfungen der Hochschulen Zürich, Münster, Bremen, Duisburg-Essen sowie des externen Dienstleisters Codiplan dargestellt.
Die Funktionalität und der Leistungsumfang von elektronischen Prüfungssystemen wird dann in Abschnitt 2.4 betrachtet. Dabei wird sich in dieser Arbeit auf die zur Zeit am verbreitetsten kommerziellen und nicht-kommerziellen Systeme beschränkt.

[1] http://www.de.toefl.eu/toefl-sites/toefl-germany/ueber-toefl/internet-based-testing/, 06.04.2010

Abschließend wird dann eine Umgebung für elektronische Prüfungen an Hochschulen aus den betrachteten Organisationsmodellen heraus definiert. In dieser Umgebung werden dann auch die Akteure und Prozesse bestimmt, anhand derer sich das spätere Sicherheitskonzept orientiert.

2.1 Begriffsdeutung

Interessanterweise hat sich bis heute kein einheitliches Vokabular für Prüfungen am Computer durchgesetzt. Neben dem Begriff der „elektronischen Prüfungen“ finden sich vor allem die Begrifflichkeiten „Online-Tests“, „Online-Klausuren“ und „E-Assessment“ wieder (vgl. u.a. [RMP06], [Wan06], [Sch04]). Bloh definiert E-Assessment als

> „*...Spektrum der auf den neuen (elektronischen) Informations- und Kommunikationstechnologien basierende Verfahren der lehrzielbezogenen Bestimmung, Beurteilung, Bewertung, Dokumentation und Rückmeldung der jeweiligen Lernvoraussetzungen, des aktuellen Lernstandes oder der erreichten Lernergebnisse/ -leistungen vor, während (Assessment für das Lernen) oder nach Abschluss (Assessment des Lernens) einer spezifischen Lehr-Lernperiode.*“ [Blo08]

Nach der Definition von Bloh orientieren sich also E-Assessments an den Lernzielen.
Im deutschsprachigen Raum haben sich neben dem E-Assessment Begriff die der „elektronische Prüfungen“ und „computergestützte Prüfungen“ durchgesetzt. Vielfach werden die Begriffe als Synonyme verwendet (siehe u.a. [Lan07]).
In [RSNSS07] werden die Vielfalt der Begriffe anhand einer Literaturanalyse diskutiert, wobei wissenschaftliche Arbeiten, Präsentationen auf Konferenzen und Workshops, sowie institutseigene Veröffentlichungen und Zeitungsartikel untersucht wurden. Die Häufigkeit der Nennungen in den über 60 deutschsprachigen Arbeiten ist in Tabelle 2.1 dargestellt.

Rüdel schlägt anhand der Untersuchung den Begriff „eAssessment“ vor, der in Anlehnung an den Begriff eLearning als Oberbegriff für alle Prüfungsformen (formativ, summativ, diagnostisch) dienen soll [RSNSS07]. Als Begründung weisen die Autoren an, dass der Begriff „Prüfung“ andere Vorstellungen erwecken würde und eben nicht den gesamten Einsatzbereich abdeckt. Brennscheidt hält dem entgegen:

Begriff	Häufigkeit der Nennung
Online-Prüfungen	14
Computergestützte Prüfungen	8
eTesting	6
eKlausuren	5
Online Klausuren	5
eAssessment	4
Elektronische Prüfung	4
Prüfung	4
Aufgaben	3
Computerbasierte Prüfung	3
Computerunterstützte Prüfung	3
PC Prüfung	3
Online Testing	2
Prüfen am PC	2
Rechnerunterstütztes formatives Prüfen	2
Weitere Begriffe jeweils	1

Tabelle 2.1: Begriffe für eAssessment [RSNSS07]

> „*Eine computergestützte Prüfung (elektronische Prüfung) ist eine Bezeichnung für eine Prüfung, welche mit Hilfe eines Computers bearbeitet wird. Sie bietet teilweise oder vollständig die Möglichkeit, eine automatische Auswertung durchzuführen. Eine computergestützte Prüfung kann über ein Computernetzwerk zur Verfügung gestellt werden, wobei es keine Rolle spielt, ob dieses Computernetzwerk ein Intranet oder das Internet ist. Es ist außerdem möglich, die Prüfung ohne ein Computernetz oder nur teilweise über ein Computernetz durchzuführen.*“ [Bre08]

In dieser Arbeit wird der Begriff „elektronische Prüfung“ und als Synonym „computergestützte Prüfung“ verwendet so wie in [Bre08] definiert. International hat sich der Begriff „eAssessment“ jedoch manifestiert. Allerdings ist der Begriff *Assessment* im deutschen Sprachgebrauch als eine Form der Bewerberauswahl bekannt (Assessment-Center), weshalb in dieser Arbeit der Begriff eAssessment nicht verwendet wird.

Des Weiteren gilt es, die für diese Arbeit verendeten Begriffe Prüfungsangabe, Prüfungslösung und Prüfungsbewertung zu definieren:

- *Prüfungsangabe*: Dies ist die vollständig zusammengestellte Prüfung

des Dozenten, die von den Studierenden bearbeitet werden muss. Die Prüfungsangabe besteht aus einzelnen Aufgaben. Im Falle von Multiple- und Single-Choice Aufgaben sind den Aufgaben mehrere Antwortmöglichkeiten zugeordnet.

- *Prüfungslösung*: Die Prüfungslösung ist die Antwort des Studierenden zu der Prüfungsangabe.
- *Prüfungsbewertung*: Das ist die vom Dozenten korrigierte Fassung der Prüfungslösung des Studierenden.

Ein Studierender, der an einer Prüfung teilnimmt, wird als Prüfungsteilnehmer oder auch kurz als Teilnehmer bezeichnet.
Aus Gründen der leichteren Lesbarkeit wird in dieser Arbeit auf eine geschlechtsspezifische Differenzierung, wie z.B. Teilnehmer/Innen, verzichtet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für beide Geschlechter.

2.2 Prüfungsfomen an Hochschulen

Herkömmliche Prüfungsformen lassen sich auf elektronische Prüfungen abbilden. Unabhängig von der Durchführungsart existieren die folgenden Prüfungsformen [Ste06]:

- Lernzielkontrolle und Selbsteinschätzung
- Kenntnisstandermittlung als Lehrgrundlage und Prüfungsvorbereitung
- Förderung der intensiven Auseinandersetzung mit den Lehrinhalten
- Leistungsbewertung
- Qualitätskontrolle und Evaluation der Lehre.

Eine abstraktere Einteilung der Prüfungsformen findet sich u.a. in [Cri07]. Hier unterscheidet man zwischen diagnostischen, formativen und summativen Prüfungen (siehe Abbildung 2.1).

Diagnostische Prüfungen dienen zum einen als Vorauswahl für zulassungsbeschränkte Studiengänge, aber auch zur Identifizierung des aktuellen Wissenstandes der Studierenden, um im Vorfeld einer Lehrveranstaltung ggf. die Lehr-/ Lerninhalte darauf anzupassen [Cri07]. Das Ergebnis einer diagnostischen Prüfung kann zum einen von dem Lehrenden genutzt werden, um die

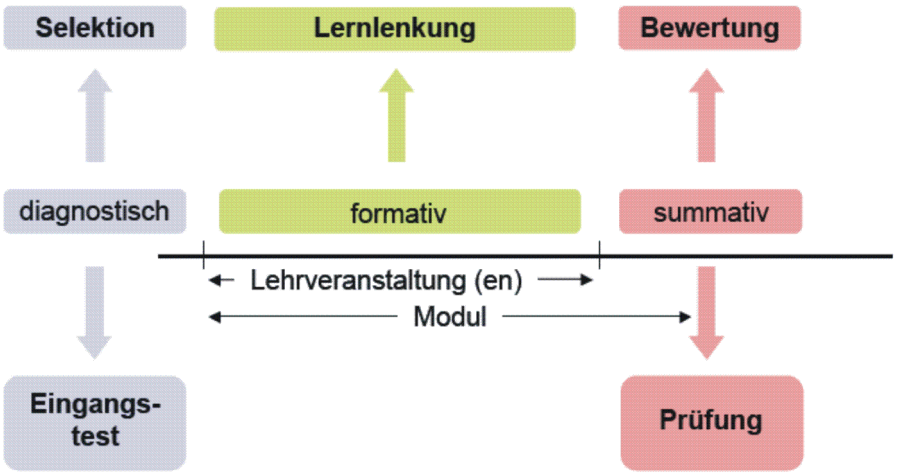

Abbildung 2.1: Einsatzbereiche von elektronischen Prüfungen (aus [Rue09] nach [Cri07])

Veranstaltung an den Wissensstand der Teilnehmer anzupassen. Zum anderen kann er den Lernenden dazu dienen, zu überprüfen, ob sie die nötigen Voraussetzungen für die Veranstaltung besitzen.
Formative Prüfungen dagegen werden während einer Lernphase eingesetzt. Sie sollen lernunterstützend wirken und können in Form von Übungen als praktische Anwendung des Vorlesungsstoffes angewendet werden. Formative Prüfungen besitzen zwar eine bewertende Komponente, die aber im Idealfall vor allem aus einem Feedback für die Studierenden besteht. Die formativen Prüfungen testen punktuelles Wissen periodisch über die gesamte Lernphase. Die Ergebnisse können den Lehrenden zeigen, ob die Lernziele erreicht wurden, wie erfolgreich die Lehr-/ Lerninhalte präsentiert wurden und wo noch eine intensivere Beschäftigung mit dem Lehr-/ Lernmaterial nötig ist. Somit können formative Prüfungen ein Instrument der Qualitätskontrolle und -sicherung der Lehre sein [PR05]. Durch das Feedback können die Studierenden ihre Lösungen reflektieren und somit ihr Lernverhalten anpassen.
Summative Prüfungen dienen der Bewertung des Leistungsstandes eines Studierenden in Form einer Note und werden nach einer Lernphase eingesetzt. Die summativen Prüfungen haben also eine direkte Konsequenz für den Lernenden. Die Anforderungen an summative Prüfungen sind im Vergleich zu den formativen und diagnostischen Prüfungen dementsprechend hoch. Dies betrifft nicht nur die organisatorischen Anforderungen sondern vor allem auch die Anforderungen an die Sicherheit und den Datenschutz.

2.3 Organisationsmodelle für elektronische Prüfungen

Der Ablauf einer klassischen Prüfung ist unterteilt in die Prozesse Konstruktion, Durchführung, Auswertung und Archivierung (siehe [RSNSS07]). Dieses eher didaktisch orientierte Modell berücksichtigt die administrativen Schritte, die bei einer elektronischen Prüfung nicht anfallen. Nachfolgend werden verschiedene Organisationsmodelle diskutiert. Die Organisationsmodelle der Universitäten Zürich, Bremen und Münster sind speziell auf die Anforderungen von elektronische Prüfungen an Hochschulen zugeschnitten und betrachten auch die administrativen Elemente.

2.3.1 Zürcher Arbeitsmodell

Das vom JISC (Joint Information Systems Committee) erstellte Arbeitsmodell für eAssessments berücksichtigt neben den administrativen Schritten auch die Motivation der Dozenten, die Administration, die Durchführung und die Evaluation [RSNSS07, Rue09]. In Abbildung 2.2 sind die einzelnen Stufen des JISC-Modells dargestellt, wie sie im Organisationsmodell der Universität Zürich angewendet werden.

Die Motivationsphase ist charakterisiert durch eine Analyse der Lehre seitens der Dozierenden und eine Beratung und Schulung derselben durch ein zentrales eLearning Team. Den Studierenden werden ebenfalls die Vorzüge der elektronischen Prüfung, auch durch die Hochschulleitung, näher gebracht. Die Dozierenden legen in der Planungsphase die Prüfungsart (diagnostisch, formativ oder summativ) und die Beurteilungskriterien fest. Außerdem erfolgen die Fragenerstellung und eine Vorrevision der Fragen sowohl sprachlich als auch inhaltlich. Dabei werden die Dozierenden durch die Informatikdienste und das zentrale eLearning-Team der Hochschule unterstützt.
In der Entwicklungsphase werden die Fragen durch die Dozierenden in das Prüfungssystem eingegeben und in einen Fragenpool eingebunden. Den Dozierenden obliegt hierbei auch eine Revision der Fragen. In der Administrationsphase melden die Dozierende ihre Prüfungen an die zuständigen Stellen in den Fakultäten (Prüfungsämter, Koordinatoren). Die Studierenden melden sich anschließend zu den Prüfungen an und die Kandidatenlisten werden erstellt. Die benötigten Räumlichkeiten werden reserviert.
Die Durchführungsphase beginnt mit einem Probelauf für die Studierenden, damit sie sich mit dem elektronischen Prüfungssystem vertraut machen können (Navigationen, Look and Feel, etc.). Die Prüfungsaufgaben werden dann von den Dozierenden im Prüfungssystem bereitgestellt und die Studierenden

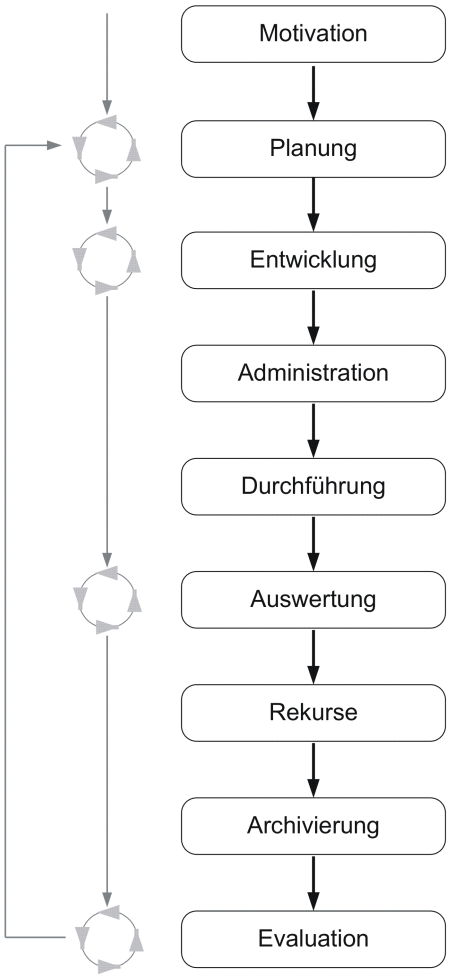

Abbildung 2.2: Entwicklungskreislauf Zürcher Arbeitsmodell [RSNSS07]

	D	F	U	Z	K	ID	H	BUR	RD	S
Motivation	X		X	X	X					X
Planung	X			X		X				
Erstellung	X									
Administration	X	X	X			X	X	X		
Durchführung	X	X				X				X
Auswertung						X				
Rekurse						X			X	X
Archivierung		X	X			X			X	
Evaluation	X			X	X	X				

Tabelle 2.2: Phasen einer elektronischen Prüfung und beteiligte Akteure [RSNSS07]

können die Prüfung durchführen. Währenddessen werden Prüfungsaufsichten eingesetzt um die Anwesenheit zu kontrollieren und um einen ordnungsgemäßen Verlauf der Prüfung zu garantieren.
Die Auswertung der Prüfung erfolgt für Multiple- und Single-Choice Aufgabentypen automatisch. Anschließend erfolgt eine manuelle Auswertung der offenen Aufgabentypen durch die Dozierenden. Nach der statistischen Auswertung und Analyse wird die endgültige Bewertung festgelegt. Die Studierenden erhalten anschließend ihr Ergebnis.
In Abbildung 2.2 folgt nach der Auswertung eine Rekursephase. Diese beinhaltet die Einsicht in die Prüfung und die Korrektur durch die Studierenden. Eine eventuelle Beschwerde der Studierenden muss durch den Rechtsdienst (Justiziar) der Hochschule bearbeitet werden. Die endgültigen Ergebnisse der Prüfung werden dann in der Archivierungsphase elektronisch aufbewahrt. Die abschließende Evaluation ermöglicht eine Nachlese der Prüfung sowie die Analyse etwaiger Verbesserungsvorschläge durch die Studierenden.
Des Weiteren wurden die folgenden sog. Stakeholder (Akteure) der Universität Zürich in das Modell mit einbezogen: Dozierende (D), Fakultäten mit Instituten (F), Universitätsleitung (U), Zentrales eLearning Team (Z), Koordinatoren in den Fakultäten (K), Informatikdienste (ID), Hörsaal-Disposition (H), Bauten und Räume (BUR), Rechtsdienst (RD) und Studierende (S). Tabelle 2.2 zeigt die Entwicklungsphasen mit den an den einzelnen Phasen beteiligten Akteure.

Das Zürcher Arbeitsmodell zeigt, dass die Prozessphasen einer Prüfung in einen hochschulweiten Kontext eingebunden werden müssen. Die Schaffung einer eLearning-Einheit (Z) als verantwortliche Stelle für die Abdeckung aller Phasen ist notwendig. In Tabelle 2.2 ist dies durch die Verbindung der zen-

AP1	Prüfungsverwaltung (Raumbuchung, Terminverwaltung und Organisation der Aufsichten)
AP2	Anmeldeorganisation
AP3	Katalogerstellung und Qualitätskontrolle
AP4	Prüfungsvorbereitung
AP5	Durchführung
AP6	Nachbereitung und Auswertung
AP7	Prüfungseinsicht

Tabelle 2.3: Arbeitspakete zur Organisation und Durchführung an der Universität Bremen [BÖ8a]

tralen eLearning-Einheit und der Informatikdienste realisiert. Jedoch bleibt anzumerken, dass das Zürcher-Modell einige Akteure nicht oder nur unzureichend berücksichtigt. Gerade im Praxiseinsatz wird üblicherweise der Großteil der Administration von den Sekretariaten der Dozierenden erledigt.
Des Weiteren ist in Tabelle 2.2 der Dozierende (D) bei der Auswertungsphase nicht berücksichtigt, sondern nur die Informatik-Dienste. Das ist aber nicht nur bei der Auswertung der offenen Fragen nötig, sondern auch bei einer möglichen Nachbewertung der Multiple-und Single-Choice Aufgaben im Falle einer fehlerhaften Lösung oder Aufgabenstellung. Außerdem wird die Erstellung bzw. Wartung des Fragenpools oftmals Mitarbeitern überlassen. Die Auswertung der Ergebnisse ist ebenso oftmals Mitarbeitern bzw. Korrektoren/ Tutoren überlassen. Lediglich die Endabnahme (Zweitkorrektur) wird durch den Dozierenden selbst erledigt.

2.3.2 Organisationsmodell der Universität Bremen

Das Organisationsmodell der Universität Bremen basiert auf einem serviceorientierten Ansatz. Der zentrale eAssessment Dienst des Zentrums für Multimedia in der Lehre (ZMML) bietet ein campusweites Angebot zur Erstellung, Organisation, Durchführung und Auswertung von computergestützten Prüfungen, Einstufungstests und Übungen [BÖ8a]. Der eAssessment Dienst umfasst dabei die in Tabelle 2.3 aufgelisteten Arbeitspakete. An jedem dieser Arbeitspakete sind verschiedene Akteure beteiligt (siehe Abbildung 2.3).

Die Prüfungsverantwortlichen (PV) sind die Dozenten der jeweiligen Veranstaltung in denen die Prüfung durchgeführt wird. Dozenten sind in der Regel auch die Prüfungsautoren. Die Aufgaben der PV sind die inhaltliche und rechtliche Verantwortung der Prüfung und die Festlegung und Weitergabe der Klausurregelungen. Eine wichtige Aufgabe der PV ist die Endabnahme

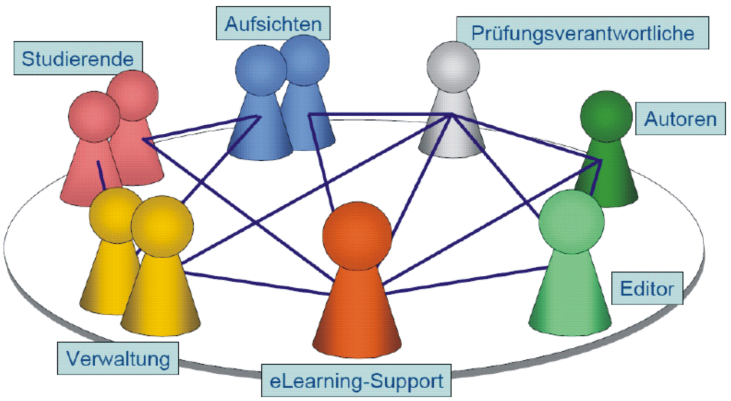

Abbildung 2.3: Akteure im Organisationsmodell Universität Bremen [ZMM09]

der elektronischen Prüfung im Testcenter zur Entlastung des eAssessment-Dienstes.
Nach der Durchführung erfolgt durch die PV die Nachbewertung und die Festlegung sowie die Weitergabe der Noten an die Verwaltung (V). Die Verwaltung erstellt das Prüfungsangebot und legt die Prüfungstermine fest. Die Verwaltung bucht das Testcenter und organisiert die Aufsichtspersonen für die Prüfungstermine und gibt sie an die Aufsichtsführenden (AF) weiter. Des Weiteren regelt die Verwaltung den gesamten Anmeldeprozess. Nach der Durchführung erhält die Verwaltung die Ergebnisse von den PV und informiert die Studierenden. Am Ende werden die Prüfungsunterlagen archiviert. Neben den PV können auch Prüfungsautoren (AU) Fragen erstellen bzw. verwalten. Außerdem dürfen sie Qualitätskontrollen der erstellten Fragenkataloge durchführen und Prüfungseinstellungen vornehmen.
Prüfungseditoren (ED) setzen die Fragenvorlagen der PV und AU in der Editorsoftware um und sind für den Transfer an den eLearning-Support (ES) verantwortlich. Nach der Prüfung sind die ED dazu berechtigt, die Fragenkataloge nach Absprache mit ES und AU zu aktualisieren, sowie die Ergebnis- und Fragenstatistiken auszuwerten. Die AF sind für die Beschaffung und Ausgabe von Klausurunterlagen (Klausurhinweise, etc.) sowie der Bekanntgabe der Klausurregelungen zuständig. Sie bieten Hilfestellungen bei Einlog-Problemen, übernehmen die Authentifizierung der Teilnehmer und kontrollieren die Studierenden (S) auf Täuschungsversuche. Der ES koordiniert die Arbeitspakete aus Tabelle 2.3. Der ES ist für die Administration des Prüfungsservers und des Testcenter-Management Tools (TCMT) verantwortlich.

Die weiteren Aufgaben des ES sind:

- Schulung und Beratung von PV und AU zu Prüfungsdidaktik, Kosten-/Nutzen-Analyse, Organisation und Nachbewertung
- Schulung und Beratung der AU und ED zur Editor-Software
- Festlegung und Controlling der Deadlines für Katalogerstellung und Qualitätskontrolle
- Beteiligung an der Qualitätskontrolle und Revision der Fragenkataloge
- Termin- und Dateiverwaltung über das TCMT
- Pflege der Daten auf dem Prüfungsserver
- Upload der Fragenkataloge
- Eingabe der Prüfungsdaten in Absprache mit V und ES
- Einrichten und Verwalten von Katalogtestaccounts
- Upload und Aktualisierung der Teilnehmerlisten in Absprache mit V
- Organisation und Durchführung von Funktionstest, Zugangstest und Prüfungsabnahme im Testcenter mit PV, AU und AF
- Bereitstellung von Checklisten und Templates
- Administrative Aufgaben während der Prüfung (Nachtragen von Teilnehmern, Lösen von Problemen beim Einloggen)
- Aufschließen, technische Vorbereitung und Abschließen des Testcenters
- Support und Rufbereitschaft während der Prüfungsdurchläufe und Einsichten
- Export, Pflege und Weiterleitung der Teilnehmerdaten, Prüfungsergebnisse und statistischer Analysen.
- Bereitstellung von Informationen bei Widersprüchen

Tabelle 2.4 zeigt, dass wie im Zürcher Arbeitsmodell ein zentraler eLearning bzw. eAssessment Dienst, in fast allen Prüfungsphasen aktiv ist. Allerdings ist das Organisationsmodell der Universität Bremen aktuell nur auf die summativen Prüfungen in einem Testcenter ausgelegt. Die Erweiterung des eAssessment Dienstes auch auf die formativen Prüfungen sind die nächsten Schritte [BÖ9].

	PV	V	AU	ED	AF	ES	S
AP1		X				X	
AP2		X					X
AP3	X		X	X		X	
AP4	X	X	X		X	X	
AP5					X	X	X
AP6	X	X	X			X	X
AP7	X					X	X

Tabelle 2.4: Akteure und Phasen im Organisationsmodell der Universität Bremen (vgl. [ZMM09])

Testcenter der Universität Bremen

Für die Errichtung des Testcenters wurde ein ehemaliges Gebäude der Bibliothek der Universität Bremen umgebaut. Finanziert wurde der Umbau u.a. über das Computer-Investitions-Programm (CIP) nach dem Hochschulbauförderungsgesetz (HBFG). Seit Dezember 2007 ist das Testcenter in Betrieb. In Abb. 6 ist der Grundriss des Testcenters dargestellt. Das Gebäude besteht aus einem Stockwerk. Der Zugang zum Prüfungssaal ist nur über den Vorraum möglich. Der Prüfungsaal misst 17 mal 17 Meter und besteht aus 4 Doppelreihen mit insgesamt 120 Arbeitsplätzen. Die Arbeitsplätze sind mit einem 19Zoll TFT-Bildschirm, Maus und leiser Tastatur, Headset und Webcam ausgestattet. Als Betriebssystem wird Windows XP eingesetzt. Des Weiteren ist der Raum mit drei Beamern, einer Sound-Anlage und einer Multimediasteuerung ausgestattet. Außerdem stehen zwei Drucker und zwei Scanner zur Verfügung [VS09]. Ferner existieren drei Technikräume. In einem der Technikräume befinden sich zehn weitere PCs für den Austausch im Falle eines Defektes. Über diesen Raum gelangt man in den Serverraum, in dem acht Server betrieben werden:

- 1 Webserver mit dem Prüfungssystem
- 1 Webserver mit dem Prüfungssystem als Replacement Server zur Ausfallsicherheit und Testumgebung
- 1 Management Server für die Client PCs
- 2 Datenbankserver
- 1 Backupserver
- 1 Server für Übungsklausuren

Abbildung 2.4: Grundriss des Testcenters der Universität Bremen [Sch08]

- 1 Server für die Firewall

Alle Server sind mit einer unterbrechungsfreien Stromversorgung (USV) ausgestattet. Die Kosten des Umbaus betrugen ca. 300.000 EUR, die von der Universität Bremen getragen wurden. Der Umbau umfasste die Klimatisierung, einen neuen Boden, die Verkabelung (Strom und Netz), Schließ- und Alarmanlagen sowie das Mobiliar. Die IT-Ausstattung wurde über den CIP Antrag finanziert. Die Kosten belaufen sich auf ca. 315.000 EUR und beinhalten die Kosten für die Clients, Server, das Netzwerk, Medien und die Software (Betriebssysteme, Office-Produkte, etc.). Hierbei ist aber anzumerken, dass die im Testcenter verwendete kommerzielle Prüfungssoftware LPLUS (siehe Unterabschnitt 2.4.1). der Universität Bremen kostenfrei im Rahmen eines Kooperationsvertrages zur Verfügung gestellt wird. Die PCs im Testcenter werden von dem Managementserver aus gemeinsam gestartet. Als Benutzer ist auf den Rechnern ein Klausur-User eingerichtet, der keinerlei Zugriffsrechte auf das Dateisystem oder Anwendungen hat. Die Durchführung der Prüfungen erfolgt per Autostart im Kiosk-Modus.

2.3.3 E-Competence Agentur der Universität Duisburg-Essen

Das Rektorat der Universität Duisburg-Essen (UDE) hat im August 2008 eine E-Strategy beschlossen mit dem Ziel, sämtliche an der UDE vorhandenen und

Abbildung 2.5: Multifunktionaler Arbeitsplatz PC Hall UDE [ZIM09]

sinnvollen digital umsetzbaren Dienste über das Internet zu realisieren.

> „*Unsere Universität strebt an, sich als E-University zu profilieren. Wir möchten unsere gute Position auf dem Gebiet der digitalen Services für Forschung, Lehre und Management ausbauen und sehen dies als eine Chance, uns damit im Wettbewerb mit anderen Hochschulen zu profilieren.*“ [Str08]

Das Testcenter, das als „PC Hall“ an der UDE bezeichnet wird, stellt mit 198 Plätzen (196 Arbeitsplätze+2 Aufsichtsplätze) den zur Zeit größten PC-Prüfungsraum in Deutschland dar. Der dafür notwendige Raum wurde für 500.000 EUR saniert und ausgebaut. Insgesamt belaufen sich die Kosten für die PC Hall auf über 1 Million EUR [ZIM09]. Aufgrund der hohen Investition sieht das Raumnutzungskonzept vor, den Raum sowohl für computergestützte Prüfungen als auch für papierbasierte Prüfungen einzusetzen. Dazu wurden spezielle Computertische angeschafft wie in Abbildung 2.5 dargestellt. Außerhalb der Prüfungszeiten dient die PC Hall als Computerarbeitsraum. Zur Zeit wird der Raum durch den Medien- und Kundenservices des Zentrums für Informations- und Mediendienste (ZIM) administriert. Das ZIM unterstützt außerdem die Lehrenden und Aufsichten durch eine E-Competence Agentur. Die Studierenden werden durch Demo-Prüfungen an das Prüfungssystem LPLUS (siehe Unterabschnitt 2.4.1) gewöhnt. Der Standort der PC Hall ist der Campus Essen, an dem auch ein Kompetenzzen-

trum für PC-gestützte Prüfungen geschaffen wird. Das Kompetenzzentrum soll die Möglichkeit schaffen Online-Klausuren in einer gesicherten Umgebung durchzuführen.

2.3.4 Organisationsmodell für Lernfortschrittskontrolle an der Universität Münster

Die Universität Münster definiert universitäre Lernforschrittskontrollen als einen Überbegriff für Klausuren, Übungen und Tutorien, Selbststudium, Fachar beiten und mündlichen Prüfungen [EGK08]. Eine Untersuchung von existierenden Systemen hatte ergeben, dass den Systemen die expliziten Anforderungen des Hochschulalltages fehlen und eine einheitliche Lernfortschrittskontrolle nicht möglich ist [EGK08]. Deshalb wurde eine Eigenentwicklung mit Namen EASy (E-Assessment-System) angestrebt. Dabei wurden die Akteure Administratoren, Verwaltungsangestellte, Lehrbeauftragte, Korrektoren, Tutoren, Studierende sowie Hochschulinformationssysteme und E-Learning Plattformen identifiziert. In Abbildung 2.6 sind die Use-Cases in vereinfachter Form dargestellt.

In Abbildung 2.7 ist die technische Architektur dargestellt. Es handelt sich um eine 4-Schichten-Architektur um die Datenhaltung, Logik, Sicherheitsaspekte und die Präsentation flexibel gestalten zu können. In der Datenhaltungsschicht werden die Daten redundant gespeichert. Dies wird durch einen RAIDb-Datenbank-Cluster realisiert. Für die Speicherung von Aufgaben mit Multimediaelementen ist ein Medienserver vorgesehen. Die Logikschicht bildet den Kern der Architektur und bildet alle Prozesse ab, die im Zusammenhang mit Lernfortschrittskontrollen stehen. Hierbei wird ein J2EE-Applicationserver mit EJB 3 eingesetzt (JBOSS). Die Kommunikationszwischenschicht ist aufgrund der Integrationsanforderungen an ein universitär einsetzbares System, sowie der speziellen Anforderungen an ein Klausursystem notwendig. Die Zwischenschicht ist für die Kommunikation zwischen der Logikschicht und der Präsentations- und Interaktionsschicht zuständig und verfolgt einen dienstorientierten Ansatz mithilfe von Webservices und Portlets. Das Klausursystem ist dabei nur innerhalb des Universitätsnetzwerkes erreichbar. Die Präsentations- und Interaktionsschicht stellt die verschiedenen Zugänge zu den verschiedenen Anwendungen zur Lernfortschrittskontrolle dar.

In Abbildung 2.6 wurden die Use-Cases auf den allgemeinen Anwendungsaspekt der Lernfortschrittskontrolle hin dargestellt. Für den sensiblen Bereich

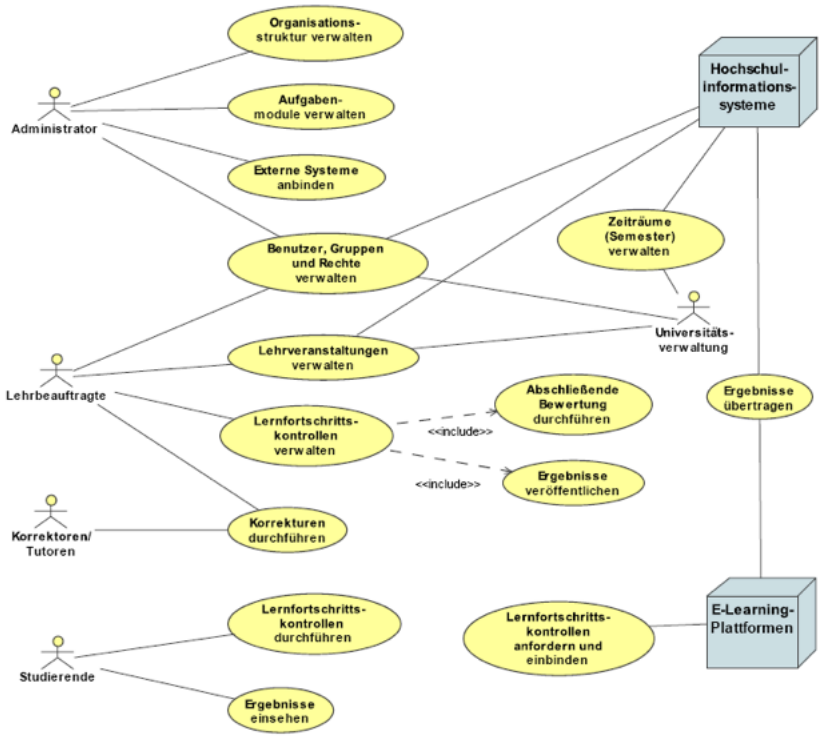

Abbildung 2.6: Anwendungsfälle [EGK08]

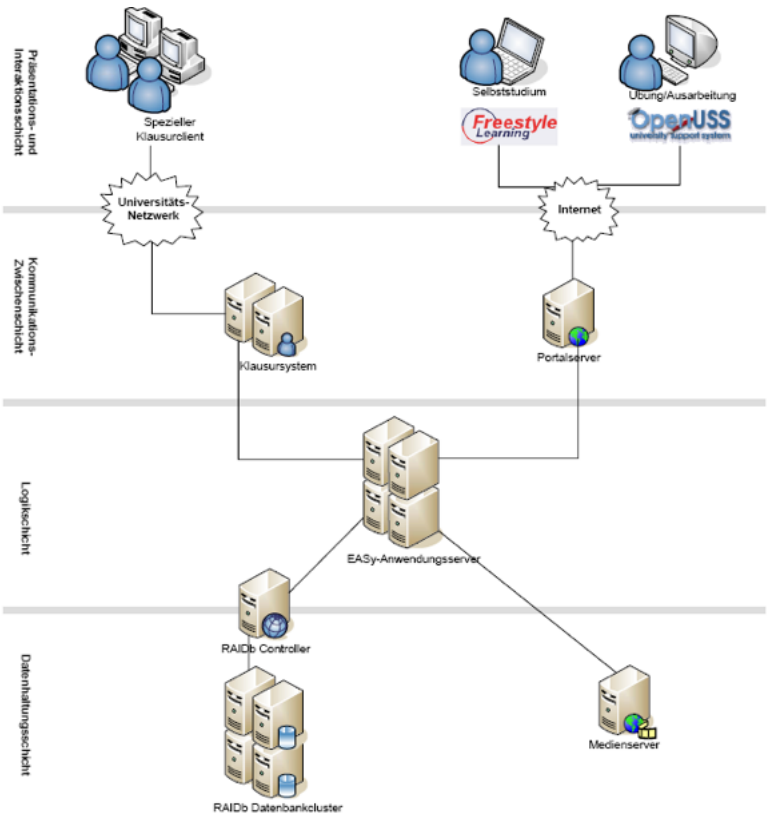

Abbildung 2.7: Vier-Schichten-Architektur EASy [EGK08]

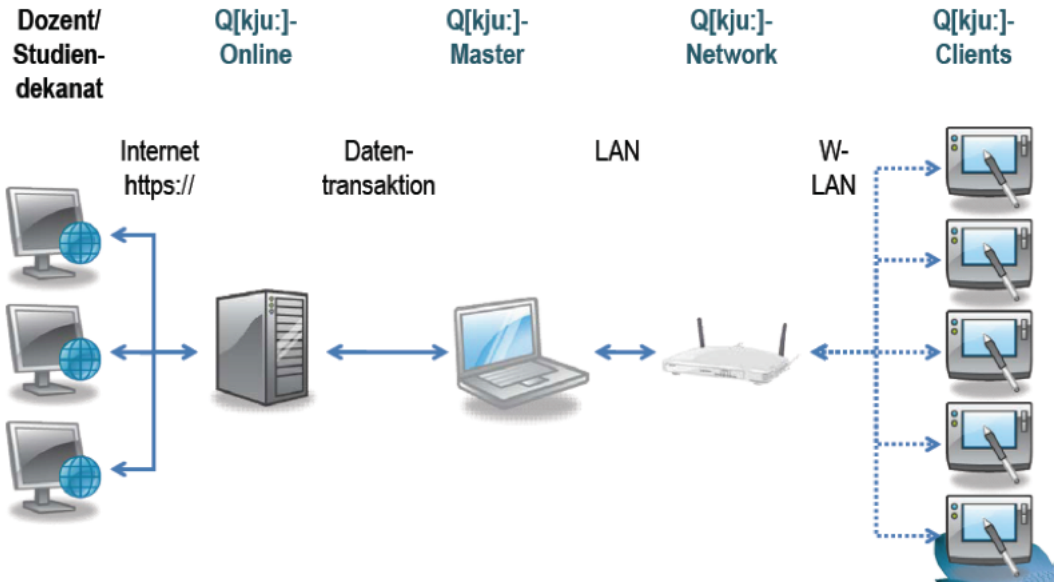

Abbildung 2.8: Systemarchitektur Codiplan Q[kju:] [MÖ9]

der elektronischen Klausuren finden sich in [Ree08a] die entsprechenden Anwendungsfälle.

2.3.5 Codiplan Q[kju:]

Die Firma Codiplan bietet mit ihrem System Q[kju:] ein System zur Durchführung von elektronischen Prüfungen. Dabei stellt Codiplan die gesamte Hard- und Software zur Verfügung und nutzt die gewöhnlichen Hörsäle der Hochschule zur Durchführung. Somit werden von Codiplan mobile Server, W-LAN Router, Tablet PCs und die Prüfungssoftware aus einer Hand geliefert [MÖ9]. Neben den mobilen Servern wird eine zentrale Installation an der Hochschule angeboten, auf der die Dozenten ihre Klausuren erstellen können. Die Klausuren werden auf die mobilen Server im Vorfeld übertragen und die Klausur wird durchgeführt.

Das Besondere bei der Durchführung ist die Verwendung von TabletPCs auf Seiten der Studierenden. Die TabletPCs kommunizieren dabei über W-LAN mit den mobilen Servern. Bis zu vier verschiedene Klausuren sind gleichzeitig durchführbar. Nach der Durchführung dieser Prüfung werden alle Clients eingesammelt und die Server werden heruntergefahren. Codiplan übernimmt alle Aufgaben der Organisation. Einzig die Aufsichtspersonen müssen von

der Hochschule gestellt werden. Nachteilig wirken sich die eingeschränkten Aufgabentypen und die Problematik, dass die Klausuraufgaben und die Studierendenlösungen bei einem externen Dienstleister gehostet werden aus (siehe [VS09]).
Die Kosten berechnen sich in Abhängigkeit der Anzahl der durchgeführten Klausuren. Je mehr Teilnehmer pro Klausur, desto höher die Kosten.

2.4 Systemlösungen für elektronische Prüfungen

Am Markt befinden sich zum Zeitpunkt der Erstellung der Arbeit zahlreiche Anbieter für elektronische Prüfungssysteme. Neben den kommerziellen Anbietern sind auch die Open-Source Lösungen interessante Kandidaten für den Hochschuleinsatz. Dabei handelt es sich vor allem um in Lern-Management Systemen (LMS) integrierte Prüfungsmodule. Die kommerziellen Anbieter hingegen setzen dabei oft auf reine Prüfungssysteme mit Schnittstellen zu LMS und Hochschulinformationssystemen.

2.4.1 LPLUS

Das kommerzielle Produkt LPLUS wird u.a. an den Universitäten Münster und Bremen für computergestützte Prüfungen eingesetzt. Für ihre Planung und Durchführung von Prüfungen werden zwei Modelle unterschieden. Das LPLUS Test Management System (LTMS) ermöglicht Prüfungen in einem lokalen Netz mittels einer Client-Server-Lösung. Demgegenüber steht die webbasierte ASP-Variante LPLUS Test Studio (LTS) via Internet. Der ASP (Application Service Provider) betreut hierbei die gesamte Hard- und Software und übernimmt die gesamte Administration der verwendeten Programme. Somit entstehen für den Kunden keine Anschaffungskosten und Aktualisierungen, Wartungen entfallen ebenfalls.
Neben einem umfangreichen Administrationsmanagement umfasst das System eine Reihe weiterer Funktionen, wie z.B. Kommunikationsplattformen, die eine interne Verständigung zwischen den Benutzern ermöglichen soll. Fraglich hierbei ist allerdings, inwieweit solche Kommunikationssysteme während eine Prüfung durch die Teilnehmer missbraucht werden können. Zusätzlich zu elektronischen Prüfungen (LTS-Examination) können auch andere Einsatzgebiete wie LTS-Trainings (autodidaktische Prüfungsvorbereitung, Prüfungssimulation) und LTS-Assessments (wie z.B. Bewerbervorauswahlen, Lernstandsermittlung) durch LPLUS realisiert werden. Die Prüfungen können dabei am Computer oder auch in konventioneller schriftlicher Form abge-

legt werden. Aufgabendesign und -management der Prüfungsfrage ist durch den integrierten TM-Editor möglich und erleichtert die Verwaltung der Aufgabenpools. Die Evaluation erfolgt größtenteils automatisch. Bei manchen Fragetypen ist aber eine manuelle Nachbewertung nötig. Außerdem bietet LPLUS eine Import- bzw. Exportfunktion, womit Teilnehmerlisten oder Fragenkataloge ins System importiert bzw. in zahlreiche Formate exportiert werden können. Laut LPLUS stehen ebenfalls verschiedene Schnittstellen zur Integration des Testsystems in eine bestehende Infrastruktur zur Verfügung [VS09].
Die Universität Münster verwendet die Client-Server-Lösung (LTMS). Zur Gestaltung, Erstellung und Verwaltung von Aufgaben wird das Autorensystem TM-Editor verwendet. Komplexe Aufgabentypen wie Multimedia-Objekte, Animationen oder Office-Dokumente, lassen sich zur interaktiven Bearbeitung leicht einbinden und mit Hilfe von Meta-Daten verwalten. Aber auch Lückentexte oder Zuordnungsaufgaben können mit diesem System entwickelt werden. Die Fragen werden entweder in einer festen Reihenfolge oder zufällig aus mehreren Fragenpools einer Klausur zugeordnet. Bei einer zufälligen Zusammenstellung kann zumindest mit Hilfe eines Rasters die Mindestanzahl der Aufgaben pro Themengebiet festgelegt werden. Auch die Festlegung der Reihenfolge der Antwort-Optionen bei Multiple-Choice-Aufgaben kann dynamisch geschehen. Dieses erschwert das Abschreiben für den Sitznachbarn [VS09].
In der Aufgabenerstellungsphase hat der Dozent die Möglichkeit, Benutzerrechte für die einzelnen Fragenkataloge zu vergeben. So können die erstellten Fragen durch andere Personen eingesehen und verwendet werden. Anhand einer Versionskontrolle kann genau nachvollzogen werden, zu welchem Zeitpunkt ein Verfasser Änderungen vorgenommen hat.
Eine automatische Kontrolle prüft, ob alle Fragestellungen vollständig sind (ob bspw. Musterlösungen oder alle erforderlichen Grafiken vorhanden sind). Ein einfaches Backup-Management schützt vor Datenverlusten und ermöglicht eine Reproduktion der Daten im Falle eines Verlustes.
Vor der Prüfung müssen sich die Studenten wie gewöhnlich über das Prüfungsamt für die Klausur anmelden. Die Anmeldung kann über Prüfungsverwaltungssysteme wie z.B. FlexNow erfolgen, wobei diese Systeme in der Regel über eine Export-Schnittstelle (Excel-Tabellen) verfügen um die Teilnehmerdaten in das Prüfungssystem zu übertragen. Aufgrund der begrenzten Arbeitsplätze werden die Teilnehmer in Gruppen aufgeteilt. Die Teilnehmerlisten können dann gemeinsam mit den erstellten Fragenkatalogen in LPLUS importiert werden [Ree06].
Nach einer persönlichen Kontrolle durch die Aufsichtsperson können sich die Teilnehmer mit ihrer Matrikelnummer und einer von dem System generierten

PIN-Nummer einloggen. Der Arbeitsplatzrechner wird während der Prüfung durch einen Secure-Browser in einen sog. Prüfungsmodus (auch Kiosk-Modus genannt) versetzt. Das bedeutet, dass nur der LPLUS-Client ausgeführt werden kann, der Zugriff auf unerwünschte Programme ist nicht möglich. Während der Prüfung werden keine Daten auf dem lokalen Rechner gespeichert. Somit kann bei einer technischen Störung die unterbrochene Sitzung ohne Datenverlust an einem beliebigen Rechner fortgesetzt werden. Um Serverausfällen entgegenzuwirken, wurden in der Architektur redundante Systeme eingerichtet, die eine sofortige Umschaltung auf einen Replacement-Server ermöglichen [Ree06].
Nach Beendigung der Prüfung werden die Prüfungsangaben ausgedruckt und durch die Studierenden unterschrieben. Im Anschluss an die Prüfung werden die Antworten durch das System automatisch ausgewertet. Manche Fragetypen verlangen allerdings eine manuelle Auswertung, die in diesem Fall von einem Prüfungsberechtigten durchgeführt wird.
Die Notenlisten mit den erreichten Punkten werden als Excel-Tabelle exportiert und können an das Prüfungsamt weitergeleitet werden. Die Antworten der einzelnen Teilnehmer werden als PDF-Dokument gespeichert und können mittels eines webbasierten Moduls (zu einer bestimmten Zeit von festgelegten Rechnern) eingesehen werden. Für den Zugriff auf die entsprechende PDF-Datei benötigt der Teilnehmer seine Matrikelnummer und die vor der Prüfung ausgegebene PIN. Zur Archivierung werden die PDF-Dateien auf einer CD oder DVD gesichert [Ree06].

2.4.2 Questionmark Perception

Perception ist ein umfassendes Assessment Management-System der Firma Questionmark für die Erstellung, Planung, Bereitstellung und Auswertung von Umfragen, Quizzes, Tests und Prüfungen. Der Vertrieb von Perception für Deutschland, Österreich und die Schweiz erfolgt durch die Firma Telerat in Berlin. Questionmark Perception ist ein Managementsystem für das Verfassen, Verteilen und Auswerten von Umfragen, Tests und Prüfungen. Fragen und Prüfungen können einfach erstellt, modifiziert und gelöscht werden [VS09].
Für das Erstellen von Fragen stehen zwei Alternativen zur Verfügung. Die erste Möglichkeit besteht darin, die Prüfungen oder Ähnliches über eine windowsbasierte Software zu erstellen. Die Alternative besteht in der Erstellung der Prüfung über einen Browser mit einem serverseitigen System.
Das browser-basierte Autorensystem benötigt keine Installation und ermöglicht eine einfache Bereitstellung von Prüfungen im großen Rahmen. Das windowsbasierte Autorensystem erstellt sowohl Fragen als auch Prüfungen und

publiziert diese. Außerdem können unterstützende Inhalte wie Bild und Ton verwaltet werden. Es bietet die Möglichkeit, Fragen bzw. Aufgaben in Prüfungen, Befragungen, Umfragen, etc. zusammenzufassen und im Anschluss online oder offline zu verteilen [VS09].
Autoren haben die Möglichkeit, verschiedene Fragetypen mit Hilfe von Assistenten oder einem Frageneditor zu erstellen. Assistenten helfen außerdem bei der Fragenzusammenstellung zu einer Prüfung. Es ist möglich, ein hilfreiches Feedback in die Fragen einzubauen, um die Merkfähigkeit zu erhöhen. Die Fragen und Aufgaben können inklusive ihrer Ressourcen in Datenbanken (Repositories) gespeichert und verwaltet werden. Die Datenbanken können lokal und verteilt verwaltet werden.
Multimediale Elemente können ebenfalls in Fragen und Aufgaben eingebettet werden. Zu den Hauptfunktionen bezüglich der Sicherheit von Prüfungen stehen rollenbasierte Sicherheitselemente für gemeinsame Multi-Autoren-Umgebungen und die Verwaltung des Workflows (Ablaufs) zur Verfügung. Autoren und Administratoren werden bestimmte Zugriffsrechte zugeordnet, so dass mehrere Autoren zusammen in einer kontrollierten Umgebung arbeiten können.
Die Administratoren können verschiedene Rollen für diverse Benutzerprofile festzulegen. Der Zugriff von Autoren auf bestimmte Themenbereiche oder Prüfungsordner kann eingeschränkt werden. Abhängig vom Subworkflow können Rechte von Autoren variiert werden. Administratoren können verschiedene Ebenen für Multi-Autoren bestimmen, die unterschiedliche Zugriffsrechte aufweisen.
Um den Workflow verwalten zu können, steht ein Workflow-Editor zur Verfügung, der dabei hilft, die einzelnen Stadien (Erstellung bis Freigabe der Klausur) zu definieren, zu kontrollieren und zu dokumentieren. Die Fragen können den Status *Normal*, *Experimentell*, *Beta*, *Unvollständig* oder *Zurückgezogen* erhalten. So kann gewährleistet werden, dass nur zulässige Fragen in der Prüfung verwendet werden.
Das Workflow-Management trägt zur Strukturierung der Multi-Autoren Umgebung bei und macht diese bequemer. Die wichtigste Eigenschaft des Workflow-Managements ist die Überwachbarkeit der Prüfung. Mit ihr geht auch eine Protokollierung oder Aufzeichnung der gesamten Prüfung einher, welches für die Rechtssicherheit der Prüfung einen wesentlichen Faktor darstellt.
Nach Fertigstellung der Klausur können auf dem Perception Server Teilnehmer, Gruppen und Termine verwaltet werden. Das *Questionmark Web Integrated Services environment* (QMWISe) bietet die Möglichkeit, Termine und Teilnehmer mit anderen Management Systemen zu synchronisieren. QMWISe unterstützt mehr als 30 verschiedene Webservices um die Daten aus Prüfungsverwaltungssystemen oder Lern-Management Systemen zu importieren.

Zur Durchführung sicherer Prüfungen kann der Secure-Player eingesetzt werden. Dieser angepasste Browser bietet ähnliche Funktionalitäten wie ein Secure-Browser. Die Prüfungen auf dem Server können so konfiguriert werden, dass sie nur über den Secure-Player aufgerufen werden können. Die Ergebnisse der Prüfungen werden dann von dem Werkzeug *Enterprise Reporter* verwaltet.

2.4.3 ILIAS Testmodul

ILIAS ist eine Open-Source Lernplattform, die mittlerweile weltweit an Hochschulen eingesetzt wird. Als flexibles LMS findet sich ILIAS aber auch in verschiedenen Bereichen der beruflichen Aus- und Weiterbildung wieder. Für den Betrieb von ILIAS werden ein Webserver, eine MySQL-Datenbank und PHP benötigt.
Neben dem Learning-Managament bietet ILIAS auch ein Modul zur Erstellung und Durchführung von elektronischen Prüfungen. Das Test und Assessment Modul kann aber auch für Selbsteinschätzungstests, Übungsklausuren oder anonymisierte Umfragen verwendet werden und bietet eine Vielzahl verschiedener Fragetypen an. Die Fragen werden in so genannten Fragenpools verwaltet und können in einer Prüfung statisch oder zufällig angeordnet werden.
Für die Durchführung wird die Prüfungen mit einem Passwort versehen, das unmittelbar vor Beginn der Durchführung den Teilnehmern bekannt gegeben wird. Damit sichergestellt wird, dass nicht von einem unautorisierten Rechner aus auf eine Prüfung zugegriffen wird, lassen sich zu jedem Teilnehmer die IP-Adresse zuordnen[2]. Über einen gesicherten Prüfungsrechner erhalten die Teilnehmer Zugriff auf ILIAS. Über den persönlichen Arbeitsbereich kann ein Teilnehmer dann nach dem Einloggen die Prüfung öffnen. Die Prüfungsaufsicht bestimmt den Startzeitpunkt der Prüfung und kann während der Prüfung den Status eines jeden Teilnehmers einsehen. Eine Aufgabenübersichtsseite gibt Auskunft über den Bearbeitungsstatus der einzelnen Aufgaben. Bis zum Ablauf der Bearbeitungszeit hat der Teilnehmer die Möglichkeit, frei zwischen den Aufgaben zu navigieren und die Prüfungsantworten zu korrigieren.
Bei Abgabe der Prüfung wird automatisch eine Übersicht der abgegebenen Antworten erstellt. Diese Antworten können dann ausgedruckt und durch den Teilnehmer unterschrieben werden. Dabei werden alle notwendigen Identifikationsmerkmale wie Name, IP-Adresse und Matrikelnummer mit ausgedruckt.

[2]http://wiki.uni-giessen.de/eklausur/index.php/ILIAS_Testmodul, aufgerufen am 22.04.2010

Für die eindeutige Zuordnung von Prüfung und Teilnehmer hat die Universität Mainz einen interessanten Ansatz entwickelt [Wet08a]: Hierbei erhält jeder Teilnehmer eine eindeutige Klausurnummer. Die generierte Übersicht am Ende der Prüfung enthält neben den o.g. Merkmalen auch diese Klausurnummer. Jeder Student erhält vor der Klausurdurchführung ein Beiblatt, auf dem allgemeine Bedienungshinweise aufgeführt sind. Nach Abgabe der Klausur muss der Teilnehmer das Beiblatt ausfüllen. Dort vermerkt er neben seinen persönlichen Angaben seine persönliche Klausurnummer und bestätigt die Angaben mit seiner Unterschrift.

2.4.4 OLAT 6

OLAT ist die Abkürzung für *Online Learning And Training* und ordnet sich der Kategorie der Lernmanagement-Systeme (LMS) zu [GRSF09]. OLAT wird seit 1999 an der Universität Zürich entwickelt und wird unter der Apache 2.0 Open-Source-Lizenz kostenfrei bereitgestellt, sowohl zur reinen Nutzung als auch zur Weiterentwicklung.
OLAT wird im heterogenen Netzwerk der Universität Zürich eingesetzt und ist hochgradig modularisiert, so dass Anpassungen leicht möglich sind und die Gesamtarchitektur erweiterbar ist. Neben einem Kurssystem bietet OLAT kursunabhängige und auch kursübergreifende Funktionen, zum Beispiel die Verwaltung von Lernressourcen und Editoren für Tests und Fragebögen. Die wichtigsten Funktionen sind [GRSF09]:

- Unterstützung von kollaborativen Arbeiten in Gruppen
- Groupwaretools wie Diskussionsforen, Chat, Kalender, Wiki, E-Mail-Formulare und Dokumentenablagen
- Didaktische Freiheit im Kurssystem durch flexible Verwendung von folgenden Kursbausteinen: Struktur, einzelne Seite, externe Seite, Wiki, IMS-CP- und SCORM-Lerninhalt, Test,
- Selbsttest, Fragebogen, Bewertung, Aufgabe, Dateidiskussion, Einschreibung, Kontaktformular, Forum und Ordner
- Benachrichtigungsservice via E-Mail oder RSS
- Einfach zu bedienende und personalisierte Oberfläche
- Mehrsprachige Benutzerführung (DE, FR, IT, EN und viele weitere mehr)
- Anbindung an externe Informationssysteme

OLAT ist in der Programmiersprache Java entwickelt, so dass es unter verschiedenen Systemen wie Windows, Linux und MacOS ohne Anpassungen funktionsfähig ist. In der Datenhaltung können verschiedene Datenbankmanagementsysteme wie MySQL, Postgres oder Oracle zum Einsatz kommen. Im Hinblick auf Wartbarkeit und Erweiterbarkeit wird bei der Entwicklung auf eine Trennung und hohe Wiederverwendbarkeit der Softwarekomponenten geachtet. Um bei Webapplikationen Interaktivität und Intuition, die vergleichbar mit Desktopapplikationen sind, zu erreichen, wird mittels des AJAX-Konzeptes das traditionelle seitenbasierte Konzept aufgelöst und das gezielte (Nach-)Laden von Elementen innerhalb der Seite eingeführt. OLAT-Benutzern stehen beide Varianten zur Verfügung wobei in dieser Arbeit nur die für elektronische Prüfungen relevanten Punkte vorgestellt werden.

Benutzerverwaltung und Systemadministration

Die Benutzerverwaltung legt entweder Personen einzeln an oder importiert diese zum Beispiel aus einer externen Tabelle. Des Weiteren ist es möglich, dass sich Personen über einen zentralen Login anmelden und so automatisch ein Benutzeraccount angelegt wird. Jedem Benutzer sind ein Profil und Verbindungen zu Systemrollen zugewiesen [GRSF09]. In mehreren Hierarchiestufen gibt es die Rollen für Gäste, die als anonyme Benutzer nur eingeschränkte Rechte haben, Benutzer, die u.a. ihre Benutzeroberfläche anpassen und als Teilnehmer einen Kurs starten können und Autoren, die Lernressourcen bearbeiten können. Weiterhin gibt es in der Verwaltungsebene Gruppenverwalter, Benutzerverwalter und Administratoren [GRSF09]. Möglich ist sowohl das Erstellen kursinterner Gruppen, als auch kursübergreifender Gruppen. So können sich Personen eines Jahrgangs in einer kursübergreifenden Gruppe zusammenfinden. Bezüglich der Systemadministration liefert OLAT Programme zur Anzeige der Systeminformation, also Listen aktiver Benutzer, Fehlermeldungen und Systemmeldungen. Mit der Quota- Verwaltung können Speicherlimite gesetzt werden, so dass nicht jeder Benutzer beliebig viele Dateien hochladen kann. Ferner werden Tools zur Übersetzung der Oberfläche in andere Sprachen, zur Benachrichtigung über Ereignisse, zur Volltextsuche und Überwachung/ Monitoring des System angeboten [GRSF09].

Lernressourcen

In der Lernressourcen-Ablage sind die von den Autoren erstellten Kurse, Tests und Fragebögen zusammengefasst und je nach Zugriffsrechten nur ihm selbst, allen Autoren, allen Benutzern oder auch Gästen zugänglich. Eine Suche nach den in den Metadaten hinterlegten Informationen, wie Autor,

Name und Beschreibung, ist vorhanden. In einem Katalog werden dann die publizierten und freigegebenen Lernressourcen angeboten. Solch ein Katalog kann das Lernangebot des Campus widerspiegeln und ein herkömmliches Vorlesungsverzeichnis ersetzen.

Testeditor

Der in OLAT eingebaute Testeditor erlaubt das Erstellen von Single- und Multiple-Choice-Aufgaben, sowie von Lückentext- und K-Prim-Fragen. Die Fragen werden im standardisierten IMS-QTI-Format gespeichert, so dass ein Im- und Export der Fragen in/aus anderen Systemen unterstützt wird. Die Wiederholbarkeit der Tests, Zeitlimite und Punkte können eingetragen werden. Neben Textbausteinen als Frage und Antwort können diese auch aus multimedialen Daten wie Bildern und Filmen bestehen.

Kurssystem

Wie die Systemrollen-Architektur ist auch das Kurskonzept hierarchisch aufgebaut, so dass einzelnen Kursbausteinen bestimmte Sichtbarkeits- und Zugangsbeschränkungen zugewiesen sind, die dem gewünschten didaktischen Konzept entsprechen. Lernressourcen können daher in Abhängigkeit einer bestimmten Punktzahl von vorhergehenden Tests, einem Datum, Gruppenzugehörigkeit und auch Rollen verknüpft werden und damit einen Lernfluss definieren. Änderungen an Kursen sind jederzeit an einer Kopie möglich und werden selektiv freigeschaltet und somit für den Benutzer freigegeben [GRSF09]. Lerninhalte werden in der Regel nicht mit OLAT erstellt, sondern von externen Programmen eingefügt. OLAT liefert einen WYSIWYG-Editor für HTML-Seiten, der auch in Content-Management-Systemen Verwendung findet. Traditionelle Lerninhalte aus Text und Grafiken werden dem Benutzer in der Regel als PDF-Datei zur Verfügung gestellt. Ein Kurs wird über Struktur-Elemente (Sektion, Kapitel, Lernschritt und Übung) gestaltet. Inhalte bestehen aus Einzelseiten, Ordnern und Lerninhalten. Eine Bewertung findet manuell durch Betreuer statt oder auf Basis automatischer Tests, Fragebögen und Aufgaben mit einem Rückgabeordner zur Abgabe einer Arbeit, Musterlösung und einer Bewertung. Testresultate, Selbsttestresultate, Fragebogenresultate Kursresultate, Logdateien und Abgabeordner können archiviert werden. Über ein Bewertungswerkzeug können die erreichten Punkte von Kursteilnehmern eingesehen und editiert werden [GRSF09]. Änderungen werden dabei auch transparent in Logdateien festgehalten.

2.5 Ableitung der Organisationsmodelle

Der Einsatz von neuen Medien bei der Durchführung von elektronischen Prüfungen bedarf einer Vielzahl an Akteuren. Bei den papierbasierten Prüfungen z.B. oblag die Geheimhaltung der Prüfungsfragen größtenteils dem Dozenten selbst. Bei der elektronischen Variante liegen die Prüfungsfragen zentral auf Servern des Rechenzentrums.
Waren bei den Papierklausuren nur die Durchführungs- und Auswertungsphase in Bezug auf die Sicherheit von entscheidender Bedeutung, so sind es bei den elektronischen Klausuren nahezu alle Phasen des Prüfungsprozesses.
Aus den in Abschnitt 2.3 dargestellten Organisationsmodellen lassen sich die Prozesse und Akteure wie folgt zusammenfassen:
Die Prozesse werden unterteilt in vorbereitende (Planung, Entwicklung, Administration), durchführende (Prüfungsdurchführung) und nachbereitende Maßnahmen (Auswertung, Einsicht, Archivierung). Die Akteure innerhalb dieser Prozesse sind (vgl. Abschnitt 2.3): Prüfungsverantwortliche, Studierende, Korrektor, Prüfungsamtsmitarbeiter, Sekretariat, Prüfungsaufsichten, Administrator, eAssessment Support.
Jedoch sind im praktischen Umfeld der Prüfungen die Akteure oftmals in verschiedenen Rollen aktiv. Ein Studierender kann in einer Prüfung als Prüfungsteilnehmer und in einer anderen Prüfung (im Rahmen einer studentischen Hilfskraftstelle) als Prüfungsaufsicht auftreten. In Abbildung 2.9 sind die Use-Cases des Prüfungssystems, die externen Anwendungen sowie die Akteure dargestellt.
Ein Mitarbeiter kann in die Rollen der Aufsicht, Korrektor oder Autor (AT) treten. Die Lehrperson kann ebenfalls als Autor auftreten oder eben als Prüfungsverantwortlicher (PV), Korrekteur (KO) bzw. Aufsicht (AU). Beim eAssessment Dienst ist nur die Rolle des Administrators als Nutzer im Prüfungssystem aktiv. Die Prüfungsanmeldung wird über ein Hochschul-Informationssystem geregelt, ebenso die Archivierung der Prüfungsdaten. Die Anbindung von Autorenwerkzeugen oder Learning-Management Systemen ist optional.
Im Vorfeld der Prüfung werden in der Planungsphase mögliche Prüfungsfragen gesammelt. Dies erfolgt in der Praxis nicht nur durch den Dozenten, sondern oftmals auch durch Mitarbeiter (Autoren). Ein zentraler eAssessment Dienst stellt neben der technischen auch eine didaktische und medientechnische Unterstützung bereit, um zu klären, wie Fragen umgesetzt werden können.
In der Entwicklungsphase werden die Aufgaben durch Autoren in das Prüfungssystem eingegeben. Dabei werden Aufgaben oftmals mit Hilfe von externen Autorenwerkzeugen eingegeben oder aus Learning-Management Systemen importiert. Eine Revision der Fragen inkl. grafischer Umsetzungen

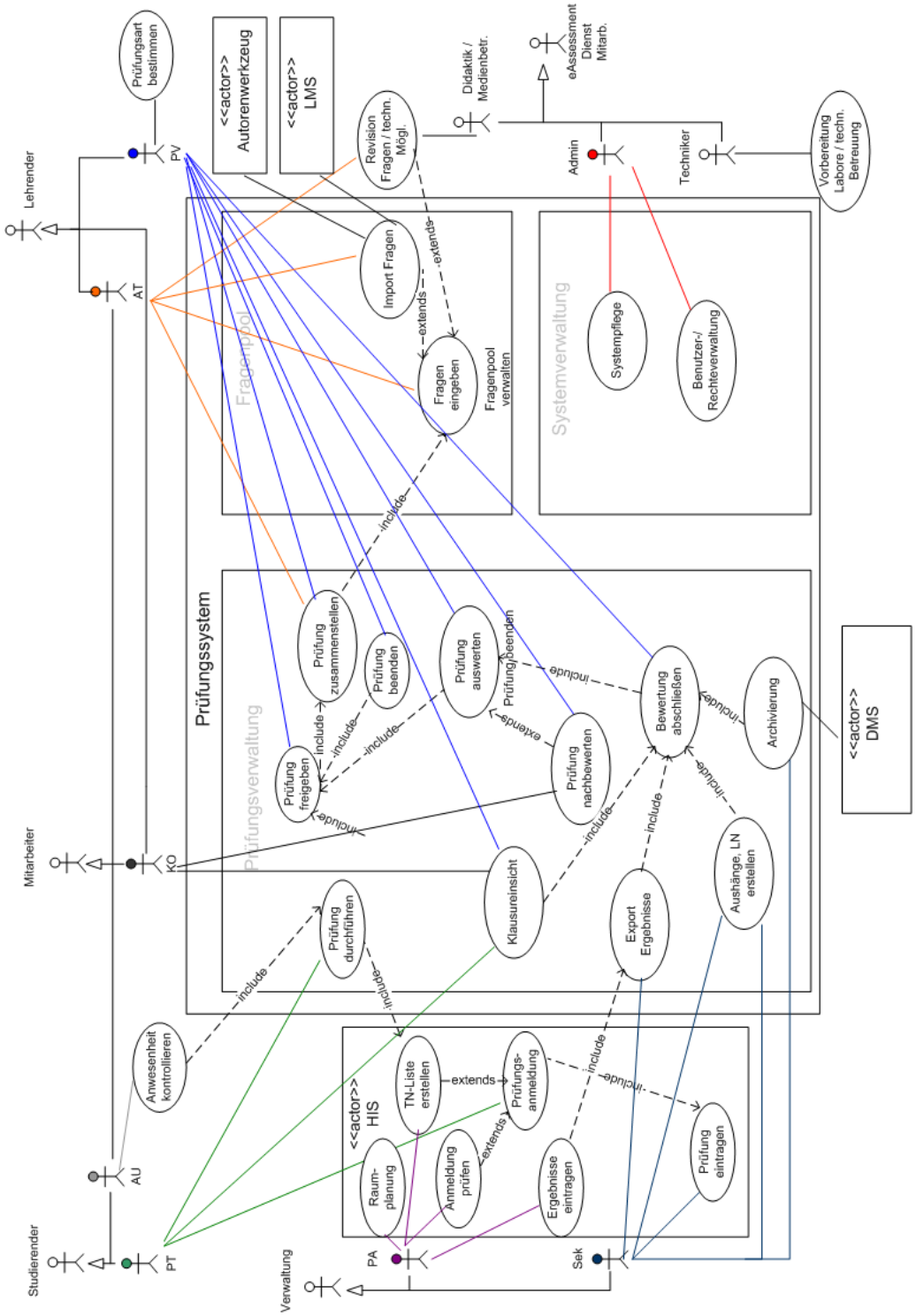

Abbildung 2.9: Use Cases

durch den eAssessment Dienst sollte auch hier stattfinden. Eine sprachliche und inhaltliche Revision ist für die Qualität der Fragen entscheidend. Der Fragenpool kann ständig erweitert oder angepasst werden.
Die Zusammenstellung der Prüfung erfolgt durch den Prüfungsverantwortlichen. Die administrative Phase läuft teilweise parallel zur Entwicklungsphase. Die Benutzerkennungen und die Rechteverwaltung obliegen hierbei dem Administrator, aber auch dem Prüfungsverantwortlichen, der AT, KO und AU einrichten möchte. Welche Prüfungen angeboten werden, erfolgt durch Angabe der entsprechenden Daten im Hochschulinformationssystem. Diese Angaben werden durch das Sekretariat der Lehrperson erfolgen.
Die Prüfungsteilnehmer melden sich über ein Hochschul-Informationssystem zu den Prüfungen an. Innerhalb dieses Systems wird durch das Prüfungsamt die Berechtigung zur Prüfung überprüft. Nach dem Anmeldezeitraum werden die Teilnehmerlisten erstellt und in das Prüfungssystem importiert. Für die Authentifizierung der Prüfungsteilnehmer kann durch den Administrator für jeden Einzelnen ein Account im System eingerichtet werden oder aber der Zugang erfolgt über eine Schnittstelle zu einem zentralen Verzeichnisdienst (z.B. LDAP). Des Weiteren sind in der Administrations-Phase die Labore zu reservieren und die Aufsichten zu koordinieren.
Zu Beginn der Durchführungsphase müssen die PC-Labore vorbereitet werden, was durch techn. Mitarbeiter des eAssessment Dienstes erfolgt. Wenn die Prüfungsteilnehmer sich erfolgreich am System angemeldet haben, schaltet der Prüfungsverantwortliche die Prüfung frei. Die angemeldeten Teilnehmer können dann mit der Prüfungsdurchführung beginnen. Während der Durchführung betreuen technisches Personal und die eingeteilten Aufsichten die Prüfungsteilnehmer. Die Aufsichten kontrollieren die Anwesenheit während der Prüfung und dabei werden die Teilnehmer anhand von Lichtbildausweisen authentifiziert. Nach der Durchführungszeit beendet der Prüfungsverantwortliche die Prüfung.
Die Auswertung der Prüfung wird durch den PV angestoßen. Dabei ist zwischen den automatisiert auswertbaren Fragen und den manuell auswertbaren Fragen zu unterschieden. Die manuell auswertbaren Fragen wie z.B. offene Fragen können neben dem PV auch durch Korrektoren nachbewertet werden. Die abschließende definitive Bewertung erfolgt dann wieder durch den PV. Hierbei fallen auch mögliche Änderungen, z.B. an den Bestehensgrenzen an. Anschließend werden die PT durch Aushänge an den Sekretariaten oder aber über den Zugang zum Prüfungssystem über die Bewertung informiert. Offiziell ist diese Benachrichtigung aber erst, wenn diese durch das Prüfungsamt erfolgt. Die anonymisierten oder pseudonymisierten Ergebnisse können dann für statistische Analysen dienen.
Nach der Bekanntgabe der Bewertung besitzt der PT ein Recht auf Akteneinsicht.

sicht. Die Akteneinsicht wird durch den PV oder einen KO begleitet. Mögliche Einsprüche der PT gegen das Prüfungsergebnis müssen im Rahmen der in den Prüfungsordnungen festgeschrieben Aufbewahrungszeiten durch den PT möglich sein. Bei der Archivierung werden die gesamten Prüfungsdaten (Fragen, Angaben, Lösungen, Bewertungsschemata, etc.) in einem Dokumenten-Management System abgespeichert.

Kapitel 3

Datenschutz und Datensicherheit elektronischer Prüfungen

Die größten Herausforderungen bei der Umsetzung und Durchführung von elektronischen Prüfungen sind der Datenschutz und die Datensicherheit.

> „*Besonders große Bedenken betreffen die Rechtssicherheit elektronischer Prüfungen!*“ [WKD09].

Vor allem die summativen Prüfungen sind oftmals Grund für Beschwerden, Anfechtungen oder Rechtstreits [VS09]. Bei den elektronischen Prüfungen kommen dazu noch die Virtualität der Prüfungsangaben sowie der Studierendenlösungen. Neben technischen Schwierigkeiten wie Rechnerabstürzen und Serverausfällen kommen oftmals die Zweifel hinzu, ob die Ergebnisse auch korrekt abgespeichert wurden. „Die mir zugeordneten Antworten wurden so nicht von mir getätigt“, „Das System hat meine Angaben nicht korrekt abgespeichert“- so könnten Klagen der Teilnehmer lauten. Dazu kommen noch die Datenschutzsaspekte. „Wer kann meine Daten einsehen bzw. welche Daten kann er sehen?“. Aber auch die Dozenten könnten Bedenken bei der Sicherheit der elektronischen Prüfungen haben: „Wie kann ich sicherstellen, dass nur die Studenten die Fragen sehen, die dazu berechtigt sind?“, „Sind die Prüfungsdaten wirklich vorher nicht einsehbar?“ und „Ist eine elektronische Klausurdurchführung überhaupt erlaubt?“.

3.1 Sicherheitsanalyse

Die Sicherheit von elektronische Prüfungen bezieht sich nicht nur auf die technischen Anforderungen, sondern es ist notwendig, die gesamte Umgebung mit einzubeziehen (siehe u.a. [Eib08a]). Dazu gehören auch die orga-

nisatorischen Prozesse und die beteiligten Nutzer. Åhlfeldt definiert ein Sicherheitsmodell, das die organisatorische und administrative Sicherheit mit einbezieht [ASS07]. Dazu wird das Modell in drei Hauptteile aufgegliedert: technisch, formal, informal. Neben der technischen Sicherheit werden in der formal-adminstrativen Sicherheit vor allem Regelungen, Gesetze und Vorschriften betrachtet. Die informal-administrative Sicherheit versucht über die Relevanz von Sicherheitsmaßnahmen aufzuklären, wobei hierunter Schulungen etc. zu sehen sind.
Dieses Modell wird als Basis zur Ermittlung und Einordnung der Anforderungen für die elektronischen Prüfungen verwendet. Ausgehend von den allgemeinen Schutzzielen *Authentizität*, *Datenintegrität*, *Vertraulichkeit*, *Verfügbarkeit*, *Verbindlichkeit* und *Anonymisierung/ Pseudonymisierung*, werden anschließend die prüfungsrechtlichen und datenschutzrechtlichen Anforderungen betrachtet. Die Dreiteilung der Sicherheitsziele nach Åhlfeldt kann also so aussehen, dass die allgemeinen Schutzziele sowie einige prüfungsrechtliche Anforderungen im Wesentlichen der technischen Sicherheit zuzuordnen sind. Vorschriften und Gesetze aus prüfungsrechtlicher und datenschutzrechtlicher Sicht sind wiederum vorwiegend der formal-administrativen Sicherheit zuzuordnen. Bei den elektronischen Prüfungen ist der informal-administrative Aspekt geprägt von Schulungsmaßnahmen sowie Informationen um die Studierenden und Dozierende an die neue Prüfungsart zu gewöhnen.
Für ein Sicherheitskonzept spielen alle drei Bestandteile eine wichtige Rolle, jedoch sind für eine softwaretechnische Umsetzung nur die technischen und formal-administrativen Anforderungen von Bedeutung[1].

3.1.1 Definition der allgemeinen Schutzziele

Authentizität

Die Authentizität unterscheidet nach [Eck06] zwischen der Authentizität eines Subjektes (im allgemeinen Benutzer) und der Authentizität von Objekten (Server, Access-Points, Software, etc.). Die Authentizität wird durch Maßnahmen der Authentifikation sichergestellt. Die Authentizität eines Subjektes oder Objektes erfolgt über einen Identitätsnachweis, der einem anderen Subjekt oder Objekt seine Identität zweifelsfrei nachweisen kann.
Betrachtet man beispielsweise die Authentizität von Daten, so soll überprüft werden, ob eine Nachricht nachweislich von einer bestimmten Instanz stammt und ob Sie beim Empfänger unverändert angekommen ist (siehe Datenintegrität). Entscheidend dabei ist, dass die Bindung an die Identität auch nach

[1] Beispiele und Anregungen wie die informal-administrativen Anforderungen umgesetzt werden können finden sich u.a. in [VS09]

der erfolgreichen Authentifikation überprüfbar bleibt [BMB+05].

Datenintegrität

Für den Empfänger der Daten muss eindeutig erkennbar sein, ob die Daten während der Übertragung verändert wurden [BMB+05]. Zum einen erfordert dies die Festlegung von Berechtigungen für die Daten, indem Rechte an Subjekte vergeben werden. Zum anderen müssen Techniken verwendet werden, um Veränderungen, wie z.B. absichtliche Veränderungen etc., eindeutig zu erkennen.

Vertraulichkeit

Die Vertraulichkeit der Daten ist gewährleistet, wenn sichergestellt ist, dass Subjekte nicht unautorisiert Kenntnis von diesen Daten erlangen können [Eck06]. Dies erfordert die Festlegung von Berechtigungen, speziell für personenbezogene bzw. personenbeziehbare Daten, wobei zwischen der Speicherung und der Übertragung von Daten zu unterscheiden ist. Bei den abgespeicherten Daten sind die genannten Zugriffsrechte zu vergeben. Bei den zu übertragenden Daten hingegen ist die Verschlüsselung der Daten nötig, um die Vertraulichkeit zu gewährleisten.

Verfügbarkeit

Ein System gewährleistet die Verfügbarkeit, wenn authentifizierte und autorisierte Instanzen in der Wahrnehmung ihrer Berechtigungen nicht unautorisiert beeinträchtigt werden können [Eck06]. Für die elektronischen Prüfungen ist die Verfügbarkeit während der Prüfungsdurchführung unabdingbar. Selbst Ausführungsverzögerungen können nur in einem sehr kleinen Rahmen toleriert werden.

Verbindlichkeit

Die Verbindlichkeit (Nichtabstreitbarkeit) einer Menge von Aktionen ist gewährleistet, wenn es nicht möglich ist, dass ein Subjekt im Nachhinein die Durchführung einer solchen Aktion abstreiten kann [Eck06]. Für die elektronischen Prüfungen bedeutet dies, dass ein Teilnehmer im Nachhinein die gemachten Lösungen und ein Prüfungsverantwortlicher die gestellten Aufgaben nicht abstreiten kann.
Die Verbindlichkeit setzt die Authentizität und Integrität voraus.

Anonymisierung und Pseudonymisierung

> „*Unter der Anonymisierung versteht man das Verändern personenbezogener Daten der Art, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können.*“[2] [Eck06]

Eine schwächere Form der Anonymisierung ist die Pseudonymisierung. Hierbei werden personenbezogene Daten z.B. durch die Verwendung von Pseudonymen derart verändert, so dass die Einzelangaben über eine natürliche Person nicht zugeordnet werden können[3] [Eck06]. Die Anonymität steht zum Teil im Widerspruch zur Authentizität bei elektronischen Prüfungen. Aber gerade bei der Bewertung der Prüfungsleistungen durch den Prüfungsverantwortlichen könnte eine Anonymität oder Pseudonymität die Objektivität der Bewertung erhöhen.

3.1.2 Prüfungsrechtliche Anforderungen

P 1 (Formvorschrift)**.**

Im Vorfeld einer Prüfung ist zuerst einmal zu überprüfen, ob eine Prüfung überhaupt elektronisch durchgeführt werden darf. Dazu ist zu überprüfen, ob eine schriftliche Prüfung durch eine elektronische Form ersetzt werden kann. Die rechtlichen Anforderungen, die an eine schriftliche papierbasierte Prüfung gestellt werden, müssen auch für elektronische Prüfungen gelten [KF08]. Die papierbasierte Durchführung kann nach § 126 Abs. 3 Bürgerliches Gesetzbuch (BGB) durch die elektronische Form gemäß § 126a BGB ersetzt werden:

> „*Soll die gesetzlich vorgeschriebene schriftliche Form durch die elektronische Form ersetzt werden, so muss der Aussteller der Erklärung dieser seinen Namen hinzufügen und das elektronische Dokument mit einer qualifizierten digitalen Signatur nach dem Signaturgesetz versehen.*“

Daraus ergibt sich, dass für eine rechtssichere Durchführung einer elektronischen Prüfung die Verwendung von qualifizierenden digitalen Signaturen nach dem Signaturgesetz (SigG) verpflichtend ist.

[2] § 3 Abs. 6 Bundesdatenschutzgesetz (BDSG)
[3] § 3 Abs. 6a Bundesdatenschutzgesetz (BDSG)

P 2 (Anpassung der Prüfungsordnung)**.**

Die Umsetzung von Anforderung P1 ermöglicht die rechtliche Gleichstellung einer elektronischen Durchführung mit der herkömmlichen papierbasierten Durchführung. Dennoch ist die elektronische Durchführung als neue eigenständige Prüfungsform in den Prüfungsordnungen zu erwähnen [KF08]. In [KF08] finden sich entsprechende Formulierungsvorschläge.

P 3 (Antwort-Wahl-Verfahren)**.**

Neben der Prüfungsform muss auch das Antwort-Wahl-Verfahren (Multiple-Choice) in die Prüfungsordnung mit aufgenommen werden. Denn gerade das Antwort-Wahl-Verfahren wird in elektronischen Prüfungen sehr häufig eingesetzt, weil es eine vollständige automatisierte Auswertung erlaubt. Wird das Antwort-Wahl-Verfahren nicht in die Prüfungsordnung aufgenommen, kann der Anspruch auf Durchführung eines rechtsfehlerfreien Prüfungsverfahrens verletzt sein[4] [KF08].

P 4 (Betriebssicherheit)**.**

Eine weitere Anforderung ist die Betriebssicherheit der verwendeten Technik. Zum einen muss die Verfügbarkeit während der Durchführung gewährleistet sein (siehe Unterabschnitt 3.1.1) und zum anderen müssen Regelungen getroffen werden, was bei einem Systemausfall oder bei Verzögerungen vor bzw. während der Prüfung passiert [KF08, Ree08a]. Denn Teilnehmer haben die Möglichkeit, bei Prüfungen, bei denen sich die Durchführung zu lange verzögert oder ein Systemausfall während der Prüfung für lange Verzögerungen gesorgt hat, Einspruch zu erheben [KF08].
Die Auswirkungen eines solchen Zwischenfalls werden an folgendem Vorfall deutlich: Bei der ersten elektronischen Klausur an der Berliner Charité 2004 stürzte der Zentralrechner während der Prüfung ab. Die 150 Teilnehmer bekamen ihre Leistungsnachweise ohne Gegenleistung, weil die Hochschulleitung mögliche Einsprüche der Studierenden fürchtete [Kri05].
Der Ausfall oder die Fehlfunktion von technischen Komponenten darf nicht dazu führen, dass ein Teilnehmer die Prüfung nicht beenden kann. In Richtlinien muss festgelegt werden, welche Maßnahmen bei einem Systemausfall getroffen werden. Außerdem sind die verantwortlichen Stellen (eAssessment Dienst usw.) und ihre Aufgaben in den Regelungen festzuhalten.

[4] OVG Bautzen, Beschluss vom 10. Oktober 2002 - 4 BS 328/02 - NVwZ-RR 2003, 853 ff.

P 5 (Eindeutige Zuordnung Teilnehmer-Prüfung)**.**

Für die Durchführung muss im Prüfungssystem eine eindeutige Zuordnung zwischen Teilnehmer und Prüfung existieren [Ree08b]. Diese Zuordnung erfolgt durch eine Anmeldung des Studierenden zur Prüfung. Die Anmeldungen und die damit einhergehende Überprüfung, ob der Studierende an der Prüfung teilnehmen darf, erfolgt durch die Prüfungsämter (siehe Abbildung 2.9). Die Zuordnungen Teilnehmer-Prüfung müssen dann in das Prüfungssystem übertragen werden bzw. in diesem zur Verfügung stehen. Die Zuordnungen Teilnehmer-Prüfung und Teilnehmer-Prüfungslösung sind selbstredend sehr sensible Daten, die es vor unautorisierten Zugriffen zu schützen gilt.
Die Zuordnung der Prüfungsdaten zu einem Teilnehmer bedarf einer eindeutigen Authentifizierung des Teilnehmers am Prüfungssystem. Außerdem sind die Zuordnung und Prüfungsdaten mit geeigneten Maßnahmen vor unautorisierten Zugriffen zu schützen. Ein Administrator ist dabei kein autorisierter Nutzer.

P 6 (Betrugssicherheit)**.**

Zur Durchführung müssen sich die Teilnehmer am Prüfungssystem über eine eindeutige Benutzerkennung anmelden, um die Teilnehmer nachweisbar eindeutig zu identifizieren (siehe Abschnitt 3.1.1) [Ree08a]. Dennoch muss die Authentizität der Teilnehmer durch geeignete Maßnahmen überprüft werden um sicher zu stellen, dass ein Teilnehmer seine Benutzerkennung nicht an eine andere Person weitergegeben hat. Wie bei den papierbasierten Verfahren ist dies durch eine Ausweiskontrolle (Studierenden- und Personalausweis) möglich oder aber durch den Einsatz von biometrischen Authentifizierungssystemen wie Fingerprint, Gesichtserkennung etc.[5].
Damit einher geht auch die Anforderung die Aufsichtsregeln einzuhalten. Während sich bei den papierbasierten Verfahren die Täuschungsmöglichkeiten auf das Abschreiben bzw. das sog. „Spicken“ beschränken, sind diese bei den elektronischen Prüfungen vielfältiger (siehe u.a. [Ree08b,Cri07]). So darf es einem Teilnehmer nicht möglich sein, durch eine erneute Anmeldung am Prüfungssystem die Prüfung noch einmal durchzuführen. Dies gilt auch für das Ausschalten oder Neustarten des Arbeitsrechners. Auch die unerlaubte Verwendung von Hardware und Software ist zu unterbinden.
Falls Täuschungsversuche erkannt werden, ist es entscheidend, diese auch belegen und damit eine evtl. Bestrafung begründen zu können. Dazu ist eine Dokumentation des gesamten Prüfungsablaufes nötig [Ree08a].

[5] Auf eine detaillierte Betrachtung dieser Technologien wird in dieser Arbeit verzichtet. Siehe dazu u.a. [Eck09]

P 7 (Gleichheitsgrundsatz)**.**

> „*Bei allen Arten von Prüfungen ist der Grundsatz der Chancengleichheit zu berücksichtigen. Dieser Grundsatz wurde aus dem Gleichheitssatz aus Art. 3 Abs. 1 Grundgesetz (GG) entwickelt und ist mittlerweile zum zentralen Kontrollmaßstab von Prüfungsentscheidungen geworden.*“ [KF08]

Bei den elektronischen Prüfungen ist die Chancengleichheit sowohl für die äußeren Prüfungsbedingungen, als auch für den Prüfungsinhalt, also die Fragen, sicher zu stellen. Denn gerade bei einigen elektronischen Prüfungssystemen können die Prüfungen randomisiert zusammengestellt werden. Dazu werden die Fragen nach dem Zufallsprinzip aus einem Pool ausgewählt. Hierbei muss sichergestellt sein, dass jeder Teilnehmer eine Prüfung mit einem vergleichbaren Schwierigkeitsgrad wie alle anderen erhält. Das bedeutet also entweder die gleiche Prüfung für alle oder individuelle, aber gleichwertige Prüfungen für alle [VS09].
Bei den äußeren Prüfungsbedingungen sind auch die Ausstattungen der Rechner bzw. der verwendeten Software zu beachten. Ob allerdings der Einfluss von Faktoren wie z.B. verschiedene Betriebssysteme, 15 Zoll oder 22 Zoll Monitor etc. groß genug sind um zwischen „bestehen“ oder „nicht bestehen“ zu entscheiden, ist fraglich und nicht allgemeingültig zu bewerten [VS09]. Dennoch sollten die technischen Bedingungen wie z.B. Softwareausstattung, Netzwerkanbindung etc. für alle Teilnehmer vergleichbar sein. Des Weiteren sind unterschiedliche PC-Kenntnisse seitens der Prüfungsteilnehmer als unvermeidbar hinzunehmen [KF08].
Chancengleichheit muss aber auch bei der Bewertung gelten. Die teilweise vollautomatische Auswertung bei elektronischen Prüfungen trägt hierbei zur Gleichbehandlung bei. Bei offenen Aufgaben wie z.B. Aufsätzen oder Essays ist jedoch eine manuelle Korrektur nötig. Eine anonyme bzw. anonymisierte Bewertung würde auch hierbei die Chancengleichheit gewährleisten. Jedoch besteht kein rechtlicher Anspruch auf Anonymität [ZB07].

P 8 (Nichtabstreitbarkeit)**.**

Eine der wichtigsten rechtlichen Anforderung ist die Nichtabstreitbarkeit (Verbindlichkeit) der Prüfungsangaben des Teilnehmers (vgl. Abschnitt 3.1.1). Das bedeutet, ein Teilnehmer darf im Nachhinein seine gemachten Prüfungsangaben nicht abstreiten können. Das Gleiche gilt aber auch für den Dozenten: Der Teilnehmer muss sicher sein, dass die Klausurfragen vom Dozenten im Nachhinein nicht abgestritten werden können. Was bei den handschriftlichen Prüfungen durch entsprechende Gutachten bewiesen werden kann, muss

bei der elektronischen Variante durch standardisierte und rechtsgültige Verfahren geregelt werden. Damit geht die Anforderung einher, dass sowohl die Prüfungsangaben als auch die Prüfungslösungen vollständig und korrekt vorliegen. Die in der Anforderung P1 beschriebenen qualifizierenden digitale Signaturen erfüllen die Anforderung der Nichtabstreitbarkeit.

P 9 (Einsicht und Archivierung)**.**

Weitere rechtliche Anforderungen sind die zu gewährende Einsichtnahme in die Prüfung inkl. Bewertung und die Archivierung [KF08]. Dem Teilnehmer muss die Möglichkeit gegeben werden, Einsicht in seine Prüfungsangaben und die Bewertung zu nehmen. Die Archivierung der Prüfungslösungen bzw. deren Angaben und Bewertungen dienen dazu, den Teilnehmern eine Einsicht in ihre Lösungen und deren Bewertung zu geben, um nachvollziehen zu können wie die Bewertung ihrer Leistung zustande gekommen ist. Auch aus rechtlicher Sicht haben die Teilnehmer das Recht zur Akteneinsicht nach §29 Verwaltungsverfahrensgesetz (VwVfG), außerdem ist dies aus den Hochschulgesetzen der Länder (z.B. §94 Abs. 2 Ziff. 15 HG NRW) ableitbar [ZB07].
Die Fristen, in denen das Einsichtsrecht des Teilnehmers möglich sind, sind dem Teilnehmer durch eine Rechtsmittelbelehrung mitzuteilen. Diese Belehrung muss dem Teilnehmer unmittelbar nach Bekanntgabe der Note mitgeteilt werden und muss den Teilnehmer über seine Rechte zur Anfechtung der Prüfung informieren. In der Belehrung muss auch die Frist enthalten sein, in der der Teilnehmer Widerspruch einlegen muss (siehe [ZB07]). Erfolgt keine Rechtsmittelbelehrung so ist ein Widerspruch innerhalb der nach §58 Abs. 2 Verwaltungsgerichtsordnung (VwGO) angegebenen Jahresfrist möglich.
Die Archivierungspflichten sind in den jeweiligen Prüfungsordnungen festgesetzt. Dabei gelten die weiteren Anforderungen, dass die Prüfungslösungen und die Prüfungsangaben zuverlässig und nachweisbar unverändert abgespeichert werden. Die Dokumentation des Prüfungsverlaufes (siehe Anforderung P6) ist nur für die Dauer der Einspruchsfrist vorzuhalten. Eine Musterlösung zur Prüfung muss nicht Bestandteil der Archivierung sein, weil diese nicht das konkrete Prüfungsverfahren des Teilnehmers betrifft [ZB07]. Die archivierten Daten dürfen nicht unautorisiert zugänglich sein.

Urheberrechtliche Anforderungen

Wenn in einer elektronischen Prüfungsumgebung Aufgaben von Dritten verwendet oder erstellt werden, so sind hierbei die urheberrechtlichen Bestimmungen zu beachten. Dies trifft in noch größerem Ausmaß bei eLearning-Systemen zu, in denen fremde Materialien eingestellt werden, die Quellen wissenschaftlicher, literarischer oder künstlerischer Abbildungen sind [KF08].

Des Weiteren werden oftmals die Erstellung von Prüfungsfragen wissenschaftlichen Angestellten überlassen. Bei alledem gilt der Grundsatz nach dem Urhebergesetz (UrhG), dass der Schöpfer eines Werkes der Urheber der Werkes ist[6].
Erfolgt die Erstellung von Prüfungsfragen aus dem Angestelltenverhältnis eines wissenschaftlichen Mitarbeiters heraus, so erhält der Arbeitgeber aufgrund des Arbeitsvertrages und § 43 UrhG die Nutzungsrechte an den Prüfungsfragen, wenn sie in Erfüllung der Dienstpflichten geschaffen wurden [TH08].
Werden in einer Prüfung z.B. Teile eines Werkes Dritter oder einzelne Beiträge aus Zeitungen, Zeitschriften o.ä. verwendet, so ist dies grundsätzlich nach § 52a Abs. 1 Nr. 1 UrhG (Recht der öffentlichen Zugänglichmachung für Unterricht) möglich. Voraussetzung hierbei ist allerdings, dass das verwendete Werk oder der Beitrag bereits veröffentlicht wurde. Beispiele hierfür sind ein im Verlag erschienenes Buch oder ein ausschließlich im Internet veröffentlichter Text [TH08].
Für die weiteren Betrachtungen der urheberrechtlichen Aspekte speziell an Hochschulen sei u.a. auf die Quellen [KF08, TH08] und [Kal08] verwiesen.

3.1.3 Datenschutz

Beim Datenschutz unterliegen die Hochschulen ihren jeweiligen Landesdatenschutzgesetzen (LDSG). Da sich diese Arbeit jedoch nicht auf ein bestimmtes Bundesland beschränkt, wird das Bundesdatenschutzgesetz (BDSG) verwendet.
Der Zweck des Bundesdatenschutzgesetzes (BDSG) ist in § 1 Abs. 1 BDSG definiert:

> „*Zweck dieses Gesetzes ist es, den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.*“

Personenbezogene Daten sind nach § 3 Abs. 1 BDSG definiert als *„...Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener).“*
Welche Form des Umgangs mit den personenbezogenen Daten gemeint ist, lässt sich anhand der Grundprinzipien des BDSG erklären: Prinzip des Erlaubnisvorbehalts, Zweckbindungsprinzip, Erforderlichkeitsprinzip und die Prinzipien der Datenvermeidung und der Datensparsamkeit.

[6] § 7 UrhG

Prinzip des Erlaubnisvorbehalts

Das Prinzip des Erlaubnisvorbehalts besagt, dass schutzwürdige personenbezogene Daten nur aufgrund von Gesetzen oder einer Einwilligung des Betroffenen erhoben und verarbeitet werden dürfen[7]. Das bedeutet, dass die Verwendung von personenbezogenen Daten verboten ist. Sie ist nur dann erlaubt, wenn eine gesetzliche Erlaubnis oder eine ausdrückliche Einwilligung der betroffenen Person vorliegt [HH08]. Daraus lässt sich schließen, dass die Verwendung der bei den elektronischen Prüfungen grundsätzlich erhobenen Daten einer Einwilligung der Teilnehmer bedarf. Allerdings gilt die Einwilligung des Teilnehmers nur dann, falls sie auf der freien Entscheidung des Teilnehmers beruht[8]. In wie weit diese Freiwilligkeit im Rahmen von Prüfungen bzw. dem Verhältnis Lehrender zu Student gegeben ist, ist fraglich (vgl. [LH09]).
Es bedarf auch hier der Anpassung der Prüfungsordnungen, um eine entsprechende Rechtsgrundlage zu schaffen (siehe Anforderung P2). Diese Problematik beschränkt sich aber nicht nur auf die elektronischen Prüfungen, sondern auch auf alle Informationssysteme an Hochschulen (siehe u.a. [LH09, HH08, Wet08b]).
Dennoch ist das informationelle Selbstbestimmungsrecht für alle beteiligten Akteure nicht unbedingt durch Gesetze und Vorschriften aufzuweichen. Vielmehr sollten alle Akteure selbst bestimmen können, wer welche Daten wann zu Gesicht bekommt und damit „Herr über ihre Daten“ bleiben können. Allerdings gilt es zu bedenken, dass die gesamten Prüfungsdaten nicht nur Daten des Teilnehmers sind, sondern auch Daten des Prüfers.

D 1 (Erlaubnisvorbehalt)**.** *Jeder Nutzer eines elektronischen Prüfungssystems muss „Herr seiner Daten“ bleiben. Dazu zählt auch, dass die Zuordnungen Teilnehmer-Prüfung (siehe rechtliche Anforderung P5), die Prüfungsangaben sowie die Prüfungslösungen und die Prüfungsbewertungen nur explizit autorisierten Nutzern zugänglich sind. Administratoren sind* ***keine*** *autorisierten Nutzer.*

Zweckbindungsprinzip

Das Zweckbindungsprinzip beschreibt, dass die einmal erhobenen Daten nur für den Zweck verarbeitet und genutzt werden dürfen, für den sie ursprünglich erhoben wurden. Der Zweck der Erhebung muss in der Einwilligung oder der gesetzlichen Erlaubnis exakt bezeichnet werden [Bor08]. Dieses Prinzip ermöglicht eine Transparenz der Erhebung und Verarbeitung der Daten, so

[7] § 4 Abs. 1 BDSG
[8] § 4a Abs. 1 BDSG

dass nachvollziehbar ist, wer wann welche Daten einsehen konnte [LH09].
Problematisch ist dies aber vor allem bei LMS, die über ein integriertes Prüfungsmodul verfügen (wie. z.B. Moodle). Hierbei werden vor allem die persönlichen Daten für die Zwecke des eLearnings und für die elektronischen Prüfungen verwendet. Das bedeutet, dass der Dozent oftmals die Tätigkeiten der Prüfungsteilnehmer innerhalb des LMS mit der erbrachten Leistung in der Prüfung vergleichen kann (siehe [Eib08b, Eib08a]).

Erforderlichkeitsprinzip

Das Erforderlichkeitsprinzip beschreibt, dass nur die Daten erhoben, verarbeitet und genutzt werden dürfen, die für den eigentlichen Zweck benötigt werden.
Daraus ergibt sich auch die Tatsache, dass die Daten, die nicht erforderlich sind, nicht verwendet werden dürfen und somit auch nicht einmal erhoben werden dürfen [Bor08].

D 2 (Zweckbindung und Erforderlichkeit)**.** *Alle Daten, die innerhalb des elektronischen Prüfungssystems erhoben werden, sind ausschließlich für die Zwecke der Prüfungsdurchführung, Auswertung, Einsicht und Archivierung zu verwenden. Falls weitere Verwendungen wie z.B. Evaluation, statistische Analysen etc. nach der Durchführung bzw. Auswertung geplant sind, müssen die Betroffenen dem zustimmen bevor diese Daten überhaupt erhoben werden.*
Alternativ wären die Daten durch Anonymisierung / Pseudonymisierung so zu verändern, dass kein Personenbezug mehr möglich ist, womit die Daten vollständig verwendet bzw. weitergegeben werden können.

Datenvermeidung und Datensparsamkeit

Das Ziel der Datenvermeidung und der Datensparsamkeit ist es, der Gefahr der Profilbildung von Betroffenen vorzubeugen. Die Datenvermeidung kann allerdings gerade bei elektronischen Prüfungen nur schwer realisiert werden. Dennoch kann sie erreicht werden, indem die personenbezogenen Daten anonymisiert werden (vgl. § 3a BDSG). Somit wird der Personenbezug aufgehoben und die Daten stellen keine Gefahr mehr für das informationelle Selbstbestimmungsrecht dar [Bor08].
Eine Abschwächung der Datenvermeidung aus datenschutzrechtlicher Sicht ist die Datensparsamkeit, die vor allem durch das Pseudonymisieren von personenbezogenen Daten erreicht werden kann.

D 3 (DATENMINIMIERUNG)**.** *Die Daten sind soweit wie möglich zu anonymisieren bzw. pseudonymisieren.*
Alle personenbezogenen Daten müssen aus dem Prüfungssystem nach einer bestimmten Frist vollständig gelöscht werden (Löschungsverpflichtung).

3.1.4 Einordnung der Anforderungen

In Tabelle 3.1 sind die Anforderungen zusammengefasst dargestellt. Die Anforderungen sind in drei Realisierungskategorien aufgeteilt: formal, administrativ und technisch. Diese Aufteilung orientiert sich an dem Modell von Åhlfeldt (siehe Abschnitt 3.1) und zeigt, dass die Sicherheit von elektronischen Prüfungen nur durch formale, administrative und technischer Maßnahmen umzusetzen sind. Eine formale Umsetzung bedeutet, dass hier Änderungen durch Verordnungen zu realisieren sind wie bei (P2) und des Antwort-Wahl-Verfahrens (P3).
Unter administrativen Umsetzungen sind Änderungen im Gesamtkonzept der elektronischen Prüfungen zu sehen. Darunter fallen auch bauliche Maßnahmen wie z.B. bei P64 die Errichtung von Trennwänden, um das „Spicken“ zu verhindern. Die technischen Maßnahmen werden unterteilt in Maßnahmen die nur durch Anpassungen am Prüfungssystem direkt zu realisieren sind (z.B. P65)[9] und Maßnahmen, die durch ein Sicherheitskonzept realisiert werden können. Viele dieser technischen Anforderungen können durch die allgemeinen Schutzziele bzw. Kombinationen davon repräsentiert werden. Jedoch widersprechen sich die aufgeführten Anforderungen teilweise komplett.
Wie in Tabelle 3.1 zu erkennen, steht die Anforderung der Authentizität der Teilnehmer (P61) der Datenschutzanforderung nach Anonymität (D21) entgegen. Die Protokollierung des gesamten Prüfungsverlaufes (P67) sowie die Archivierung (P92) widersprechen der allgemeinen Datenschutzanforderung der Datenvermeidung bzw. -sparsamkeit.

Das Ziel für ein software-basiertes Sicherheitskonzept für die elektronischen Prüfungen muss sein, dass trotz der Widersprüche zwischen einigen Anforderungen, die Anforderungen als Ganzes umzusetzen sind. Dazu werden im folgenden Kapitel mögliche Sicherheitsmaßnahmen für die einzelnen Anforderungen diskutiert.

[9] Die Maßnahmen die durch Anpassungen am Prüfungssystem zu erfolgen haben sind in der Spalte „technisch“ kursiv dargestellt

Anforderung	Gliederung	Umsetzung		
		formal	administrativ	technisch
P1: Formvorschrift	P11: Rechtliche Gleichstellung Papier u. elektronisch		PKI, Smartcards, Smartcardreader	Verbindlichkeit (Qualif. dig. Sign. nach SigG)
P2: Anpassung der Prüfungsordnung	P21: elektronische Prüfungsform in Prüfungsordnungen aufnehmen	Änderung der Prüfungsordnungen		
P3: Antwort-Wahl-Verfahren	P31: Antwort-Wahl Verfahren in Prüfungsordnungen aufnehmen	Änderung der Prüfungsordnungen		
P4: Betriebssicherheit	P41: Sicherstellung der Verfügbarkeit des Prüfungssystems	Richtlinien für den Fall des Systemausfalls	Lastverteilung, USV	Verfügbarkeit
	P42: Ausfallsicherheit des Clients	Richtlinien für den Ausfall eines Clients	Ersatzclients bereitstellen	
P5: Zuordnung Teilnehmer-Prüfung	P51: Eindeutige Zuordnung Teilnehmer-Prüfung	Anmeldung über Prüfungsamt	Import Teilnehmerliste ins Prüfungssystem	Zugriffskontrolle
	P52: Zuordnung Teilnehmer - Prüfungslösungen			Zugriffskontrolle
P6: Betrugssicherheit / Dokumentation	P61: Eindeutige Identifizierung und Autorisierung der Teilnehmer		Ausweiskontrolle durch Aufsichten	Authentizität
	P62: Rechtzeitigkeit der Prüfungsangaben			Vertraulichkeit
	P63: Vollständigkeit der Prüfungsangaben zu Prüfungsbeginn			Verbindlichkeit, Integrität
	P64: Vollständigkeit der Prüfungslösungen nach Durchführung			Verbindlichkeit, Integrität
	P65: Verhinderung des Unterschleifs		Trennwände, im Tisch versenkbare Monitore etc.	Vertraulichkeit, *Randomisierte Aufgabenstellung*
	P66: Verhinderung der Mehrfachdurchführung bzw. Mehrfachlogin			Rechtemanagement, *Sessionbasierter Login*
	P67: Verhinderung der Benutzung unerlaubter HW/SW	Maßnahmen/ Richtlinien für Aufsichten	Sichere Prüfungsumgebung (Secure Browser Konzept)	
	P68: Nachvollziehbarkeit des Prüfungsverlaufes			Protokollierung
	P69: Nachträgliche Veränderung bzw. Mehrfachabgabe der Lösungen verhindern			Rechtemanagement, Verbindlichkeit
P7: Gleichheitsgrundsatz	P71: Vergleichbare technische Bedingungen für alle Teilnehmer		Gleichwertige Ausstattung (HW/SW)	

Fortsetzung der Tabelle auf der nächsten Seite ...

Fortsetzung der Tabelle				
		Umsetzung		
Anforderung	Gliederung	formal	administrativ	technisch
	P72: Gleichwertige Umgebungen für alle	Richtlinien was bei äußerlichen Störungen (z.B. Baulärm) zu tun ist	Geräuscharme Eingabegeräte, Lüfter, etc.	
	P73: Anonyme/ anonymisierte Bewertung der Lösungen			*Ausblenden der Personendaten der Teilnehmer bei der Bewertung*
P8: Nichtabstreitbarkeit	P81: Teilnehmer dürfen ihre Lösungen im Nachhinein nicht abstreiten dürfen			Verbindlichkeit
	P82: Dozent darf Angaben u. Bewertungen im Nachhinein nicht abstreiten dürfen			Verbindlichkeit
P9: Einsicht und Archivierung	P91: Teilnehmer hat Recht auf Einsicht seiner Lösungen und Bewertungen			Verbindlichkeit
	P92: Angaben, Lösungen und Dokumentation sind zu archivieren	Archivierungsdaten festlegen	DMS	Verbindlichkeit, Vertraulichkeit, *Exportfunktionalität*
D1: Erlaubnisvorbehalt	D11: Nutzer bleiben Herr ihrer Daten			Zugriffskontrolle, Rechtemanagement
D2: Zweckbindung und Erforderlichkeit	D21: Personendaten der Teilnehmer nur für Prüfungszwecke erheben	Datenschutzbestimmungen, Verfahrensverzeichnis		Vertraulichkeit
	D22: Statistische Analyse der Prüfungsdaten ermöglichen			Anonymisierung / Pseudonymisierung

Tabelle 3.1: Anforderungen an elektronische Prüfungen

3.2 Sicherheitsmaßnahmen

Im Folgenden werden die in Unterabschnitt 3.1.1, Unterabschnitt 3.1.2 und Unterabschnitt 3.1.3 aufgeführten Anforderungen an die Sicherheit und den Datenschutz umgesetzt. Dabei werden mögliche Technologien aufgeführt, die später zu einem Sicherheitskonzept zusammengefasst werden. Für das allgemeine Verständnis werden deshalb zuerst einige kryptografische Grundlagen erklärt, bevor dann auf die Realisierungsmöglichkeiten der einzelnen Anforderungen eingegangen wird.

3.2.1 Kryptografische Grundlagen

Definition Kryptosystem

Unter einem Kryptosystem versteht man ein System das festlegt, wie Klartexte in Chiffretexte verschlüsselt und wie die Chiffretexte wieder in Klartexte entschlüsselt werden. Ein Kryptosystem kann wie folgt beschieben werden [Eck06]:

> Gegeben seien zwei endliche Zeichenvorräte (Alphabete) A_1 und A_2. Ein Kryptosystem ist gegeben durch ein Tupel
>
> $$KS = (M, C, EK, DK, E, D)$$
>
> mit
>
> 1. der nicht leeren endlichen Menge von Klartexten $M \subseteq A_1^*$ wobei A_1^* die Menge aller Worte über dem Alphabet A_1 beschreibt,
> 2. der nicht leeren endlichen Menge von Krypto- bzw. Chiffretexten $C \subseteq A_2^*$,
> 3. der nicht leeren Menge von Verschlüsselungsschlüsseln EK,
> 4. der nicht leeren Menge von Entschlüsselungsschlüsseln DK sowie einer Bijektion $f : EK \to DK$. Die Bijektion assoziiert zu einem Verschlüsselungsschlüssel $K_E \in EK$ einen dazu passenden Entschlüsselungsschlüssel $K_D \in DK$, d.h. $f(K_E) = K_D$,
> 5. dem linkstotalen und injektiven Verschlüsselungsverfahren
>
> $$E : M \times EK \to C$$
>
> 6. dem Entschlüsselungsverfahren
>
> $$D : C \times DK \to M$$
>
> mit der Eigenschaft, dass für zwei Schlüssel $K_E \in EK, K_D \in DK$ mit $f(K_E) = K_D$ gilt:
>
> $$\forall m \in M : D(E(M, K_E), K_D)) = M.$$

Aus der obigen Definition eines Kryptosystems ergibt sich, dass Klartexte und Chiffretexte Worte über einem endlichen Zeichenvorrat sind. Die Alphabete A_1 und A_2 können dabei unterschiedlich sein. Die Eigenschaft 6. der Definition besagt, dass ein beliebiger Klartext M, der mit einem Verschlüsselungsschlüssel aus dem Schlüsselraum verschlüsselt wurde, $E(M, K_E)$, anschließend wieder mit dem dazu passenden Entschlüssellungsschlüssel $K_D = f(K_E)$ entschlüsselt werden kann, $M = D(E(M, K_E), K_D)$ [Eck06].

Symmetrische Verfahren

Bei den symmetrischen kryptografischen Verfahren verwenden Sender und Empfänger den gleichen, geheimen Schlüssel um Nachrichten zu ver- und entschlüsseln. Das erfordert, dass sich die Kommunikationspartner über diesen gemeinsamen Schlüssel verständigen müssen. Somit hängt die Sicherheit der Kommunikation nicht nur von der Stärke der verwendeten Verfahren, sondern auch von der sicheren Aufbewahrung der Schlüssel ab.
Für eine vertrauliche Kommunikation vereinbaren die Kommunikationspartner Alice A und Bob B zuerst einen gemeinsamen, geheimen Schlüssel $K_E = K_D = K_{A,B}$. Um einen Klartext M von Alice an Bob zu versenden, verschlüsselt Alice den Text M mittels $K_{A,B}$, also $C = E(M, K_{A,B})$, und sendet C an Bob. Bob entschlüsselt C mittels $K_{A.B}$, also $M = D(C, K_{A,B}) = D(E(M, K_{A,B}), K_{A,B})$ [Eck06].
Symmetrische Verfahren teilen sich in Blockchiffren und Stromchiffren. Die Blockchiffren verschlüsseln Blöcke fester Länge in einem Durchgang und die Stromchiffren verschlüsseln in jedem Arbeitsschritt ein einzelnes Zeichen. Auf die detaillierte Betrachtung der beiden Chiffren wird hier verzichtet und auf einschlägige Literatur verwiesen [BMB+05, Eck06].

Asymmetrische Verfahren

Die symmetrischen Verfahren haben den Nachteil, dass der gemeinsame symmetrische Schlüssel sicher und geheim ausgetauscht werden muss. Jeder Dritte, dem es gelingt, in den Besitz dieses Schlüssels zu gelangen, kann die Daten zwischen Alice und Bob entschlüsseln und sich sogar als einer der beiden Teilnehmer ausgeben und Daten in dessen Namen verschlüsseln und versenden. Als Konsequenz würde das bedeuten, dass bei einem Netzwerk mit n Teilnehmern, sich jeder Teilnehmer $n-1$ Schlüssel merken muss. Dazu existieren im Netz $n(n-1)$ Schlüssel [BMB+05].
Die Kernidee der asymmetrischen Verfahren besteht nun darin, dass jeder Kommunikationspartner ein Schlüsselpaar bestehend aus einem geheimen (privaten) und einem öffentlichen Schlüssel besitzt. Der öffentliche Schlüssel

ist allen Teilnehmern bekannt zu geben, der private Schlüssel jedoch sicher zu verwalten. Der Unterschied zum symmetrischen Verfahren liegt nun darin, dass der geheime Schlüssel nicht mit den anderen Teilnehmern ausgetauscht werden muss bzw. darf. Eine Kommunikation unter Nutzung des asymmetrischen Verfahrens läuft wie folgt ab [Eck06]:
Die Kommunikationspartner Alice A und Bob B erzeugen ihre eigenen Schlüsselpaare (K_E^A, K_D^A) bzw. (K_E^B, K_D^B), wobei K_D^A und K_D^B die jeweils geheimen Schlüssel sind.
Die Schlüssel K_E^A und K_E^B werden öffentlich bekannt gegeben. Dies kann z.B. durch eine öffentliche Datenbank, über die eigene Webseite, per eMail oder über einen öffentlich zugänglichen Schlüssel-Server erfolgen.
Wenn nun Alice einen Klartext M, der für Bob bestimmt ist, verschlüsselt, so verwendet sie dazu Bobs öffentlichen Schlüssel: $E(M, K_E^B) = C$, und sendet C an Bob.
Bob entschlüsselt C mit seinem geheimen Schlüssel, also: $D(C, K_D^B) = M$

Hybride Verfahren

Bei der Verschlüsselung von größeren Datenmengen ist ein asymmetrisches Verfahren nicht sinnvoll. Denn der Aufwand zur Chiffrierung der Daten ist bei den asymmetrischen Verfahren um ein vielfaches größer als bei den symmetrischen Verfahren (siehe u.a. [Eck06, BMB+05, BSW04].
Die Vorteile der beiden Verfahren werden durch hybride Verfahren kombiniert, die sich wie folgend charakterisieren lassen [BMB+05]:

> Voraussetzung: Alice A und Bob B besitzen ihre eigenen Schlüsselpaare (K_E^A, K_D^A) bzw. (K_E^B, K_D^B), wobei K_D^A und K_D^B die jeweils geheimen Schlüssel sind. Bob möchte an Alice Daten schicken. Dazu benötigt Bob den öffentlichen Schlüssel von Alice K_E^A, den er direkt von Alice oder aus einem Verzeichnis bekommen kann.
> Bob generiert dann einen zufälligen Sitzungsschlüssel r_{AB} für ein symmetrisches Verfahren. Mit diesem Schlüssel werden später die Daten zwischen Alice und Bob ausgetauscht.
> Bob verschlüsselt den Sitzungsschlüssel mit $E(r_{AB}, K_E^A) = S$ und schickt S an Alice. Alice kann S nun mit ihrem geheimen Schlüssel entschlüsseln und erhält r_{AB} durch $D(S, K_D^A) = r_{AB}$.
> Somit besitzen sowohl Alice als auch Bob einen gemeinsamen geheimen symmetrischen Schlüssel r_{AB}, mit sie die Daten chiffrieren und austauschen können.

Bei längerer Kommunikation sollten Alice und Bob diesen Vorgang wiederholen, so dass der Sitzungsschlüssel häufiger gewechselt wird.

Hashfunktionen

Hashfunktionen (bzw. genauer *kryptografisch sichere Hashfunktionen*) berechnen sogenannte digitale Fingerabdrücke von Daten und werden mit diesen Daten zusammen verschickt oder abgespeichert. Eine kryptografische Hashfunktion H bildet eine Nachricht M beliebiger Länge auf einen Hashwert $h = H(M)$ fester Länge ab, wobei die Hashfunktion zwei wesentliche Charakteristika aufweisen soll [HK06, BMB+05]:

1. Es ist praktisch nicht möglich aus dem Hashwert h eine zugehörige Nachricht M zu berechnen (Einwegeigenschaft).

2. Es ist praktisch nicht möglich, zwei Nachrichten M_1 und M_2 zu finden, die auf den gleichen Hashwert $H(M_1) = H(M_2)$ abgebildet werden (Kollisionsresistenz).

Somit ist es möglich, die Integrität (siehe Unterabschnitt 3.1.1) der Daten zu überprüfen und nicht autorisierte Modifikationen an den Daten aufdeckbar zu machen [Eck06]. Der Ablauf einer Integritätskontrolle sieht wie folgt aus [Eck06]:

1. Der Urheber eines Dokumentes oder einer Nachricht M berechnet den Hashwert $h = H(M)$ und speichert diesen Wert zusammen mit M ab, bzw. überträgt M und $h = H(M)$ an den Empfänger.
2. Der Empfänger kontrolliert die Integrität von M, in dem der Hashwert $h' = H(M)$ berechnet wird und das Ergebnis mit h verglichen wird.
3. Falls $h = h'$, dann kann davon ausgegangen werden, dass auch $M = M'$. Damit handelt es sich also bei M' um das unmodifizierte Originaldokument.

Hashfunktionen dienen aber in der Praxis nicht nur zur Überprüfung der Integrität eines Dokumentes. Sie werden sehr häufig in Kombination mit Signaturverfahren eingesetzt, um eine eindeutige Urheberschaft nachweisen zu können [Eck06].

3.2.2 Elektronische Signaturen

Aus der Anforderung P1 aus Unterabschnitt 3.1.2 ergibt sich die Bedingung, qualifizierende digitale Signaturen zu verwenden, um die elektronische Durchführungsform der papierbasierten Form rechtlich gleichzusetzen. Eine so genannte elektronische Unterschrift ist nötig, um elektronische Dokumente zweifelsfrei einer natürlichen oder juristischen Person zuzuordnen. Grundsätzlich lassen sich die Bedingungen zur Gleichstellung einer handschriftlichen Unterschrift mit einer elektronischen Signatur anhand der Anforderungen an handschriftliche Unterschriften beschreiben [Eck06]:

1. Die Unterschrift gibt Auskunft über die Person des Unterzeichners (Identifikation).
2. Die Unterschrift bezeugt, dass das Dokument dem Aussteller vorgelegen hat und von ihm anerkannt wurde (Echtheit).
3. Die Unterschrift erklärt, dass der Inhalt des Dokumentes inhaltlich richtig und vollständig ist (Abschluss).
4. Die Notwendigkeit, dass eine Unterschrift geleistet werden muss, zeigt dem Anwender die rechtliche Bedeutung auf (Warnung).

Die Übertragung der Anforderungen auf die elektronischen Signaturen bedeutet, dass (vgl. [BMB+05])...

- ...die Signatur die Identität des Unterzeichners zweifelsfrei bestätigt (Authentizität).
- ...die Signatur eindeutig mit dem unterzeichnenden Dokument verbunden und nur mit diesem gültig ist.
- ...die Signatur nicht wiederverwendbar sein darf.
- ...das signierte Dokument nicht veränderbar sein darf bzw. nachträgliche Änderungen erkennbar sein müssen müssen (Integrität).
- ...der Unterzeichner als Urheber der Signatur nicht in der Lage sein darf, seine Signatur nach deren Erstellung abzustreiten (Nichtabstreitbarkeit/ Verbindlichkeit).

Für die elektronischen Prüfungen bedeutet dies, dass sowohl die Anforderung der Authentizität, Verbindlichkeit, Integrität als auch die rechtlichen Anforderungen der Formvorschrift (P1) und der eindeutigen Zuordnung von

Teilnehmer und Prüfung (P5) umgesetzt werden können.
Die Erstellung und Verifikation von elektronischen Signaturen basiert auf asymmetrischen Verfahren, d.h. eine Nachricht wird mit einem privaten Schlüssel verschlüsselt und kann dann mit dem öffentlichen Schlüssel des Signierers wieder entschlüsselt werden. Einige asymmetrische Verfahren wie z.B. das RSA-Verfahren können zum Verschlüsseln und auch zum Signieren eingesetzt werden, wohingegen Verfahren wie DSA (Digital Signature Algorithm) ausschließlich zum Signieren verwendet werden können.
Das Protokoll der Erstellung und Verifikation einer elektronischen Signatur lässt sich wie folgt beschreiben [Eck09]:

1. Sei K_D^A, K_E^A das Schlüsselpaar von Alice. K_D^A der Signaturschlüssel und K_E^A der öffentliche Verifikationsschlüssel.

2. Alice hinterlegt K_E^A in einer öffentlichen Datenbank oder Verzeichnis.

3. Alice signiert ein Dokument M durch Verschlüsseln mit ihrem privaten Schlüssel, $D(M, K_D^A) = SIG$ und sendet das signierte Dokument SIG an Bob.

4. Bob ruft den benötigten Verifikationsschlüssel K_E^A aus der Datenbank oder Verzeichnis ab

5. und verifiziert die Signatur $SIG, M = E(SIG, K_E^A)$.

In der Praxis wird aber nicht das ganze Dokument M mit dem geheimen Schlüssel verschlüsselt, sondern über das Dokument M wird ein Hashwert $h = H(M)$ berechnet, den dann Alice mit ihrem geheimen Schlüssel verschlüsselt. Für Bob ist sowohl M als auch $H(M)$ zugänglich. Bob verwendet den öffentlichen Schlüssel von Alice, entschlüsselt den Hashwert und berechnet seinerseits den Hashwert $h' = H(M)$. Wenn $h' = h$, dann weiß Bob, dass die Nachricht unverändert ist, und er kann sicher sein, dass die Nachricht von Alice stammt. Denn nur Alice hat mit ihrem privaten Schlüssel die Möglichkeit, M zu signieren. Im Gegenzug bedeutet das, dass Alice gegenüber Bob nicht abstreiten kann, die Nachricht unterschrieben zu haben.

Signaturgesetz

Das Signaturgesetz (SigG) und die Signaturverordnung (SigV) definieren den rechtlichen Rahmen der elektronischen Signatur in Deutschland. Der Zweck des SigG ist es, Rahmenbedingungen für elektronische Signaturen zu schaffen. Die SigV konkretisiert das Signaturgesetz vor allem in technischen Fragen.

Grundlegende Bestimmungen

Das Signaturgesetz (SigG) definiert drei verschiedene Arten von elektronischen Signaturen. Gemäß § 2 SigG sind

1. „elektronische Signaturen“ Daten in elektronischer Form, die anderen elektronischen Daten beigefügt oder logisch mit ihnen verknüpft sind und die zur Authentifizierung dienen,

2. „fortgeschrittene elektronische Signaturen“ sind elektronische Signaturen nach Nummer 1, die

   (a) ausschließlich dem Signaturschlüssel-Inhaber zugeordnet sind,

   (b) die Identifizierung des Signaturschlüssel-Inhabers ermöglichen,

   (c) mit Mitteln erzeugt werden, die der Signaturschlüssel-Inhaber unter seiner alleinigen Kontrolle halten kann, und

   (d) mit den Daten, auf die sie sich beziehen, so verknüpft sind, dass eine nachträgliche Veränderung der Daten erkannt werden kann,

3. „qualifizierte elektronische Signaturen“ elektronische Signaturen nach Nummer 2, die

   (a) auf einem zum Zeitpunkt ihrer Erzeugung gültigen qualifizierten Zertifikat beruhen und

   (b) mit einer sicheren Signaturerstellungseinheit erzeugt werden.

Demnach ist nach § 2 Nr.1 SigG bereits eine eingescannte Unterschrift eine einfache elektronische Signatur, denn sie dient als Authentifikation und wird den anderen elektronischen Daten beigefügt. Diese Signatur ist weder fälschungssicher noch muss sie fest mit den anderen Daten verknüpft sein [Roß09]. Dennoch können einfache Signaturen durchaus sicher sein. Durch die Verbindung mit Verschlüsselungstechniken kann das Sicherheitsniveau gesteigert werden. Außerdem sind die einfachen digitalen Signaturen unter bestimmten Voraussetzungen als Beweismittel geeignet und rechtsgültig [GHK+07]. Die sich aus der rechtlichen Anforderung P1 ergebenden Anforderungen der Schriftform und der Nichtabstreitbarkeit können mit der einfachen Signatur aber nicht realisiert werden.
Die Voraussetzungen, die an fortgeschrittene Signaturen nach § 2 Nr. 2 SigG gestellt werden, können nur durch asymmetrischer Kryptographie und mit einer Public-Key-Infrastruktur (PKI) oder aber das beim PGP-Verfahren eingesetzte „Web of Trust“, gewährleistet werden [Roß09,HK06]. Allerdings existieren im SigG keine Anforderungen an die verwendeten Algorithmen, an die

Zertifizierungsdiensteanbieter oder aber an die zu verwendeten Erstellungs- und Anwendungskomponenten [Roß09]. Für formfreie Rechtsgeschäfte sind fortgeschrittene elektronische Signaturen einsetzbar, die für die elektronischen Prüfungen verlangte Schriftform wird jedoch nicht erfüllt [GHK+07]. Nur die qualifizierenden digitalen Signaturen sind als Beweismittel vor Gericht und als Ersatz für die eigenhändige Unterschrift zugelassen, so dass der Gesetzgeber den Umgang damit reguliert [Eck06].

Qualifizierte elektronische Signatur

Wie aus § 3 Nr. 3 SigG ersichtlich, muss die qualifizierte Signatur die Anforderungen an die fortgeschrittene Signatur erfüllen. Zum Zeitpunkt der Erzeugung einer qualifizierenden digitalen Signatur muss diese zusätzlich noch auf einem qualifizierenden Zertifikat beruhen und mit einer sicheren Signaturerstellungseinheit erzeugt werden.

Zertifikate Ein Zertifikat ist nach § 2 Nr. SigG eine elektronische Bescheinigung, mit den Signaturprüfschlüssel einer Person zugeordnet werden und die Identität dieser Person bestätigt wird. Ein Signaturprüfschlüssel entspricht nach SigG dem öffentlichen Schlüssel eines kryptografischen Schlüsselpaares[10]. Der Signaturschlüssel ist dementsprechend nach SigG der private Schlüssel eines kryptografischen Schlüsselpaares[11].
Der Inhalt eines solchen Zertifikates wird durch den X.509 Standard[12] festgelegt, dessen allgemeine Struktur in Tabelle 3.2 festgehalten ist.
Die Zertifizierungsstelle (CA) (engl. *Certification Authority* oder *Trust Center*) ist für die Eindeutigkeit der Zertifikate verantwortlich. Eine CA verwaltet die Zertifikate, d.h. sie stellt diese aus und ruft sie ggf. wieder zurück. Bei der Ausstellung der Zertifikate verwendet die CA ihrerseits eine Signatur um die Zuordnung zwischen einem öffentlichen Schlüssel und seinem Besitzer zu beglaubigen. Die CAs sind Bestandteil von Public-Key-Infrastrukturen (PKI), die die Gesamtheit der Komponenten beschreiben, die zur Erzeugung und Verwaltung von Zertifikaten benötigt werden [Eck06]. Die Komponenten einer PKI sind (siehe Abbildung 3.1) (vgl. [HK06, Eck06, BMB+05]):

- Zertifizierungsinstanz (CA)
- Registrierungsinstanz (RA)

[10] § 2 Nr. 5 SigG
[11] § 2 Nr.4 SigG
[12] http://www.ietf.org/rfc/rfc2459.txt, geprüft am 08.01.2010

Inhalt	**Beschreibung**
version	Versionsnummer des Zertifikatformates
serialNumber	Seriennummer des Zertifikates, die in Verbindung mit issuer eindeutig ist
issuer	ID des Zertifikatausstellers
signature	verwendeter Algorithmus
validity	Gültigkeitsdauer des Zertifikates
subject	Name des Zertifikatsbesitzers
subjectPublicKeyinfo	öffentlicher Schlüssel des Zertifikatsinhabers
issuerUniqueIdentifier	erweiterte ID des Zertifikatausstellers
subjectUniqueIdentifier	erweiterte ID des Zertifikatbesitzers
extensions	Erweiterungen (z.B. *SubjectDirectoryAttributes* das auch ein Bild des Zertifikatinhabers speichern kann)

Tabelle 3.2: Inhalt eines X.509v3 Zertifikates (vgl. [Eck06, BMB+05])

- Schlüsselgenerator (KG)
- Verzeichnisdienst (DIR)
- Zeitstempeldienst (TSA)

Die Regisitrierungsinstanz (RA) (engl. *Registration Authority*) stellt die Schnittstelle zwischen Anwender und der PKI dar. Der Anwender beantragt bei der RA seine Zertifikate, wobei bei der erstmaligen Registrierung die Identität des Anwenders festgestellt werden muss, um die Korrektheit der Angabe im Zertifikat zu gewährleisten. Dazu gehört auch die Aufnahme von zusätzlichen Attributen durch die Attributquelle (AS). Anschließend werden durch den Schlüsselgenerator (KG) die Schlüsselpaare erzeugt. Die Erzeugung der Schlüssel kann durch einen externen Schlüsselgenerator erfolgen oder aber beim Einsatz von Chipkarten direkt auf der Karte. Das hat den Vorteil, dass der geheime Schlüssel die Chipkarte (Signaturerstellungseinheit) niemals verlässt. Die zur Ausstellung eines Zertifikates benötigten Informationen werden dann an die CA weitergeleitet, die dann die Zertifikate erstellt.
Die Zertifikate werden durch den Verzeichnisdienst (DIR) nachprüfbar und abrufbar gemacht. Ebenso die Sperrlisten - eine Liste widerrufener Zertifikate (CRL) (engl. *Certificate Revocation List*). Widerrufsgründe können z.B. sein [BMB+05]:

- Zertifikat wird nicht mehr benutzt

dass dieser vom Angreifer gefälschte Signaturen positiv verifiziert und die vom Signierenden erstellten korrekten Signaturen für falsch hält. Damit solche Angriffe nicht erfolgreich sind, setzt man Zertifikate ein, die insbesondere den öffentlichen Schlüssel und den Namen des Schlüsselinhabers tragen (vgl. Abbildung 19) und von einer vertrauenswürdigen Zertifizierungsinstanz signiert werden. Durch die Signatur wird eine Bindung zwischen dem öffentlichen Schlüssel und dem Schlüsselinhaber erzeugt. Neben dieser Zertifizierungsinstanz benötigt man weitere Komponenten zur Ausgabe und Verwaltung von Zertifikaten. Die Gesamtheit dieser Komponenten und die zugehörigen Prozesse bezeichnet man als Public-Key-Infrastruktur (PKI).

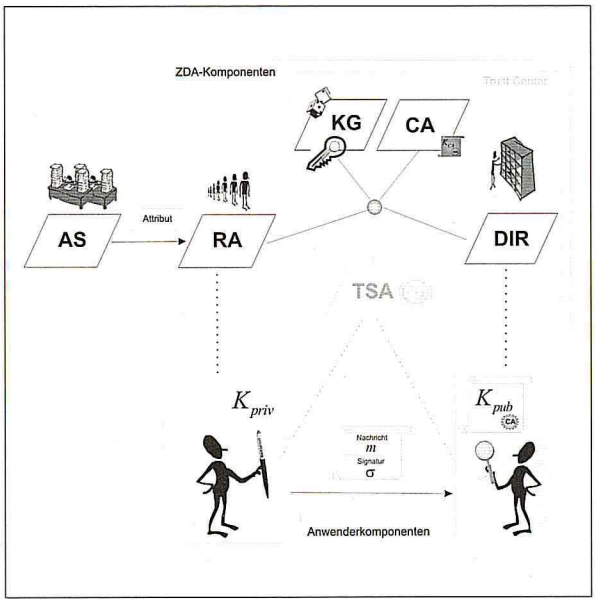

ABBILDUNG 16. Komponenten einer PKI

Abbildung 3.1: Komponenten einer PKI [HK06]

Wie in Abbildung 5 skizziert, umfasst eine solche PKI neben den Anwenderkomponenten (vgl. Abschnitt 3.2.1) auch eine Reihe von Infrastrukturkomponenten, die abgesehen von der Attributquelle (engl. Attribute Source (AS), vgl. Abschnitt 3.2.3) im Verantwortungsbereich des Zertifizierungsdiensteanbieters (ZDA) liegen. Zu diesen ZDA-Komponenten zählen

- die Registrierungsinstanz (engl. Registration Authority (RA)),
- der Schlüsselgenerator (engl. Key Generator (KG)),
- die Zertifizierungsinstanz (engl. Certification Authority (CA)),
- der Verzeichnisdienst (engl. Directory (DIR)) und möglicherweise
- der Zeitstempeldienst (engl. Time Stamping Authority (TSA))

- Privater Schlüssel ist nicht mehr nutzbar
- Privater Schlüssel wurde kompromittiert
- Schlüsselparameter sind nicht mehr adäquat (z.B. Schlüssellänge nicht mehr angemessen)

Der Zeitstempeldienst (TSA) datiert die Signaturen, so dass z.B. Mehrfachvorlagen eines signierten Dokumentes erkannt werden können. Damit elektronische Signaturen erstellt und überprüft werden können, werden eine Signaturanwendungskomponente und eine sichere Signaturerstellungseinheit (SSEE) benötigt [HK06].

Signaturanwendungskomponenten und Anwenderinfrastrukturen
Signaturanwendungskomponenten sind nach § 2 Nr. 11 SigG, Software- oder Hardwareprodukte die Daten dem Prozess der Erzeugung qualifizierter elektronischer Signaturen zuführen oder qualifizierte Signaturen bzw. qualifizierte Zertifikate prüfen und die Ergebnisse anzeigen. D.h., Editoren, Viewer und Komponenten zur Ansteuerung der Signier- und Prüffunktionen sowie Schnittstellen zu externen Diensten, die z.B. Zeitstempel und Sperrauskünfte anfordern bzw. empfangen können [Por03]
Um Daten zu signieren bzw. zu verifizieren sind weitere Komponenten nötig.

Dazu zählen neben Softwarekomponenten (z.B. Treiber) auch Hardwarekomponenten wie z.B. ein Smartcardleser, falls Smartcards als sichere Signaturerstellungseinheit verwendet werden. Die Summe der Software- und Hardwarekomponenten zum Signieren und Verifizieren wird als Anwenderinfrastruktur bezeichnet (siehe [Por03]).

Sichere Signaturerstellungseinheit (SSEE) SSEE sind nach § 2 Nr. 10 SigG, *Software- oder Hardwareeinheiten zur Speicherung und Anwendung des jeweiligen Signaturschlüssels*. Dabei muss eine SSEE auch die Anforderungen nach § 17 (Produkte für qualifizierte elektronische Signaturen) oder § 23 SigG (Ausländische elektronische Signaturen und Produkte für elektronische Signaturen) und die Vorschriften der Rechtsverordnung nach § 24 des SigG erfüllen.
Praktisch werden als SSEE Chipkarten oder USB-Tokens eingesetzt. Hierbei werden die kryptografischen Schlüssel im ROM (Read-Only-Memory) der SSEE gespeichert und können nur mit speziellen Lesegeräten ausgelesen werden [Eck06]. Der geheime Schlüssel darf dabei niemals ausgelesen werden können.
Damit der Inhaber auf seine Smartcard zugreifen kann, muss er sich über eine PIN gegenüber der Smartcard authentisieren. Je nach Klassifizierung des Smartcardlesers erfolgt die Eingabe der PIN entweder direkt am Lesegerät oder über die Eingabegeräte des angeschlossenen Gerätes (wie z.B. PC).

3.2.3 Authentifikation

Begriffsdeutung: Authentifikation ist die Überprüfung, ob jemand der ist, der er vorgibt zu sein. Der Begriff Authentifizierung hat die gleiche Bedeutung. Die Authentisierung beschreibt den Vorgang aus Sicht des Überprüften. So authentisiert sich ein Teilnehmer am Prüfungssystem, aber das Prüfungssystem (PS) authentifiziert den Teilnehmer (vgl. [Sch09c]). Bei der Teilnehmer-PS-Konstellation ist vor allem die Authentifizierung des Teilnehmers gegenüber dem PS wichtig. Bei der Konstellation Autor bzw. Prüfungsverantwortlicher (PV)-PS ist es aber auch wichtig, dass sich der PV sicher sein kann, dass es sich nicht um ein kompromittiertes PS handelt.
Grundsätzlich kann die Authentifizierung durch drei Authentifikationstechniken durchgeführt werden:

- Wissen: Das PS kontrolliert, ob der Nutzer eine bestimmte Information kennt. Beispiel hierfür sind Passwörter, One-Time-Pads (Einmalpasswörter), geheime Schlüssel etc.

- Besitz: Das PS kontrolliert, ob der Nutzer einen nicht (bzw.) schwer zu fälschenden Gegenstand besitzt wie z.B. einen Ausweis (Smartcard, eToken etc.)

- Eigenschaft: Das PS kontrolliert, ob der Nutzer ein unverwechselbares, schwer zu fälschendes persönliches Merkmal besitzt. Beispiele hier sind biometrische Merkmale wie Fingerabdruck oder Iris.

Verteilte Authentifizierung

Single-Sign-on Ein elektronisches Prüfungssystem ist im Grunde nur ein weiteres Anwendungssystem in der Hochschullandschaft. Mehrere Systeme bedeuten, dass sich die Nutzer an jedem einzelnen System authentisieren müssen. Die Authentifizierungsart der einzelnen Systeme kann dann von Passwort über Smartcard bis hin zur Biometrie reichen. Allerdings sind gerade die passwortbasierten wegen ihrer Einfachheit am weitesten verbreitet. Das Merken der vielen Passwörter ist mühsam und kann zu Sicherheitsproblemen führen, wenn Nutzer sich die Passwörter nicht merken können oder wollen und sich diese aufschreiben. Vielfach tendieren Nutzer dazu, für alle Systeme das gleiche Passwort oder aber schwache Passwörter zu verwenden (siehe dazu [BW07]).
Single-Sign-On Techniken (SSO) ermöglichen das einmalige Authentifizieren und dann den Zugriff auf mehrere Anwendungen ohne erneute Authentifizierung. SSO Techniken sind zwar sehr anwenderfreundlich aber auch sicherheitskritisch, weil Angreifer nur eine Hürde überwinden müssen, um auf alle Anwendungen zugreifen zu können [FS03]. Es existieren mehrere Ansätze zur Realisierung von SSO (siehe u.a. [Sch09c, Vog05]):

- *Lokales SSO:* Der Benutzerrechner hat einen speziellen SSO-Client installiert, der alle Logins abgreift und die Logins mit Benutzername und Passwort füllt. Der Benutzer authentisiert sich nur gegenüber dem SSO-Client.

- *Ticket-SSO:* Anstelle des lokalen SSO gibt es einen Authentifizierungsserver. Benutzer authentisieren sich gegenüber einem Authentifizierungsserver, der wiederum den direkten Zugang zu den Servern der Anwendungen vermittelt. Dabei benutzt der Authentifizierungsserver sog. Tickets, die Benutzernamen und die Zugangsberechtigung zum jeweiligen Server enthalten. Das Ticket wird vom Authentifizierungsserver digital signiert oder so verschlüsselt, dass nur der jeweilige Server es

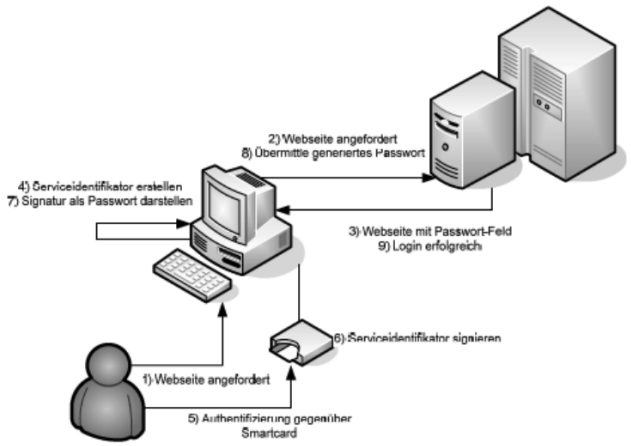

Abbildung 3.2: SSO mit Signaturen [RZ06]

lesen kann. Dies funktioniert aber nur, wenn alle Beteiligten ein standardisiertes Protokoll verwenden. *Kerberos* ist ein solcher Standard.

- *SSO mit Signaturen:* Der Einsatz von elektronischen Signaturen kann auch zur Realisierung eines SSO dienen. In [RZ06] ist eine Möglichkeit beschrieben, wie elektronische Signaturen zur Authentifizierung bei Passwortsystemen genutzt werden können (siehe Abbildung 3.2). Damit sich ein Benutzer am Prüfungssystem authentifizieren kann, berechnet ein browserbasiertes Plugin einen Dienstidentifikator unter der Verwendung der URL des Prüfungssystems und des Benutzernamens des Teilnehmers (1-4). Der Teilnehmer authentifiziert sich gegenüber der Signaturerstellungseinheit durch die Eingabe einer PIN (5). Danach wird der Dienstidenifikator mit Hilfe der Signaturerstellungseinheit signiert (6) und das Ergebnis der Signatur als Passwort kodiert (7). Das Passwort wird dann an das Prüfungssystem, das die Authentifizierung angefordert hat, übertragen (8). Das Prüfungssystem erlaubt dann den Zugriff (9).

Kerberos Kerberos ermöglicht die Authentifizierung von zwei Beteiligten (z.B. Teilnehmer und Prüfungsserver), die keinen gemeinsamen geheimen Schlüssel besitzen, wobei aber nicht auf Mittel der asymmetrischen Kryptografie zurückgegriffen wird. Es wird stattdessen mittels eines dritten bzw. vierten Protokollbeteiligten realisiert [Sch09c]. Kerberos authentisiert sowohl

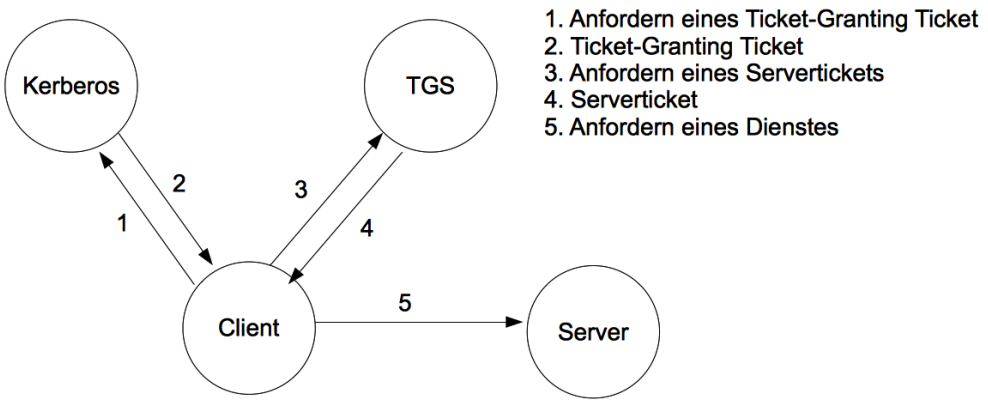

Abbildung 3.3: Authentifizierung bei Kerberos [Sch06c]

den Server gegenüber dem Client, als auch den Client gegenüber dem Server, um Man-In-The-Middle-Angriffe zu unterbinden. In Abbildung 3.3 sind die einzelnen Auhentifizierungsschritte dargestellt.
Nach der Anmeldung des Benutzers am Client fordert der Client von Kerberos ein Ticket-Granting-Ticket (TGT) für einen Ticket-Granting-Service (TGS) an. Wenn der Client nun einen bestimmten Server benutzen will, fordert er vom TGS ein Ticket für diesen Server an. Der Client präsentiert dem Server das Ticket und wenn alles in Ordnung ist, dann erhält er vom Server die Erlaubnis zum Zugriff [Sch06c]. Tickets und Authentikatoren sind zwei Arten der Legimitation, die Kerberos verwendet [Sch06c]. Ein Ticket dient dazu, die Identität des Clients sicher an den Server zu übertragen. Das Ticket enthält außerdem noch Informationen, damit der Server überprüfen kann, ob das Ticket auch wirklich für den Client ausgegeben wurde, der es gerade benutzt. Ein Kerberos-Ticket hat die folgende Form (siehe [Sch06c, Eck06]):

$$T_{C,S} = S, E((C, A, V, K_{C,S})K_S)$$

Dabei ist $T_{C,S}$ das Ticket vom Client C für die Verwendung des Servers S. Der Inhalt des Tickets besteht also aus dem Servernamen S und den mit dem geheimen Schlüssel des Servers K_S verschlüsselten Elementen: Clientname S, Netzadresse des Clients A, Beginn und Ende der Ticketgeltungsdauer V sowie den Sitzungsschlüssel $K_{C,S}$ für Client und Server.
Ein Ticket bezieht sich somit auf einen einzelnen Server und einen einzelnen Client. Der Client kann solange auf den Server zugreifen, bis das Ticket abgelaufen ist. Des Weiteren kann das Ticket bei der Übertragung weder abgehört noch geändert werden [Sch06c].

Der Authentikator wird jedesmal generiert, wenn der Client einen Dienst auf dem Server nutzen möchte. Der Authentikator hat die folgende Form:

$$A_{C,S} = E((C, T, X)K_{C,S})$$

Der Authentikator enthält den Namen des Clients (C), einen Zeitstempel (T) und optional einen weiteren Sitzungsschlüssel (X). Alles zusammen wird mit dem gemeinsamen Sitzungsschlüssel $K_{C,S}$ chiffriert. Der Authentikator kann nur einmal verwendet werden und muss je nach Bedarf vom Client neu erzeugt werden. Auf eine detaillierte Betrachtung des Kerberos-Protokolls wird in dieser Arbeit verzichtet und auf einschlägige Literatur verwiesen (siehe u.a. [Sch09c, Eck09, Sch06c]).

3.2.4 Zugriffs- und Informationskontrolle

Bell-LaPadula

Das Bell-LaPadula Modell gilt als das erste vollständig formalisierte Sicherheitsmodell. Es unterscheidet Objekte (Dateien, Verzeichnisse, etc.) und Subjekte (Benutzer, Prozesse etc.) [Eck09]. Jedem Subjekt wird eine Sicherheitsklasse (Clearence) und jedem Objekt eine Sicherheitsklassifikation (Classification) zugeordnet.
Der Zugriff auf die Objekte o durch ein Subjekt s wird durch die Regeln *no-read-up* und *no-write-down* beschrieben. Die *no-read-up* Regel besagt, dass ein Lesezugriff auf o durch s nur dann zulässig ist, wenn die Klassifikation von o kleiner oder gleich der Sicherheitsklasse von s ist (siehe Abbildung 3.4). Die *no-write-down* Regel besagt, dass ein schreibender Zugriff auf o durch s nur zulässig ist, wenn die Klassifikation von o größer oder gleich der Sicherheitsklasse von s ist (siehe Abbildung 3.4).
Das Problem bei diesem Modell ist, dass z.B. ein Subjekt auf ein höher eingestuftes Objekt schreibend zugreifen darf, aber anschließend diese Veränderung nicht mehr lesen darf. Daraus ergibt sich die Problematik, dass ein Subjekt ein Objekt beliebig verändern darf [Eck09]. Diese Verletzung der Integritätsanforderung wäre für den Einsatz bei den Prüfungen fatal.
Dennoch könnte das Bell-LaPadula Modell für die elektronischen Prüfungen interessant sein. Denn ein angemeldeter Teilnehmer könnte für eine Prüfung in eine höhere Sicherheitsstufe wie alle nicht angemeldeten Studierenden eingestuft werden und somit die Prüfungsangaben des Dozenten lesen. Durch die *no-write-down*-Regel kann dann der Teilnehmer die Lösungen schreiben, diese aber nicht mehr lesen und verändern und auch nicht anderen Studierenden zur Verfügung stellen. Für eine weitere detaillierte Betrachtung

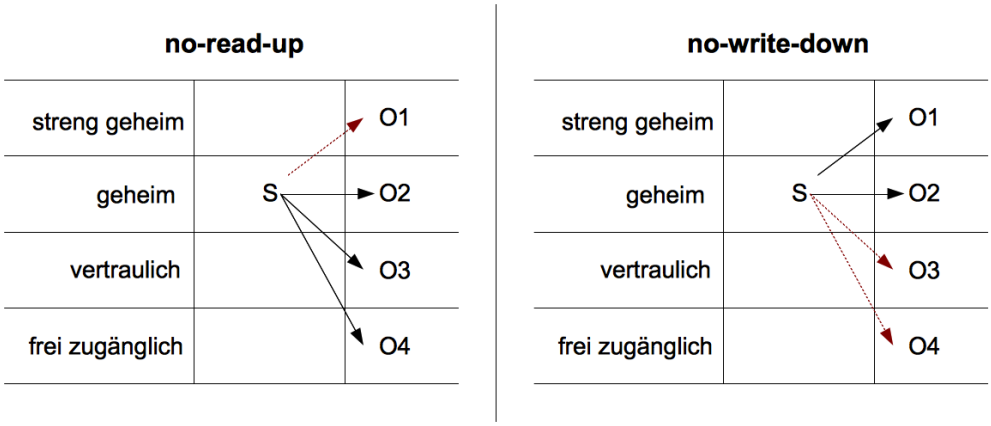

Abbildung 3.4: Bell-LaPadula Regeln (vgl. [Beu05])

des Bell-LaPadula Modell wird auf weiterführende Literatur verwiesen (siehe u.a. [Eck09, SBBH08]).

3.2.5 Vertraulichkeit

Bei der Vertraulichkeit wird zwischen Vertraulichkeit der Kommunikation und Vertraulichkeit der Daten unterschieden. Die Verfahren zur Vertraulichkeit der Daten wurden bereits in Unterabschnitt 3.2.4 erwähnt. Zur Sicherstellung der Vertraulichkeit der Kommunikation existieren u.a. die Standards SSL/TLS und Virtuelle Private Netzwerke (VPN), die nachfolgend beschrieben werden.

SSL/TLS

Secure Socket Layer (SSL) stellt seit der Version 3.0 einen de facto Internet-Standard für sichere Verbindungen dar, der von allen gängigen Internet-Browsern unterstützt wird [Eck09]. Das Protokoll HTTP über SSL wird als HTTPS[13] bezeichnet. Das SSL-Protokoll besteht im Wesentlichen aus den Teilschritten Handshake- und Record-Protokoll. Das Handshake-Protokoll authentifiziert beim Verbindungsaufbau die Kommunikationspartner und handelt die kryptografischen Parameter zwischen den Kommunikationspartnern aus. In Abbildung 3.5 sind die Schritte des Handshake-Protokolls dargestellt. Bevor eine Verbindung aufgebaut werden kann, werden die benötigten kryptografischen Parameter unverschlüsselt übertragen. Diese Informationen werden bereits mit der Hello-Nachricht vom Client an den Server übermittelt.

[13] HTTP Secure, siehe RFC 2818

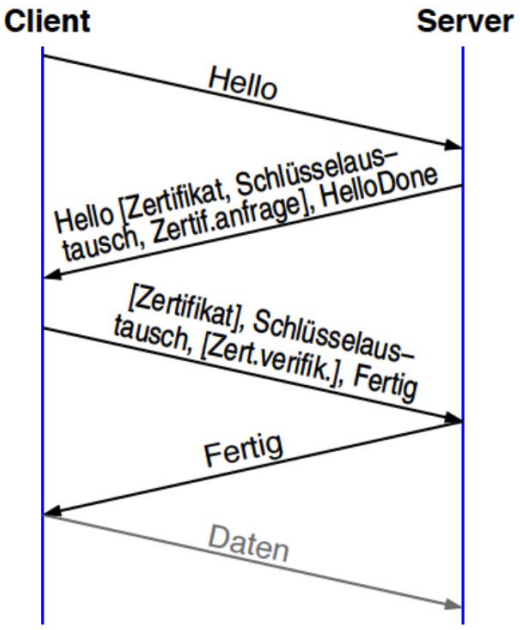

Abbildung 3.5: SSL-Handshake [SBBH08]

Dazu zählen ein Zeitstempel, eine Zufallszahl, eine SitzungsID und eine Liste mit Verschlüsselungs- und Kompressionsverfahren, die der Client unterstützt. [Eck09].
Der Server authentifiziert sich mit einer Hello-Nachricht, die die gleichen Parameter enthält wie das Client-Hello zuvor. Jedoch sind die Verschlüsselungs- und Kompressionsverfahren auf diejenigen beschränkt, die auch der Server unterstützt. Bei der Authentifizierung des Servers lässt der Server dem Client sein Zertifikat (in der Regel ein x.509 Zertifikat) zukommen. Fordert der Server auch eine Authentifikation des Clients, so schickt der Client sein Zertifikat in analoger Weise an den Server. Sind die Kommunikationsparameter ausgehandelt, können die Daten übertragen werden. Dies wird durch das Record-Protokoll geregelt.
Das Record-Protokoll dient zur Absicherung der Verbindung sowie zur Fragmentierung der Datenpakete. Die Absicherung der Verbindung erfolgt mittels Ende-zu-Ende Verschlüsselung durch einen symmetrischen Schlüssel, der zuvor im SSL-Handshake ausgehandelt wurde. Des Weiteren dient das Record-Protokoll der Komprimierung der Daten sowie der Sicherung der Integrität und Authentizität [Eck09].
Die Unterschiede zwischen SSL und Transport-Layer-Security (TLS) sind

marginal. TLS verwendet im Gegensatz zu SSL das HMAC-Verfahren zur Berechnung der MAC-Werte. Außerdem verwendet TLS ein verändertes Schlüsselerzeugungsverfahren, das robuster gegenüber Angriffen auf Hashwerte sein soll [Eck09].

VPN

Eine weitere Form der Verbindungs- und Ende-zu-Ende-Verschlüsselung kann durch virtuelle private Netzwerke (VPNs) realisiert werden. Die Kommunikation findet dabei über einen sog. Tunnel statt. Unter einem Tunnel versteht man *...die Nutzung einer Netzinfrastruktur zum Transfer von Daten in einem Netzwerk zu einem anderen* [Eck09]. In einem VPN werden die zu transportierenden Frames bzw. Pakete durch einen zusätzlichen Header gekapselt. Dieser Header ermöglicht die Kommunikation zwischen dem Sender-Gateway und dem Gateway des Empfängers. Dazu werden die Daten im Sender-Gateway verpackt und über das private Netz zum Empfänger-Gateways weitergeleitet. Hier werden sie wiederum entpackt und zum Empfänger weitergeleitet.

3.2.6 Anonymisierung / Pseudonymisierung

Ein großer Vorteil der elektronischen Prüfungen gegenüber der Papiervariante ist die Möglichkeit, Prüfungen zu bewerten ohne die Identität des Teilnehmers preisgeben zu müssen. Wie bereits in Anforderung P7 erwähnt, ist die anonyme Bewertung ein *Kann*-Kriterium, das aber zur Erfüllung des Gleichheitsgrundsatz erheblich beiträgt.
Da die Anonymisierung aber im Widerspruch zur Authentifizierung und Verbindlichkeit steht, müssen entsprechende Maßnahmen getroffen werden, die die Authentifizierung bei der Durchführung und die Anonymisierung bei der Auswertung bzw. Nachbewertung gleichermaßen erfüllen. Dies kann z.B. durch das Trennen der persönlichen Angaben von den Prüfungsangaben erfolgen. Die Verknüpfung der persönlichen Angaben und der Prüfungsangaben erfolgt über eine sog. Kennziffer (Pseudonym). Die Prüfungsdaten werden dem Korrekteur unter dieser Kennziffer präsentiert.
Eine Maßnahmen wäre, die persönlichen Angaben in der Darstellung des Korrekteurs auszublenden. Allerdings besteht wiederum bei manchen Prüfungen die Notwendigkeit, die Identität des Teilnehmers zu kennen. Denn gerade bei dem zweiten Wiederholungsversuch kann der Korrekteur bei der Auswertung etwas andere Bewertungsmaßstäbe ansetzen [ZB07]. Allerdings sei angemerkt, dass hierzu zwar die Information über den zweiten Wiederholungsversuch nötig ist, die persönlichen Daten wie der Name für den Korrekteur jedoch nicht nötig sind.

3.2.7 Ausfallsicherheit

Die Anforderung der Ausfallsicherheit bezieht sich bei den elektronischen Prüfungen nicht nur auf den Backend-Bereich (Server), sondern auch auf die Clients. Etablierte Verfahren zur Sicherstellung der Ausfallsicherheit werden unter die Rubriken Redundanz und Lastverteilung zusammengefasst. Während die Redundanz auf eine verteilte Speicherung der Daten hinausläuft, ist die Lastverteilung für die Skalierbarkeit der Anwendung zuständig. Die Redundanz schafft eine hohe Ausfallsicherheit der Anwendung, während die Lastverteilung ein skalierbares Antwortverhalten der Anwendung sicherstellt. Was für die Hochverfügbarkeit der Dienste durch das Clustering erzielt werden kann, muss bei den elektronischen Prüfungsdaten gesondert betrachtet werden. Denn die Prüfungsdaten werden während einer Prüfung durch die Teilnehmer ständig gelesen und geschrieben. Eine verteilte Speicherung der Daten auf mehrere Server hätte dann zur Folge, dass die Daten ständig repliziert werden müssten.
Eine Möglichkeit wäre, die Daten neben der Speicherung auf dem Server auch auf den Clients zu speichern. Bei einem Netzwerk- oder Serverausfall könnte dann die Prüfung lokal durchgeführt werden. Voraussetzung hier ist aber, dass sich die Prüfungsfragen und alle notwendigen Materialien (Bilder, Ton, etc.) zu Prüfungsbeginn auf dem Client befinden. Nach der lokalen Durchführung müssten dann die Daten aber auch wiederum zur Auswertung auf den Server übertragen und vollständig vom Client entfernt werden. In [Bre08] ist ein solches Konzept dargestellt (siehe auch Abschnitt 7.4). Hierbei werden die Daten lokal auf einem USB-Stick zwischengespeichert. Nach der Prüfung werden die USB-Sticks eingesammelt und die Daten könnten dann auf den Server übertragen werden, wo sie ausgewertet werden.

3.2.8 Betrugssicherheit

Maßnahmen zur Sicherstellung der Betrugssicherheit sind nicht nur durch das Prüfungssystem realisierbar. Obgleich z.B. das zufällige Anordnen der Prüfungsfragen bzw. -antworten eine funktionale Eigenschaft ist, ist die Betrugssicherheit vor allem ein administrativer Aspekt.
Damit die Teilnehmer während der Prüfung keine unerlaubten Hilfsmittel einsetzen, müssen auch bei elektronischen Prüfungen Aufsichten eingesetzt und die auch bei Papierklausuren üblichen Maßnahmen getroffen werden. Aber bei den webbasierten Prüfungssystemen muss eine sichere Prüfungsumgebung geschaffen werden. Dies ist durch den Einsatz eines Secure-Browsers möglich. Ein Werkzeug aus dieser Kategorie ist das Open-Source Tool *Safe*

Exam Browser (SEB)[14]. Der *Safe Exam Browser* wurde speziell für die sichere Durchführung webbasierter Prüfungen entwickelt.
Der SEB basiert auf einem adaptierten Firefox-Browser, der den Rechner in einen Vollbildmodus versetzt und nur zu der vorgegebenen Startadresse des Prüfungsservers eine Verbindung herstellt. Des Weiteren werden alle Systembefehle (wie z.B. CTRL-ALT-DEL) gesperrt und auch der Vollbildmodus kann nur durch eine vorher eingestellte Tastenkombination aufgehoben werden. Somit sind während der Prüfungen auch keine unerlaubten Programme ausführbar bzw. nur die Programme, die dazu vorher freigegeben wurden [Sch06b, VS09].
Zum Zeitpunkt dieser Arbeit ist der SEB nur für Windows-Systeme erhältlich.

3.2.9 Nachvollziehbarkeit und Archivierung

Wie detailliert die Dokumentation des Prüfungsverlaufes zu sein hat, ist nicht abschließend zu klären. Einige Prüfungssysteme registrieren jede Mausbewegung des Teilnehmers und speichern diese ab [Ree08b]. Andere speichern neben der Start- und Endzeit nur die IP-Adressen mit ab. Grundsätzlich gilt zwar zur Nachvollziehbarkeit „je mehr desto besser“, aber dies steht im direkten Widerspruch zu der Datenschutzanforderung der Datensparsamkeit. Wie viel Dokumentation ist gerade noch nötig, um die Nachvollziehbarkeit zweifelsfrei aufzuzeigen?
Eine Möglichkeit wäre, die Protokolldaten, wie Zeitpunkt und getätigte Angabe, auf einem Protokollierungsserver zu speichern. Dies hätte zur Folge, dass die Prüfungsdaten und die Protokolldaten physisch getrennt aufbewahrt werden. Auf die Protokollserver haben der Prüfungsverantwortliche und die Korrekteure keinen Zugriff. Nach der Auswertung können die Protokolldaten dann mit allen relevanten Daten zusammen archiviert werden.
Zu den relevanten Daten gehören:

- Veranstaltungsdaten
- Prüfungsdatum und -uhrzeit
- organisatorische Aspekte (Labore, etc.)
- Korrekteure, Aufsichten
- Prüfungsangaben (inkl. aller Materialien)

[14] http://www.safeexambrowser.org, zuletzt aufgerufen am 17.01.10

- Musterlösungen zu den Prüfungsangaben
- Prüfungsteilnehmer (Name, Matrikelnr., Studiengang)
- Prüfungslösungen
- Protokolldaten

Die Daten müssen in einem sog. unveränderlichen Dateiformat wie PDF/A[15] vorliegen. PDF/A kann im Gegensatz zu PDF keine Skripte oder verborgene Texte beinhalten. Damit die Daten auch rechtsverbindlich archiviert werden können, müssen diese in signierter Form abgespeichert werden. Hierbei muss das Prinzip „What you see is what you sign!“ gelten. Denn nicht immer kann der Signierer genau erkennen, was er signiert (siehe [Por03]). In der Regel werden die Prüfungssysteme von einem zentralen Informations- und Mediendienst der Hochschule betrieben, der dann als Betreiber seinerseits die Pflicht hat, die Daten zu archivieren.
Eine Idee wäre die Signierung der Daten durch den Prüfungsverantwortlichen und ihre anschließende Archivierung in einem zentralen Archivsystem. Archivsysteme bieten die Technologie zur Langfristaufbewahrung von elektronischen Dokumenten (mehr als 10 Jahre) und sichern die Daten gegen unberechtigten Zugriff und Veränderbarkeit ab. Oftmals bieten die Archivsysteme auch Schnittstellen an, um einen direkten Zugriff aus den Anwendungen heraus zu gewährleisten. Bei elektronischen Prüfungen ist jedoch ein ständiger Zugriff (lesend und schreibend) auf die archivierten Prüfungsdaten nicht nötig. Somit könnte sich eine mögliche Anbindung eines Prüfungssystems an ein Archivsystem auf einen schreibenden (bzw. erzeugenden) Zugriff beschränken.

3.2.10 Zusammenfassung

Die dargestellten Maßnahmen decken zwar in ihrer Gesamtheit alle Anforderungen an die Prüfungen ab. Jedoch ist z.B. Single-Sign-On (SSO) mit Signaturen einfach umzusetzen, aber eine feingranulare Informationskontrolle ist damit nicht möglich.
Kerberos ist ein reiner Authentifikations- und Schlüsselaustauschdienst. Eine Zugriffskontrolle oder die Vergabe von Rechten ist damit nicht möglich [Eck09]. Aber gerade die Zugriffskontrolle ist für die Prüfungen von absoluter Wichtigkeit.

[15] http://www.pdfa.org/, aufgerufen am 26.01.10

Die bestehenden Systeme (siehe Abschnitt 2.4) verwenden rollenbasierte Zugriffskontrollen, wobei der Administrator Zugriff auf alle Daten des Prüfungssystems hat, was eine verschlüsselte Speicherung der Daten erfordert.
Die Konsequenzen der Maßnahmen orientieren sich vor allem an der Notwendigkeit einer Public-Key-Infrastruktur für die qualifizierenden digitalen Signaturen. Ebenso ist die Einführung einer sicheren Signaturerstellungseinheit (SSEE) wie z.B. eines elektronischen Studierendenausweises notwendig.

Kapitel 4

State-of-the-Art existierender Sicherheitskonzepte

In diesem Kapitel werden zuerst die Sicherheitskonzepte existierender Prüfungssysteme untersucht. Anschließend werden mehrere Sicherheitskonzepte für elektronische Prüfungen untersucht, die weitestgehend unabhängig vom verwendeten Prüfungssystem sind.
Aufgrund der Anforderung, dass qualifizierende digitale Signaturen zu verwenden sind und damit auch die Verwendung von Signaturkarten, werden abschließend existierende Signaturkartenkonzepte betrachtet und auf eine mögliche Anpassung an die Prüfungen hin bewertet.

4.1 Sicherheitskonzepte bestehender Prüfungssysteme

Die in Abschnitt 2.4 dargestellten Softwaresysteme für elektronische Prüfungen an Hochschulen werden im Folgenden anhand der in Kapitel 3 erarbeiteten Sicherheitsanforderungen hin untersucht.

4.1.1 Umsetzung der Anforderungen

Elektronische Signaturen (P1) werden derzeit von keinem der Werkzeuge eingesetzt. Deshalb wird die Nichtabstreitbarkeit (P8) der Prüfungsangaben mit Ausdruck und Unterschrift realisiert. Die Studenten führen die Prüfung online durch und ihre Angaben werden in eine Datenbank geschrieben. Nach der Prüfungsdurchführung werden die Angaben der Studenten ausgedruckt, anschließend unterschreibt jeder Student seinen Ausdruck. Die Prüfung wird elektronisch ausgewertet und die elektronischen Daten werden gelöscht. Nur

die Papier-Ausdrucke werden archiviert. Aber selbst der Ausdruck der gemachten Angaben stellt rechtlich gesehen eine Grauzone dar. Denn laut der Zivilprozessordnung (ZPO) ist ein Computerausdruck nur dann eine Urkunde, wenn der Ausdruck im unmittelbaren Herrschaftsbereich des Ausstellers liegt [HHS99]. Wenn nun die Angaben über einen Netzwerkdrucker ausgedruckt werden, stellt sich schon die Frage, ob der Netzwerkdrucker, auf dem auch die anderen Studenten ihre Angaben ausdrucken, im unmittelbaren Herrschaftsbereich des Studenten liegt. Ganz zu schweigen von den möglichen technischen und organisatorischen Problemen wie z.B. Druckerausfall und Koordinierung der Unterschriften. Außerdem stellt diese Variante einen kompletten Medienbruch dar.
Deshalb geht die Universität Mainz mit einem angepassten ilias3 System einen anderen Weg. Hierbei erhalten die Prüflinge vor der Durchführung ein Beiblatt das allgemeine Benutzungs- und Datenschutzanweisungen enthält. Auf diesem Beiblatt müssen die Studierenden ihre persönlichen Daten eintragen. Zusätzlich müssen sie nach der Abgabe der Prüfung dann einen Hashwert, der über ihre Prüfungsangaben generiert wurde, abschreiben und auf dem Beiblatt eintragen. Am Schluss bestätigen sie die Richtigkeit der Angaben mit der Unterschrift auf dem Beiblatt [Wet08a].
Es stellt sich jedoch die Frage, was passiert, wenn ein Prüfling den angezeigten Hashwert nicht korrekt abschreibt. Auch hier werden wiederum verschiedene Medien verwendet, um die Nichtabstreitbarkeit zu gewährleisten.

Die Verfügbarkeit (P4) ist bei der Durchführungsphase ein absolutes Muss-Kriterium. Die Hersteller geben oftmals keine Garantien für die Anwendung, denn zurecht verweisen sie auf den Einsatzzweck und -umfang der Anwendung, die von Hochschule zu Hochschule unterschiedlich sein kann. Die Hochschulen setzen deshalb eigene Konzepte wie Clustering, Redundanz etc. ein, um die Verfügbarkeit sicherzustellen.
Das Clustered OLAT Konzept ist in Abbildung 4.1 dargestellt. OLAT setzt dabei den Open-Source Messaging-Broker Apache Active MQ 5.2 [1] ein. Bei LPLUS werden durch Lasttests im Vorfeld derDurchführung Beispielprüfungen simultan an mehreren Stationen durchgeführt. Denn die unterschiedlichen Aufgabenformate innerhalb eines Tests erzeugen unterschiedliche Netzlasten. So erzeugen aufwendige Grafiken oder Multimediadokumente eine wesentlich höhere Netzlast als reine Textfragen.

Alle Systeme verfügen über die gängigen Authentifizierungsmöglichkeiten wie Benutzername und Passwort. Es besteht auch die Möglichkeit, bei den

[1] http://activemq.apache.org/, aufgerufen am 24.01.10

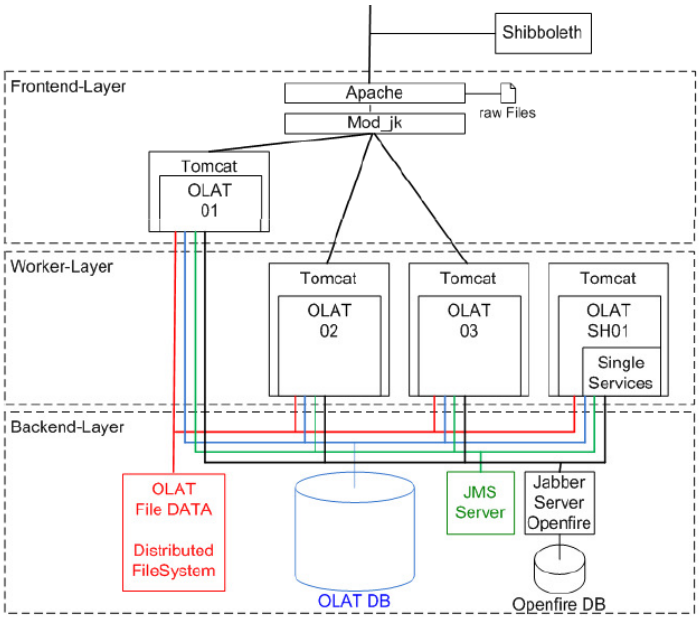

Abbildung 4.1: OLAT Cluster-Modell [BG09]

meisten Systemen den Login über einen bestehenden zentralen Authentifizierungs- bzw. Verzeichnisdienst zu gewährleisten. Dazu zählen vor allem LDAP, Kerberos und CAS-Logins. Dadurch ist der Zugang über die Hochschulkennungen möglich. OLAT verwendet zusätzlich das Shibboleth-Protokoll[2], das ein webbasiertes Single-Sign-On realisiert.
Die Universität Bremen setzt beim Einsatz von LPLUS auf die Vergabe von temporären Zugängen zum Prüfungssystem. Und zwar nur zur Prüfungsdurchführung werden für jeden Teilnehmer Zugänge erzeugt und dann in ein spezielles Dateiaustauschportal (nicht Bestandteil von LPLUS) bereitgestellt. Jeder Teilnehmer hat über seinen Hochschulaccount einen Zugang für das Portal und kann sich dann kurz vor der Prüfung seine Zugangsdaten abrufen.
Alle webbasierten Systeme setzen auf HTTPS als Übertragungsprotokoll, was bedeutet, dass auch die Integrität der Daten während der Übertragung sichergestellt ist (siehe Abschnitt 3.2.5). Bei der Vertraulichkeit und Integrität der gespeicherten Prüfungsdaten bzw. -fragen ist dies weniger eindeutig, weil z.B. bei LPLUS der Administrator die Möglichkeit hat, die Zugriffsrechte beliebig an Benutzer zu vergeben. So können Klausurdaten zum einen durch den Administrator und zum anderen durch nicht berechtigte Benutzer einge-

[2]http://shibboleth.internet2.edu/, 21.01.2010

sehen oder manipuliert werden (siehe [Bod09, BÖ8b]).
Außerdem ist der Einsatz des Secure-Browsers Konzeptes bei allen Systemen verbreitet. LPLUS verwendet aber ebenso eine eigene Entwicklung, wie auch Questionmark mit *Questionmark Secure*.

Zur Archivierung der Prüfungsdaten bieten die Systeme die Möglichkeit, die Daten aus der Datenbank zu exportieren. Die Dokumentation der Prüfung erfolgt z.B. bei LPLUS durch Ausdrucke, die die Beantwortung und Auswertung nachvollziehbar machen sollen.

Bei den Prüfungssystemen werden keine speziellen Datenschutzkonzepte eingesetzt. Alle Systeme setzen auf ein Rollenkonzept, in dem allerdings ein Missbrauch des Administratorzugangs oftmals den Zugriff auf sämtliche im Prüfungssystem vorgehaltenen Daten ermöglicht [BÖ8b].

4.1.2 Zusammenfassung

Keines der Systeme setzt elektronische Signaturen ein, um die Verbindlichkeit der Prüfungsangaben bzw -lösungen sicherzustellen. Die Verbindlichkeit wird statt dessen lückenhaft durch Medienbrüche umgesetzt. Bei der Verfügbarkeit sind jedoch gerade die Ansätze bei OLAT sehr interessant. Das Clustered OLAT Konzept verteilt dabei die Anfragen auf mehrere Nodes.

4.2 Stufenmodell Universität Hannover

In [St06] wird ein Sicherheits-Stufenmodell für Online-Prüfungsverfahren beschrieben, das verschiedene Prüfungsverfahren klassifiziert (Selbsteinschätzung, Studien- und Prüfungsvorbereitung, Auswahl und Vorauswahl, Leistungsnachweis) und auf drei Sicherheitsstufen (Keine zusätzliche Sicherheit, Daten-/Kommunikationssicherheit, Sichere Prüfungsumgebung) verteilt (siehe Abbildung 4.2). Das Stufenmodell orientiert sich an der Tatsache, dass summative Prüfungen einen sehr viel höheren Aufwand an organisatorischen, rechtlichen und technischen Aspekten haben als z.B. Tests zur Selbsteinschätzung.

Dieses Modell findet in der Durchführung von Online-Prüfungen an der Uni Hannover mit dem ilias3-System Anwendung. Unter die Stufe 1 fallen alle Prüfungsarten, die keiner zusätzlichen Sicherheit bedürfen, wie z.B. anonyme, unbeaufsichtigte Selbsttests. Unter der zusätzlichen Sicherheit werden

Stufenmodell für Online-Prüfungsverfahren		Prüfungsart				Bewertung Korrektur					Didaktik				Zeit Ort	
		Selbsteinschätzung	Studien- und Prüfungsvorbereitung	Auswahl und Vorauswahl	Leistungsnachweis	Online-Bewertung	Online-Teilbewertung	Konventionelle Bewertung	Automatische Korrektur	Manuelle (Nach)Korrektur	Formative Prüfung	Summative Prüfung	Offenes Szenario	Geschlossenes Szenario	Zeitunabhängig	Ortsunabhängig
Stufe 1	Keine zusätzliche Sicherheit	x	x			x			x		x		x		x	x
Stufe 2	Daten-, Kommunikationssicherheit		x	x			x	x	x	x	x	x	x		x	x
Stufe 3	Sichere Prüfungsumgebung			x	x	x			x	x		x		x		

Abbildung 4.2: Stufenmodell der Universität Hannover [Ste06]

hier erweiterte Sicherheitsmaßnahmen verstanden, die nicht bereits vom verwendeten Prüfungssystem angeboten werden. Die Stufe 2 ist für Vorauswahlverfahren oder fachspezifische Kenntnisprüfungen vorgesehen. Dabei erfolgt hier eine Online-Teilbewertung, die in Ergänzung mit konventionellen Verfahren wie z.B. Gespräche, Zeugnisse etc. die abschließende Bewertung ergibt. Stufe 3 ist für summative Prüfungen als alleinige verbindliche Bewertungsgrundlage vorgesehen.
Das Stufenmodell bildet zwar die verschiedenen Sicherheitsanforderungen der einzelnen Prüfungsarten ab, allerdings ist in diesem Modell die Verwendung von elektronischen Signaturen nicht vorgesehen.

4.3 Framework für Online-Lernerfolgskontrollen

In [Gra03] wird ein Testframework für Online-Lernerfolgskontrollen dargestellt. Hierbei werden Prüfungen in einzelne Prüfungsprozesse untergliedert. Das Testframework verwendet eine RMI Middleware, bei der die Testmaterialien vollständig und verschlüsselt vom Server auf die Clients übertragen werden. Die Materialien werden dann im Browser-Cache zwischengespeichert, bevor sie mittels Java-Script in das Test-Applet geladen werden. Durch die-

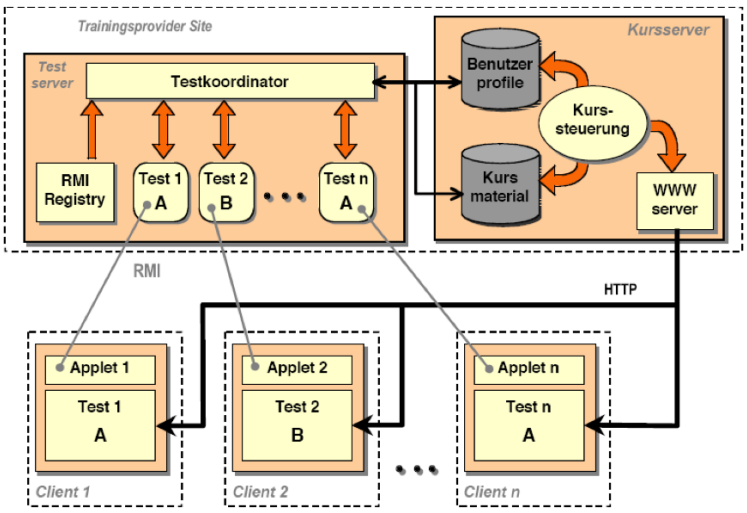

Abbildung 4.3: Frameworkarchitektur nach [Gra03]

sen Vorgang wird verhindert, dass während der Klausur Änderungen an den Fragen vorgenommen werden. Denn die gesamte Klausur befindet sich bereits auf den Clients. In Abbildung 4.3 sind die Komponenten des Frameworks zu erkennen: Den Testserver und die Testapplets. Der Testserver verwaltet über den Testkoordinator die Tests und die Anbindung an das Lernsystem. Außerdem wird über den Testkoordinator für jeden aktiven Test ein eigenes Objekt erzeugt.

Der Testserver verwendet ein abstraktes Interface um serverseitig auf Testmaterial oder das Benutzerprofil seitens des Lernsystems zuzugreifen. Die RMI-Registry wird benötigt, um jedes Testobjekt über RMI mit einem zugehörigen Testapplet im Browser der Clients zu verbinden. Die elementaren Sicherheitsmechanismen werden durch Testservices realisiert:

- Sichere und garantierte Materialauslieferung
- Sichere Testauswertung
- Zuverlässige Zeitüberwachung.

Um eine sichere und garantierte Materialauslieferung zu gewährleisten, wird auf der Clientseite das Test-Applet geladen, das eine RMI-Verbindung zum Testserver aufbaut. Auf dem Testserver wird ein Testobjekt abgelegt, das die Verwaltung der Tests übernimmt. Beim Teststart fordert das Testobjekt die relevanten Materialien vom Lernsystem an und liefert es an das Test-Applet.

Das Testmaterial wird verschlüsselt im Browsercache des Clients abgelegt. Die Verschlüsselung wird durch den Testserver durchgeführt, wobei dieser für jeden Test einen zufälligen symmetrischen Schlüssel generiert und das Testmaterial damit verschlüsselt. Das Applet fordert erst unmittelbar vor Testbeginn den Schlüssel vom Testserver an. Dann wird das Testmaterial auf dem Client entschlüsselt und in einem neuen Frame angezeigt. Dieses Verfahren gewährleistet somit, dass alles Testmaterial garantiert zu Testbeginn zur Verfügung steht und dass beim Laden des Materials keine Verzögerungen erfolgen. Jedes Testereignis wird an das serverseitige Testobjekt weitergeleitet. Dabei protokolliert das Testobjekt alle Testereignisse in einer Aktionsliste wodurch der gesamte Testverlauf nachvollziehbar ist.
Graf beschreibt in seiner Arbeit nicht nur die Sicherheitsaspekte für elektronische Prüfungen, sondern vor allem die Sicherheitsmechanismen für eine Lernumgebung. Durch diesen „Rundumschlag“ werden zwar einige prüfungsrechtliche Anforderungen umgesetzt ($P6$ (Betrugssicherheit und Dokumentation) und Teile von $P4$ (Betriebssicherheit)), die Authentifizierung und die Verbindlichkeit durch qualifizierende Signaturen werden aber durch dieses Framework nicht sichergestellt.

4.4 Secure Interactive Online eXam (SIOUX)

Das Projekt SIOUX wurde 2008 an der ETH Zürich ins Leben gerufen, ausgehend von der Tatsache, dass bestehende Prüfungssysteme sich nur bedingt für die Durchführung von summativen Prüfungen eignen, weil sie nicht alle Sicherheitsanforderungen umsetzen:

> „*Diesen Systemen fehlt eine digitale Signierung der Resultate, um nachträgliche Manipulation systematisch zu erkennen.*“[3]

SIOUX wurde aber nicht nur als Sicherheitskonzept entwickelt, sondern als vollständiges Prüfungssystem. Es deckt alle Prüfungsphasen von der Fragenpool-Erstellung über die Durchführung, Korrektur und Auswertung ab und kann auch für formative Prüfungen (z.B. Lernstandserhebungen) und Evaluationen genutzt werden. In dieser Arbeit wird sich auf die Beschreibung des Sicherheitskonzeptes beschränkt. Für die Beschreibung der funktionalen Eigenschaften wird auf die Projektwebseite[4] verwiesen.

[3] http://www.cta.ethz.ch/computerbased_assessment/sioux/ziele, aufgerufen am 18.606.2010

[4] http://www.cta.ethz.ch/computerbased_assessment/sioux, aufgerufen am 18.606.2010

In Abbildung 4.4[5] ist das Sicherheitskonzept dargestellt. Der Ablauf der sicheren Prüfungsbereitstellung kann wie folgt beschrieben werden:

1. Alle Studierenden beziehen ein persönliches Schlüsselpaar vom Certificate Authority (CA) Server und speichern es auf dem elektronischen Studierendenausweis (Legi-Card).

2. Die Studierenden melden sich mit ihrem privaten Schlüssel am SIOUX Client an.

3. Der SIOUX Client überträgt den öffentlichen Schlüssel an den SIOUX Server.

4. Der SIOUX Server überprüft die Gültigkeit des öffentlichen Schlüssels, stellt die Prüfung zusammen und verschlüsselt diese mit dem öffentlichen Schlüssel des Studierenden.

5. Die Prüfung wird auf dem SIOUX Client mit dem privaten Schlüssel geöffnet und gestartet.

6. Alle Prüfungslösungen der Studierenden werden wiederum mit ihrem privaten Schlüssel verschlüsselt und an den Server geschickt. Diese werden dort persistent gespeichert.

Kritisch ist die Tatsache, dass die Prüfung entweder unverschlüsselt auf dem Server liegt oder entschlüsselt werden muss, um sie dann wieder mit den einzelnen öffentlichen Schlüssel der Studierenden zu verschlüsseln. Dabei stellt sich ebenfalls die Frage, wie lange der Vorgang der Ent- und Verschlüsselung dauert, wenn sich viele Studierende gleichzeitig anmelden.
Bei der Übertragung der Prüfungslösungen der Studierenden auf den Server werden dann die Lösungen der Studierenden mit ihrem privaten Schlüssel verschlüsselt und dann zum Server übertragen. Damit wird die Verbindlichkeit der einzelnen Prüfungslösungen sichergestellt, aber ob auch die Gesamtheit der Prüfungslösungen eines Studierenden signiert wird, ist nicht klar. Ebenfalls unklar bleibt, wie die Signierung während der Prüfung stattfindet: Müssen die Studierenden zur Signierung jeder Lösung eine PIN zur Authentifizierung eingeben?
Trotz der offenen Fragen, ist der Ansatz des SIOUX-Konzeptes interessant. Denn auch hier war die Ausgangssituation, dass eine rechtlich sichere Prüfung nur durch den Einsatz von digitalen Signaturen durchführbar ist. Allerdings wurde dazu gleich ein komplett neues Prüfungssystem entwickelt (siehe [Hei08]).

[5] http://www.cta.ethz.ch/computerbased_assessment/sioux/security

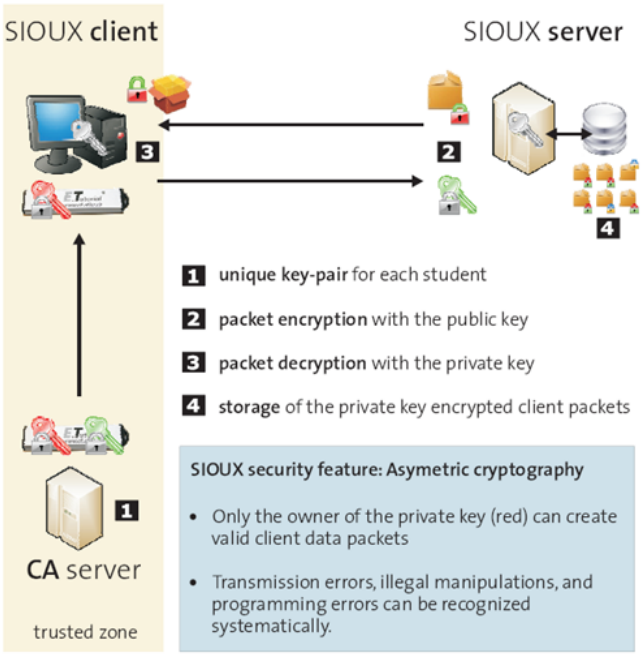

Abbildung 4.4: Sicherheitskonzept SIOUX

4.5 Proxy-Server für transparente, digitale Signierung von E-Learning Inhalten (proXsi)

proXsi ist ein Proxy-Server, der das digitale Signieren von e-Learning Inhalten in Learning Management Systemen (LMS) ermöglicht. Dabei werden die Daten vom LMS signiert und die Signatur von den eingehenden Daten überprüft.
Dazu wurde an der Universität Siegen im Rahmen einer Projektgruppe ein HTTP-basierter Proxy-Server entwickelt, der für verschiedene LMSe verwendet werden kann [HNP+09]. Dabei wurde vor allem auf die einfache Anwendung des *proXsi* geachtet. Der Signier- und Verifikationsvorgang läuft dabei automatisch und für den Benutzer völlig transparent ab.
Die Signatur basiert auf einer qualifizierten digitalen Signatur, wobei in der Umsetzung aber auf eine fortgeschrittene elektronische Signatur gesetzt wurde, um den Proxy-Server auch ohne Public-Key Infrastruktur bzw. entsprechenden Trägermedien zu betreiben. Jedoch ist bei der Verwendung von Signaturen der Zugriff auf den geheimen Schlüssel des Benutzers nötig. In der Regel ist der Schlüssel mit einem Passwort geschützt. Aufgrund der Einfachheit der Benutzung des Proxys, wird der geheime Schlüssel im Speicher

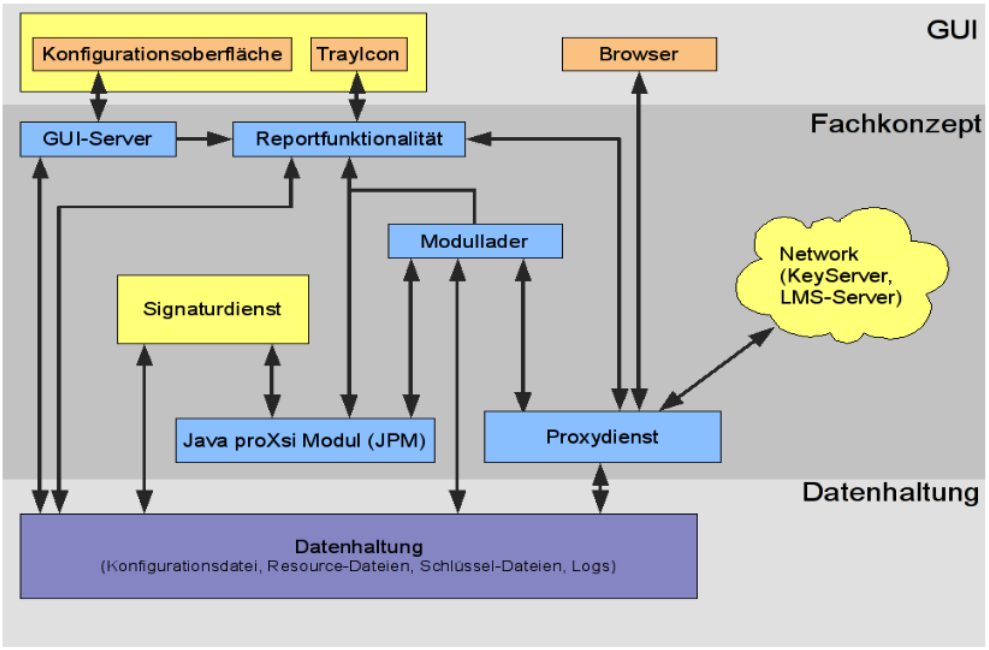

Abbildung 4.5: Struktur von proXsi [HNP+09]

gehalten um eine ständige Eingabe des Passwortes für den Zugriff auf den geheimen Schlüssel zu vermeiden [EvSS07].
Entwickelt wurde der Proxy-Server mit Hilfe der java.net Bibliotheken. Durch die Verwendung von dynamischen Java-Modulen kann der Proxy um verschiedene Module erweitert werden. Dabei sind weitere Learning-Management Systeme anzubinden, wobei jedes LMS ein eigenes Modul verwendet.
Die Proxy-Software wird lokal auf dem Client gestartet. Ein beliebiger Browser kommuniziert dann über den Proxy mit dem LMS. Die Konfiguration des Proxy-Servers kann über eine HTML-Oberfläche auf einem bestimmten Port (55556) aufgerufen werden. Die „Drei-Schichten-Architektur“ ist in Abbildung 4.5 dargestellt.
Ausgehende Nachrichten werden automatisch durch den Signaturdienst signiert und eingehende Nachrichten mit Hilfe des öffentlichen Schlüssels verifiziert. Durch die Verifikation können der Sender der Nachricht und die Unveränderbarkeit der Nachricht festgestellt werden. Die Implementierung des Signaturdienstes basiert auf dem PGP-Standard und dem Hash-Verfahren SHA265.
Der Proxydienst erkennt die Ziel-Adresse des LMS-Servers unter der Voraussetzung, dass die LMS-Angaben und das entsprechende Modul korrekt

definiert worden sind. Nur die Anfragen und Antworten zwischen den beiden Teilnehmern werden betrachtet. Andere Daten werden ungehindert weitergeleitet und durch den Proxy nicht weiter bearbeitet.
Der Proxy wird als freie Software mit offenem Quellcode vertrieben und auf der Projekt-Webseite zur Verfügung gestellt[6]. Auf die Verschlüsselung wurde verzichtet, jedoch ist eine Anpassung an den HTTPS-Standard vorgesehen. Allerdings ist die Speicherung des geheimen Schlüssels im Arbeitsspeicher oder auf der Festplatte kein sicheres Verfahren.

4.6 Signaturkartenkonzepte

Nachfolgend werden aktuelle Signaturkartenkonzepte betrachtet und auf ihre mögliche Verwendung für die elektronischen Prüfungen hin diskutiert.

4.6.1 Elektronische Studierendenkarte

Die elektronischen Studierendenkarten (eSK) sind in Deutschland bereits an vielen Hochschulen verbreitet. Dies zeigt eine Liste von Chipkartenprojekten deutscher Hochschulen auf den Infoseiten zu eSK der Ruhr-Universität Bochum[7], die allerdings seit 2006 nicht mehr aktualisiert wurde.
Allerdings sind nicht alle eSKs mit einer Signaturfunktionalität ausgestattet. Die eSK der Justus-Liebig Universität Gießen (JLU-Chipkarte) stellte zum Zeitpunkt dieser Arbeit das Non-Plus-Ultra der eSKs dar.

JLU-Chipkarte

Die JLU-Chipkarte verfügt über die folgenden Funktionen[8]:

- Lichtbildausweis
- Semesterticket für den ÖPNV
- Bibliotheksausweis für die Universitätsbibliothek
- Bezahlfunktion für Dienste des Studentenwerkes (Mensa, Kaffeeautomaten, Waschmaschinen in Studentenheimen, Kopierer)
- Verschlüsselung und Signierung von E-Mails

[6] http://www.die.informatik.uni-siegen.de/pgproxy, aufgerufen am 26.01.2010
[7] http://www.ruhr-uni-bochum.de/dezernat6/chipkarte/, aufgerufen am 15.04.2010
[8] http://www.uni-giessen.de/uni/chipkarte/, aufgerufen am 15.04.2010

- Rechtsverbindliche Anmeldung zu Prüfungen und Veranstaltungen
- sicherer Zugang zu personalisierten Webdiensten (zum Beispiel Lernplattform, Benutzerdatenbank)
- Zugangskontrolle für begrenzten Parkraum und sensible Bereiche.

Diese Multifunktionalität dieser Karte wird auch durch einen RFID-Chip komplettiert, der für die bargeldlose Bezahlfunktionen und für die Zugangskontrollen zu Parkplätzen oder Labore genutzt wird. In dem kontaktlosen RFID-Chip werden dabei nur die eindeutige Ausweisnummer, eine elektronische Geldbörse, ein Abrechnungsspeicher (Zähler) für Kopierdienste, der Inhaberstatus (Studierender, Beschäftigter, etc.) und die Gültigkeitsdauer des Ausweises gespeichert also keine personenbezogenen Daten.
Im kontaktgebundenen Kryptochip werden gespeichert:

- Vorname, Nachname
- Geheimer Schlüssel
- Zertifikat (inkl. öffentlicher Schlüssel)
- PIN (persönliche Geheimzahl, gehasht)
- PUK (Personal unblock key zum Entsperren, gehasht)
- Kyptografische Kennzahl zur Erzeugung von Einmalpasswörtern.

Die JLU Gießen bietet ihren Studierenden mit Smartcard-Lesern ausgestattete Labore an, um die Funktionalitäten der JLU-Chipkarte voll nutzen zu können. Dennoch existieren auch Missbrauchsmöglichkeiten besonders im Bereich des RFID-Chips. Denn die Funktion des RFID-Chips ist aufgrund praktischer Erwägungen nicht mit einer PIN geschützt[9]. Daher ist es für jeden, der in Besitz einer JLU-Chipkarte ist möglich, den Geldbetrag der elektronischen Geldbörse zu verbrauchen. Dies ist durch die Geldkartenfunktionalität der Chipkarte gegeben. Des Weiteren ist der Zugang zu Räumen, etc. möglich und auch der Ausdruck von Bescheinigungen. Daher empfiehlt das Hochschul-Rechenzentrum der JLU die eSK mit der gleichen Sorgfalt wie EC- oder Kreditkarten zu verwahren und zu benutzen.
Gerade die Möglichkeiten des Missbrauchs und auch die Ungewissheit der Anwender, was mit Ihren Daten geschieht, führt oftmals dazu, dass gerade

[9]http://www.uni-giessen.de/uni/chipkarte/missbrauch.html, aufgerufen am 15.04.2010

von Seiten der Studierendenschaft eine ablehnende bzw. skeptische Haltung gegenüber einer solchen eSK besteht[10].

4.6.2 Elektronischer Personalausweis

Die Einführung des elektronischen Personalausweises (ePA) wird nach dem Beschluss des Bundeskabinetts vom 23.7.2008 für November 2010 angestrebt. Ab dann soll der ePA flächendeckend in Deutschland eingeführt werden.
Im Vorfeld wurde der Ausweis u.a. auch im Rahmen eines Campusprojekts an der TU Darmstadt in Verbindung mit dem Fraunhofer Institut für Sichere Informationstechnologie getestet[11].
Der Ausweis wird mit einem ISO 14443-konformen RFID-Chip ausgestattet und ist für Personen ab dem 24. Lebensjahr 10 Jahre und für Personen unter 24 Jahre 6 Jahre gültig. Auf dem Pass werden bestimmte Merkmale wie Lichtbild, Größe, Augenfarbe und Alter optisch aufgedruckt. Diese Daten werden aber ebenfalls auch auf dem Chip der ePA gespeichert. Ob auch der Fingerabdruck des Besitzers auf dem Chip abgelegt werden soll, bestimmt der Besitzer bei Ausstellung des ePA selbst.
Der ePA verfügt über eine eID-Funktion zum Nachweis der digitalen Identität und die Verwendung von qualifizierenden digitalen Signaturen nach dem deutschen Signaturgesetz (SigG). Die Verwendung des qualifizierenden digitalen Signatur ist jedoch optional und der Besitzer entscheidet, ob er ein Zertifikat nachladen möchte oder nicht (siehe [Eck09]).
Die eID-Funktionalität liefert einen digitalen Identitätsnachweis des Ausweisinhabers, so dass dieser sich beim Online-Shopping, Online-Banking, etc. sicher identifizieren kann. Dazu werden die folgenden Daten auf dem RFID-Chip des Ausweises abgelegt [Eck09]:

- Vorname, Nachname
- Doktorgrad
- Geburtsdatum und -ort, sowie eine Auskunft, ob ein bestimmtes Alter über- bzw. unterschritten ist
- Anschrift, sowie eine Funktion, die überprüft, ob der Wohnort mit einem bestimmten Wohnort übereinstimmt

[10] http://astamuenster.wordpress.com/2009/12/07/chips?-bitte-nur-aus-der-tute-von-der-studicard-und-ihren-scheinbar-unbegrenzten-moglichkeiten/, aufgerufen am 15.04.2010

[11] https://www.bsi.bund.de/DE/Themen/ElektronischeAusweise/Projekte/-projekteCampus/CampusPilot_node.html, aufgerufen am 10.04.2010

- Dokumentenart
- ausstellendes Land
- Funktion zur Abfrage der Gültigkeit des ePA.

Allerdings bestimmt der Besitzer bei der Beantragung selbst, welche Datenfelder zum Auslesen verfügbar sein sollen und welche nicht. Eine detaillierte Beschreibung der eID-Funkionalität und des dabei verwendeten PACE (Password Authenticated Connection Establishment) Protokolls findet sich in [Eck09].
Besonders interessant bei dem ePA ist, dass dieser nicht an internationale Mindeststandards gebunden ist und damit mehr Möglichkeiten, bestehen Zusatzanwendungen zu verwenden. So sind die ePAs in anderen Ländern bereits Wirklichkeit und sie zeigen, wie z.B. in Malaysia, wie multifunktional ein Einsatz sein kann. Denn in Malaysia wird der ePA (MyKad) als Bankkarte, Bezahlkarte, Führerschein und für die digitale Signatur benutzt [Sch09a]. Interessant ist ebenfalls, dass in einigen Staaten der ePA auch zusätzlich als Krankenversichertenkarte dient.

4.6.3 Elektronische Gesundheitskarte

Die elektronische Gesundheitskarte (eGK) wird aufgrund des Gesetzes zur Modernisierung der gesetzlichen Krankenversicherung eingeführt. Durch die Nutzung einer Chipkarte sollen viele Anwendungsfälle der medizinischen Versorgung, wie die elektronische Übertragung von Rezepten, papierlos ablaufen. Um diesen Austausch zu gewährleisten, muss die eGK auf einer leistungsfähigen Infrastruktur basieren, damit die Patientendaten sicher transportiert werden können. Die bestehende Infrastruktur der Primärsysteme in Krankenhäusern, Arztpraxen und Apotheken können und sollen beibehalten werden, so dass die Anwendungen dieser Primärsysteme nur erweitert werden, ansonsten aber wie bisher genutzt werden können.
Die Infrastruktur muss hoch verfügbar und performant sein und dazu noch sicherstellen, dass ein Zugriff auf Daten nur mit der Einwilligung des Versicherten erfolgen kann („Patient bleibt Herr seiner Daten“) [Cau06]. Auch der Zugriff seitens des Leistungserbringers (Arzt, Apotheker, etc.) kann nur in Verbindung mit einem chipkartenbasierten Heilberufeausweis erfolgen. Der Versicherte kann für die Daten rollenbasierte (z.B. alle Apotheken) oder sogar auf Einzelpersonen bezogene Rechte vergeben.
Basis dieser Infrastruktur sind qualifizierte digitale Signaturen, die auf einem gültigen Zertifikat beruhen (siehe Abschnitt 3.2.2). Die Signaturen stellen die Verbindlichkeit (Nichtabstreitbarkeit) der Kommunikation, die Authentizität

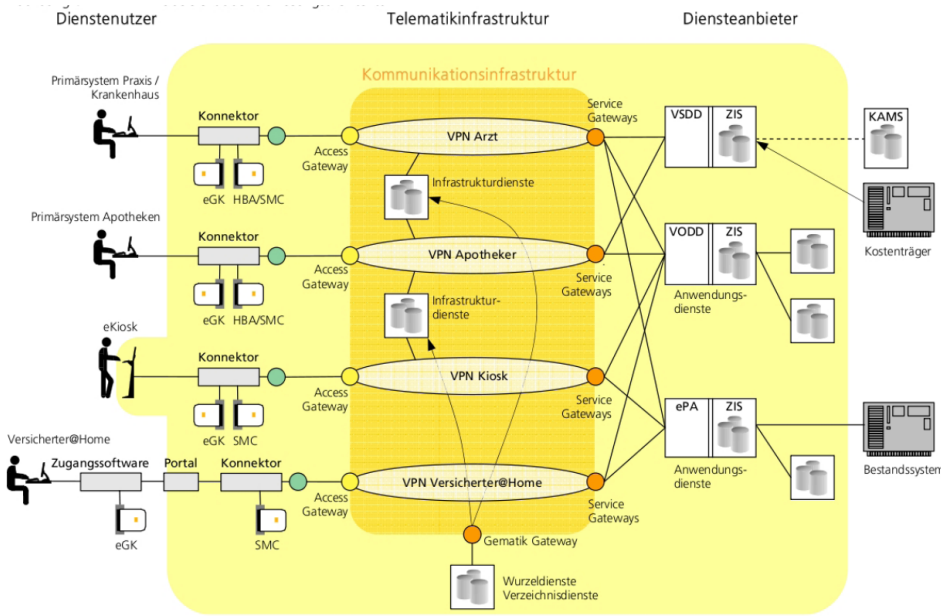

Abbildung 4.6: Lösungsarchitektur eGK [Fra05]

der Anwender und die Integrität der Daten sicher.
In Abbildung 4.6 ist die Lösungsarchitektur der eGK dargestellt. Die zentralen Komponenten, die auch für elektronische Prüfungen von Interesse sind, sind die Konnektoren, die Access/Service-Gateways und die Zugangs- und Integrationsschicht (ZIS). Die Konnektoren mit angeschlossener VPN-Box dienen dazu, die Primärsysteme an die Kommunikationsinfrastruktur anzubinden. Dazu nimmt die VPN-Box Kontakt zu einem Access Gateway auf, das wiederum anhand der Zertifikate der Kommunikationsteilnehmer entscheidet, welches VPN benutzt werden soll. Die VPN enden dann in den jeweiligen Service Gateways, welche die Berechtigungen zu den entsprechenden Diensten kontrollieren. Die Dienste wiederum greifen auf die Datenbanken über die Zugangs- und Integrationsschicht zu.

Die Zugangs- und Integrationsschicht (ZIS) ist die Schnittstelle zwischen dem Informatiksystem und den Datenspeichern. Durch sie wird der Zugriff auf die Daten transparent gestaltet, so dass es aussieht, als ob im Hintergrund ein gemeinsames virtuelles Dateisystem existieren würde, obwohl die Daten auf vielen Servern und sogar auf der eGK gespeichert sein können.
Auf die Daten auf den Servern dürfen nur autorisierte Personen zugreifen,

was durch ein Rechtemanagement sichergestellt wird. Aufgrund der Sensibilität der Daten und der Struktur des Informatiksystems sind die Daten auch vor den Betreibern und Administratoren der Datenserver geschützt. Dazu gehört auch, dass es nicht möglich ist, den Datensatz einer bestimmten Person zuzuordnen. Um Daten vor einem Einblick zu schützen, werden diese verschlüsselt. Da bei einer größeren Datenmenge eine asymmetrische Verschlüsselung zu rechenintensiv ist, werden die Daten hybrid verschlüsselt. Der symmetrische Schlüssel, mit dem die Daten verschlüsselt wurden, wird mit dem öffentlichen Schlüssel der eGK verknüpft. Dies kann nur durch den zugehörigen privaten Schlüssel rückgängig gemacht werden, wozu der Versicherte seine Erlaubnis mit der Eingabe seiner persönlichen PIN gibt.
Um an die Datensätze zu gelangen, wird dem Leistungserbringer (Arzt, Apotheker, etc.) ein Ticket für diesen Datensatz ausgestellt. Um ein Ticket zu erstellen, muss die eGK anwesend sein, weil nur in dieser der benötigte geheime Schlüssel des Versicherten gespeichert ist, womit der Schlüssel für das Datenobjekt gewonnen werden kann. Die Einwilligung zum Erstellen eines Tickets gibt der Versicherte dadurch, dass er sich mit seiner PIN authentifiziert. Zum Erstellen eines Tickets ist es nicht nötig, dass der Heilberufeausweis anwesend ist.
Jedes Objekt, ob Verzeichnis oder Datensatz ist einem sog. tnode zugeordnet. Ein tnode ist eine Metabeschreibung, die die Zugriffsrechte für das Objekt beinhaltet (siehe Abbildung 4.7). Außerdem gibt der tnode an, wo das Objekt gespeichert ist. Der lokalisierte Datensatz wird dann im verschlüsselten Zustand an den Konnektor gesendet, wo dann die Entschlüsselung erfolgt. Die ZIS hat weder die Berechtigung, noch die benötigten Algorithmen, um Datensätze zu ver- oder entschlüsseln.
Aufgrund der Verteiltheit der Daten benötigt man Mechanismen zur Lokalisierung der Daten. Jedes Datenobjekt bekommt einen eindeutigen Identifizierer und wird einem oder mehreren Verzeichnissen zugewiesen. Dies geschieht über die Speicherung einer Liste von Verzeichnis-IDs in dem Datenobjekt. Die ZIS muss andererseits auch wissen, auf welchem Server die Objekte gespeichert sind. Daher werden in einem Verzeichnisobjekt Adressen von Serverprozessen, die das referenzierte Objekt behandeln, hinterlegt. Da ein Verzeichnis keine Hinweise auf eine Person enthält und auch nicht direkt auf die Datenobjekte verweist, werden die Verzeichnisobjekte nicht verschlüsselt.
Jeder Versicherte hat genau ein Wurzelverzeichnis, dem eine ID, die mit der Krankenversichertennummer übereinstimmt, zugeordnet wird. Dieses Wurzelverzeichnis kann über einen speziellen Dienst der ZIS gefunden werden (Query-Dienst).

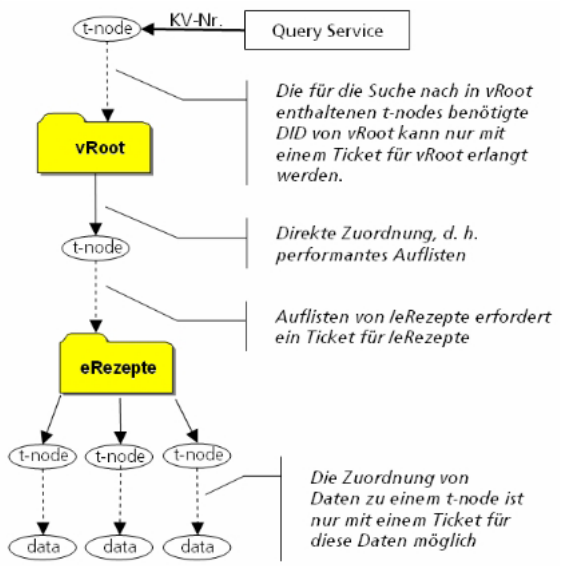

Abbildung 4.7: Hierarchische Verzeichnisstruktur der eGK [Fra05]

4.7 Zusammenfassung

Die existierenden Prüfungssysteme setzen nur Teile der prüfungsrechtlichen und datenschutzrechtlichen Anforderungen um. Aufgrund der Tatsache, dass einige Systeme nicht nur für die Durchführung von Prüfungen an Hochschulen konzipiert wurden (z.B. Perception oder LPLUS), setzen diese Werkzeuge nicht alle Sicherheitsanforderungen um. Aber diese Werkzeuge sind bereits in der deutschen Hochschullandschaft weit verbreitet. Deshalb ist ein Sicherheitskonzept nötig, das alle Anforderungen umsetzt aber dennoch die Anpassungen an die Prüfungssysteme auf ein Minimum beschränkt.
Das SIOUX-Konzept der ETH Zürich (siehe Abschnitt 4.4) basiert sogar auf einem neu entwickelten Prüfungssystem, u.a. aufgrund der nicht zufriedenstellenden Sicherheitsumsetzungen der bestehenden Prüfungssysteme.
Grundsätzlich ist für die Realisierung eines Sicherheitskonzeptes eine Public-Key Infrastruktur absolut notwendig. Nur durch den Einsatz der qualifizierenden, digitalen Signaturen sind rechtssichere Prüfungen möglich. Daraus ergibt sich die Konsequenz der Verwendung einer sicheren Signatur-Erstellungseinheit (SSEE). Eine SSEE muss dabei über Signaturfunktionen verfügen, weshalb auch verschiedene Signaturkartenkonzepte betrachtet wurden.
Die elektronischen Studierendenausweise sind mittlerweile sehr verbreitet. gerade die multifunktionale Anwendung einer solchen Karte bieten viele Vor-

teile für die Studierenden. Dennoch sind es gerade diese Funktionsvielfalt, die oftmals Anlass für Proteste seitens der Studierenden sind. Wie am Beispiel der JLU-Gießen gezeigt, sind auch für die rechtssichere Durchführung von elektronischen Prüfungen alle Voraussetzungen erfüllt.
Bei der Authentifizierung bei papierbasierten Prüfungen wird der Studierendenausweis mit einem Lichtbildausweis (z.B. Personalausweis) zur Überprüfung herangezogen. Die Einführung des elektronischen Personalausweises bietet somit auch für die Prüfungen interessante Perspektiven, denn jeder Bundesbürger wird in Zukunft mit diesem Ausweis ausgestattet. Somit wäre eine Einführung eines elektronischen Studierendenausweises - aus Sicht der Prüfungen - nicht zwingend notwendig, denn der elektronische Personalausweis könnte auch als Signaturkarte für die Prüfungen verwendet werden. Denjenigen die über keinen elektronischen Personalausweis verfügen oder aber die Signaturfunktionalität nicht wünschen, müssten aber dann entsprechende Alternativrealisierungen angeboten werden.
Die Anforderungen der eGK-Lösungsarchitektur kann im Großen und Ganzen auf die Anforderungen der elektronischen Prüfungen transferiert werden. In [Ber08] wurde die grundsätzliche Machbarkeit einer Umsetzung für die Prüfungen bereits diskutiert. Auch wenn die Telematik-Infrastruktur zu umfangreich ist, so ist aber die Zugangs- und Integrationsschicht und ganz speziell das virtuelle, ticketbasierte Dateisystem, auf die Prüfungen zu transferieren [Ber08]. Entscheidend ist, dass die Anpassungen an den bestehenden Prüfungssystemen gering bleiben. Genau dies war auch bei der Einführung der eGK eines der wichtigsten Ziele. Denn hier war eine der wichtigsten Anforderungen, dass die existierenden Primärsysteme in den Praxen, Apotheken und Krankenhäuser nicht bzw. nur minimal verändert werden.
Zusammenfassend lässt sich also sagen, dass das Lösungskonzept der eGK die Anforderungen an elektronische Prüfungen im Bezug auf die Verbindlichkeit (durch die digitalen Signaturen) und den Datenschutz (durch das ticketbasierte, virtuelle Dateisystem) vollständig abdeckt und gleichzeitig in eine bestehende Systemlandschaft integriert werden kann. Da das Lösungskonzept der elektronischen Gesundheitskarte (eGK) jedoch aus einer sehr umfangreichen Informations-, Kommunikations- und Sicherheitsinfrastruktur besteht und eine derartige Telematikinfrastruktur nur sehr bedingt für den Hochschuleinsatz tauglich ist, ist eine vollständige Transformation der Methoden nicht sinnvoll, sehr wohl aber die Anpassung auf den Problemkontext der elektronischen Prüfungen.

Kapitel 5

Virtuelles, ticketbasiertes Dateisystem

In diesem Kapitel erfolgt die Anpassung des virtuellen, ticketbasierten Dateisystems der eGK auf die elektronischen Prüfungen an Hochschulen. Dazu wird in Abschnitt 5.1 begründet, warum eine Anpassung überhaupt sinnvoll ist. In Abschnitt 5.2 wird dann das virtuelle, ticketbasierte Dateisystem (vtD) der eGK analysiert und in Abschnitt 5.3 dann auf die elektronischen Prüfungen adaptiert.
Zu erwähnen ist, dass es sich bei der analysierten Fassung des Lösungskonzeptes der eGK um die erste Fassung der Spezifikation Version 1.0 vom 14. März 2005 handelt [Fra05].

5.1 Begründung für die Adaption des Lösungskonzeptes

Durch die Notwendigkeit der qualifizierenden digitalen Signaturen wurden in Abschnitt 4.6 existierende Signaturkartenkonzepte betrachtet. In diesem Kapitel werden die allgemeinen Sicherheitsanforderungen und die Anforderungen an das Berechtigungskonzept der eGK mit den Anforderungen an die elektronischen Prüfungen im Detail verglichen. Das Ergebnis ist, dass sich die Sicherheitsanforderungen der eGK mit denen der Prüfungen vergleichen lassen. Das Berechtigungskonzept der eGK kann somit auf die Prüfungen übertragen werden [HW08].
Das Lösungskonzept der elektronischen Gesundheitskarte (eGK) wurde in [Ber08] analysiert und auf eine mögliche Verwendung für die elektronischen Prüfungen hin betrachtet.

> „*Zwischen elektronischen Prüfungen und der elektronischen Gesundheitskarte können viele Parallelen gezogen werden. Die Lösungsarchitektur kann nicht ganz übernommen werden, da die zugrunde liegende Anwenderzahl weit auseinander liegt. Jedoch ist das ticketbasierte Konzept der Zugangs- und Integrationsschicht sehr gut zu übernehmen.*“ [Ber08]

5.1.1 Allgemeine Sicherheitsanforderungen

Die in [BB04] definierten grundlegenden Sicherheitsanforderungen im Gesundheitswesen decken sich mit denen der in Kapitel Abschnitt 3.1 definierten Anforderungen an die Prüfungen. Die Anforderungen an die Sicherheit und an den Datenschutz wurden im Rahmen einer Veröffentlichung der Datenschutzbeauftragten der Länder und des Bundes zum Thema *Datenschutz in der Telemedizin* bereits im Jahr 2002 formuliert (siehe [BWB+02]):

Authentizität (Zurechenbarkeit) Die *Authentizität* der erhobenen Daten muss gewährleistet sein.

> „*Der Urheber von patientenbezogenen bzw. der Verantwortliche für patientenbezogene Daten sowie der Auslöser eines Verarbeitungsvorganges bzw. der Verantwortliche eines Verarbeitungsvorganges muss jederzeit eindeutig feststellbar sein [...] Medizinische Dokumente, die ihren Urheber bzw. Verantwortlichen nicht erkennen lassen, sind als Grundlage für Behandlungen und Begutachtungen ungeeignet.*grqq [BWB+02].

Diese Anforderungen ist genauso auch auf die Prüfungen anzuwenden. Die Authentizität der Prüfungsteilnehmer ist notwendig, um die Prüfungslösungen genau diesem Teilnehmer zuordnen zu können ($\rightarrow P_5$). Ebenso aber auch die Prüfungsangaben und die Prüfungsbewertung, die dem Dozenten bzw. Korrekteur eindeutig zuzuordnen ist.

Nutzungsfestlegung

> „*Medizinische Datenverabeitungssysteme müssen es ermöglichen, für jedes patientenbezogene Dokument den Nutzerkreis sowie abgestufte Nutzungsrechte festzulegen und Nutzungsausschlüsse zu definieren.*“ [BWB+02]

Bei den Prüfungen möchte der Dozent, dass auch nur die Studierenden die Prüfungsdaten einsehen und damit die Prüfung durchführen können, die ordnungsgemäß zur Prüfung angemeldet sind ($\rightarrow P_5$). Im Gegenzug möchte der

Studierende wissen, welche Personen Zugang zu seinen Prüfungsdaten haben ($\rightarrow D_1$).

Vertraulichkeit Bei der eGK spielt die Vertraulichkeit der Daten vor allem im Hinblick auf die Einhaltung der ärztliche Schweigepflicht eine wichtige Rolle. Denn diese in der ärztlichen Berufsordnung und dem Strafgesetzbuch[1] verankerte Bestimmung soll das Vertrauensverhältnis zwischen Arzt und Patient gewährleisten [BWB+02]. Der Arzt muss die ihm anvertrauten Daten vertraulich behandeln und nur Befugte dürfen Kenntnis personenbezogener Daten erlangen. Bei der verteilten Architektur der eGK muss natürlich auch die vertrauliche Übermittlung der Daten gelten.
Die Vertraulichkeit der Prüfungsdaten ist selbstredend von entscheidender Wichtigkeit. Dies gilt aber auch vor allem für die vertrauliche Datenhaltung im Vorfeld einer Prüfung. Werden Prüfungsdaten vor der eigentlichen Prüfungsdurchführung bekannt, würde das den Anforderungen der Betrugssicherheit ($\rightarrow P_6$), des Gleichheitsgrundsatzes ($\rightarrow P_7$) sowie des der Datenschutzanforderung ($\rightarrow D_1$).

Integrität Die Echtheit, Vollständigkeit und Korrektheit der medizinischen Daten ist für eine erfolgreiche Behandlung absolut notwenig.

> „*Eine Verfälschung oder Unvollständigkeit der Daten kann zu falschen medizinischen Entscheidungen mit u.U. lebensbedrohenden Folgen für den Patienten führen, verbunden mit rechtlichen Konsequenzen für den Mediziner.*“ [BB04]

Die Integrität aller Prüfungsdaten ist absolute Voraussetzung für eine sichere Prüfung. Eine Verfälschung oder Unvollständigkeit hat bei den Prüfungsdaten sicherlich keine lebensbedrohenden Folgen für den Studierenden, jedoch könnte es Auswirkungen auf die weitere berufliche Zukunft nehmen. Eine gewollte und vorsätzliche Verletzung der Datenintegrität, um ein nicht gültiges Prüfungsergebnis zu erzeugen, ist ebenfalls auszuschließen ($\rightarrow P_6$).

Nicht-Abstreitbarkeit und Datenübermittlung Als Voraussetzung für die Revisionsfähigkeit gewährleistet die Nicht-Abstreitbarkeit, dass ein Arzt, der ein patientenbezogenes Dokument erstellt hat, dieses nicht abstreiten kann. Der Patient, bzw. auch ein anderer Leistungserbringer (wie z.B. Arzt, Apotheker), muss wiederum sicher sein, dass das Dokument von der erwarteten Person erstellt wurde.

[1] § 203 Strafgesetzbuch (StGB) Abs. 1

Die Nicht-Abstreitbarkeit von patientenbezogenen Daten ist genauso auf die Prüfungsdaten anzuwenden ($\rightarrow P_8$). Denn auch hier darf weder der Dozent noch der Prüfungsteilnehmer die Daten abstreiten, die er getätigt hat.

Revisionsfähigkeit Die Verarbeitung der Daten muss lückenlos nachvollzogen werden können. Außerdem muss festgestellt werden können, welche patientenbezogenen Daten von wem auf welche Weise verarbeitet wurden. Eine lückenlose Dokumentation der Behandlung ist wichtig, damit nachvollziehbar ist, wer welche Diagnose gestellt und welche Therapie verordnet hat. Die Authentizität ist eine notwendige Voraussetzung für die Gewährleistung der Revisionsfähigkeit.
Für die Prüfungsdaten gelten die gleichen Anforderungen: Die Dokumentation des Prüfungsablaufes ist notwendig, um im Streitfall das Vergehen etc. begründen und nachweisen zu können ($\rightarrow P_6$). Die Revisionssicherheit bei den Prüfungen verlangt auch, dass die Dokumentation archiviert wird ($\rightarrow P_9$). Wichtig dabei ist dann wie bei den medizinischen Daten auch, dass die Datenschutzanforderungen des Erlaubnisvorbehaltes ($\rightarrow D_1$) und der Zweckbindung und Erforderlichkeit ($\rightarrow D_2$) erfüllt sind.

Rechtssicherheit Jeder Verarbeitungsvorgang und dessen Ergebnisse sind durch den Verursachenden und Verantwortlichen beweiskräftig nachweisbar zu machen. Wenn die Rechtssicherheit nicht gegeben ist, dann können eventuelle Schadensersatzansprüche durch den Patienten nicht geltend gemacht werden. Außerdem können Mediziner u.U. die Korrektheit ihres Handelns nicht nachweisen. Die Gewährleistung der Revisionsfähigkeit ist eine notwendige Voraussetzung der Rechtssicherheit, aber die Revisionsfähigkeit gewährleistet noch nicht die beweiskräftige Überprüfbarkeit von Verarbeitungsvorgängen in gerichtlichen Verfahren [BWB+02].
Grundsätzlich sind für die Rechtssicherheit gesetzliche Vorgaben zu erstellen und einzuhalten ($\rightarrow P_1$, $\rightarrow P_2$ und $\rightarrow P_3$). Die Umsetzungen der Vorgaben muss durch standardisierte Verfahren gewährleistet sein.

Validität Die Validität der personenbezogenen Daten muss gewährleistet sein. D.h., die Daten müssen in der für den Nutzungszweck angemessenen Qualität verarbeitet werden. Bei der eGK betrifft dies vor allem Bilddateien (Röntgenbilder, etc.), bei denen es auf Farbindex und Bildauflösungen ankommt. Die Integrität der Daten ist nur ein Teil der Anforderung. Denn die Bilddateien können zwar vollständig und unversehrt sein, aber die Darstellungsqualität und auch Aktualität können für die medizinische Verwendung unzureichend sein.

Bei den Prüfungen sind die Anforderungen der Validität in der Forderung nach Gleichheit zusammengefasst und betreffen vor allem unterschiedliche hardware- oder softwaretechnische Ausstattungen der Prüfungsrechner ($\rightarrow P_7$).

Verfügbarkeit Eine zeitgerechte Verfügbarkeit der Daten ist für eine erfolgreiche Behandlung essentiell. Stehen bestimmte Daten nicht oder nur verzögert zur Verfügung, kann eine Handlungsunfähigkeit dazu führen, dass - im schlimmsten Fall - das Leben des Patienten gefährdet ist. Damit einher geht natürlich auch eine rechtliche Konsequenz für den Mediziner. Unter die Verfügbarkeit fällt auch die ordnungsgemäße Funktionsweise der zur Verarbeitung erforderlichen Komponenten des IT-Systems [BWB+02].
In der gleichen Weise ist die Verfügbarkeit des Prüfungssystems selbstredend ein absolutes MUSS-Kriterium ($\rightarrow P_4$).

5.1.2 Anforderungen und Vorgaben an das Berechtigungskonzept der eGK

In diesem Abschnitt werden zuerst die Anforderungen und Vorgaben an das Berechtigungskonzept bzgl. der Anwendungen der eGK beschrieben, die über die in Unterabschnitt 5.1.1 definierten allgemeinen Anforderungen hinausgehen. Im Anschluss werden die Parallelen zu den Prüfungen aufgezeigt und warum eine Adaption sinnvoll und möglich ist.
Die Anwendungen, die mit der elektronischen Gesundheitskarte verbunden sind, werden in freiwillige und verpflichtende Anwendungen unterschieden. Die wichtigste verpflichtende Anwendung ist das elektronische Rezept (eRezept). Die freiwilligen Anwendungen sind u.a. [CWF+06]:

- *Elektronische Patientenakte*, die Informationen zu Untersuchungen, Diagnosen und Therapien verwaltet.
- *Arzneimitteldokumentation*, die die Medikamentenhistorie eines Versicherten speichert, um z.B. Wechselwirkungen zwischen Medikamenten zu vermeiden.
- *Notfalldatensatz*, der im Notfall wichtige Informationen z.B. über Allergien enthält.

Die Anforderungen und Vorgaben an das Berechtigungskonzept der elektronischen Gesundheitskarte (eGK) sind in §291a Sozialgesetzbuch (SGB V) spezifiziert. Darin ist für jede Anwendung festgelegt, welche Anwender auf welche Daten unter welchen Voraussetzungen zugreifen dürfen. In der Regel

erfordert ein Zugriff auf die Daten die eGK des Patienten und einen Heilberufeausweis (HBA) eines Heilberufers (Arzt, Apotheker, Psychotherapeuten etc.) [Cau06]. Dies gilt vor allem für Verordnungen (Rezepte).
Dabei ist aber klar definiert, dass Befunde, Diagnosen oder Behandlungsberichte sowie Impfungen nur durch Autorisierung der Versicherten zugreifbar sein dürfen. In §291a SGB V ist auch geregelt, dass der Zugriff auf ärztliche Verordnungen nicht nur mit einem Heilberufeausweis sondern auch mit einem entsprechenden Berufsausweis erfolgen kann, wenn dieser Ausweis qualifizierende digitale Signaturen verwendet[2].
Wichtig ist, dass sowohl das Identifizieren von Daten, der Zugriff und das Lesen bzw. Ändern der Daten nur durch vom Patienten berechtigten Personen möglich ist. Die weiteren Anforderungen sind die Beibehaltung von existierenden Infrastrukturen und die Beibehaltung der fachlogischen Abläufe (siehe [Cau06, CWF+06]).
Alle diese Anforderungen, die für diese Anwendungen nötig sind, setzen eine rollenbezogene Rechtevergabe voraus. Jedoch sind ebenfalls personenbezogene Rechtevergaben umgesetzt, wodurch ganz spezielle Berechtigungen formuliert werden können, z.B. [Cau06]:

- Jeder Apotheker, dem ich meine eGK gebe, darf alle Rezepte sehen, nur nicht die Mitarbeiter der Schwanen-Apotheke in Bremen.
- Die Einträge in der Patientenakte, die mit meinem Haarausfall zusammenhängen, darf nur Dr. Schulz lesen. Jeder Arzt darf aber neue Laborwerte anfügen.

Wie fein die Granularität der Berechtigungen im Detail zu formulieren ist, wird durch die gesetzlichen Vorgaben nicht bestimmt.
Das Berechtigungskonzept der eGK kann also sowohl anwendungsbezogen (z.B. alle Berechtigungen für eRezepte sind identisch), als auch sehr feingranular angewendet werden. So könnte dann zum Beispiel für jedes eRezept eine individuelle Rechtevergabe realisiert werden. Die Verwendung der qualifizierenden digitalen Signaturen spielt dabei eine Schlüsselrolle. Die digitalen Signaturen ermöglichen durch ihre zertifikatsbasierte Zugriffsmöglichkeiten eine personenbezogene Rechtevergabe. Andererseits muss aber aus Gründen des Datenschutzes eine Zugriffsmöglichkeit ohne Zertifikatsbeigabe geschaffen werden, um Datenspuren und Bewegungsprofile zu vermeiden [Cau05].

[2] §291a SGB V Abs. 5

5.1.3 Anforderungen an ein Berechtigungskonzept für elektronische Prüfungen

Bei den Prüfungen dürfen nur die Studierenden an der Prüfung teilnehmen, die zu der Prüfung ordentlich angemeldet sind. In der Regel wird eine solche Anmeldung durch das Prüfungsamt durchgeführt. Das Prüfungsamt ermittelt dann nach dem Ende der Anmeldefrist, wer an der Prüfung teilnimmt. Jedoch existieren bei der Zulassung häufig auch Spezialfälle, die durch die reine formelle Prüfung des Prüfungsamtes nicht berücksichtigt werden.
So ist eine Zulassung zu einer Prüfung laut Prüfungsordnung möglich, jedoch basiert die Zulassung zusätzlich noch z.B. auf Ergebnissen von Übungen o.ä.. Andererseits kann aber eine formelle Zulassung zur Prüfung nicht möglich sein (wie z.B. bei einem geplanten Studiengangwechsel), dafür kann aber nach Absprache mit dem Dozenten der Studierende trotzdem die Erlaubnis zur Teilnahme bekommen. Für ein Berechtigungskonzept bedeutet dies eine eindeutige personenbezogene Rechtevergabe für die Prüfung. Diese muss in der Hoheit des Dozenten liegen, denn die Prüfungsfragen liegen in seinem Hoheitsbereich und der Dozent muss bestimmen können, wer die Daten einsehen kann und wer nicht.
Außerdem hat jeder Teilnehmer der Prüfung das Recht zu bestimmen, ob er seine Prüfungsangaben vernichtet und damit einer Bewertung entzieht[3] oder aber zur Bewertung abgibt. Grundsätzlich sollte aber der Dozent zumindest als „erlaubte Person“ eingetragen sein um zu vermeiden, dass ein Student seine Prüfungsangaben versehentlich nicht freigibt.
Die Prüfungsfragen sind erst dann dem Studenten zugänglich zu machen, wenn der Dozent diese freigegeben hat. Das bedeutet, die Teilnehmer besitzen ein zeitlich eng begrenztes Zugriffsrecht.

5.1.4 Zusammenfassung

Betrachten wir das Berechtigungskonzept der eGK und vergleichen es mit den Anforderungen an die elektronischen Prüfungen, so ergeben sich interessante Parallelen: Die allgemeinen Anforderungen an die eGK (siehe Unterabschnitt 5.1.1) können nahezu vollständig auf die Anforderungen der elektronischen Prüfungen (vgl. Abschnitt 3.1) gespiegelt werden. Des Weiteren sind ebenfalls bei den Prüfungen die bestehenden Infrastrukturen an den Hochschulen sowie die fachlogischen Abläufe beizubehalten. D.h. es ist ein Sicherheitskonzept nötig, das sich in die existierenden Strukturen integrieren lässt und möglichst wenig Anpassung in den Strukturen benötigt. Jedoch er-

[3] Ähnlich dem Zerreißen des Prüfungsbogens bei den papierbasierten Prüfungen

gibt sich an den Hochschulen eine wesentlich einfachere Möglichkeit, die bestehenden Systeme zumindest in einem gewissen Rahmen anzupassen. Dies ist bei der eGK so nicht möglich.
Die Rechtssicherheit bei der Verarbeitung der medizinischen Daten ist wie bei den Prüfungsdaten nur mittels qualifizierender digitaler Signaturen möglich. Ebenfalls vergleichbar ist die Forderung nach der Umsetzung aller Anforderungen, auch wenn dies zu Zielkonflikten führt. Denn bei der eGK besteht z.B. der Konflikt, dass zwar der Zugriff auf die Daten nur mittels eGK und HBA möglich sein soll, aber gleichzeitig besteht die Anforderung aus den Anwendungsfällen, dass die Erteilung einer Berechtigung und die Nutzung einer Berechtigung zeitlich zu entkoppeln sind [Cau06]. Dies tritt z.B. bei einem Krankenhausaufenthalt auf, bei dem der Patient bei seiner Einweisung die Daten seiner Patientenakte freigibt, die behandelnden Ärzte jedoch erst Tage später - wenn der Patient dann schon in der Narkose liegt - darauf zugreifen müssen.
Bei den Prüfungen ergibt sich ein ähnlicher Konflikt, denn die Prüfungszulassung durch das Prüfungsamt erfolgt meistens Tage vor der eigentlichen Prüfungsdurchführung und damit auch die Rechtevergabe durch den Dozenten. Der eigentliche Zugriff darf aber erst zum exakten Prüfungstermin erfolgen.

5.2 Analyse des virtuellen, ticketbasierten Dateisystems der eGK

Die Funktionalitäten der in Unterabschnitt 5.1.2 dargestellten Anwendungen werden über serverbasierte Dienste bereitgestellt. Jeder Dienst unterstützt zwei Schnittstellen, die je nach Anwendung erweitert werden können [CWF+06]:

- CRUD-Schnittstelle[4] zum Zugriff auf die medizinischen Daten.
- Rechte-Management Schnittstelle für die Administration der Zugriffsrechte auf die Daten.

Die beiden Schnittstellen werden auf die Zugangs- und Integrationsschicht (ZIS) abgebildet. Die ZIS dient der Verwaltung der Anwendungsdaten und der darauf geltenden Zugriffsrechte (siehe Unterabschnitt 4.6.3).

[4] CRUD = create, read, update, delete

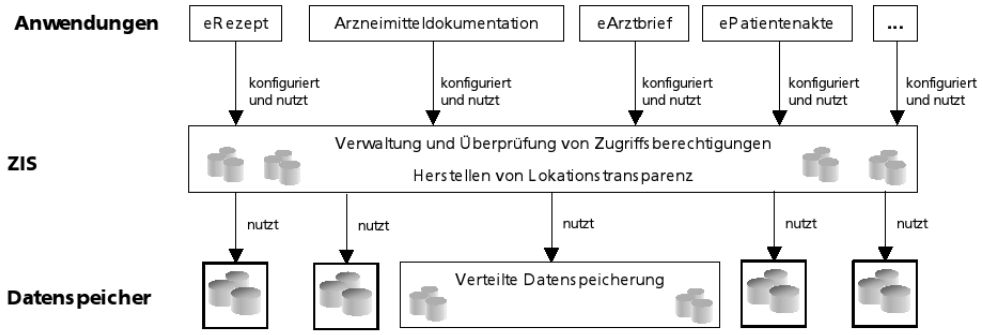

Abbildung 5.1: Zugangs- und Integrationsschicht [Fra05]

5.2.1 Zugangs- und Integrationsschicht (ZIS)

Die ZIS ist die Schnittstelle zwischen den Anwendungen und den Datenspeichern. Abbildung 5.1 zeigt die schematische Darstellung der ZIS. Die ZIS realisiert die Anforderungen an den Datenschutz und die Datensicherheit mit einem durch kryptografische Methoden unterstützten Berechtigungskonzept ab. Das Berechtigungskonzept wird durch einen ticketbasierten Ansatz (Ticketkonzept) realisiert. Die Daten selbst werden dabei in beliebigen Datenspeichern vorgehalten (virtuelles Dateisystem).
Die Lokationstransparenz ermöglicht es, dass der Zugriff auf die Daten so aussieht, als ob ein gemeinsames Dateisystem existieren würde. In Wirklichkeit sind die Daten aber auf vielen Servern gespeichert oder auf der eGK selbst. Diese Kapselung hat den Vorteil, dass auch existierende Datenbestände weiter verwendet werden können [CWF+06].

5.2.2 Konnektor

Während die ZIS die Schnittstelle zwischen den serverseitigen Anwendungen und den verteilten Datenspeichern regelt, werden die Primärsysteme in den Praxen, Krankenhäuser, Apotheken etc. über einen so genannten Konnektor an die Telematikinfrastruktur angekoppelt (siehe Abbildung 4.6) [Gem09]. Der Konnektor regelt auch die Anbindung der eGK und des HBA sowie einer Security Module Card (SMC) [Gem09].
Eine SMC dient der Identifikation einer berechtigten Institution (Arztpraxis, Apotheke etc.)[5]. Somit kann z.B. eine Apotheke auch den Angestellten die Möglichkeit geben, auf ein eRezept zuzugreifen ohne dass die Angestellten eine eigene Karte besitzen. Der Inhaber der Apotheke ist verantwortlich dafür,

[5]http://de.wikipedia.org/wiki/Security_Module_Card, aufgerufen am 02.03.2010

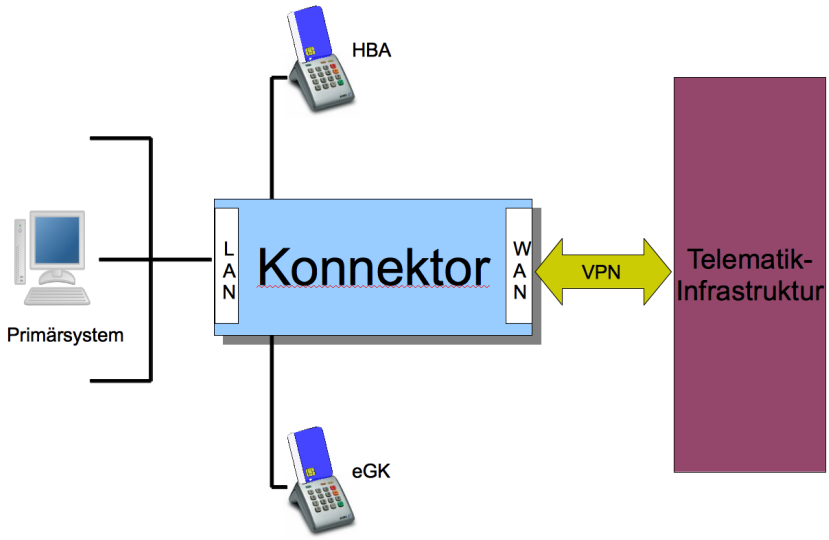

Abbildung 5.2: Konnektor Einordnung in die Infrastruktur

dass nur die dazu berechtigten Angestellten über die SMC Zugriff auf die Daten des Versicherten erhalten. Der Zugriff durch Personal auf die eGK bzw. die Daten des Versicherten ohne persönlichen Ausweis ist nach § 291a SGB V technisch zu ermöglichen. Allerdings setzt die Nutzung der SMC durch das Personal eine Card-2-Card Authentifizierung zwischen HBA des Apothekers und der SMC der Apotheke voraus. Eine technische Beschreibung der Card-2-Card Authentifizierunsgmechanismen von eGK, HBA und SMC findet sich in [Ö06]. Der Unterschied zwischen HBA und SMC liegt darin, dass der HBA mehr Rechte besitzt als die SMC [Gem09].
Der Konnektor verbindet sich über ein IPSec basiertes VPN mit der Telematikinfrastruktur, sowie über das LAN der Institution mit dem Primärsystem(en) (siehe Abbildung 5.2). Der Konnektor besitzt dafür zwei Netzwerkkarten. Des Weiteren sind die Kartenterminals für die eGK und den HBA an den Konnektor angeschlossen. Der Konnektor selbst besteht aus den Funktionsblöcken Netzkonnektor (KONN:NK), Anwendungskonnektor (KONN:AK) und Signaturanwendungskonnektor (KONN:SAK) (siehe Abbildung 5.3) [Gem09].
Der Netzkonnektor ist für die Verbindungsorganisation zur Telematikinfrastruktur (TIS) über VPN zuständig. Ein spezieller Paketfilter schützt dabei die Komponenten des Konnektors und die TIS vor ungültigen Anfragen aus

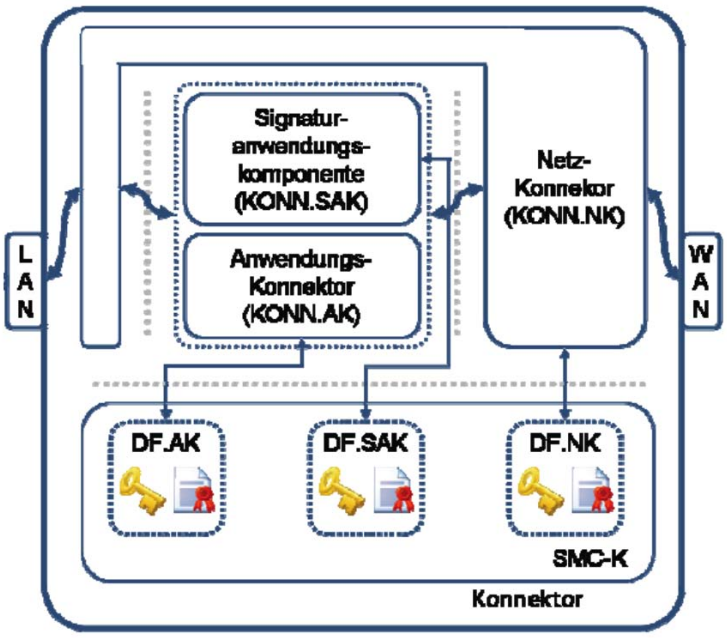

Abbildung 5.3: Funktionsblöcke des Konnektors [Gem09]

dem LAN der Institution.
Der Anwendungskonnektor steuert die fachlichen Anwendungsfälle. Außerdem überwacht er die Kommunikation zwischen LAN und der TIS und verwaltet die Kartenterminals. Der Anwendungskonnektor ist für die sichere Kommunikation zwischen Konnektor und Kartenterminal bzw. der darin eingelegten Chipkarte verantwortlich. Dazu wird eine TLS Verbindung zwischen dem Primärsystem und dem Anwendungskonnektor über das LAN der Institution aufgebaut [Gem08].

Die Signaturanwendungskomponente (SAK) erzeugt und prüft die qualifizierenden digitalen Signaturen, die im Rahmen der Anwendungen genutzt werden. Dazu sendet die SAK die zu signierenden Daten verschlüsselt an den entsprechenden Ausweis. Außerdem stellt die SAK einen Extended Trusted Viewer bereit, um die zu signierenden Daten und Zertifikate auf den Terminals der Institution anzeigen zu können [Gem09].
Jede der drei Komponenten besitzt zur Erfüllung der Aufgaben individuelle kryptografische Schlüssel und Zertifikate, die in einem sicheren Schlüsselspeicher der SMC hinterlegt sind. Selbstverständlich sind die Schnittstellen LAN und WAN des Konnektors durch Firewalls geschützt. Es existieren verschiedene Anbieter von Konnektoren, die jedoch erst durch eine Zertifizierung des Bundesamt für Sicherheit in der Informationstechnik (BSI) die Genehmigung

Abbildung 5.4: Konnektor der Firma Siemens AG

erhalten, als Konnektoren in den Institutionen eingesetzt zu werden. In Abbildung 5.4 ist beispielhaft ein Konnektor der Firma Siemens dargestellt.

Eine Verwendung eines solchen Hardwarekonnektors für die Prüfungen kommt aus verschiedenen Punkten nicht in Frage: Zum einen ist eine Prüfungssituation keine 1 : 1 Beziehung wie zwischen Heilberufler und Patient. Sondern eine $1 : n(n >= 1)$ Beziehung Dozent und Prüfungsteilnehmer. Die Konnektoren müssten also für jeden Prüfungsrechner installiert werden, wobei dies die Anwesenheit des Dozentenausweises an jedem Arbeitsplatz verlangen würde, was unmöglich ist. Eine Alternative könnte die Realisierung von Funktionsteilen des Konnektors mittels Software sein. Dies würde die „Verteilung“ des Konnektors auf die einzelnen Prüfungsrechner erheblich erleichtern. Dazu wäre dann nur die Signaturanwendungskomponente und der Anwendungskonnektor zu realisieren.
Aufgrund der Tatsache, dass viele Prüfungszentren über ein eigenes Subnetz verfügen, könnte ein Konnektor als Hardware auch zwischen dem LAN und den Servern geschaltet sein. Dies bedingt aber zusätzlich immer noch den Einsatz von Kartenterminals an den einzelnen Prüfungsrechnern. Der Dozent würde mit seinem Ausweis dann den gesamten Prüfungsverlauf signieren können. Nachteilig ist dabei jedoch, dass ein solcher Proxy-Konnektor einen Single-Point-Of-Failure bedeuten würde.
Sinnvoller scheint für die Prüfungsdurchführung die softwaretechnische Lösung in Form eines Client-Proxy zu sein. Anders sieht es bei der Anwendung in den Prüfungsämtern aus. Hier würde der Einsatz eines Hardware-

Konnektors Sinn machen. Parallelen sind hierbei zwischen den Prüfungsamtsmitarbeitern und den Apothekenmitarbeiter zu ziehen: Die Angestellten des Prüfungsamtes müssen somit über keinen eigenen Ausweis verfügen. Nur der Prüfungsamtsvorsitzende müsste sich - analog zum HBA des Apothekers - mit seinem Ausweis am Konnektor authentisieren. Die eingebaute SMC des Konnektors würde dann die Sicherheitsdienste für die Prüfungsamtsmitarbeiter bereitstellen.
Der Hardware-Konnektor ist eine - auch physisch geschlossene - Komponente an deren Sicherheitsniveau ein rein softwarebasierter Konnektor nicht herankommt. Allerdings ist hierbei die Frage, inwieweit das Sicherheitsniveau eines HW-Konnektors bei elektronischen Prüfungen erreicht werden muss. Neben den Kosten ist auch das organisatorische Element zu beachten: Während eine Installation eines Software-Konnektors trivial ist, so ist die Inbetriebnahme und die Wartung der Hardware sehr viel aufwendiger. Denn letztere muss auch gegen Vandalismus, Diebstahl usw. geschützt sein. Dies kann durch bauliche Maßnahmen erreicht werden oder aber die gesamten HW-Konnektoren würden nur für die Prüfungsdurchführung in den Laboren verwendet und sonst an einem sicheren Ort aufbewahrt werden. Für die Dozenten würden die HW-Konnektoren aber permanent im Büro installiert werden. Das Arbeiten (Prüfung erstellen, auswerten etc.) von zu Hause aus würde allerdings dadurch sehr erschwert, weil dies die Verwendung des Konnektors voraussetzt.
Der HW-Konnektor ermöglicht aber dahingehend erweiterte kryptografische Operationen als der SW-Konnektor. So können Sitzungsschlüssel über eine gesicherte Verbindung auf den HW-Konnektor übertragen und dort temporär gespeichert werden. In einem SW-Konnektor würde dies eine nicht unerhebliche Sicherheitslücke bedeuten.

Zusammenfassend lässt sich sagen, dass für die Prüfungsdurchführung der Einsatz eines SW-Konnektors unter Abwägung aller Sicherheitsaspekte, administrativen Aspekten und der Kosten sinnvoll ist. Gleiches gilt auch für den Dozenten-Konnektor. Bei den Prüfungsämtern ist aber der Einsatz von HW-Konnektoren zu realisieren. Denn durch die Verwendung einer SMC im Konnektor müssen nicht alle PA-Mitarbeiter mit einer Smartcard ausgestattet sein. Außerdem wären die Primärsysteme in den Prüfungsämtern weiter einsetzbar.

5.2.3 Ticketkonzept der eGK

Das Ticketkonzept wurde vom Fraunhofer ISST entwickelt und realisiert alle Anforderungen, die durch die Datenschutzbeauftragten der Länder und des

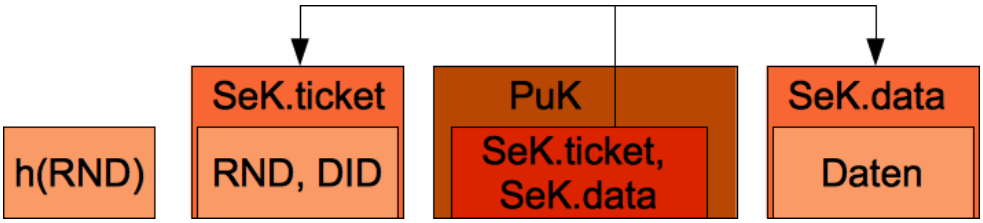

Abbildung 5.5: Beispielhafte Darstellung eines Ticket-Toolkits

Bundes definiert wurden (siehe Unterabschnitt 5.1.2, [BWB+02]). Auch die auftretenden Zielkonflikte können damit aufgelöst werden. So ist z.B. auch eine Anonymität trotz Authentizität möglich und „der Patient bleibt Herr seiner Daten“ [Cau06]. Die Innovation des Konzeptes beruht darauf, dass sowohl das Auffinden, Zugreifen und Entschlüsseln als drei entkoppelte Aktionen zu begreifen und durch verschiedene Sicherheitskonzepte zu realisieren ist [Cau06].
Die Daten werden dabei mit einem Hybridverfahren verschlüsselt und die Autorisierung und Authentifizierung wird durch ein Challenge-Response Verfahren umgesetzt. Beim Challenge-Response Verfahren beruht die Sicherheit darauf, dass sich ein Teilnehmer dadurch authentifiziert, indem er ein „Rätsel“ löst, dass nur er lösen kann (siehe u.a. [Eck09, BSW04]). Die Daten selbst werden in einem virtuellen Dateisystem verwaltet (siehe Unterabschnitt 5.2.5).

Ticket

Grundlage für das Ticketkonzept ist, dass jeder Datensatz (egal ob Anwendungsdaten oder Verzeichnis) mit einem spezifischen symmetrischen Schlüssel verschlüsselt ist. Dieser symmetrische Schlüssel ist wiederum verschlüsselt und kann nur durch Verwendung der eGK des Versicherten entschlüsselt werden. Auf den Datensatz kann nur derjenige zugreifen, der diesen symmetrischen Schlüssel bereitgestellt bekommt. Diese Bereitstellung erfolgt mit speziellen Zugriffsberechtigungen bzw. Ticket oder „Eintrittskarte“ genannt [Fra05]. In Abbildung 5.5 ist der Aufbau eines solchen Datensatzes beispielhaft dargestellt. Wenn ein Datensatz erstellt wird, wird zusätzlich eine Zufallszahl RND erzeugt. RND wird mit einem zufällig generierten symmetrischen Schlüssel Sek.ticket verschlüsselt. Der Datensatz wird mit einem weiteren symmetrischen Schlüssel SeK.data verschlüsselt. Beide symmetrischen Schlüssel werden wiederum mit dem öffentlichen Schlüssel des Versicherten

(PuK) verschlüsselt. Diese symmetrische Verschlüsselung mit SeK.data und Sek.ticket ist nötig, damit ein Angreifer nicht durch Ausprobieren doch einen indirekten Bezug zu einer Person herstellen kann. SeK.ticket und SeK.data werden aus Performancegründen zusammen verschlüsselt, denn dadurch ist nur ein Ver- und Entschlüsselungsvorgang nötig. Falls der tnode auf ein Verzeichnis verweist, so wird der SeK.data nicht benötigt. Nur wenn der tnode auf einen Datensatz verweist, kommt der SeK.data zur Anwendung.
Bei einem Zugriff auf die Daten wird dem Patienten die verschlüsselte Zufallszahl geliefert. Nur der Patient kann diese Zufallszahl entschlüsseln, die er dann wiederum an den Server zurückschickt. Dieser vergleicht die vorhandene Zufallszahl mit der empfangenen Zufallszahl und gibt den Datensatz bei Gleichheit frei. Die Entschlüsselung der Zufallszahl authentifiziert also die Person, für die das Ticket erzeugt wurde ohne dass der Server das Zertifikat der Person zu sehen bekommt. Somit bildet die verschlüsselte Zufallszahl also so etwas wie einen Ticket-Bausatz (Ticket-Toolkit), die zurückgelieferte Zufallszahl (RND) dann das Ticket. RND wird nicht im Klartext auf dem Server gespeichert, sondern als Hashwert.
Grundsätzlich existieren zwei Arten von Berechtigungen: rollenbezogene (Default Ticket-Toolkits) und personenbezogene (Personal Ticket-Tookits) Berechtigungen (siehe Abschnitt 5.2.3).

tnode Modell

Im Ticketkonzept wird jeder Datensatz durch einen so genannten *tnode* beschrieben. Ein tnode entspricht dabei einem beschreibenden Datensatz (Metadatensatz) in einer Datenbank, wobei die Daten selbst in einer anderen Datenbank vorgehalten werden können. Die Trennung von Daten und Metadaten ermöglicht eine vollständige Verteilung aller Daten zu einem Versicherten.
Ein tnode besteht aus den folgenden Elementen [Fra05]:

```
DefType tnode (tid: OID,
               did: hashedOID,
               flags: Boolean[8],
               kurztext: Char[256],
               klassifizierung: base64Text,
               tnodeTyp: {"dir", "pers", "transp", "protocol"},
               dateErstellung: TS,
               dateAenderung: TS,
               dateZugriff: TS,
               dateGueltigkeitsende: TS,
```

```
dienstReferenz: 4ByteServiceKey,
hashedOid: HashedOID,
2ndKey: EncryptedKeys,
arm:ARM,
dtt: TicketToolkit,
ptt: PersonalTicketToolkit [ ] )
```

tid Dies ist eine weltweit eindeutige tnode-ID (TID). Diese wird als Identifizierer eines tnodes verwendet um diesen anwendungsübergreifend referenzieren zu können. Die TID ist vom Typ Object-ID (OID). Die Vergabe der IDs erfolgt durch den zentralen Object-ID-Dienst (OIDD) der Lösungsarchitektur [Fra05].

did Vom Typ hashedOID ist die Verzeichnis-ID (DID). In jedem tnode können eine oder mehrere DIDs existieren, die den übergeordneten Verzeichnissen (also zu denen der tnode und seine zugehörigen Daten gehören) zugeordnet sind. Falls der tnode ein Wurzelverzeichnis ist, wird 0 angegeben. Die DIDs entkoppeln also die Daten von den Metadaten (Ticket-Informationen), wodurch eine erhöhte Flexibilität beim Zugriff ermöglicht wird.

flags Der erste Wert der 8 booleschen Werte gibt an, ob der tnode verborgen wurde oder nicht. Der zweite Wert, ob der tnode gesperrt ist. Die letzten 6 Werte können durch die Anwendungsdienste frei genutzt werden.

kurztext Ein unverschlüsselter beschreibender Text. Der Kurztext beinhaltet keine Informationen, die einen Personenbezug herstellen lassen könnten.

klassifizierung Ein XML-kodierter Text zur Klassifizierung des med. Datenobjektes, der z.B. für Suchdienste genutzt werden kann. Dieser Tag ist optional.

tnodeTyp: Hierbei kann die Klassifizierung des Datenobjektes angegeben werden, ob es sich also um ein Verzeichnis („dir“), persistente Daten („pers“), Transportdaten („transp“) oder Protokolldaten („protocol“) handelt.

dateErstellung Erstellungsdatum (Zeitstempel) des Verzeichnisses bzw. des Datenobjektes.

dateAenderung Änderungsdatum (Zeitstempel) des Verzeichnisses bzw. des Datenobjektes.

dateZugriff Datum (Zeitstempel) des letzten Zugriffs auf das Verzeichnis bzw. Datenobjekt.

dateGueltigkeitsende Ende der Gültigkeit (Zeitstempel) des Verzeichnisses bzw. des Datenobjektes.

dienstReferenz Eine Referenz des Dienstes, auf dem die Daten verwaltet werden.

hashedOid Hashwert über die DID, falls es sich um ein Verzeichnis handelt oder über die OID, falls es sich um Anwendungsdaten handelt.

2ndKey Dies ist ein sog. Zweitschlüssel. Er besteht aus den symmetrischen Schlüsseln SeK.data und SeK.ticket (siehe Abbildung 5.5), die mit dem privaten Schlüssel einer zweiten Karte verschlüsselt sind. Im Falle eines Kartenverlustes oder einer Neuausstellung können so die Daten weiter zugreifbar sein. Ohne diesen Zweitschlüssel käme ein Kartenverlust der Zerstörung der Daten gleich. Denn niemand könnte diese Daten wiederfinden bzw. entschlüsseln [Cau06]. Die Verschlüsselung erfolgt mit einer 2. Karte, die entweder eine private Signaturkarte oder aber z.B. auch die eGK des Ehepartners etc. sein kann.

arm Die Access-Restriction-Matrix beinhaltet die rollenspezifischen Zugriffsrechte, die bei Anwesenheit der eGK zum Zugriff auf die Daten möglich sind (siehe Tabelle 5.1).

dtt: TicketToolkit Default-Ticket-Toolkits (DTTs) sind Tickets, die für eine Rolle und nicht für eine bestimmte Person erstellt werden. Ein DTT regelt den Zugriff auf den tnode und auf die mit dem tnode verknüpften Datenobjekte über eine Access-Restriction-Matrix. Eine solche ARM ist in Tabelle 5.1 beispielhaft dargestellt.

Die ARM beschreibt, welche rollenspezifischen Zugriffsrechte auf die Daten vergeben werden dürfen. Sie bestimmt dabei nicht welche Rechte eine bestimmte Person besitzt, sondern die Kodierung bestimmt für jede Rolle welche Zugriffsart erlaubt ist [Fra05]:

- 00: Das Recht darf der Rolle nicht eingeräumt werden.

Rolle	list	read	write	delete	auth	tnode	t-auth
Arzt	01	01	10	00	ticket	00	protocol
Zahnarzt	01	01	10	00	ticket	00	protocol
Apotheke	10	10	00	00	ticket	00	protocol
Krankenhaus	01	01	10	00	ticket	00	protocol
sonstige LE	01	01	10	00	ticket	00	protocol
eKiosk	10	10	00	10	ticket	11	fullAuth
home	10	10	00	10	ticket	00	protocol

Tabelle 5.1: Beispiel für eine ARM [Fra05]

- 01: Das Recht darf der Rolle eingeräumt werden, ist aber nicht der Normalfall.
- 10: Das Recht darf der Rolle verweigert werden, ist aber der Normalfall.
- 11: Das Recht ist der Rolle immer gewährt und darf ihr auch nicht entzogen werden.

Außerdem wird in der ARM die Form der Authentifizierung für die Standardoperationen als auch für die Modifikation des tnodes angegeben:

- Ticket: Wer das Ticket vorweist, erlangt ohne weitere Überprüfung den Zugang zu den Daten.
- Protocol: Neben dem Ticket wird zusätzlich ein Zertifikat vom Einlösenden verlangt, um die Rollenzugehörigkeit zu überprüfen. Der Zugriff wird protokolliert.
- FullAuth: Neben dem Ticket wird zusätzlich ein Zertifikat vom Einlösenden verlangt, welches dann über einen Verzeichnisdienst überprüft wird. Der Zugriff wird protokolliert.

Ein DTT ist in Abbildung 5.6 dargestellt. Dabei sind die Elemente Typ und Ablaufdatum neu. Typ bestimmt, ob nach einem Zugriff das Ticket-Toolkit durch ein neues Ticket-Toolkit ersetzt werden kann (flushAccess). Das Element Typ kann auch einen zweiten Wert, nämlich multipleAccess annehmen. Falls multipleAccess verwendet wird, können innerhalb der festgelegten Gültigkeitsdauer (Element Ablaufdatum) beliebig viele Schreib- und Leseoperationen durchgeführt werden.
Außerdem erhält der Datensatz eine Daten-ID (DID), die den Datensatz eindeutig referenziert (in Abbildung 5.6 (35)). Diese Daten-ID wird aus Performancegründen zusammen mit der Zufallszahl RND mit dem symmetrischen

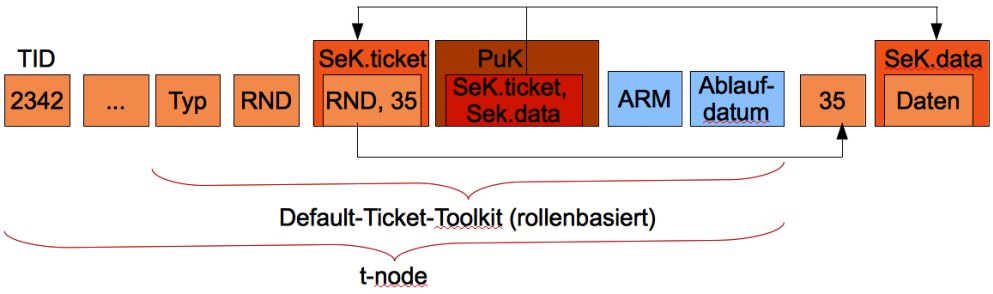

Abbildung 5.6: Default-Ticket-Toolkit innerhalb eines tnodes (vgl. [Cau05])

Schlüssel SeK.ticket verschlüsselt. Inwieweit DTT für die Prüfungen zu verwenden sind, ist fraglich. Denn der Zugriff ist zwar rollenbezogen definiert, jedoch nur unter der Verwendung der eGK. Eine rollenbezogene Rechtevergabe bei den Prüfungen in der Form: „nur Dozenten dürfen meine Prüfungsangaben sehen und korrigieren“ oder „nur Studierende der Fachbereiche 5 und 12 dürfen an meiner Prüfung teilnehmen“ schränken den Benutzerkreis stark ein, aber nur durch die PTT ist eine exakt definierte Benutzerliste bestimmbar.
DTT in der Form der eGK sind für die Prüfungen nur in der Form als Zugang für den Dozenten bzw. Studenten als Besitzer der Datensätze sinnvoll.

ptt: PersonalTicketToolkit [ ] Neben den DTTs können sog. Personal-Ticket-Toolkits (PTTs) in einem tnode enthalten sein. Diese PTTs sind jeweils an eine einzelne Person gebunden. Ein PTT besteht aus den Datenelementen des DTT und wird durch weitere Felder ergänzt (siehe auch Abbildung 5.7). Zum einen erhält jedes PTT eine eigene Zufallszahl als Verifizierer (RND2). Des Weiteren werden die Session-Keys mit dem öffentlichen Schlüssel derjenigen Person verschlüsselt, die ein PTT erhalten soll (PuK.x). Dazu wird das Zertifikat der Person dem PTT beigefügt (Cert.x). Außerdem wird jedem PTT eine Acces-Control-List (ACL.x) beigefügt, die die für diese Person erlaubten Berechtigungen darstellt. Pro tnode kann es beliebig viele PTTs geben.
Das Element Typ erhält zusätzlich die Option singleAccess. Diese Option ermöglicht es, dass mit diesem Ticket lediglich einmalig eine Schreib- oder Leseoperation ausgeführt werden kann. Nach der Durchführung der Operation wird das TicketToolkit gelöscht.

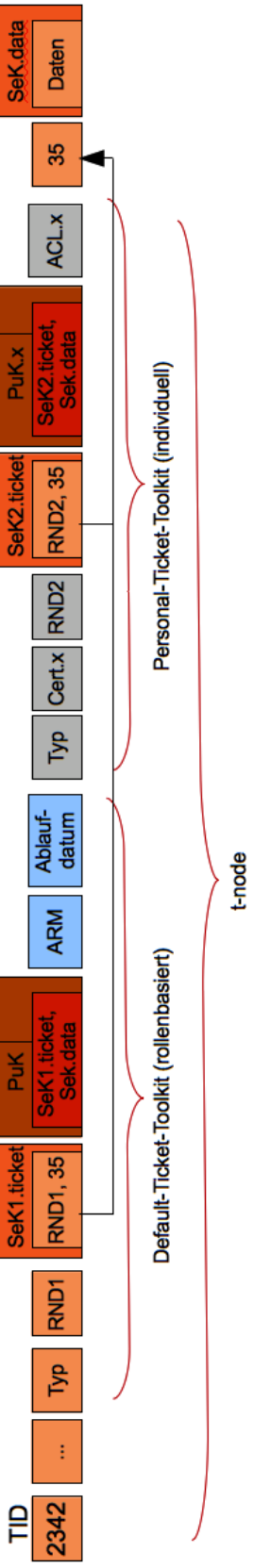

Abbildung 5.7: Personal-Ticket-Toolkit innerhalb eines tnodes (vgl. [Cau05])

Anwendungsdatensatz

Ein Anwendungsdatensatz ist durch das folgende Informationsmodell beschreibbar [Fra05]:

```
DefType ApplicationData {
    oid: OID,
    expDate: TS,
    data: Base64Block
    <<state>> gültig,
    <<state>> abgelaufen,
    <<state>> verwaist
}
```

Die Objekt-ID (OID) wird im ISO OID Format dargestellt und hat eine Größe von 30 Byte [Fra05]. Der Aufbau der OID ist auf einen europäischen Anwendungsraum hin konzipiert. Die Spezifikation des OID-Dienstes ist in [Fra05, S. 260 ff.] beschrieben.
Das Element expDate ist ein Zeitstempel, der das Ende der Gültigkeit des Datensatzes angibt. In Abhängigkeit von expDate kann ein Datensatz die Zustände gültig (Ablaufdatum noch nicht erreicht), abgelaufen (Ablaufdatum ist um weniger als 4 Wochen überschritten) und verwaist (Ablaufdatum um mehr als 4 Wochen überschritten) annehmen. Der Zustand verwaist zeigt an, dass der Datensatz nicht mehr verwendet wird und daher aus dem Datenspeicher gelöscht werden kann.

Verzeichnisdatensatz

Ein Verzeichnisdatensatz dient dazu, die Speicherorte der tnodes eines Verzeichnis zu ermitteln.

```
DefType DirectoryData {
    did: OID,
    datastorekey: 4ByteServiceKey[]
}
```

Die tnode Struktur ist auch für die Prüfungen zu verwenden. Anpassungen könnten vor allem bei den Elementen flags, kurztext, sowie tnodeTyp erfolgen. Auf den *Kurztext* kann verzichtet werden, die *Klassifizierung* ist jedoch sinnvoll, denn sollte das Ticketkonzept auch für weitere Anwendungen verwendet werden (elektronische Studierendenakte etc.), wäre ein beschreibender Datensatz für Suchdienste o.ä. hilfreich. Der tnodeTyp kann sich auf

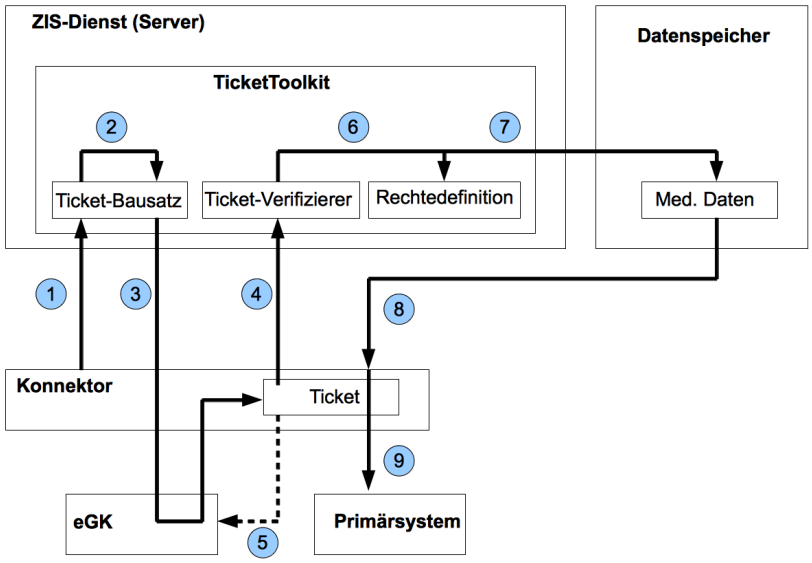

Abbildung 5.8: Ticketing (vgl. [Fra05])

die Elemente „dir“ für die Verzeichnisse und „pers“ für die Anwendungsdaten beschränken.

5.2.4 Ticketing

In Abbildung 5.8 ist der grobe Ablauf des Ticketing dargestellt.

1. Anforderung des Ticket-Bausatzes. Der Konnektor fordert den ZIS-Dienst an, für den Benutzer einen Ticket-Bausatz zum Zugriff auf die medizinischen Daten zu liefern.

2. Der ZIS-Dienst schaut nach, ob für den Benutzer ein Personal-Ticket- oder Default-Ticket-Toolkit vorhanden ist.

3. Der ZIS-Dienst gibt den Bausatz (siehe Abbildung 5.9) des jeweiligen Ticket-Toolkits des Benutzers und die tnodeID an den Konnektor zurück. Der Bausatz wird mit Hilfe der eGK entschlüsselt und das Ticket, bestehend aus tnodeID, DID und RND, wird erstellt.

4. Das Ticket wird eingelöst, indem die tnodeID, DID und RND an den ZIS-Dienst geschickt werden. Die Ticketverifizierung erfolgt durch Ver-

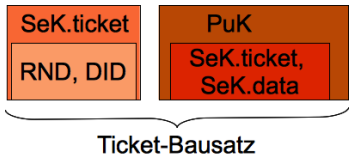

Abbildung 5.9: Ticket-Bausatz

gleich von RND mit dem im Ticket-Toolkit abgespeicherten Zufallswert. Ist dieser gleich, gilt das Ticket als gültig und kann eingelöst werden.

5. (optional) Die Speicherung des Tickets (tnodeID, DID, RND) kann auch zeitlich befristet auf der Karte erfolgen, um für spätere Zugriffe wieder zu verwenden, um sich den Aufwand der Traversierung und Ticketerzeugung zu ersparen.

6. Jetzt wird überprüft ob das Ticket-Toolkit noch gültig ist; dazu wird die Datumsangabe mit dem aktuellen Datum verglichen.

7. Anschließend überprüft der ZIS-Server die Zugriffsrechte, die dem Benutzer für dieses TicketToolkit zugeordnet sind. In Abhängigkeit davon wird die Freigabe zum Zugriff auf die Daten (z.B. lesend) erteilt und anhand der DID und der Dienst-Referenz der Daten wird der Datensatz ermittelt.

8. Der verschlüsselte Datensatz wird an den Konnektor geschickt.

9. Im Konnektor wird der Datensatz mit Hilfe des SeK.data (siehe Abbildung 5.9) entschlüsselt. Die entschlüsselten Daten werden an das Primärsystem weitergeleitet, mit dem die Daten durch den Benutzer entsprechend verarbeitet werden können.

5.2.5 Virtuelles Dateisystem

Das Ticketkonzept stellt die Datensicherheit und den Zugriffsschutz sicher. Das virtuelle Dateisystem stellt sicher, dass ein Berechtigter die ihm zugeordneten Daten finden kann und durch die Lokationstransparenz die Möglichkeit alle Daten vollständig verteilt zu speichern.
Voraussetzung ist eine Zuordnung zwischen Daten und Personen. Allerdings darf diese Zuordnung nicht eindeutig erkennbar sein und nur mit Hilfe der

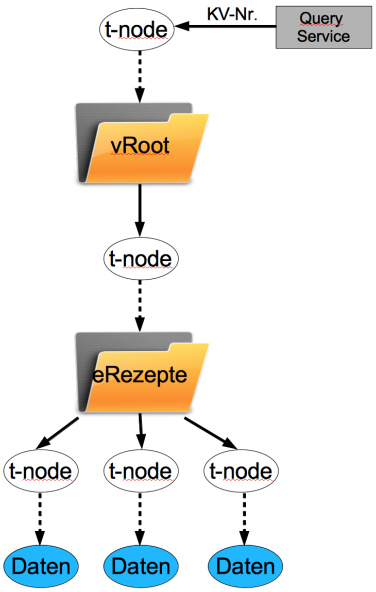

Abbildung 5.10: Aufbau eines Dateibaumes (vgl. [Fra05])

eGK berechenbar sein [Cau06]. Des Weiteren ist der 1:n Charakter der Zuordnungen zwischen Personen und Daten und die Umsetzung der verschiedenen Anwendungen der Grund, warum die Daten eines Versicherten als hierarchischer Baum angeordnet werden. Der Baum besteht aus Ordnern und Datensätzen und kann sich über verschiedene Server sowie zentrale und dezentrale Speichermedien erstrecken [Cau06].

In Abbildung 5.10 ist der Aufbau eines Dateibaumes dargestellt. Die dargestellte Hierarchie zeigt, dass die gestrichelten Linien eine verschlüsselte Referenzierung und die geschlossenen Linien eine offene Referenzierung darstellen. Die Zuordnung der Daten zu einem tnode ist also nur mit einem Ticket für diese Daten möglich. Das Auflisten des Ordners eRezepte ist nur mit einem Ticket für eRezepte möglich.

Jedes virtuelle Dateisystem verwaltet also drei Arten von Daten [Fra05]:

1. Anwendungsdaten: Jedem Anwendungsdatum ist eine eindeutige ID zugeordnet, die eine Referenzierung auf das Datum ermöglicht.

2. Metadaten zu den Anwendungsdaten (tnodes): Die tnodes enthalten Ticket-Toolkits, die Tickets zum Zugang zu den Daten im Konnektor erzeugen und dann auf einem Server eingelöst werden können. Ein Zu-

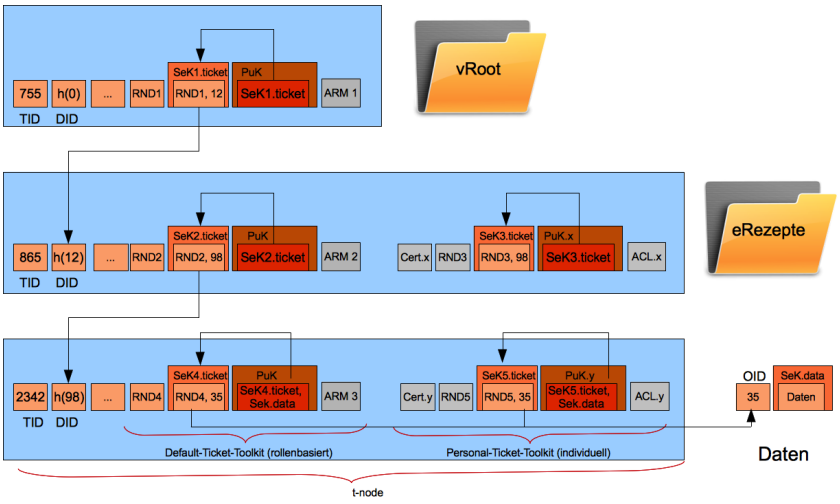

Abbildung 5.11: Beispielhafte Darstellung eines Dateibaumes mit tnodes (vgl. [Fra05])

griff auf das zum tnode gehörende Datum ist nur durch Einlösen eines Tickets möglich, dass über diesen tnode ausgestellt wurde.

3. Verzeichnisse über die tnodes: Alle Daten eines Versicherten sind in hierarchisch strukturierten Verzeichnissen zusammengefasst. Ein tnode kann auch mehreren Verzeichnissen zugeordnet sein.

Abbildung 5.11 zeigt anhand eines Beispiels, wie die Gestaltung des Dateibaumes konkret aussieht und funktioniert. In der Darstellung wurde aber auf einige Elemente der Übersicht wegen verzichtet.
Jeder tnode ist über eine tnode ID (TID) eindeutig gekennzeichnet. Der Zugriff auf ein Datenobjekt ist nur und ausschließlich über den zugehörigen tnode und die dort realisierten Berechtigungen möglich. Jedes Verzeichnis verfügt über eine eindeutige Verzeichnis-ID (DID). Der tnode, der auf das Verzeichnis verweist, enthält die DID dieses Verzeichnisses nur in verschlüsselter Form, wobei eine Entschlüsselung nur über ein gültiges Ticket möglich ist [Fra05]. Einem tnode können dann mehrere Verzeichnisse zugeordnet sein, die dann die gleiche DID besitzen.
Ein Aufwärtstraversieren des Dateibaumes ist nicht möglich, weil es keine Möglichkeit gibt, zu einer DID den zugehörigen übergeordneten tnode zu finden. In Abbildung 5.11 ist es z.B. nicht möglich, von dem untersten tnode (Datenverzeichnis) auf den übergeordneten tnode (Anwendungsverzeichnis) zuzugreifen. Die in der Abbildung angegebene DID 98 liegt im Anwendungsverzeichnis nämlich verschlüsselt vor [Cau06].

Wichtig hierbei ist, dass es z.B. auf der tnode-Ebene der eRezepte meherere Rezepte geben kann, die wiederum einen eigenen tnode haben der dem tnode eRezepte zugeordnet ist. D.h. für diese tnodes sind die Werte des Feldes DID alle gleich. Durch diese Zugehörigkeit ist z.B. das Auflisten aller Rezepte eines Patienten möglich, wenn dies für diejenige Person durch ARM bzw. die ACL erlaubt ist.
Ein weiterer Vorteil ist, dass die TID des Wurzelordners (vRoot) eines Versicherten nicht geheim sein muss, weil die Zuordnung eines verschlüsselten Datensatzes zur vRoot des Versicherten nicht möglich ist. Denn es gibt keine Möglichkeit, den Pfad vom Datensatz zur Wurzel zu finden [Cau06].
Das Abwärtstraversieren ist nur möglich, wenn für jede Ebene eine entsprechende Erlaubnis (Ticket) nach dem in Unterabschnitt 5.2.3 dargestellten Berechtigungskonzept vorhanden ist. Wichtig hierbei ist noch, dass die Dienstreferenzangabe in jedem tnode, den Speicherort der diesem tnode untergeordneten tnodes bzw. Datenobjekte angibt. Jedes Datenobjekt ist über eine Objekt-ID (OID) eindeutig gekennzeichnet. Der zum Datenobjekt zugehörige tnode enthält die OID des Datensatzes nur in verschlüsselter Form und eine Entschlüsselung und damit der Zugriff auf den Datensatz ist nur über ein gültiges Ticket möglich. Die Berechtigungen um auf diesen tnode bzw. Datenobjekt zuzugreifen werden durch die ARM des Default-Ticket-Toolkits oder der ACL der Personal-Ticket-Toolkits in dem übergeordneten tnode verwaltet.

5.3 Ein virtuelles, ticketbasiertes Dateisystem für die Prüfungen

In Abschnitt 5.2 wurden die möglichen Anpassungen für die einzelnen Komponenten des virtuellen, ticketbasierten Dateisystems bereits diskutiert. In diesem Abschnitt wird gezielt auf die Adaption des virtuellen, ticketbasierten Dateisystems auf die elektronischen Prüfungen eingegangen.

5.3.1 Voraussetzungen

Die Anpassung des virtuellen, ticketbasierten Dateisystems erfordert bestimmte Voraussetzungen. So muss jeder Student bei der Immatrikulation eine elektronische Studierendenkarte (eSK) ausgehändigt bekommen, auf der seine persönlichen Daten, ein privater Schlüssel und ein öffentlicher Schlüssel mit einem zugehörigen Zertifikat gespeichert sind. Ebenfalls bekommen die Mitarbeiter des universitären Lehrbetriebes je eine elektronische Prüferkarte

(ePK), auf der ebenfalls persönliche Daten, der Status (Professor, Privatdozent, wissenschaftlicher Mitarbeiter) und ein zertifiziertes Schlüsselpaar gespeichert sind. Weiterhin muss eine hochschulweite Public-Key Infrastruktur (PKI) existieren und alle beteiligten Rechner müssen mit entsprechenden Smartcard-Readern ausgestattet sein.
Eine Alternative zur Smartcard ist der Einsatz von eToken. eToken sind USB-Speichersticks kombiniert mit Smartcardfuktionalitäten. Der Vorteil einer solchen Token-basierten Lösung ist, dass nicht alle Rechner mit einem extra Smartcard-Reader ausgestattet werden müssen. Der Zugriff auf den eToken erfolgt über die USB-Schnittstellen. Außerdem wäre ein Mehrfachnutzen durch den Speicher des Tokens gegeben.
Allerdings belaufen sich die Speichergrößen der gängigen eTokens zum Zeitpunkt dieser Arbeit auf 128/ 256 MB. Nachteilig ist außerdem, dass ein eToken als Lichtbildausweis nicht in Frage kommt. Des Weiteren hätte eine Smartcard den Vorteil, dass auch Bezahlfunktionen über eine integrierte Geldkartenfunktionalität realisiert werden könnten. Auf eine weitere Diskussion über die Verwendung von Smartcards oder eTokens als sichere Signaturerstellungseinheit wird in dieser Arbeit verzichtet. Es sei jedoch darauf hingewiesen, dass in [Sch09b] beschrieben wurde, dass Ansätze wie eTokens, virtuelle Ansätze, Implantate oder Web of Trust durchaus Alternativen zu einem elektronischen Ausweis sein können, aber die Gesamtfunktionalität einer Smartcard nicht ersetzen können.

5.3.2 Anpassung des virtuellen, ticketbasierten Dateisystem

Allgemein

Einer der größten Unterschiede bei der Anpassung des virtuellen, ticketbasierten Dateisystems (vtD) an die elektronischen Prüfungen ist, dass bei der eGK nur der Versicherte mit einem Dateisystem ausgestattet ist. Bei den Prüfungen muss aber auch der Dozent über ein entsprechendes Dateisystem verfügen. Denn zum einen stellt der Dozent dem Studierenden die Berechtigung aus, auf die Prüfungsfragen zuzugreifen. Zum anderen will der Studierende aber auch genau bestimmen können, wer auf seine Prüfungsangaben zugreifen darf.
Ein weiterer Unterschied zum Lösungskonzept der eGK ist, dass im Gesundheitssystem alle 80 Millionen Bürger der Bundesrepublik als Anwender betrachtet werden müssen. Bei Hochschulen beläuft sich die Anwenderzahl zumeist im fünfstelligen Bereich. Während das System der eGK auf Grund der großen Teilnehmerzahl sehr gut skalieren muss, um alle Akteure bedienen

zu können, ergeben sich für die elektronischen Prüfungen Vereinfachungen. Dazu zählt vor allem die Realisierung der Konnektoren als Software, anstatt als Hardware-Konnektor wie im Konzept der eGK vorgesehen. Wie in Unterabschnitt 5.2.2 beschrieben, ist allerdings dennoch der Einsatz eines Hardware-Konnektors im Bereich der Prüfungsämter sinnvoll. Diese Arbeit beschränkt sich aber nur auf die Software-Variante, wobei diese natürlich in ähnlicher Weise auch in einem Hardware-Konnektor vorhanden sein würde. Der Konnektor wird als Software-Proxy implementiert.
Auf die umfangreiche Telematik-Infrastruktur der eGK kann größtenteils verzichtet werden. Die Grobarchitektur ist in Abbildung 5.13 dargestellt. Die Ticketstruktur für die elektronischen Prüfungen benötigt ein paar Änderungen gegenüber zum Ticketkonzept der eGK. Der wichtigste Unterschied ist, dass nicht nur die Studierenden einen Dateibaum besitzen, sondern auch die Dozenten und dass die Prüfungsämter auch Zugriff auf die Studentenbäume besitzen, für die sie zuständig sind.

Virtuelles Dateisystem

In Abbildung 5.12 ist der für die Prüfungen angepasste Dateibaum dargestellt. Beide Dateibäume besitzen zwar eine ähnliche Struktur, jedoch unterscheiden sie sich sowohl in ihrer Breite als auch Tiefe. Der Dozentenbaum beginnt mit dem virtuellen Root-Verzeichnis (*vRoot*), das mit Hilfe der DozentenID auffindbar ist. Falls der Zugreifende ein Ticket für das vRoot besitzt, kann er die nächste Verzeichnisebene *ePrüfung* auswählen. Unterhalb dieser Ebene erfolgt die Gliederung nach Semestern (Verzeichnisebene *Semester*). Denn die Prüfungen werden einem Semester zugeordnet. Unterhalb der Semester-Ebene befinden sich die einzelnen Prüfungsordner, die durch die jeweilige PrüfungsID gekennzeichnet sind (Verzeichnis *PrüfungsID*). Jedes Prüfungsverzeichnis ist wiederum in zwei Unterordner zugeordnet: *Angabe* und *Auswertung*. *Angabe* verweist auf die vom Dozenten angelegte Prüfung. *Auswertung* verweist auf die durch den Dozenten ausgewerteten Lösungen der Studierenden, wobei hier der Klassifizierer des tnodes die gehashte Matrikelnummer des Studierenden besitzt (*h(Matrikelnr)*), um die Auswertung zu einem Studierenden zu finden.
Der studentische Dateibaum verzweigt sich nach dem vRoot in *ePrüfung* und *eSA* (elektronische Studierendenakte). Die *eSA* wird in dieser Arbeit nicht weiter betrachtet und ist optional. Auch der Bereich unterhalb des Verzeichnisses *eSA* könnte wiederum unterteilt sein. Die Struktur des Verzeichnisses *ePrüfung* ähnelt sehr der Struktur des Dozentenbaumes. Allerdings werden hier nur die getätigten Prüfungslösungen zu einer Prüfung abgespeichert.

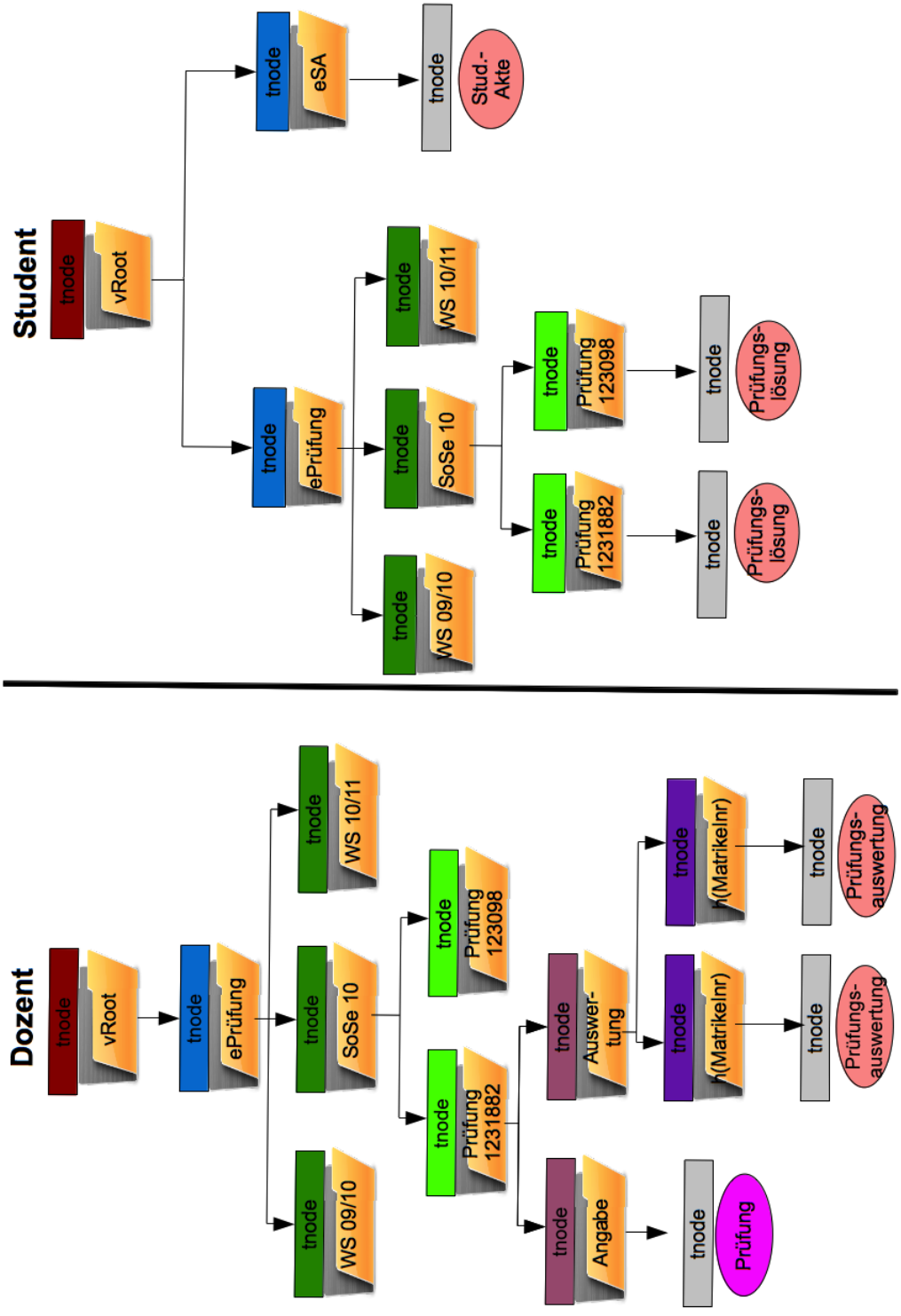

Abbildung 5.12: Angepasster Dateibaum

Ticketkonzept

Basis der Zugriffsregelung sind die TicketToolkits, die aus einem Ticket-Bausatz, einem Ticket-Verifizierer und einer Rechtedefinition bestehen (siehe Unterabschnitt 5.2.5).
Ein tnode besteht aus zumindest einem DTT und ggf. aus weiteren PTTs, die eine personenbezogene Rechtevergabe ermöglichen. Nachfolgend werden nur die Änderungen gegenüber der tnode-Struktur des Lösungskonzeptes beschrieben. Die Typ-Definition für den tnode lautet:

```
DefType tnode {
     TID: OID,
     DID: OID,
     Klassifizierer: char[256],
     tnodeTyp: ('dir', 'data'),
     dienstRef: 4ByteServiceKey,
     datumErstellung: TS,
     datumZugriff: TS,
     datumAenderung: TS,
     datumGueltigkeitsende: TS,
     ARM: Vector,
     2ndKey: byte[],
     dtt: DefaultTicketToolkit,
     ptt: PersonalTicketToolkit[] )
```

Der *tnodeTyp* wird auf die Felder „dir“ und „data“ beschränkt, um die Unterscheidung zwischen Datensatz und Verzeichnis-Datensatz zu ermöglichen.
Die Access-Restriction-Matrix (ARM) gibt bei der eGK an, welche Berechtigungen für eine Rolle zum Zugriff auf einen tnode erlaubt sind. Für die Prüfungen ergibt sich die in Tabelle 5.2 dargestellte ARM. Die ARM wird nicht im Konnektor erzeugt, sondern wird bei einem create() vom Elterntnode vererbt. Der Aufbau und die Anwendung der ARM ist in Abschnitt 5.4 detailliert beschrieben.

Die ACL wird aufgrund der veränderten Eigenschaften der Prüfungen gegenüber der eGK angepasst. So wird bei den Prüfungen für jede Operation das Zertifikat des Aufrufers verwendet. Somit sind die Standardoperationen Ticket, Protocol und fullPath nicht notwendig. Der Aufbau einer ACL ist in Tabelle 5.3 dargestellt. Sie verwendet aus Konsistenzgründen die gleiche Notation wie die ARM in Tabelle 5.2. Die Dienstreferenz ist wie bei der eGK als 4ByteServiceKey realisiert, der den Speicherort des tnodes angibt.

Rolle	create	read	update	delete	list
Dozent	01	01	10	00	01
Student	01	00	10	00	01
Prüfungsamt	01	01	10	00	00

Tabelle 5.2: Beispiel für eine ARM

create	read	update	delete	list
01	01	10	00	01

Tabelle 5.3: Beispiel für eine ACL

DTT Ein DTT für die elektronischen Prüfungen ist wie folgt aufgebaut:

```
DefType DefaultTicketToolkit {
hashedRND: CHAR (64), //RND mit SHA-256 gehasht
              TicketVerifizierer: byte[],
              TicketSet: byte[],
              ACL: Vector,
              Typ: enum(flush, multiple)
}
```

Bei der eGK ist in der Regel die Anwesenheit der Patientenkarte bei allen Aktionen nötig. Bei den elektronischen Prüfungen ist dies aber nicht der Fall, denn hier muss z.B. ein Zugriff auf die Studentenlösungen durch den Dozenten möglich sein, ohne das der Student anwesend ist. Daraus ergibt sich, dass das DTT in seiner ursprünglichen Form so nicht verwendet werden kann. Denn ein rein rollenbasierter Zugriff existiert bei den Prüfungen nicht, weil die Berechtigungen personenbezogen sind.
Aus diesem Grund wird für das DTT eine Access-Control-List (ACL) eingeführt, wie sie auch bei den PTTs vorhanden ist. Sie wird deshalb auch für die DTTs benötigt, damit ebenfalls für den Eigentümer eines Verzeichnisses oder Datensatzes die Zugriffsrechte festgelegt werden können. Denn die ARM regelt bei der eGK die rollenbezogenen Zugriffe, während sie bei den Prüfungen die Detailrechte vorgibt (siehe Abschnitt 5.4).

PTT Die Vergabe von Berechtigungen ist bei den Prüfungen selbstredend immer personenbezogen. Zum einen können an einer Prüfung nur die Studierenden teilnehmen, für die ein persönliches Ticket-Toolkit vorhanden ist.

Zum anderen dürfen nur die Dozenten bzw. Korrekteure auf die Prüfungslösungen der Teilnehmer zugreifen, die dazu durch die Teilnehmer mittels PTT berechtigt wurden.
Der Aufbau des PTT besteht aus den Elementen des DTT plus zwei Datumsangaben, die für das PTT den Zeitraum festlegen, in dem der PTT gültig sein soll. Außerdem wird das Element Typ durch den Wert *single* erweitert. Dies bedeutet, dass ein Zugriff auf das PTT nur einmal erfolgen darf.
Das Zertifikat ordnet das PTT einer eindeutigen Person zu. Jede Anfrage, die an den Ticketserver gerichtet wird, enthält als Parameter das Zertifikat des Anfragenden. Dadurch kann ein PTT für den Anfragenden gefunden und zurückgegeben werden.

```
DefType PersonalTicketToolkit[] {
               hashedRND: CHAR (64), //RND mit SHA-256 gehasht
              TicketVerifizierer: byte[],
              TicketSet: byte[],
              ACL: Vector,
              Cert.User: Certificate,
              datumBis: TS,
              datumNichtVor: TS,
              Typ: enum(flush, multiple, single)
}
```

5.3.3 Architektur

Die grundlegende Architektur, die sich aus den Anpassungen ergibt, ist in Abbildung 5.13 dargestellt. Der Client (unabhängig ob Prüfungs- oder Dozentenclient) kommuniziert mit dem Prüfungssystem (PS) und dem Ticket-Server (TS) ausschließlich über den Konnektor. In der Abbildung sieht man, dass die Datenhaltung des PS nicht über den Ticketserver realisiert ist. Dies ist dadurch begründet, dass die Anpassungen bei der Einführung eines Sicherheitskonzeptes auf ein Minimum beschränkt werden sollen. Wichtig zu erwähnen ist, dass jedwede Kommunikation, die vom bzw. zum Konnektor gesendet wird, verschlüsselt ist.
Nachfolgend werden die beiden zentralen Komponenten Konnektor und Ticket-Server beschrieben. Damit einher geht die Darstellung des für die Prüfungen angepassten virtuellen, ticketbasierten Dateisystems.

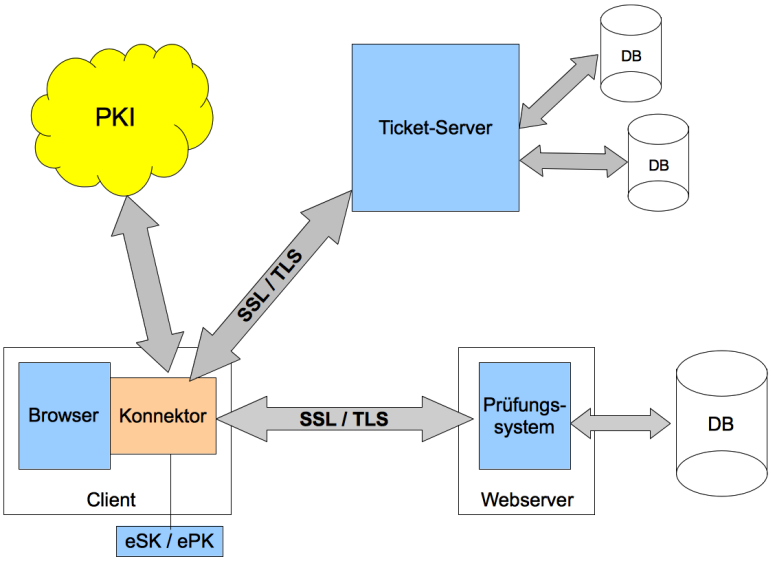

Abbildung 5.13: Grobarchitektur

Konnektor

Der Konnektor besteht aus den Teilkomponenten *Netzwerk*, *Ticket-Server* und *Signatur-/ Verschlüsselung* (siehe Abbildung 5.14). Über die *Netzwerk-Komponente (NW-Komponente)* werden sämtliche Verbindungen zum Browser bzw. Client-Anwendung und zur WAN-Schnittstelle geregelt. Das bedeutet, dass jegliche Kommunikation zwischen dem Client und den Servern über die NW-Komponente erfolgt.
Die *Ticket-Server Komponente (TS-Komponente)* ist für die Kommunikation mit dem Ticket-Server (TS) verantwortlich. Sie kommuniziert mit dem OID-Dienst des TS und ermöglicht die Traversierung des Dateibaumes des Anwenders mit Hilfe des Query- und tnode-Dienstes des TS. Zur Entschlüsselung der Default-Ticket-Toolkits bzw. Personal-Ticket-Toolkits werden diese an die Signatur-/Verschlüsselungskomponente (SV-Komponente) weitergegeben. Die TS-Komponente ist ebenfalls für die Erstellung von tnodes und die Erstellung von Datensätzen (in Verbindung mit der SV-Komponente) verantwortlich.
Die Signatur-/Verschlüsselungskomponente (SV-Komponente) ist für die Erstellung und Verifizierung von Signaturen zuständig und kommuniziert dazu über die NW-Komponente mit der PKI und über eine entsprechende Schnitt-

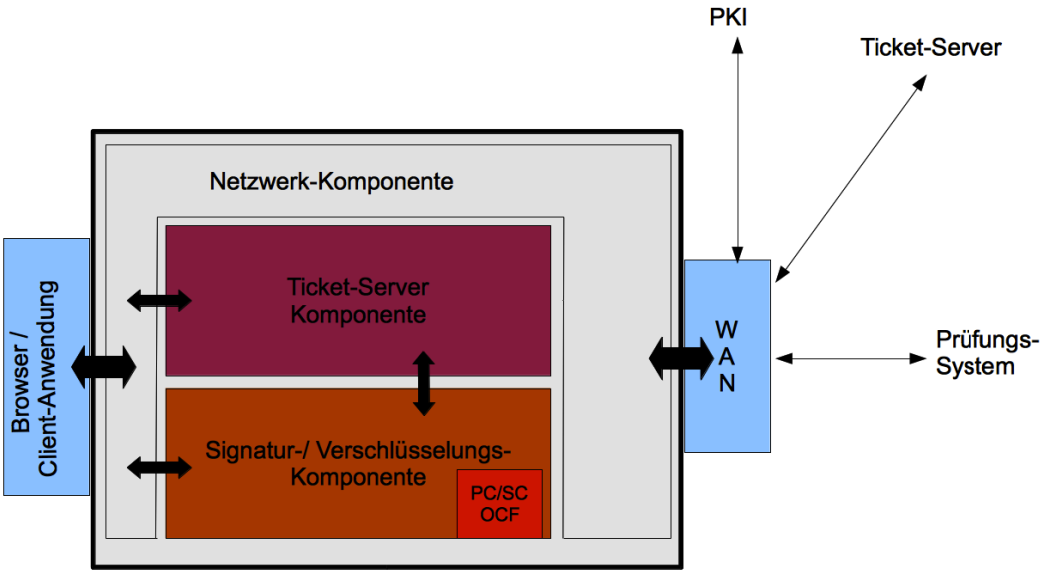

Abbildung 5.14: Konnektor Aufbau

stelle mit der Smartcard des Nutzers. Die in Abbildung 5.14 dargestellte PC/SC bzw. OCF -Schnittstelle dient zur Kommunikation mit einer Smartcard. Das Gleiche gilt für Ver- und Entschlüsselungsoperationen. Die SV-Komponente regelt außerdem die sichere Kommunikation mit der Smartcard des Nutzers.

Ticket-Server

Abbildung 5.15 zeigt den Aufbau des Ticketservers (TS) mitsamt Schnittstellen. Der TS ist zustandslos, d.h., es werden keine Zustandsinformationen gespeichert. Der TS besteht aus den Anwendungsdiensten, der Zugangs- und Integrationsschicht (ZIS) und der Methoden zum Zugriff auf die Datenspeicher. Die ZIS wird unterteilt in die Dienste:

- Query-Dienst (QD): Namensdienst, der die Speicherorte der vRoot verwaltet.
- tnode-Dienst (TND): Verwaltet sämtliche Zugriffe auf die Daten und Verzeichnisse.
- Referenz-Dienst (REF): Verwaltet die Dienstreferenzen (Speicherorte) in Form eines 4ByteServiceKey.

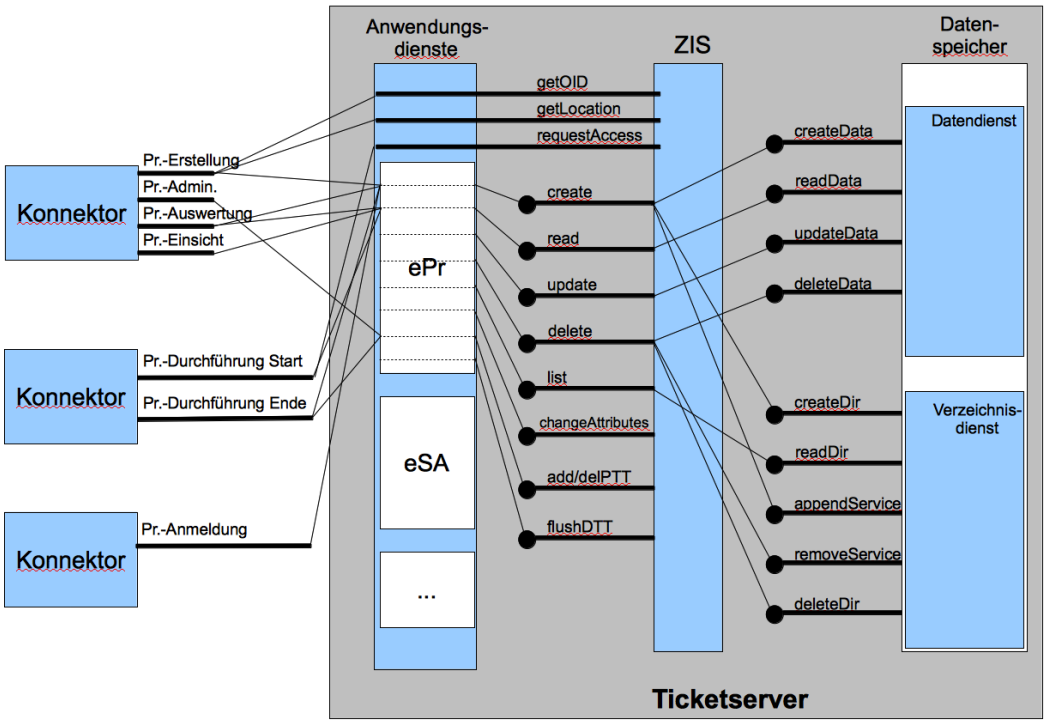

Abbildung 5.15: Dienste des Ticketservers (vgl. [Fra05, S. 175])

- ObjectID-Dienst (OID): Verwaltet und generiert alle ObjectIDs (OID) bzw. DatenIDs (DID).

Die kryptografischen Operationen seitens des Ticketservers werden über entsprechende Methoden realisiert. Ein Möglichkeit wäre der Einsatz eines Hardware-Security-Module (HSM) (siehe u.a. [Fox09]).

5.3.4 Schnittstelle Konnektor - Ticketserver

Zum besseren Verständnis der folgenden Kapitel sei die Definition des *Tickets* noch einmal klargestellt. Ein Ticket besteht im Grunde nur aus der entschlüsselten Zufallszahl RND und der entschlüsselten DirectoryID DID eines Ticket-Bausatzes (siehe Abbildung 5.9). Im Folgenden wird ein Ticket definiert als: $Ticket := (RND, DID, tnodeID, dienstRef)$. Das bedeutet, dass ein Ticket neben RND und DID auch noch aus der tnodeID des tnodes und dem Speicherort (dienstRef) des tnodes besteht.

In Abbildung 5.15 ist u.a. die Schnittstelle des Konnektors zum Ticketserver dargestellt. Die nachfolgenden Operationen können vom Konnektor aus auf dem Ticketserver aufgerufen werden. Dabei werden die Operationen über einen Anwendungsdienst geleitet, der die Verwaltung und Verwendung der ARMs übernimmt (siehe Abschnitt 5.4).
Das Einlösen eines Tickets und die entsprechende Operation ist abhängig von den Rechten, die der aufrufenden Person in der ARM bzw. der ACL eines PTT und DTT zugeteilt sind. Der Ticketserver bietet die nachfolgenden Schnittstellen für die Kommunikation mit dem Konnektor an. Dabei sei erwähnt, dass diese Schnittstellen fast vollständig den Schnittstellen der Zugangs- und Integrationsschicht der eGK entsprechen (vgl. [Fra05, S. 195-218]:

Speicherort des vRoots

getLocation ermittelt den Speicherort des vRoots des Benutzers. In einem vRoot ist die tnodeID auch die entsprechende UserID, also DozentenID bzw. die Matrikelnummer.

4ByteServiceKey getLocation(UserID: char[])

- UserID: Matrikelnummer bzw. DozentenID

Rückgabewert ist der Speicherort des vRoots.

Ticket-Bausatz anfordern

requestAccess fordert einen Ticket-Bausatz zur Erstellung eines zum Datenzugriff berechtigten Tickets an.

TicketBausatz requestAccess(tid: Reference, cert: Certificate)

- tid: Referenz auf den tnode, auf den zugegriffen werden soll. Diese Referenz besteht dabei aus der tnodeID des tnodes und der 4ByteService Kennung des Speicherortes.
- cert: Zertifikat des Aufrufers.

Rückgabewert ist der Ticket-Bausatz des angefragten tnodes. Welcher Ticket-Bausatz (PTT, DTT) zurückgegeben wird, wird anhand des Zertifikates entschieden.

Anlegen eines Verzeichnis oder Datensatz

Verzeichnisse oder Datensätze können mit Hilfe von *create* über den Ticketserver auf den Datenspeichern angelegt werden.

Ticket create(ticketDir: Ticket, tnodeLoc: 4ByteServiceKey, tndata: TNodeSetupData, data: encryptedData, cert: Certificate

- ticketDir: Ticket (mit *create*-Rechten) zu dem Verzeichnis, über das das neue Verzeichnis bzw. Datensatz referenzierbar sein soll (Parent-Directory).
- tnodeLoc: Die Dienstkennung des Dienstes, über den der tnode verwaltet wird.
- tndata: Ein Datensatz, aus dem im Ticketserver ein tnode erzeugt werden kann (s.u.).
- data: Die verschlüsselten Daten, wenn es sich um einen Datensatz handelt. Wenn nur ein Verzeichnis angelegt werden soll, dann wird anstelle der Daten NULL angegeben.
- cert: Zertifikat des Aufrufers.

```
DefType TNodeSetupData {
              oid: OID,
              hashedOID: HashedOID,
              Klassifizierer: char[256],
              tnodeTyp: ('dir', 'data'),
              2ndKey: EncryptedKey,
              datumGueltigkeitsende: TS,
              arm: ARM,
              dtt: TicketToolkit,
              ptt: PersonalTicketToolkit[]
              }
```

Rückgabewert ist ein Ticket zu dem eingestellten Verzeichnis oder Datensatz.

Beim *create* werden die einzelnen tnode-Elemente an den Ticketserver geschickt. So werden z.B. die Datumswerte $datumErstellung$, $datumZugriff$, $datumAenderung$ durch den Anwendungsdienst gesetzt. Initial erhalten diese Datumswerte alle das Datum der Erstellung. Des Weiteren zählt dazu die ARM, die in Abschnitt 5.4 im Rahmen des Rechtemanagement detailliert beschrieben wird.

Lesen eines Datenobjektes

Mit *read* kann ein Datenobjekt (z.B. Prüfungsangaben) aus den Datenspeichern ausgelesen werden. Dabei bezieht sich *read* aber nur auf Datensätze und nicht auf Verzeichnisse. Das „Lesen“ eines Verzeichnisses zum traversieren wird über die *list*-Methode geregelt.

encryptedData read(ticket: Ticket, cert: Certificate)

- ticket: Ticket, das mitsamt des Zertifikates zum Lesen des Datensatzes berechtigt.
- cert: Zertifikat des Aufrufers.

Rückgabewert sind die verschlüsselten Daten.

Löschen eines Datenobjektes

delete löscht das Datenobjekt mitsamt des zugehörigen tnodes.

encryptedData delete(ticket: Ticket, cert: Certificate)

- ticket: Ticket, das mitsamt des Zertifikates zum Löschen des Datensatzes berechtigt.
- cert: Zertifikat des Aufrufers.

Rückgabewert sind die verschlüsselten Daten, um diese evtl. zu archivieren oder aber um im Falle einer versehentlichen Löschung die Löschung rückgängig zu machen.

Aktualisierung eines Datenobjektes oder Verzeichnisses

Mittels *update* kann ein Datenobjekt über den Ticketserver im Datenspeicher aktualisiert werden.

Ticket update(ticket: Ticket, newData: encryptedData, optNewDTT: TicketToolkit, cert: Certificate)

- ticket: Ticket das mitsamt des Zertifikates zum Aktualisieren des Datensatzes berechtigt.
- newData: Die aktualisierten, verschlüsselten Daten.

- optNewDTT: Ein optionales neues DTT, mit dem das existierende DTT ersetzt wird.

- cert: Zertifikat des Aufrufers.

Rückgabewert ist ein Ticket zu den aktualisierten Daten.

Auflisten eines Verzeichnisses

Mittels *list* können die Klassifizierer-Texte und Ticket-Bausätze aller in einem Verzeichnis enthaltenen Datensätze gelesen werden.

(dienstRef, tnodeID, String, TicketBausatz)[ ] list(ticketDir: ticket, query: String, cert: Certificate)

- ticketDir: Ticket für das Verzeichnis, das mit dem Zertifikat zum Auflisten des Verzeichnisses berechtigt.

- query: Query-String, der weitere Argumente für die Abfrage besitzen kann

- cert: Zertifikat des Aufrufers.

Rückgabewerte sind die Klassifizierer-Texte und die Ticket-Bausätze der im Verzeichnis enthaltenen und für den Aufrufer listbaren Datenobjekte. Ob ein Datenobjekt für den Aufruf listbar ist, entscheidet die ARM des DTT und die ACL des PTT (falls es ein PTT für den Aufrufer gibt). Dazu werden zuerst die PTTs eines jeden Datenobjektes nach einem für den Aufrufer ausgestellten PTT durchsucht. Wenn ein PTT gefunden wurde, wird die ACL betrachtet ob ein List-Recht vorhanden ist. Wenn kein List-Recht für das PTT des Aufrufers existiert, wird dieses Datenobjekt nicht angezeigt.

5.3.5 Schnittstelle ZIS - Datenspeicher

Die Schnittstelle zwischen Konnektor und Ticketserver besteht nur über die in Unterabschnitt 5.3.4 aufgeführten Methoden. Somit ist ein Zugriff vom Konnektor auf die Verzeichnis- und Datensätze des virtuellen Dateisystems ausschließlich nur über die ZIS des Ticketservers möglich. In der ZIS wird zwischen Operationen auf Verzeichnisse und auf Anwendungsdaten unterschieden.

Schnittstellen der Anwendungsdaten

Erzeugen eines Anwendungsdatensatzes Das Einstellen von Anwendungsdatensätze erfolgt über die Operation *createData*:

Boolean createData(oid: OID, data: enryptedData)

- oid: Object-ID unter der der Datensatz referenzierbar ist
- data: Einzustellender verschlüsselter Datensatz

Rückgabewert ist true, wenn das Einstellen erfolgreich wahr, sonst false.

Lesen eines Anwendungsdatensatzes Das Auslesen von Anwendungsdatensätze erfolgt über die Operation *readData*:

encryptedData readData(oid: OID)

- oid: Object-ID des zu lesenden Datensatzes

Rückgabewert ist der gespeicherte Datensatz. Falls die Daten aufgrund eines Fehlers nicht gelesen werden können, wird NULL zurückgegeben.

Ersetzen eines Anwendungsdatensatzes Über die Operation *updateData* können bestehende Datensätze ersetzt werden.
encryptedData updateData(oid: OID, dataNew: enryptedData)

- oid: Object-ID des zu ersetzenden Datensatzes
- dataNew: Einzustellender neuer verschlüsselter Datensatz

Rückgabewert ist der Datensatz, der durch den verschlüsselten Datensatz ersetzt werden soll.

Löschen eines Anwendungsdatensatzes *deleteData* löscht den Anwendungsdatensatz aus den Datenspeichern:

encryptedData deleteData(oid: OID)

- oid: Object-ID des zu löschenden Datensatzes

Rückgabewert sind die vor der Löschung gespeicherten Anwendungsdaten.

Schnittstellen der Verzeichnisdaten

Erzeugung eines Verzeichnisdatensatzes Verzeichnisdaten werden über die Operation *createDir* angelegt. Ein neu angelegter Verzeichnisdatensatz ist initial leer.

Boolean createDir(did: OID)

- did: Directory-ID unter der der Verzeichnisdatensatz verwaltet werden soll.

Der Rückgabewert zeigt an, ob der Verzeichnisdatensatz erfolgreich angelegt wurde.

Lesen eines Verzeichnisdatensatzes Verzeichnisdaten werden über die Operation *readDir* aus dem Datenspeicher ausgelesen.

4ByteServiceKey[] readDir(did: OID)

- did: Directory-ID, unter der der Verzeichnisdatensatz verwaltet wird.

Der Rückgabewert besteht aus einer Liste, die angibt, welche Speicherorte für die tnodes des Verzeichnisses existieren.

Löschen eines Verzeichnisdatensatzes Verzeichnisdaten werden über die Operation *deleteDir* aus dem Datenspeicher gelöscht.

4ByteServiceKey[] deleteDir(did: OID)

- did: Directory-ID des zu löschenden Verzeichnisdatensatz.

Der Rückgabewert besteht aus einer Liste der vor der Löschung in dem Datensatz gespeicherten Speicherorte.

Management von Zugriffsrechten und Metadaten

Das Management der Zugriffsrechte eines tnodes darf nur durch den Eigentümer des tnodes erfolgen. Das bedingt also, dass zum Auslesen und Ändern der Zugriffsrechte bzw. Metadaten ein DTT für den Aufrufer existiert und in der ACL des DTTs die entsprechenden Berechtigungen gesetzt sind (siehe Abschnitt 5.4).

Lesen der tnode-Daten *readAttributes* ermöglicht das Auslesen der in einem tnode gespeicherten Verwaltungsdaten (ohne Ticket-Bausätze und Verifizierer).

String readAttributes(ticket: Ticket, cert: Certificate)

- ticket: Ein gültiges Ticket für das DTT, das read-Zugriff auf den tnode erlaubt.
- cert: Zertifikat des Aufrufers

Rückgabewert ist ein XML-Dokument, in dem alle nicht vertrauliche Daten des tnodes zusammengefasst sind. Dazu gehören z.B. auch alle öffentlichen Schlüssel, für die ein PTT ausgestellt wurde.

Modifizieren der tnode Daten Alle Verwaltungsdaten des tnodes können mittels *changeAttributes* geändert werden.

changeAttributes(ticket: Ticket, tnodedata: String, cert: Certificate)

- ticket: Ein gültiges Ticket für das DTT, das update-Zugriff auf den tnode erlaubt.
- cert: Zertifikat des Aufrufers

Austausch des DTT *flushDTT* ermöglicht den Austausch des DTT eines tnodes. Dabei wird auch der Verifizierer geändert.

flushDTT(ticket: Ticket, newDTT: TicketToolkit, opt2ndkey: encryptedKeys, cert: Certificate)

- ticket: Ein gültiges Ticket für das DTT, das update-Zugriff auf den tnode erlaubt.
- newDTT: Neues DTT
- opt2ndkey: optionaler Zweitschlüssel
- cert: Zertifikat des Aufrufers

Hinzufügen bzw. Löschen von Personal-Ticket-Toolkits PTTs können mit den Operationen *addPTT*() bzw. *deletePTT*() zu einem tnode hinzugefügt bzw. gelöscht werden.

addPTT(ticket: Ticket, certHolder: Certificate, ptt: TicketToolkit, cert: Certificate

- ticket: Ein gültiges Ticket, das zum Zugriff auf den tnode berechtigt. Dazu muss der Aufrufer entweder ein create oder update-Berechtigung und ein gültiges DTT besitzen. Außerdem muss er Eigentümer des tnodes sein.
- certHolder: Zertifikat desjenigen, der ein PTT erhält
- ptt: PTT für den tnode
- cert: Zertifikat des Erstellers

Zum Löschen eines PTT wird die Methode *deletePTT*() verwendet, die lediglich den öffentlichen Schlüssel des zu löschenden PTT benötigt.

deletePTT(ticket: Ticket, pukHolder: PublicKey, cert: Certificate

- ticket: Gültiges Ticket, das update-Zugriff auf den tnode ermöglicht.
- pukHolder: Öffentlicher Schlüssel des zu löschenden tnodes
- cert: Zertifikat des Aufrufers

5.4 Rechtemanagement

Die in Unterabschnitt 5.3.4 dargestellen Operationen bedingen, dass der Aufrufer zum Ausführen der Operation berechtigt ist. Dies wird in den Verzeichnissen durch die Access-Control-Lists der TicketToolkits geregelt. Dieser Abschnitt stellt dar, wer wem ein Ticket-Toolkit ausstellen darf und wer welche Rechte bei den einzelnen Operationen besitzt. Die Operationen auf das vtD sind in Abschnitt 5.5 dargestellt.
Die Zugriffs- bzw. Rechtedefinitionen sind selbstredend auf die elektronischen Prüfungen anzupassen. So ist z.B. die Umsetzung der Anforderung, dass der Student seine abgegebene Lösungen nachträglich nicht ändern darf, bzw. im Nachhinein eine neue Lösung abgibt $P69$ durch das Rechtekonzept zu berücksichtigt.

In Abbildung 5.16 ist der Verzeichnisbaum eines Studenten vereinfacht dargestellt. Die Verzeichnisbäume werden durch eine zentrale Instanz der Verwaltung der Hochschule initialisiert. Im Studentenbaum wird dies nur bis zur Anwendungsebene hin realisiert. Ab der Anwendungsebene kann das Prüfungsamt, dem der Studierende zugeordnet ist, die jeweiligen Semesterverzeichnisse anlegen. Innerhalb der Semesterverzeichnisse werden dann die Prüfungsordner mit der jeweiligen PrüfungsID (PID) erzeugt.

Die Verwendung der ARM bei den Prüfungen unterscheidet sich in soweit von der bei der eGK, dass bei der eGK die Verzeichnisstrukturen grundsätzlich fest vorgegeben sind. Es werden durch die Heilberufler nur Datensätze (wie z.B. als eRezept) angelegt. Eine dynamische Veränderung des Dateisystems durch Erzeugung von Verzeichnissen und Unterverzeichnissen etc. wird nur für die Anwendung der elektronischen Patientenakte vorgesehen, die in der für diese Arbeit verwendeten Spezifikation des Lösungskonzeptes als Mehrwertdienst eingeordnet wurde und deren genauere Beschreibung nicht erfolgte.
Bei den Prüfungen sieht dies etwas anders aus: Die Strukturen der Dateisysteme ändern sich öfter als bei der eGK. Die Teilnehmer dürfen bei der Prüfungsdurchführung nur Dateien anlegen, diese aber nicht mehr ändern. Das Prüfungsamt hingegen darf nur Verzeichnisse im Teilbaum *ePrüfungen* erzeugen, aber keine Dateien. Das Prüfungsamt darf aber im Teilbaum *eSA* Verzeichnisse und Dateien anlegen, verändern und ggf. auch löschen. Die Prüfungsangaben hingegen sind für den Dozenten nach dem Start der Prüfungsdurchführung weder zu verändern, noch zu löschen.
Diese unterschiedlichen Berechtigungen sind in der jeweiligen ACL und der ARM abzubilden. Dazu wird die Kodierung der entsprechenden Zugriffsrechte durch 0 und 1 verwendet. Durch diese Kodierung kann bei create, read, update und delete angegeben werden, in welchem Zusammenhang die Berechtigung mit den Dateien bzw. Verzeichnissen stehen:

- create:
  - 00: keinerlei create-Berechtigungen
  - 01: create-Recht zum Anlegen einer neuen Datei
  - 10: create-Recht zum Anlegen eines neuen Verzeichnisses
  - 11: create-Recht zum Anlegen einer Datei und eines Verzeichnisses
- read:
  - 00: keinerlei read-Berechtigungen

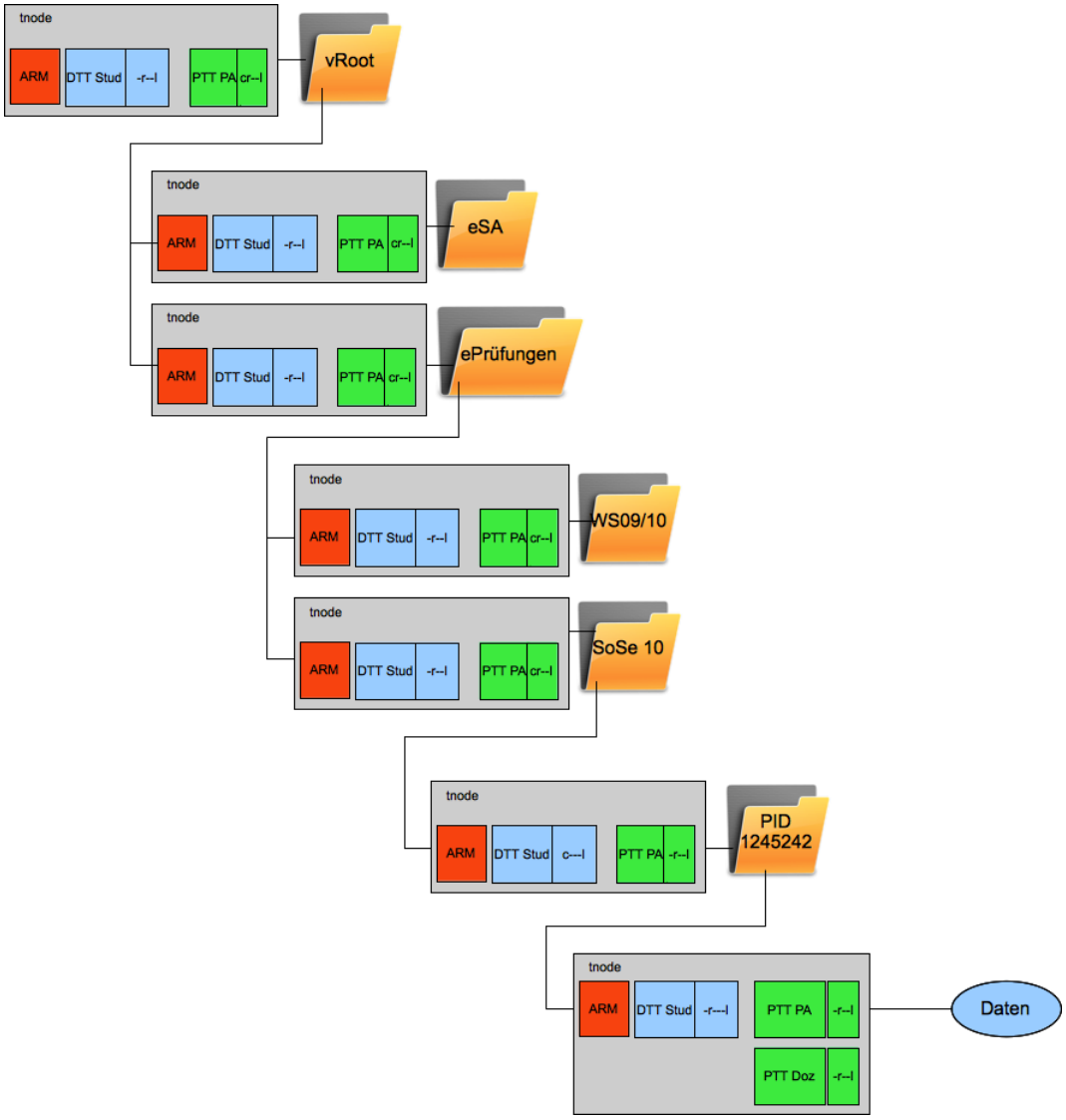

Abbildung 5.16: Verzeichnisstrukturen

  - 01: read-Recht auf Metadaten (readAttributes)
  - 10: read-Recht auf die Datei (falls vorhanden)
  - 11: read-Recht sowohl auf Metadaten als auch auf die Datei

- update:
  - 00: keinerlei update-Berechtigungen auf den aktuellen tnode
  - 01: update-Recht auf Metadaten (addPTT, delPTT)
  - 10: update-Recht auf Daten (falls vorhanden) (updateData)
  - 11: update-Recht sowohl auf Metadaten als auch auf die Daten (addPTT, delPTT, flushDTT, changeAttributes, updateData)

- delete:
  - 00: keinerlei delete-Berechtigungen auf den aktuellen tnode
  - 01: delete-Recht auf Verzeichnis (deleteDir) (falls Verzeichnis leer)
  - 10: delete-Recht auf Datei (falls vorhanden) (deleteData)
  - 11: delete-Recht sowohl auf Verzeichnis als auch auf Datei

Für list ist eine 01-Kodierung nicht notwendig. Dennoch wird in dieser Arbeit die 01-Notation auch für die list-Operation verwendet. 00 steht dabei für keine Berechtigung und 01, 10 und 11 für list-Berechtigung.

5.4.1 Bedeutung der ARM und ACLs

Die Zugriffsregelung auf die tnodes wird über die ACLs geregelt. Die ACLs geben an, welche Berechtigungen für das PTT bzw. DTT existieren. Die ACLs werden bei der Erzeugung eines Verzeichnisses oder Datensatzes im Konnektor mittels *create* angelegt, bzw. beim Hinzufügen oder Löschen von PTTs über *addPTT/ delPTT*.
Die ARM gibt die Maximalberechtigungen zu den Operationen *create*, *read*, *update*, *delete* und *list* für jede Rolle an, die für diesen tnode vergeben werden können. Für den Studentenbaum werden die ARMs der Verzeichnisse *vRoot*, *eSA* und *ePrüfungen* durch einen zentralen eAssessment Dienst fest vorgegeben und können nicht mehr verändert werden.
Zur Verdeutlichung wird die Abbildung 5.17 betrachtet, die einen detaillierteren Ausschnitt aus dem in Abbildung 5.16 dargestellten Dateibaum zeigt. Dargestellt ist das Verzeichnis „PID124252“, das die signierte und verschlüsselte Prüfungslösung des Teilnehmers enthält.

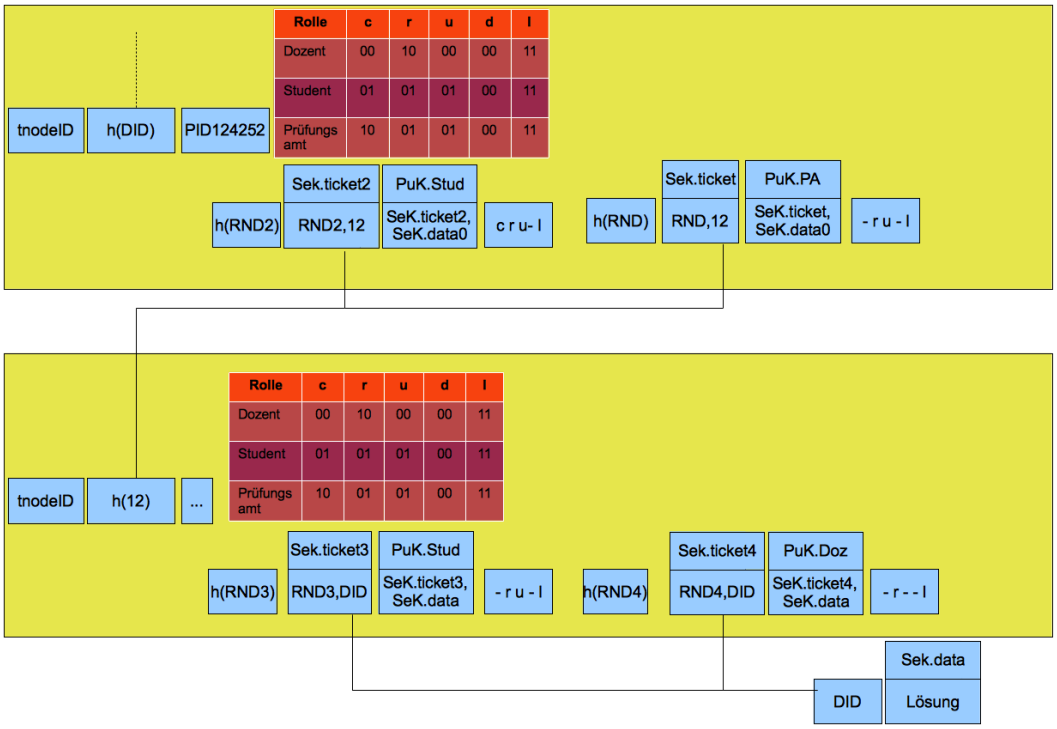

Abbildung 5.17: Detaillierte Ansicht ARM

Die ARM des Verzeichnisses „PID124252“ besagt, dass ein Student ein create-Recht auf eine neue Datei besitzen darf. Dazu auch ein read-Recht auf die Metadaten des tnodes und neben dem list-Recht auch ein update-Recht auf die PTTs. Die ACL des DTTs besagt, dass alle diese Rechte auch dem Studenten zur Verfügung stehen. Das Prüfungsamt besitzt durch das PTT nur ein read- und list-Recht. Ein create-Recht zum Erzeugen eines neuen Verzeichnisses kann dem Prüfungsamt auch vergeben werden, ist jedoch in der ACL nicht angegeben und kann somit durch das Prüfungsamt nicht wahrgenommen werden.
Nach der Durchführung der Prüfung und der anschließenden Signierung und Verschlüsselung der Lösung im Konnektor löst der Student sein Ticket für das Verzeichnis „PID124252“ ein und erzeugt mittels create() einen neuen Datensatz, wobei die einzelnen Elemente des tnodes des neuen Datensatzes (außer der ARM) im Konnektor erzeugt werden. Dazu gehört auch die ACL des neuen DTT des Studenten, die ohne create aber dafür mit read, update und list versehen wird (-ru-l).
Der Anwendungsdienst „ePrüfung“ im Ticketserver registriert anhand des create-Aufrufes des Studenten, dass ein neuer Datensatz angelegt wird. Im tnode Dienst wird dann die ARM des tnodes „PID124252“ verwendet und auf den neu angelegten tnode übertragen. Denn ein create Aufruf eines Studenten im Anwendungsdienst „ePrüfung“ ist nur dann erlaubt, wenn es sich um die Prüfungslösungen handelt. Außerdem werden beim Anlegen eines Datensatzes die Datumswerte des tnodes im Anwendungsdienst erzeugt und alle auf den Zeitpunkt der Erstellung gesetzt. Nur der Wert datumGueltigkeitsende wird im Konnektor angelegt. Welche Zeiträume dabei standardmäßig durch den Konnektor vorgegeben werden, sind sehr unterschiedlich und hängen oftmals auch von den Prüfungsordnungen ab.
Das update-Recht des Studenten auf den Datensatz ist laut ARM auf 01 gesetzt, was bedeutet, dass der Student nur die PTTs für den Zugriff auf den Datensatz bestimmen und verändern kann. Sollten die im Konnektor angegebenen ACLs mehr Berechtigungen enthalten als in der ARM definiert, so werden die einzelnen Berechtigungen im Ticketserver auf den maximal erlaubten Zustand verändert.

Die beschriebene Vererbung der ARM auf das neue Verzeichnis bzw. den neuen Datensatz bietet eine große Sicherheit bzgl. der Zugriffsrechte. Jedoch sind die Berechtigungen somit vollständig im Vorhinein festgelegt. Die Möglichkeit für ein Unterverzeichnis, einem Benutzer mehr Rechte zu vergeben als in seinem übergeordneten Verzeichnis, ist nicht möglich. Eine weitere Variante könnte sein, dass für Verzeichnisse die ARM durch den Ersteller selbst angegeben werden kann. Für die in dem Verzeichnis angelegten Datei(en) je-

doch wird die ARM von diesem Verzeichnis geerbt und kann nicht verändert werden.
Dadurch wird eine höhere Flexibilität bei der Rechtevergabe erreicht, aber gleichzeitig auch eine *Aufweichung* des Sicherheitskonzeptes.

5.5 Operationen auf das virtuelle, ticketbasierte Dateisystem

In diesem Abschnitt werden die Zugriffskontrolle, das Traversieren innerhalb des virtuellen ticketbasierten Dateisystems, das Erstellen eines Verzeichnisses sowie das Anlegen eines Datensatzes (also Prüfungsangabe, Prüfungsbewertung und Prüfungslösung) beschrieben. Dieser Abschnitt soll zur Verständlichkeit der Operationen auf das vtD beitragen, um die doch teilweise komplexen Operationen bildhafter darzustellen.
Für die folgenden Beschreibungen werden daher die Abkürzungen aus Tabelle 5.4 verwendet.

KO	Konnektor
TSK	Ticketserver-Komponente des Konnektors
SVK	Signatur- und Verschlüsselungskomponente des Konnektors
NWK	Netzwerk-Komponente des Konnektors
TS	Ticketserver
ZIS	Zugangs- und Integrationsschicht
QD	Query Dienst deZIS
REF	Referenz-Dienst der ZIS
OID	Objekt-ID Dienst der ZIS
TND	tnode Dienst der ZIS

Tabelle 5.4: Abk. für die Komponenten

5.5.1 Traversieren im Dateibaum

Der hierarchische Aufbau des vtD ermöglicht die gezielten Zugriffe nur auf bestimmte Ebenen des Dateibaumes. In Abbildung 5.18 sind die ersten Schritte der Traversierung in einem Dateibaum beschrieben. Dabei wird der Dateibaum eines Dozenten von der vRoot an bis zum Verzeichnis „ePrüfung“ durchlaufen. Die nächsten Traversierungsschritte wiederholen sich dann.

Damit eine Traversierung überhaupt möglich ist, muss der Speicherort der Wurzelebene (vRoot) festgestellt werden. Dazu wird der Query-Dienst der

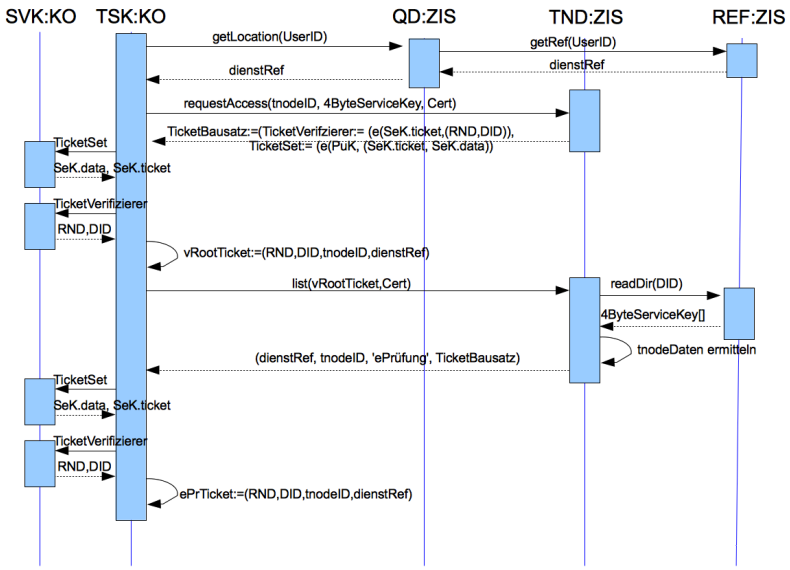

Abbildung 5.18: Sequenzdiagramm Traversierung

ZIS über die Methode *getLocation()* die UserID der Person angegeben. Die UserID eines Dozenten ergibt sich aus seiner DozentenID und die UserID eines Studierenden aus seiner Matrikelnummer. Beide UserIDs sind eindeutig und werden aus diesem Grunde als tnodeID für das jeweilige vRoot verwendet. *getLocation()* ermittelt die Dienstreferenz des vRoots und gibt diese an den Konnektor zurück.
Die TSK des Konnektors ermittelt über *requestAccess()* die Zugriffsinformationen zu dem tnode des vRoots. Der tnode-Dienst der ZIS ermittelt anhand der tnodeID und des Speicherortes den tnode. Anhand des Zertifikates des Aufrufers werden zuerst die PTTs durchsucht, ob es ein PTT für den Aufrufer gibt. Falls ja, dann wird der TicketBausatz dieses PTT zurückgegeben. Wenn kein PTT für den Aufrufer existiert, dann wird der TicketBausatz des DTT zurückgeschickt. Falls es sich bei dem Aufrufer um den Eigentümer des tnodes handelt, dann kann er den TicketBausatz entschlüsseln. Falls der Aufrufer nicht der Eigentümer ist, dann hat er keine Erlaubnis auf den tnode zuzugreifen.
Die einzelnen Elemente des TicketBausatzes werden in der SVK des Konnektors entschlüsselt. Anschließend erfolgt die Zusammenstellung des Tickets (vRootTicket). Der bei *requestAccess()* zurückgegebene tnodeTyp ist im Falle eines vRoots vom Typ „dir“.

Deshalb wird zur weiteren Traversierung die list-Methode aufgerufen. Anhand der im Ticket enthaltenen DID ruft der tnodeDienst (TND) die Methode *readDir()* auf und erhält alle Speicherorte auf denen die tnode-Daten aller Dateien oder Verzeichnisse mit der DID gespeichert sind und für die der Aufrufer eine Zugriffsberechtigung (list-Recht) besitzt. Die TSK erhält dann die tnode-Daten.
Der Aufrufer wählt dann die gewünschte Datei oder Verzeichnis aus und erstellt sich dann das Ticket. Anschließend erfolgt die weitere Traversierung wie oben beschrieben.

5.5.2 Anlegen eines Verzeichnisses oder Datensatzes

In Abbildung 5.19 ist das Sequenzdiagramm zur Erstellung der Prüfungsangabe durch den Dozenten dargestellt.
Nach der Traversierung des Dateibaumes bis zum Verzeichnis „SoSe10“ wird eine neue DID durch den OID-Dienst der ZIS geliefert. Im Konnektor werden die symmetrischen Schlüssel SeK.Data und SeK.Ticket und eine Zufallszahl erzeugt. Anschließend wird der TicketBausatz zum DTT erzeugt und ein optionaler Zweitschlüssel.
Anschließend werden die tnode-Daten zusammengestellt und mittels Aufruf der *create()*-Methode wird ein neuer Datensatz mit den verschlüsselten Daten und anschließend die Vervollständigung der tnode-Daten und deren Speicherung gestartet (siehe Abschnitt 5.3.4 und Abschnitt 5.4).
Das Anlegen eines Verzeichnisses erfolgt ebenfalls über die *create()*-Methode. Allerdings wird serverseitig die *createDir()*-Methode nur dann aufgerufen, wenn der Parameter *data* der *create()*-Methode mit NULL angegeben wird.

5.5.3 Zugriff auf einen Datensatz

Neben der in Unterabschnitt 5.5.1 beschriebenen Traversierung von vRoot aus, muss es für die Prüfungen die Möglichkeit geben direkt auf einen Datensatz oder Verzeichnis zuzugreifen.
Bei der eGK erfolgt ein direkter Zugriff nur dann wenn der Zugreifende ein Ticket für das Datenobjekt auf seiner eGK bzw. HBA hat. Dies dient als Basis zum Aufruf der list-Methode. Ansonsten erfolgt ein Zugriff nur nach vorheriger Traversierung.
Bei den Prüfungen ist aber vor allem ein direkter Zugriff für die Prüfungsdurchführung und die Prüfungsauswertung sinnvoll.
Die Realisierung des direkten Zugriffs wird durch die Verwendung der *requestAccess()*-Methode erreicht. Bei der eGK ursprünglich nur zum Zugriff auf den vRoot verwendet, kann die Methode ohne Anpassung dazu verwendet

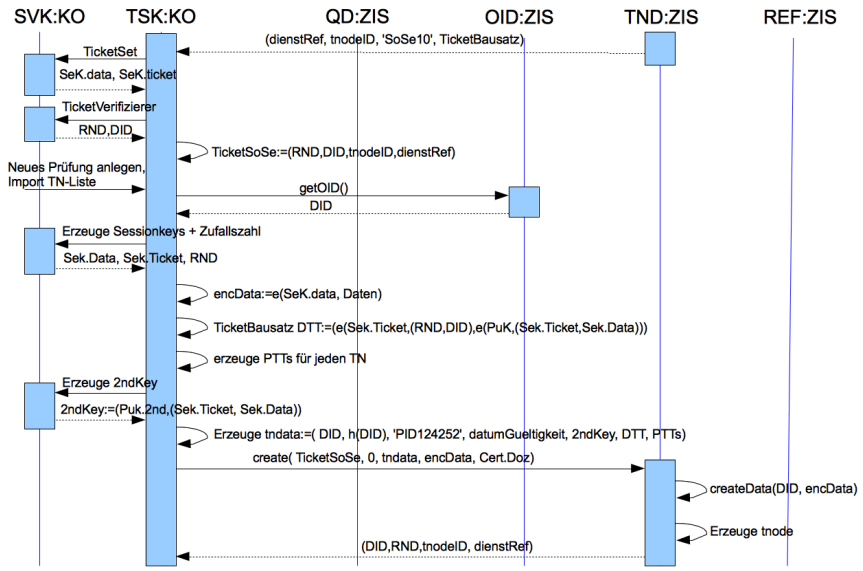

Abbildung 5.19: Sequenzdigramm Bereitstellung der Prüfungsangabe

werden, auch auf weitere tnodes direkt zuzugreifen.
Bei der Erstellung wurde im Prüfungssystem die tnodeID und die Dienstreferenz zur Prüfung abgespeichert. Das Gleiche gilt bei der Abgabe der Lösungen durch den Studenten, der die Zuordnung in einer Tabelle des Prüfungssystems abspeichert.
In Abbildung 5.20 ist das Einlösen einer Berechtigung zum Zugriff auf einen Datensatz beschrieben. Dabei sei vorausgesetzt, dass ein Prüfungsteilnehmer eine Berechtigung in Form eines PTT für die Prüfung generiert bekommen hat, sich am Prüfungssystem angemeldet hat und die Prüfung durchführen möchte.
Der eigentliche Zugriff auf den Datensatz erfolgt durch die *read()*-Methode, die als Parameter das Ticket für den lesenden Zugriff auf die Daten besitzt und das Zertifikat des Aufrufers. Der TND ruft die Methode *readData()* mit der OID des Datensatzes auf und schickt den verschlüsselten Datensatz zurück zum TSK. Dort wird der Datensatz entschlüsselt und kann verwendet werden.

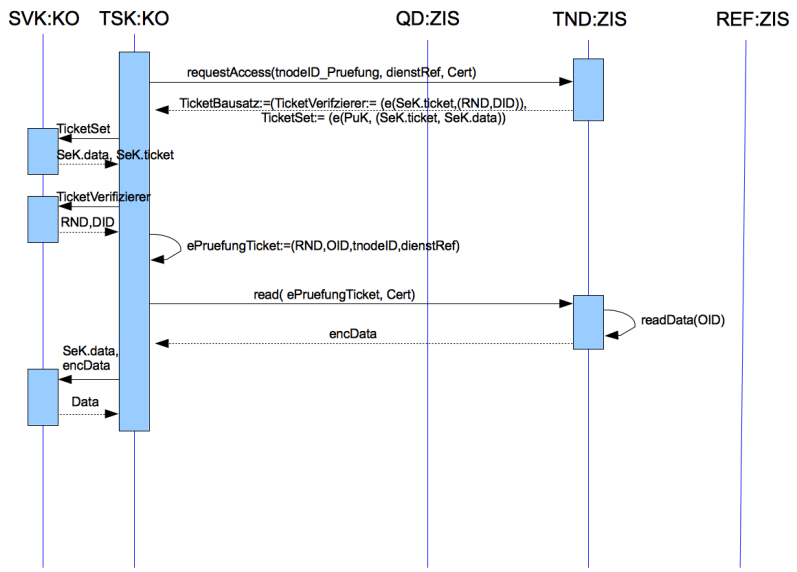

Abbildung 5.20: Sequenzdigramm Zugriff auf Datensatz

5.6 Szenario

Anhand eines Szenarios wird die Verwendung des virtuellen, ticketbasierten Dateisystems für alle Prüfungsphasen beschrieben. Das folgende Szenario beschreibt den kompletten Ablauf einer elektronischen Prüfung mittels verteiltem, ticketbasierten Dateisystem.

5.6.1 Prüfungserstellung

Die Studierenden melden sich bei ihrem Prüfungsamt zur Prüfung an. Die Prüfungsämter schicken nach Ablauf der Anmeldefrist die Listen mit den Teilnehmern an den Dozenten. Die Liste beinhaltet neben der PrüfungsID, Prüfungsdatum, Matrikelnummern auch das Zertifikat des jeweiligen Teilnehmers.
Die Prüfungsämter erzeugen dann im aktuellen Semester-Verzeichnis des Studentenbaums bei der Prüfungsanmeldung ein neues Verzeichnis für die Prüfung. Als Klassifizierer kann dabei die PrüfungsID dienen - oder falls es keine eigene ID für Prüfungen gibt - die VeranstaltungsID. Das Anlegen von Verzeichnissen durch das Prüfungsamt im Dateisystem des Studenten ist deshalb möglich, weil das virtuelle, ticketbasierte Dateisystem kein Owner-Konzept

besitzt.
Des Weiteren sei vorausgesetzt, dass der Dozent die Prüfung P im Prüfungssystem (PS) fertig zusammengestellt und in Form eines XML-Dokumentes an den Konnektor des Dozenten geschickt hat[6].
In der Prüfungserstellungsphase stellt der Dozent den für die Prüfung ordnungsgemäß angemeldeten Studierenden ein „TicketToolkit“ aus. Wird dann durch einen Studierenden aus dem TicketToolkit ein Ticket erzeugt und eingelöst, so kann er die Prüfung durchführen. Der Ablauf der Prüfungserstellung durch den Dozenten kann dann grob wie folgt beschrieben werden:

1. Traversierung des Dateibaumes des Dozenten bis zum aktuellen Semester (siehe Unterabschnitt 5.5.1)

2. Verzeichnis mit der PruefungsID als Klassifizierer anlegen (siehe Unterabschnitt 5.5.2)

3. Unterverzeichnis „Angabe“ anlegen mit PTTs für die Teilnehmer (siehe Unterabschnitt 5.5.2). Die ACLs der PTTs werden dabei auf die read-(10) und list-Berechtigung gesetzt. (siehe Abschnitt 5.4)

4. Prüfung P signieren, verschlüsseln und in „Angabe“ bereitstellen (siehe Unterabschnitt 5.5.2). Die zurückgegebene tnodeID und Dienstreferenz werden im Prüfungssystem den teilnehmenden Studenten zugeordnet.

Durch die Verwendung der qualifizierten digitalen Signatur für die Prüfung P kann der Dozent seine Prüfungsangabe nicht mehr abstreiten (P82) und erfüllt die Anforderung der Formvorschrift (P11). Bei der Prüfungserstellung ist es wichtig, die Zuordnung zwischen Prüfung und Fragen nach der erfolgreichen Erstellung und Speicherung der Prüfung im PS zu löschen. Denn nur so kann verhindert werden, dass bei einem unrechtmäßigen Zugriff auf das PS die Prüfungsangaben vor der eigentlichen Durchführung bekannt werden. Einzig die Zuordnungen Prüfung und Teilnehmer sowie die Zuordnung Prüfung, tnodeID sowie der Speicherort (Dienstreferenz) der Prüfung muss im PS vorhanden sein, um die Lösungen der Teilnehmer der Prüfung eindeutig zuordnen zu können (P51, P52).

[6]Viele Prüfungssysteme verwenden das QTI-Format, das auf dem XML-Standard beruht. QTI (Question & Test Interoperability) ist ein vom IMS Global Learning Consortium definiertes Datenformat, mit dem Tests und Quizzes zwischen verschiedenen Anwendungen ausgetauscht werden können.

Masterkey

Bei der Erstellung der Prüfung kann der Dozent nach der Signierung und Verschlüsselung die Prüfung mit einem zweiten Schlüssel (Master-Key) noch einmal verschlüsseln. Dies hat zur Folge, dass selbst diejenigen, die ein gültiges Ticket haben, die Prüfungsangaben erst dann sehen können, wenn sie den entsprechenden Master-Key kennen.
Die Verwendung des Master-Key bei der Verschlüsselung der Prüfung ist optional. Jedoch ist bei den meisten Prüfungssystemen genau diese Funktion bereits vorhanden. D.h., der Dozent gibt den anwesenden Studierenden den Masterkey (MK.Data) bekannt oder aber der Dozent gibt den Masterkey ein und schaltet somit die Prüfung zur Durchführung frei. Eine Durchführung der Prüfung ist somit nur mit der Kenntnis der beiden Schlüssel Sek.data und MK.data möglich und erfüllt damit die Anforderung der Rechtzeitigkeit der Prüfungsangaben (P62).

Einbinden von Medien

Die Einbindung von externen Medien in die Prüfungsfragen ist auch in einem signierten XML-Dokument möglich. Die externen Mediendateien werden dabei in Form einer URI ins Dokument eingebunden und die URI wird mit Hilfe von XML-Signature mit signiert.

Speicherort bestimmen

In Unterabschnitt 5.5.2 bestimmt der Dozent wo er die Prüfung bereitstellt: Entweder auf einem „eigenen“ Server oder aber auf einem Server, der vom Ticketserver verwaltet wird. Wird die Prüfung auf einem eigenen Server abgespeichert, so muss dieser beim Referenz-Dienst (mittels *appendService()*) eingetragen werden. In diesem Fall muss der Dozent aber die Verfügbarkeit des eigenen Servers während der Prüfung sicher stellen. Die Möglichkeit die Daten auf eigenen Servern zu speichern, erfüllt zumindest in Teilen die Datenschutzanforderungen (D11, D21, D22)

Alle Schreib-/ Aktualisierungsvorgänge auf die Datenhaltung werden als Transaktion durchgeführt. Das bedeutet, die Transaktion ist erst dann angeschlossen, wenn alle Elemente aktualisiert sind. Somit wird die Vollständigkeit der Daten gewährleistet (P63, P64).

5.6.2 Prüfungsdurchführung

Zur Prüfungsdurchführung authentifizieren sich die Teilnehmer am Prüfungssystem (PS). Dann wählen die Teilnehmer die Prüfung P im PS zur Durchführung aus. Durch die in der Prüfungserstellung angelegte Zuordnung zwischen tnodeID, Speicherort und PruefungsID im PS kann der für die Prüfung P erforderliche tnode ermittelt werden.
Anschließend erfolgt, wie bei jedem Zugriff auf einen tnode, die Überprüfung ob der tnode freigegeben ist oder nicht. Dazu werden durch die im tnode-Dienst der ZIS die Felder DatumBis und DatumNichtVor mit dem aktuellen Datum und Zeitwert verglichen. Wenn DatumBis und DatumNichtVor kleiner als das aktuelle Datum sind, dann wird der tnode nicht freigegeben. Ebenso wird der tnode nicht freigegeben, wenn die ACL des PTT keinen lesenden Zugriff auf den Datensatz erlaubt.
Das mehrfache Durchführen der Prüfung wird dadurch verhindert, dass der *Typ* des PTT eines jeden Studenten auf „single“ gesetzt ist. Das bedeutet, dass das Ticket nur einmal eingelöst werden darf. Anschließend wird dann das PTT gelöscht. Sollte aufgrund von technischen Problemen ein erneuter Zugriff auf den tnode erfolgen müssen, dann kann der Dozent dem betroffenen Studenten ein neues PTT ausstellen (P65).
Wenn ein Zugriff erlaubt ist, dann wird das Ticket eingelöst und es kann die Prüfung auf den Client geladen werden. Dazu benötigt der Teilnehmer noch den Masterkey MK.data, der durch den Teilnehmer kurz vor der Prüfung eingegeben wird. Anschließend wird die Prüfung entschlüsselt und die Signatur der Prüfung wird überprüft. Wenn die Signatur korrekt ist, dann wird ein PTT für den Dozenten erstellt und evtl. weitere für Korrekteure wobei der Teilnehmer bestimmen kann welcher Korrekteur seine Lösungen sehen kann und welcher nicht. Es ist sogar möglich, dass der Teilnehmer keinen Korrekteur und keinen Dozenten dabei auswählt. Dies sollte jedoch nur unter einer Bestätigung mit Hinweistext möglich sein. Dies wäre dann mit dem Nichtabgeben bzw. dem Vernichten der Papierlösungen vergleichbar (D11).
Die Durchführung kann erfolgen: Der erstellte SeK.data wird im Konnektor zwischengespeichert und die Durchführung der Prüfung beginnt. Die Lösungen werden mit dem SeK.data verschlüsselt in der Datenbank des Prüfungssystems abgelegt. Nach Klausurende erhält der Teilnehmer eine Übersicht über seine gemachten Angaben. Diese Angaben werden dann vom Teilnehmer elektronisch signiert und in ihrer Gesamtheit mit dem Sek.data verschlüsselt. Die signierte und verschlüsselte Prüfungslösung wird auf Servern des TS gespeichert (P64, P81). Die Speicherung der Prüfungslösung eines Teilnehmers erfolgt also nicht in der Datenbank des Prüfungssystems. Dies liegt darin begründet, dass eine solche Speicherung der Prüfungslösung einen größeren

Änderungsbedarf am PS bedeuten würde. Denn zum einen macht eine Speicherung eines signierten Dokumentes (und damit auch der Prüfungslösung) in einer Datenbank nur dann Sinn, wenn die Gültigkeit der Signatur vor der Speicherung überprüft wird, damit nur Dokumente mit einer gültigen Signatur abgespeichert werden. Jedoch werden in den herkömmlichen Prüfungssystemen relationale Datenbanken eingesetzt und die Lösung zu einer Aufgabe jeweils als ein Datensatz abgespeichert, damit das PS die Lösungen automatisch korrigieren kann. Somit würde das als ganzes signierte Dokument aufgeteilt und die Signatur wäre ungültig (siehe [HK03]).
Um ein Maximum an Sicherheit zu gewährleisten, könnten die im PS abgespeicherten Lösungen nach erfolgreicher Speicherung der signierten Lösungen gelöscht werden.

Spezialfall: Defekte / Fehlende eSK

Alle Operationen bei der Prüfungsdurchführung basieren auf der Anwesenheit der eSK. Für den Fall, dass ein Teilnehmer z.B. eine defekte Karte zur Prüfung mitbringt wird der folgende Ablauf gestartet:
Vor der Durchführung wird dem Prüfungsverantwortlichen (PV) durch den Dozenten ein PTT zur Prüfung ausgestellt. Wenn ein Teilnehmer über eine defekte oder fehlende eSK verfügt, dann kann der PV mit seiner Karte die Prüfungsfragen für den Teilnehmer entschlüsseln. Dabei wird vor Beginn der Prüfung die PV-Karte anstelle der eSK verwendet. Der Konnektor registriert anhand des Zertifikates von PV, dass die Authentisierung nicht über einen Teilnehmer erfolgt, sondern über den PV. Nach der PIN-Eingabe des PV wird das Ticket zur Prüfung eingelöst und der Konnektor kann die Prüfung für den Teilnehmer entschlüsseln.
Eine weitere Möglichkeit wurde in Unterabschnitt 5.3.2 beschrieben. Dabei würde der Zugriff des PV über das DTT laufen. Der PV erhält dabei vom Dozenten ein Ticket für das DTT und den SeK.Data auf seine Smartcard. Dieses Ticket hat der Dozent bei der Erstellung des Verzeichnis-Datensatzes für die Prüfung als Rückgabewert erhalten (siehe Unterabschnitt 5.5.2). Das Ticket bestehend aus SeK.ticket, RND, Speicherort und tnodeID wird zusammen in einem geschützten Bereich der Smartcard abgespeichert. Der Konnektor des Prüfungsclients registriert die fehlende eSK und versucht auf die ePK des PV zuzugreifen. Der PV authentifiziert sich gegenüber seiner ePK und der Konnektor liest dann das Ticket aus. Anschließend ruft dann der Konnektor (TSK) die Methode *read()* auf, erhält die Prüfungsangabe (siehe Unterabschnitt 5.5.3), entschlüsselt und verifiziert diese und stellt sie dem Teilnehmer zur Verfügung.
Der Konnektor erzeugt dann direkt mittels PV-Karte einen Session-Key

Sek.PV und verschlüsselt damit die Prüfungsangaben des Teilnehmers. Der Session-Key Sek.PV wird auf der PV-Karte abgespeichert in Zuordnung zur Matrikelnummer des Teilnehmers.
Die dritte Möglichkeit ist die Anwesenheit des Dozenten bei der Durchführung, der dann über sein DTT die Prüfung für den Teilnehmer freigeben kann. Auch hier wird der Session-Key zur Verschlüsselung der Prüfungslösungen des Teilnehmers auf der ePK des Dozenten in Verbindung mit seiner Matrikelnummer gespeichert.
Nach Klausurende werden die gesamten Prüfungslösungen samt Prüfungsangaben nicht vom Teilnehmer signiert (aufgrund der fehlenden eSK) sondern ausgedruckt und vom Teilnehmer unterschrieben. Die Auswertung durch den Dozenten erfolgt dann durch Entschlüsselung der Prüfungslösungen mit SeK.PV, der zuvor vom PV an den Dozenten übergeben wurde.
Die Lösungen des Teilnehmers werden dann ausgedruckt und vom Teilnehmer unterschrieben. Die elektronischen Prüfungslösungen werden dann ausgewertet und der unterschriebene Ausdruck wird archiviert.

5.6.3 Prüfungsauswertung und -einsicht

Nach der Anmeldung des Dozenten am PS, wird die Auswertung über die Prüfungslösungen von P durch den Dozenten gestartet. Die Lösungen der Studierenden liegen in verschlüsselter und signierter Art in den Datenspeichern des Ticketservers vor. Eine Möglichkeit wäre es, die im Ticketserver gespeicherte Version auf den Client des Dozenten zu laden, zu entschlüsseln und dann diese Daten in die Datenbank des PS einzuspielen und auswerten zu lassen. Aufgrund der Verwendung eines XML-Standards wie QTI ist dies möglich.
inwieweit der QTI-Standard jedoch in der Lage ist, alle zur Zeit möglichen Aufgabentypen zu unterstützen ist fraglich, die Beantwortung der Frage soll aber nicht Bestandteil dieser Arbeit sein (siehe dazu u.a. [PF07, Beh08]). Allerdings ist die Verwendung von QTI nicht unbedingt verpflichtend. So könnte bspw. auch eine eigens entworfene XML-Struktur verwendet werden, die speziell auf das PS zugeschnitten ist [Beh08].
Eine weitere Möglichkeit ist es, die im PS gespeicherten Lösungen nach der Durchführung nicht zu löschen sondern über diese Angaben die Auswertung zu starten. Um die Angaben eines jeden Teilnehmers auswerten zu können, wird der jeweilige symmetrische Sek.data benötigt. Dieser individuelle Session-Key wurde bei der Durchführung verwendet um die Angaben des Teilnehmers zu verschlüsseln. Dieser SeK.data ist im PTT des Dozenten enthalten. Um alle PTT zu finden, die für den Dozenten/ Korrekteur zur Prüfung P erstellt wurden, wurde im PS die Zuordnung PrüfungsID, Matrikelnum-

mer, tnodeID und dienstRef abgespeichert. Diese Zuordnung beinhaltet alle Teilnehmer der Prüfung P, die eine Lösung abgegeben haben.
Anschließend wird dann der SeK.data des Teilnehmers in das PS übertragen. Die Übertragung zum PS ist zwar verschlüsselt, aber im PS muss der SeK.data unverschlüsselt vorliegen um die Lösungen zu entschlüsseln. Dies birgt in so fern ein Sicherheitsrisiko mit sich, weil zu diesem Zeitpunkt neben dem unverschlüsselten SeK.data auch die tnodeID aus der Zuordnungsliste existiert. Die Kombination der beiden Werte, könnte einem Angreifer zumindest theoretisch den Zugriff und die Entschlüsselung des auf dem TS gespeicherten Prüfungslösung geben.
Nachteilig würde sich auch die doppelte Speicherung der Prüfungslösungen auswirken, die der Datenschutzanforderung der „Datenvermeidung“ widerspricht. Die im TS gespeicherten signierten Prüfungslösungen würden dann nur im Hinblick auf eine mögliche Anfechtung des Ergebnisses der Bewertung durch den TN herangezogen.
Deshalb ist die Variante, dass nach der Speicherung der signierten und verschlüsselten Lösungen auf dem TS die Lösungen im PS gelöscht werden, die sichere und elegantere. Zur Auswertung werden dann die entschlüsselten und verifizierten Lösungen der Studierenden in das Prüfungssystem anonymisiert importiert und ausgewertet. Nach der Korrektur der Prüfungslösung, werden die Prüfungsbewertungen im Dateibaum des Dozenten signiert und verschlüsselt abgelegt:

1. Traversierung bis Verzeichnis „PrüfungsID“ (Unterabschnitt 5.5.1)
2. Erstellung Verzeichnis „Auswertung“
3. Erstellung Verzeichnis „h(Matrikelnummer)“ nur mit DTT
4. Signierung, Verschlüsselung und Bereitstellung der Prüfungsbewertung der Teilnehmer (Unterabschnitt 5.5.2). Die Prüfungsbewertung enthält dabei neben der bewerteten Lösung auch die Prüfungsangabe der Teilnehmer.

Dieser Vorgang wiederholt sich für alle Teilnehmer der Prüfung, deren Lösungen durch den Dozenten bewertet werden. Nach der Korrektur aller Lösungen zur Prüfung, können alle Lösungen zur Prüfung im PS gelöscht werden. Allerdings sind diese Daten durch die Anonymisierung jetzt auch für statistische Zwecke geeignet.
In Punkt 3 wird beschrieben, dass für den jeweiligen Daten-tnode „h(Matrikelnummer)“ nur ein DTT für den Dozenten aber kein PTT für den Teilnehmer generiert werden soll. Das bedeutet, der Teilnehmer kann nur dann die korrigierte Prüfung ansehen, wenn zu diesem Zeitpunkt der Dozent/ Korrekteur

mit seiner Karte anwesend ist (P91). Dazu traversiert der Dozent einfach bis zu dem entsprechenden tnode des Teilnehmers und zeigt dem Teilnehmer die ausgewertete Prüfung.
Allerdings kann der Dozent auch PTTs für die Prüfungsauswertungen vergeben. Dabei erhält jeder Teilnehmer zu seiner Bewertung ein PTT mit einer eingeschränkten Gültigkeitsdauer und der Teilnehmer kann dann innerhalb dieser Gültigkeitsdauer seine Bewertung anschauen.

5.7 Fazit

Das virtuelle ticketbasierte Dateisystem (vtD) des Lösungskonzeptes der eGK kann auf die Anforderungen der elektronischen Prüfungen angepasst werden. Das angepasste vtD ermöglicht die Umsetzung der Anforderungen:

- P11 (Rechtliche Gleichstellung Papier <-> elektronisch)
- P51 (Eindeutige Zuordnung Teilnehmer-Prüfung)
- P52 (Zuordnung Teilnehmer-Prüfungslösungen)
- P62 (Rechtzeitigkeit der Prüfungsangaben)
- P63 (Vollständigkeit der Prüfungsangaben zu Prüfungsbeginn)
- P64 (Vollständigkeit der Prüfungslösungen nach Durchführungen)
- P65 (Verhinderung der Mehrfachdurchführung bzw. gleichzeitiger Mehrfachlogin eines Accounts)
- P73 (Anonyme/ anonymisierte Bewertung der Lösungen)
- P81 (Nichtabstreitbarkeit der Prüfungslösungen durch den Teilnehmer)
- P82 (Nichtabstreitbarkeit der Prüfungsangaben durch den Dozenten)
- P91 (Einsicht in die Prüfungsbewertung durch den Teilnehmer)
- D11 (Jeder Nutzer muss „Herr seiner Daten“ bleiben)
- D21 (Erhobene Personendaten sind nur für Prüfungszweck zu verwenden)
- D22 (Statistische Analyse der Prüfungsdaten ermöglichen)

Die Anforderung P61 (Eindeutige Identifizierung der Teilnehmer) ist nur dann vollständig durch das vtD umzusetzen, wenn biometrische Merkmale (z.B. Fingerprint) zur Authentifizierung des Teilnehmers gegenüber seiner eSK eingesetzt werden. Eine Weitergabe einer eSK ohne biometrische Merkmale an eine andere Person ist zwar möglich, würde aber im Normalfall bei der Anwesenheitskontrolle durch die Aufsichten bemerkt. Durch den Einsatz von biometrischen Merkmalen zur Authentifizierung, könnten dann auf die Überprüfung der Anwesenheit verzichtet werden und nur die reine Aufsichtskontrolle muss erfolgen.
Bei der Umsetzung des vtD ist der Konnektor das „Sicherheitszentrum“ des Konzeptes. Hier finden die Ver- und Entschlüsselungen sowie die Verifizierung und Erstellung von Signaturen durch Anbindung an die Smartcards statt. Deshalb wurde auch bei der Gesundheitskarte auf einen Hardware-Konnektor gesetzt. Dieser ist verplombt und darf nur von zertifizierten Herstellern vertrieben werden. Eine Umsetzung in Software hat zur Folge, dass die Konnektor-Software gegen Manipulationen geschützt werden muss. Die Verwendung einer gesicherten Umgebung auf einem USB-Stick o.ä. würde keine Installation des Konnektors auf den Prüfungsclients erfordern und somit die Gefahr einer Manipulation nahezu ausschließen (siehe Abschnitt 6.1).
Die Anforderungen P68 (Nachvollziehbarkeit des Prüfungsverlaufes) und P92 (Archivierung) sind so direkt nicht durch das in Kapitel 5 dargestellte vtD umzusetzen. Zur Umsetzung dieser Anforderungen müssen weitere Maßnahmen ergriffen werden (siehe Kapitel 6).
Des Weiteren ist zu überlegen wie die Umsetzungen des Konnektors und des Ticketservers konkret realisiert werden und in die bestehende Landschaft eines Prüfungssystems integriert werden können (siehe Abschnitt 7.5).

Kapitel 6

Sicherheitskonzept für elektronische Prüfungen

In diesem Kapitel werden alle Maßnahmen zur Realisierung eines Sicherheitskonzeptes zusammengeführt. Das Gesamtkonzept beschreibt alle technischen, administrativen und formale Maßnahmen wobei sich hier nur auf Richtlinien beschränkt wird. Auf juristische Formulierungen für Gesetze bzw. Gesetzesänderungen wird in dieser Arbeit verzichtet.

6.1 Gesamtkonzept

In Kapitel 5 wurde das virtuelle, ticketbasierte Dateisystem (vtD) beschrieben, das eine Vielzahl der Sicherheitsanforderungen bereits realisiert. Jedoch sind zwei wichtige technische Anforderungen $P68$ und $P92$ durch das vtD allein nicht umsetzbar. Dazu sind Erweiterungen des bisherigen Konzeptes (siehe Abbildung 5.13) durch Komponenten bzw. Funktionalitäten, die die Dokumentation und Archivierung ermöglichen nötig. Das Konzept mit den Erweiterungen ist in Abbildung 6.1 dargestellt.
Dieses Gesamtkonzept orientiert sich nicht an einem bestimmten Prüfungssystem, sondern stellt ein vom Prüfungssystem unabhängiges Sicherheitskonzept dar. Das Konzept ist daher durch verschiedene „Bausteine“ realisiert, deren Verwendung sich an das jeweilige Prüfungssystem anpassen lassen.

6.1.1 Protokollierung

In Unterabschnitt 3.2.9 wurde die Einführung eines Protokollservers vorgeschlagen, der u.a. den Zeitpunkt der Durchführung, IP- Adresse des Teilnehmerrechners, Korrekteure, Aufsichten und die Protokolldaten eines jeden

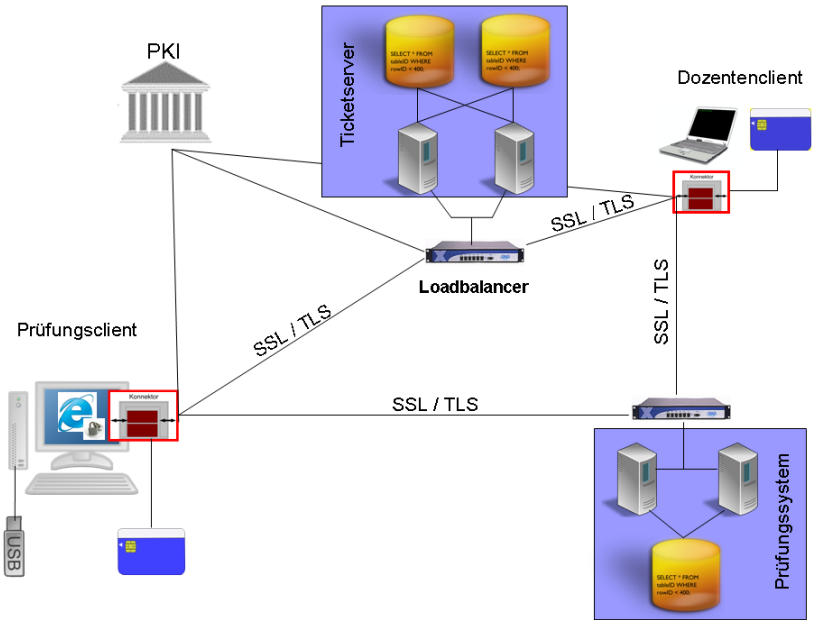

Abbildung 6.1: Gesamtarchitektur

Teilnehmers übermittelt bekommt. Die Protokollierung der Aktivitäten auf Clientseite ist deshalb wichtig, weil nur diese Aktivitäten vom Teilnehmer letztendlich zu verantworten sind.
Die clientseitige Speicherlösung dient der Sicherung der Protokoll- und Prüfungsdaten während der Prüfungsdurchführung. Dazu wird die Netzwerk-Komponente des Konnektors um die Funktionalität der Protokollierung sämtlicher Aktivitäten die über den Konnektor stattfinden erweitert.
Zu den zu protokollierenden Client-Daten gehören:

- PrüfungsID
- Matrikelnummer oder ID der elektronischen Studierendenkarte (falls diese dem Teilnehmer eindeutig zugeordnet werden kann)
- IP-/ MAC-Adresse des Prüfungsrechners
- Zeitstempel
- Bezeichnung der Aktion (Login, Start, Durchführung, Ende, Logout)
- HTTP-Anfrage (Request) bzw.
- HTTP-Antwort (Response)

- Kommunikationsinformationen vom Konnektor wie z.B. Servererreichbarkeit

Die anfallenden Daten sind personenbezogen (Matrikelnummer) bzw. personenbeziehbar (ID der eSK). Somit unterliegen sie auch den datenschutzrechtlichen Anforderungen, was eine datenschutzkonforme Protokollierung notwendig macht (vgl. [Kno06]). Somit ist sicher zu stellen, wer diese Daten, wie und wann zu welchem Zweck einsehen kann[1].
Der Idealfall wäre der, dass der Teilnehmer auch in diesem Falle „Herr seiner Daten“ bleibt, wobei sich dies nur auf das *wer darf wann auf die Protokolldaten zugreifen* beschränkt. Denn der Teilnehmer darf die Protokolldaten im Nachhinein nicht verändern und im Idealfall auch nicht löschen können. Daraus ergibt sich also die Notwendigkeit während der Durchführung die Protokolldaten nur clientseitig vorzuhalten und nach der Durchführung die Daten so abzuspeichern, dass nur der Teilnehmer bestimmt wer Zugriff auf die Daten haben kann.
Die Technologie der clientseitigen Speicherlösung kann dabei von einem USB-Stick über eTokens mit Speicherfunktionalität bis zur Speicherung der Protokolldaten auf der eSK reichen. Allerdings sind die Smartcards für einen permanenten und performanten Lese-/Schreibzyklus nicht ausgelegt. Des Weiteren liegen die Speicherkapazitäten der handelsüblichen Smartcards im Kilobytebereich, was für eine Protokollierung evtl. nicht ausreicht. Speicherkarten mit größerer Speicherkapazität (im Megabyte-Bereich) sind zum Zeitpunkt der Arbeit noch sehr teuer. Außerdem bleibt das Problem der Performance beim Zugriff auf die Karte.
eTokens mit Speicherfunktionalität wären eine Variante um sowohl die Signaturen als auch die Protokollierungsfunktion umzusetzen (siehe u.a. [Sch06a]). Des Weiteren sind keine Smartcard-Leser zu verwenden und die Kommunikation mit dem Client würde über USB laufen. Wie aber bereits in Abschnitt 4.7 angedeutet, wäre dann noch zusätzlich ein Lichtbildausweis für die Studierenden zur visuellen Authentifizierung nötig.
Als einfachste und kostengünstigste Variante käme dann neben einer eSK ein handelsüblicher USB-Stick in Frage auf dem die Protokolldaten während der Durchführung abgespeichert werden. Allerdings muss dann jeder Prüfungsrechner mit einem eigenen USB-Stick ausgestattet werden und diese müssten dann im Vorfeld der Prüfung durch die Aufsichten an den Prüfungsrechnern installiert werden. Jedoch ist ein USB-Stick für die Prüfungsdurchführung multifunktional einsetzbar (siehe Unterabschnitt 6.1.2, Abschnitt 6.2)
Die Netzwerkkomponente des Konnektors muss um eine Protokollfunktiona-

[1] http://www.thueringen.de/datenschutz/datenschutz/technisch/tods/aspekte/, aufgerufen am 27.04.2010

lität erweitert werden. Denn jedwede Kommunikation vom oder zum Client erfolgt über die Netzwerkkomponente (NW) (siehe Abschnitt 5.3.3). Dazu speichert die NW die Daten in eine Textdatei auf dem USB-Stick und fügt die nächsten Protokolleinträge ans Ende der Datei an (append). Die Protokolldatei wird dazu in einem Verzeichnis auf dem USB-Stick gespeichert, welches dem Teilnehmer eindeutig zugeordnet ist. Das Verzeichnis kann mit der Matrikelnummer oder aber mit der SmartcardID der eSK bezeichnet werden.
Kommt es zu einem Client-Ausfall während der Prüfungsdurchführung, so kann der USB-Stick einfach mit zum nächsten Rechner genommen werden. Der Rechnerausfall würde dann in der Protokolldatei sichtbar sein, aufgrund der IP- bzw. MAC-Adresse des neuen Prüfungsrechners.

Erweiterung des Dateibaumes

Der in Kapitel 5 beschriebene Dateibaum der Teilnehmer kann wie folgt erweitert werden um die Protokolldaten nach der Durchführung abzuspeichern. Dazu werden bei der Prüfungsanmeldung durch das Prüfungsamt unterhalb des Verzeichnisses „PrüfungsID“, die Datensätze „Lösung“ und „Protokoll“ erstellt. In Abbildung 6.2 ist der erweiterte Dateibaum des Teilnehmers dargestellt.
Der Teilnehmer hat nach der Prüfungsdurchführung das Recht seine Prüfungslösungen anzulegen und nun zusätzlich auch die Protokolldaten. Während aber zu den Prüfungslösungen auch der Dozent in der Regel einen lesenden Zugriff besitzt, ist dies bei den Protokolldaten nur dem Teilnehmer möglich. Allerdings besitzt der Teilnehmer im Rahmen seines DTT für die Protokolldaten nur erstellenden und lesenden Zugriff, darf die Protokolldaten aber nicht ändern oder löschen.
Bei der Prüfungsdurchführung werden die Protokolldaten zuerst auf den USB-Stick geschrieben. Nach Ende der Prüfungsdurchführung wird über die Protokolldaten ein Hashwert gebildet und der Hashwert wird zusammen mit den Prüfungslösungen signiert und verschlüsselt. Somit kann die Integrität der Protokolldaten im Nachhinein zweifelsfrei festgestellt werden.
Anschließend werden die Protokolldaten mit einem symmetrischen Schlüssel verschlüsselt und im Verzeichnisdatensatz „Protokoll“ mittels create() abgespeichert.
Wenn die Speicherung der Protokolldaten erfolgreich war, erhält der Teilnehmer eine Anzeige über die erfolgreiche Speicherung auf seinem Client. Dann werden die lokalen Protokolldaten gelöscht. Sollte die Online-Speicherung nicht erfolgreich gewesen sein, so können die USB-Sticks durch die Aufsich-

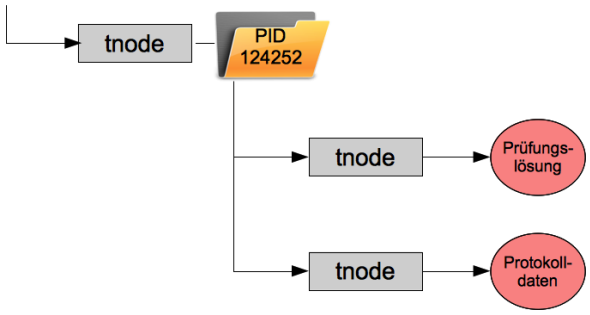

Abbildung 6.2: Erweiterter Studenten-Dateibaum

ten bzw. den Prüfungsverantwortlichen eingesammelt und gesichert werden.

Der Dozent erhält zwar lesenden Zugriff auf die Lösung des Teilnehmers (wenn er ein TicketToolkit erhalten hat), aber die Protokolldaten können von ihm nicht eingesehen werden. Des Weiteren hat der Teilnehmer nur die Berechtigung seine Protokolldaten anzulegen (create) aber nicht diese Daten zu verändern oder zu löschen.
Wenn der Teilnehmer sein Prüfungsergebnis im Nachhinein anzweifelt, indem er z.B. behauptet, dass der Server bzw. die Prüfungsfragen aufgrund von Verzögerungen ständig nicht verfügbar waren, so kann dies durch eine Analyse des Protokolls überprüft werden. Dazu muss der Teilnehmer die Protokolldaten entschlüsseln und den verantwortlichen Stellen offen legen. Denn der Teilnehmer ist in diesem Falle in der Beweispflicht.

6.1.2 Fallback

Die in Unterabschnitt 6.1.1 beschriebene Verwendung eines USB-Stick zur Protokollierung kann auch als clientseitiger Fallback (Ausfallschutz)-Mechanismus dienen. Wenn das Prüfungssystem bzw. das Netzwerk während der Durchführung ausfällt, ist eine Weiterführung der Prüfung lokal möglich. Die Prüfungslösungen des Teilnehmers werden während der Durchführung zusätzlich zu der Speicherung im Prüfungssystem auch auf den USB-Stick geschrieben, jedoch getrennt von den Protokolldaten. Denn der Zugriff auf die Protokolldaten darf nur schreibend erfolgen. Bei den Lösungen muss aber auch ein lesender Zugriff durch den Teilnehmer möglich sein. Allerdings erfordert das die Bedingung, dass die gesamten Prüfungsangaben zu Beginn der Prüfung auf dem Client vorhanden sind. In Abschnitt 5.6 wurde diese Bedingung bereits durch die signierte und verschlüsselte Bereitstellung der Prüfung über das virtuelle, ticketbasierte Dateisystem realisiert.

Falls also der Server des Prüfungssystems während der Prüfung ausfällt, erfolgt die weitere Speicherung der Prüfungslösungen des Teilnehmers auf dem USB-Stick. Nach der Durchführung überprüft der Konnektor ob das Prüfungssystem wieder erreichbar ist. Falls ja, werden die Daten vom USB-Stick im Rahmen von Transaktionen auf den Prüfungsserver hochgeladen (siehe Unterabschnitt 5.6.2). Auch hier erhält der Teilnehmer über die Abwicklung der Transaktion eine Meldung angezeigt. Wenn die Transaktion erfolgreich war, dann werden die Prüfungslösungen - wie auch die Protokolldaten (siehe Unterabschnitt 6.1.1) - vom USB-Stick vollständig gelöscht.

6.1.3 Archivierung

In Unterabschnitt 3.1.2 wurde bereits beschrieben welche Daten einer abgeschlossenen Prüfung zu archivieren sind. Dies sind im Grunde nur die Prüfungslösungen der Teilnehmer und die Prüfungsangaben der Dozenten. Die Protokolldaten der Teilnehmer sind nur für die Dauer der Einspruchsfrist vorzuhalten und nach dieser Frist zu löschen.
Das Berechtigungskonzept des virtuellen, ticketbasierte Dateisystem (vtD) aus Kapitel 5 setzt die Anforderungen an eine Archivierung der Daten bereits um. Denn der Zugriff auf die Daten ist nur durch einen genau bestimmten Benutzerkreis (Ticket-Inhaber) möglich. Einzig die Langzeitarchivierung der Daten sollte durch eine Backupstrategie ermöglicht werden. Denn bei einem größeren technischen Defekt sind die Daten nicht zugänglich. Eine solche Backupstrategie in Verbindung mit dem vtD könnte so aussehen, dass der Dozent sich den Semesterteilbaum innerhalb seines Dateibaumes exportieren kann und auf einen entsprechenden Datenträger wie DVD, BlueRay o.ä. speichern kann. Somit wären die Daten der Prüfungen für den Dozenten auch „offline“ verfügbar. Denn der Teilbaum beinhaltet nach Abschluss der Korrektur, die bewertete Lösung eines jeden Teilnehmers in der auch die Prüfungsangaben enthalten sind (siehe Abbildung 6.3). An einigen Hochschulen sind die Institute für die Archivierung ihrer Prüfungen (auch die auf Papier) selbst verantwortlich. Somit obliegt den Instituten auch die Sicherheit der auf z.B. externen Datenträgern gespeicherten Prüfungen. Ein unautorisierter Zugriff auf die Daten auf dem Datenträger ist jedoch aufgrund der Verschlüsselung nicht möglich. Der exportierte Teilbaum des Dozenten kann in genau der bestehenden Struktur abgespeichert werden, um im Falle eines Rechtsstreites den Teilbaum genau so in das vtD zurückzuspielen und somit den Zugriff über die Konnektoren und den Ticketserver zu ermöglichen.
Der Dateibaum auf dem Ticketserver kann dann gelöscht werden und im Fall einer Überprüfung o.ä. vom Datenträger wieder in das vtD eingespielt werden.

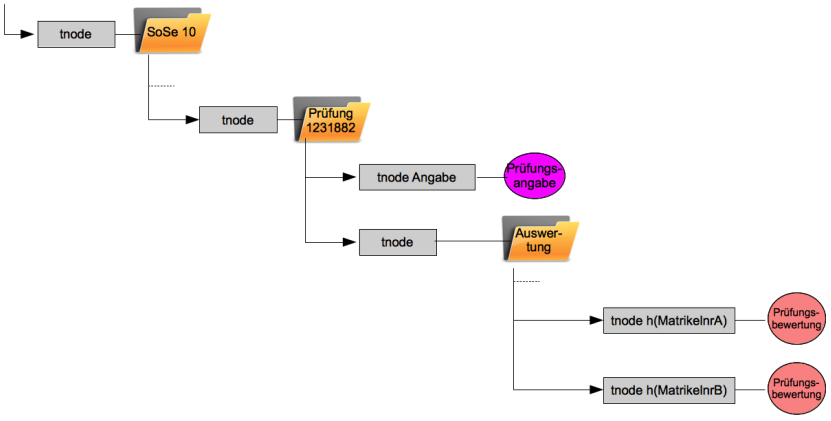

Abbildung 6.3: Zu archivierender Teilbaum

6.2 Umsetzung der administrativen Maßnahmen

Die Umsetzung der in Tabelle 3.1 aufgestellten Anforderungen ist oftmals nur durch die Zusammenarbeit von administrativen, technischen oder formalen Maßnahmen zu erreichen. Nachfolgend werden die in Tabelle 3.1 den administrativen Maßnamen zugeordneten Anforderungen realisiert.
Die Formvorschrift *P*11 erfordert durch die Notwendigkeit der qualifizierenden digitalen Signaturen eine sog. sichere Signatur-Erstellungseinheit (SSEE) (siehe Abschnitt 3.2.2). Die Verwendung eines elektronischen Studierendenausweises als SSEE in Form einer multifunktionalen Chipkarte würde dann auch den Einsatz der Signaturen ermöglichen. Aber der Einsatz der Smartcards für die elektronischen Prüfungen ist auch mit einem enormen administrativen Aufwand verbunden. So muss jeder Prüfungsrechner mit einem Smartcard-Reader ausgestattet werden bzw. diese Reader müssen administriert und gepflegt werden. Des Weiteren ist der Betrieb einer Public-Key-Infrastruktur nötig, die allerdings an nahezu allen Hochschulen zur Verfügung steht.
Aber auch hier besteht neben dem Signatureinsatz für die Prüfungen ebenfalls ein Mehrwert wie z.B. die verbindliche Anmeldung zu Prüfungen und Seminaren, sowie die Verschlüsselung und Signierung von eMails. Mit Hilfe von biometrischen Merkmalen wie Fingerabdruck, wäre z.B. auch die kostenintensive Belastung der Hochschulrechenzentren bei vergessenen Passwörtern obsolet. Denn die Passwort-Policy der Hochschulen wird immer restriktiver und führt zu einem aufwändigen Passwortmanagement (siehe [BW07]). Mit

Hilfe von Fingerprints könnte die Authentisierung des Studierenden gegenüber seiner Smartcard erfolgen und durch den Einsatz von Single-Sign-On mit Signaturen müsste sich der Studierende keine Passwörter merken (siehe Abschnitt 3.2.3). Die Authentisierung gegenüber den Systemen erfolgt dann im Prinzip nur über den Fingerprint, wobei dieser nur in kryptografischer Weise und in einem geschützten Bereich auf der Smartcard gespeichert ist und somit als Ersatz für die PIN-Eingabe dient.
Die elektronischen Prüfungssysteme sind größtenteils alle webbasiert (siehe Abschnitt 2.4 und Abschnitt 4.1). Das Prüfungssystem wird über einen Webserver bereitgestellt, der wiederum mit einem Datenbankserver verbunden ist. Diese Architektur hat allerdings den Nachteil, dass ein Ausfall des Web- oder Datenbankserversoder des Netzwerkes dazu führt, dass die Prüfung nicht weiter durchgeführt werden kann ($P41$). Dies führt dann zu der Maßnahme, zum einen die auftretende Last im Netzwerk zu verteilen und zum anderen einem Netzausfall während der Prüfung durch eine rein lokale Durchführung zu umgehen. In Abschnitt 2.4 wurde dies ja bereits durch die Verwendung von USB-Speichersticks realisiert. Ein weiterer Aspekt ist aber, dass durch das Konzept des virtuellen, ticketbasierten Dateisystems auch der Ticketserver mitsamt seinen Datenbankservern ausfallsicher sein muss. Denn nur über das vtD sind die Berechtigungen zur Zulassung zur Prüfung bzw. das verbindliche Abspeichern der Angaben und der Lösungen möglich. Für das vtD gibt es keine Alternative und somit gilt es hier entsprechende Lastverteilungssysteme für den Ticketserver einzusetzen.
In Abbildung 6.1 ist zur Lastverteilung des Ticketservers ein Hardware-Loadbalancer vorgesehen. Ein solcher Hardware-Loadbalancer ist in der Regel ein Layer 4 bis Layer 7 Switch, der auch für die SSL / TLS Kommunikation zwischen Konnektor und Ticketserver geeignet ist[2].
Eine bessere, nachhaltigere, wenn auch kostenintensivere Möglichkeit ist der Einsatz einer Blade-/Modular-Server Infrastruktur. Ein solche Infrastruktur besteht aus einem Rack-System mit mehreren Einschubfächern für sog. Moduls. Des Weiteren besteht das System aus einer Speicher- sowie Netzwerkinfrastruktur[3]. Der Vorteil eines Modular-Server gegenüber einem Blade-Server ist der, dass der Modular-Server eine Speicherinfrastruktur (StorageAreas-Network SAN) besitzt.
Ein SAN kann in Form eines RAIDs (Redundant Array of Inexpensive Disks) zwei Ziele verfolgen: Zum einen die Erhöhung der Performance der Datenzugriffe und zum anderen die Sicherstellung der Ausfallsicherheit. Dabei werden

[2]http://www.hardwareloadbalancer.com/, aufgerufen am 14.05.2010
[3]http://www.networkcomputing.de/netzwerk/datacenter-infrastruktur/artikel-6583.html, aufgerufen am 18.05.2010

die Daten redundant über mehrere Festplatten verteilt gespeichert, so dass der Betrieb der Anwendung auch beim Ausfall einer Festplatte fortgeführt werden kann.
Die Festplatten eines RAID-Systems werden auch als virtuelle Festplatte bezeichnet, weil für einen Server der an das RAID-System angeschlossen ist, es völlig verborgen bleibt, wie und wo die Daten letztendlich gespeichert sind. Für den Server stellt das RAID-System eine einzige Festplatte dar.
Ein RAID-Controller übernimmt dabei die Logik der Speicherung. Es gibt verschiedene Verfahren wie das RAID-System aufgebaut werden kann. Diese Verfahren werden als RAID-Level bezeichnet. Eine Beschreibung der einzelnen RAID-Levels findet sich u.a. in [Sch09d, TEM+08, Vad03].
Für die elektronischen Prüfungssysteme und vor allem die Ticketserver-Infrastruktur kommt vor allem der RAID-10-Level (oder auch RAID-0+1) in Frage. Dieser RAID-Level bietet neben einer sehr hohen Ausfallsicherheit eine sehr gute Leseperformance und eine gute Schreibperformance. Jedoch ist der Platzverbrauch sehr hoch, was also den Einsatz von mehreren Festplatten benötigt. Die Server werden in der Regel durch das Hochschulrechenzentrum administriert und unterliegen dabei per se nur dem physischen Zugriff durch autorisierte Mitarbeiter.
Die Ausfallsicherheit betrifft aber nicht nur die Serverseite, auch auf Clientseite muss die Durchführbarkeit der Prüfung gewährleistet sein. D.h. wenn ein Prüfungsrechner oder externe Komponenten des Rechners (Bildschirm, Eingabegeräte, etc.) nicht ordentlich funktionieren, so muss ein Ersatzarbeitsplatz oder Austauschkomponenten unmittelbar zur Verfügung stehen ($P42$). Dabei gilt es auch ausstattungstechnisch vergleichbar zu bleiben ($P71$). Dies trifft sowohl für die Hardware als auch auf die eingesetzte Software (Betriebssystem, Browser etc.) zu. Somit sind also die Prüfungsteilnehmer am besten im Vorfeld der Prüfung mit den Gegebenheiten in den Prüfungslaboren vertraut zu machen. Zum einen kann dies durch ein Merkblatt erfolgen, auf dem jeder Teilnehmer das Prozedere des Prüfungsablaufes im Vorfeld nachvollziehen kann (siehe Abschnitt 6.3).
Des Weiteren ist die Prüfungsumgebung ebenfalls für alle gleichwertig zu gestalten. Schlechte klimatische Verhältnisse, Geruchs- und Lautstärkebelästigungen sind absolut zu vermeiden und im Vorfeld der Prüfung durch Verantwortliche zu überprüfen. Dazu gehören auch die Komponenten der Prüfungsrechner wie bspw. Eingabegeräte und Lüfter. Für die Prüfungsrechner sind deshalb besonders leise Lüfter und geräuscharme Eingabegeräte zu verwenden ($P72$).
Täuschungsversuche sollten durch alle erdenklichen technischen und administrativen Maßnahmen verhindert werden. So ist bei webbasierten Prüfungssystemen eine sichere Software als Prüfungsumgebung auf den Rechner zu

installieren. Der in Unterabschnitt 3.2.8 beschriebene Secure-Browser (SEB) bietet nur eine sehr eingeschränkte Funktionalität, die aber je nach Prüfung erweitert werden kann (z.B. Freigabe von bestimmten Webseiten) ($P67$). Die Umgebung des SEB kann nur durch die Eingabe einer zuvor festgelegten Tastaturkombination verlassen werden. Hier beruht also die Sicherheit darauf, dass diese Kombination zum Deaktivieren des SEB auf den Prüfungsrechnern den Teilnehmern nicht bekannt ist.
Die Verwendung des USB-Sticks könnte auch bei der Anwendung des SEB sehr nützlich sein. Der SEB und der Konnektor befinden sich dabei auf jedem der Sticks und der SEB wird dann zur Prüfungsdurchführung von dem USB-Stick gestartet. Dies bedeutet keine Installation des SEB oder Konnektor auf dem Client. In diesem Zusammenhang ist auch der Einsatz von bootbaren USB-Sticks möglich.
Ein weiteres Problem stellen die Ausrichtung der Arbeitsplätze im Prüfungslabor dar. Oftmals werden die Rechner in mehreren Reihen hintereinander platziert. So kann der hintere Teilnehmer dem schräg vor ihm sitzenden leicht auf den Bildschirm schauen. Als Lösung kommen hier die in Unterabschnitt 2.3.3 beschriebenen, im Tisch versenkbaren, Monitore in Betracht. In Verbindung mit Trennwänden zwischen den Arbeitsplätzen ist somit auch kein „spicken“ zum Nebenmann möglich ($P65$).

6.3 Umsetzung der formalen Maßnahmen

Die Umsetzung der formalen Maßnahmen erfolgt durch Anpassungen von Prüfungsordnungen, aber auch durch Bereitstellung von Richtlinien. Auf die Diskussion bzgl. der Formulierungen von Gesetzen und Vorschriften wird in dieser Arbeit verzichtet und auf die entsprechenden juristischen Quellen verwiesen (siehe u.a. [KF08]).
Richtlinien aber dienen dazu den beteiligten Akteuren Handlungsvorgaben an die Hand zu geben, die im Falle von Problemen dazu dienen eindeutige Vorgehensweisen zu ermöglichen. Aber auch um ein einheitliches Vorgehen innerhalb des gesamten Prüfungsprozesses zu gewährleisten. Dazu gehört auch, wie z.B. bei starken Geräuschbelästigungen (z.B. durch Baulärm o.ä.) vorgegangen wird. Eine Möglichkeit ist, den vom Lärm beeinträchtigten Teilnehmern eine Schreibverlängerung einzuräumen, wie es u.a. bei juristischen Staatsexamen gehandhabt wird. Dazu muss das Prüfungssystem aber die Möglichkeit vorsehen für jeden Teilnehmer eine individuelle Schreibverlängerung einzustellen. Dies trifft im übrigen auch auf behinderte oder gehandicapte Teilnehmer zu, denen z.B. durch vergrößerte Darstellung und Schriftarten oder Braille-Eingabegeräten ebenso Rechnung getragen werden kann,

wie durch höhenverstellbare Stühle und Tische bei körperliche beeinträchtigten Teilnehmern. Des Weiteren dienen die Richtlinien dazu, gerade bei der Einführung von elektronischen Prüfungen, alle Beteiligten über den Ablauf der Prüfung im Vorfeld zu informieren.
Zuerst muss klar geregelt werden, wer für die Vorbereitung, Durchführung und Betrieb sowie die Speicherung der Daten der elektronischen Prüfungen verantwortlich ist. Hierbei gilt es auch festzulegen, wer für den Betrieb des Ticketservers und dessen Datenspeichern verantwortlich ist. In [KF08] wird empfohlen einen Verantwortlichen für die elektronischen Prüfungen zu benennen. In Abschnitt 2.3 sind die an einigen Hochschulen bereits eingerichteten Organisationseinheiten (eAssessment-Dienste) beschrieben, die sich für die elektronischen Prüfungen verantwortlich zeichnen. Ein solcher zentraler Dienst für elektronische Prüfungen ist meistens in die Struktur des Rechenzentrums der Hochschule eingegliedert.
Mit Hilfe einer solchen Institution sind die Verantwortlichkeiten klar geregelt. Der Dozent ist nur für die Prüfungsangaben und die Prüfungsbewertungen verantwortlich. Das Prüfungsamt bestimmt wer an der Prüfung teilnehmen darf und der Teilnehmer bestimmt ob und wer seine Prüfungslösungen / Protokolldaten einsehen und bewerten darf.
Eine zentrale Organisationseinheit ePrüfungen ist für die Durchführung und den Betrieb der Hardware- und Software-Komponenten verantwortlich. Dazu zählt der ordnungsgemäße Betrieb der Prüfungsrechner und Server, sowie der Smartcard-Reader und der Konnektoren. Die Organisationseinheit unterstützt die Dozenten außerdem bei der Prüfungserstellung, auch im Hinblick auf mediendidaktische Fragen.
Das Netzwerk ist Bestandteil des Hochschulnetzes und unterliegt damit der Hoheit des Hochschulrechenzentrums.

6.4 Zusammenfassung

Das dargestellte Gesamtkonzept deckt alle Maßnahmen zur Umsetzung der Anforderungen ab. Aber es ist klar, dass die Umsetzung aller Anforderungen nur durch das Zusammenspiel von (software)-technischen, administrativen und formalen Maßnahmen erfolgen kann. Denn all diese Maßnahmen können bislang die Anwesenheit von Aufsichtspersonal nicht ersetzen. Ein eAssessment-Dienst sollte sich an der Hochschule für die ePrüfungen verantwortlich zeigen. Die Hochschulen Bremen, Duisburg-Essen und Zürich sind hierbei als gute Beispiele zu nennen.
Die konkreten Implementierungen der einzelnen Maßnahmen und vor allem des virtuellen, ticketbasierten Dateisystems sind in Kapitel 7 dargestellt.

Kapitel 7

Proof-of-Concept

Die Implementierungen der in Kapitel 5 und Kapitel 6 beschriebenen Konzepte wurde unter der Betreuung des Autors durch mehrere Diplomarbeiten und Projektgruppen realisiert. In diesem Kapitel werden die wichtigsten wissenschaftlichen Arbeiten zu diesem Thema aufgezeigt. Die Verteilung der Arbeiten in Bezug zu den einzelnen „Bauteilen“ der Architektur aus Kapitel 6 sind in Abbildung 7.1 dargestellt.

7.1 Authentifizierung mit Smartcards bei elektronischen Prüfungen

In [Gro06] wurde die Machbarkeit und Anwendbarkeit einer Java-basierten, clientseitigen Sicherheitskomponente zur Authentifizierung und Verbindlichkeit von Online-Klausuren gezeigt, deren Realisierung u.a. auf einem Smartcard-basierten Ansatz beruht. Auch die Integration in ein bestehendes Online-Prüfungssystem wurde prototypisch gezeigt.

Aufbauend auf der Arbeit [Gro06] wurden in [Bre08] weitere Sicherheitsanforderungen an elektronische Prüfungen umgesetzt. Dazu gehören die Anforderungen der Ausfallsicherheit, Betrugssicherheit, Zugriffskontrolle und der Verwendung von qualifizierenden digitalen Signaturen. Damit einher geht die Verwendung von Smartcards, bzw. mobilen Plattformen wie z.B. e-Tokens, die laut Signaturgesetz Voraussetzung für die Verwendung von digitalen Signaturen sind. In [Bre08] wurde als Smartcard eine JavaCard verwendet, die es ermöglicht auf der Karte Java-Code auszuführen und somit die Smartcard als eigenständige Ausführungsplattform zu benutzen. Dadurch wäre auch die Sicherstellung von weiteren Sicherheitsanforderungen wie z.B. die Ausfallsicherheit usw. denkbar. Die Untersuchungen in [Bre08] kamen zum Ergebnis,

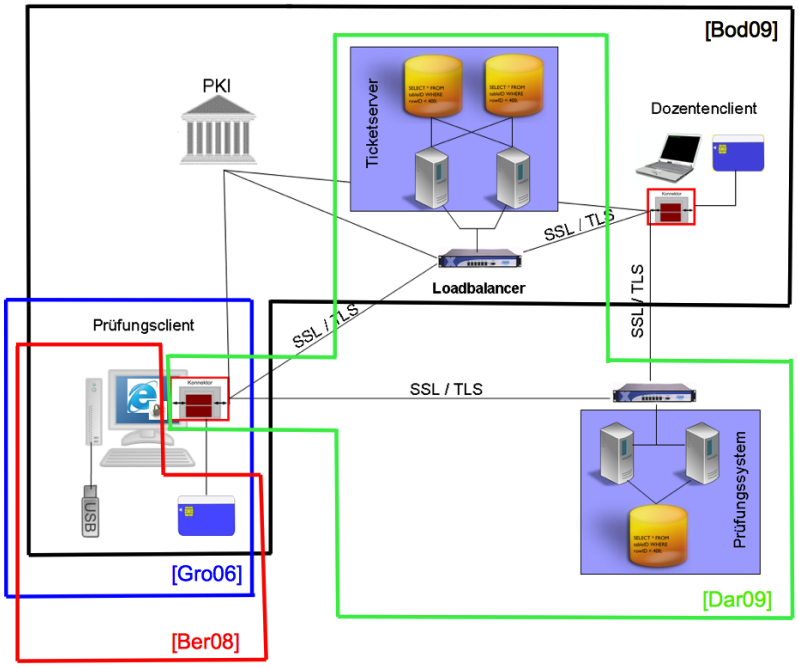

Abbildung 7.1: Aufteilung der Architektur nach Arbeiten

dass für die elektronischen Prüfungen nur wenige Funktionalitäten der JavaCard zum Einsatz kommen können. Dies liegt vor allem an den fehlenden Zertifizierungen und dem nur sehr eingeschränkten Java-Befehlssatz. Das wiederum hatte aber den positiven Nebeneffekt, dass das von ihm entwickelte Konzept auch auf Native Smartcards angewendet werden kann.

7.2 Konnektor

In [Bod09] wurde das vtD mit der dazugehörigen Infrastruktur auf die elektronischen Prüfungen hin umgesetzt. Dazu wurden Teile des in Kapitel 5 dargestellte vtD realisiert. Abbildung 7.2 zeigt die Grobarchitektur der Realisierung. Der Prüfer bzw. Student, der über den Clientrechner auf das Prüfungssystem auf dem Webserver zugreifen will, wählt seinen Weg über den Web-Proxy, der zwischengeschaltet wird. Der Web-Proxy ist eine in Teilen erfolgte Implementierung des in Abschnitt 5.3.3 dargestellten Konnektors. Der Konnektor selbst wurde dabei in [Bod09] nicht vollständig, wie in Unterabschnitt 5.3.3 beschrieben, umgesetzt. D.h., es wurden keine verschiedenen Konnektorkomponenten (SVK, NWK, TSK (siehe Unterabschnitt 5.3.3) verwendet. Zu den umgesetzten Funktionalitäten gehören:

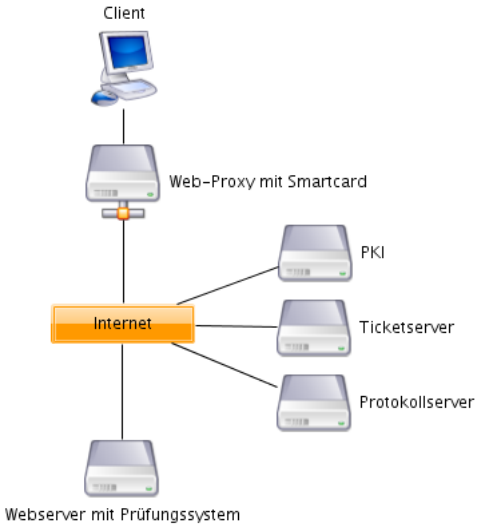

Abbildung 7.2: Grobarchitektur Realisierung vtD [Bod09]

- Alle kryptografischen Funktionen in Verbindung mit der Smartcard
  - Ver- und Entschlüsselung (symmetrisch, asymmetrisch)
  - Signierung und Verifizierung
  - Bildung von Zufallszahlen und kryptografischen Schlüsseln
- SSL-Verbindung zu Prüfungsserver und Ticketserver
- Schnittstelle zur PKI
- Protokollierung des ein- und ausgehenden Datenverkehrs
- Aufbereitung des eingehenden verschlüsselten Datenverkehrs zur Darstellung im Web-Browser

Ticketserver, PKI und Protokollserver haben in [Bod09] gemeinsam, dass sie als Webservice über das Internet erreichbar sind. Die Webservices akzeptieren allerdings nur authentifizierte Anfragen, d.h., der Proxy signiert die Anfragen für den Benutzer automatisch bei der Kommunikation mit dem Ticket- und Protokollserver. Der PKI-Server erlaubt nur einen lesenden Zugriff, so dass hier auch eine Abfrage ohne Signatur möglich ist. Des Weiteren dient der Web-Proxy zur Umsetzung der Ver- und Entschlüsselung, sowie der Signierung und Verifizierung von Daten.

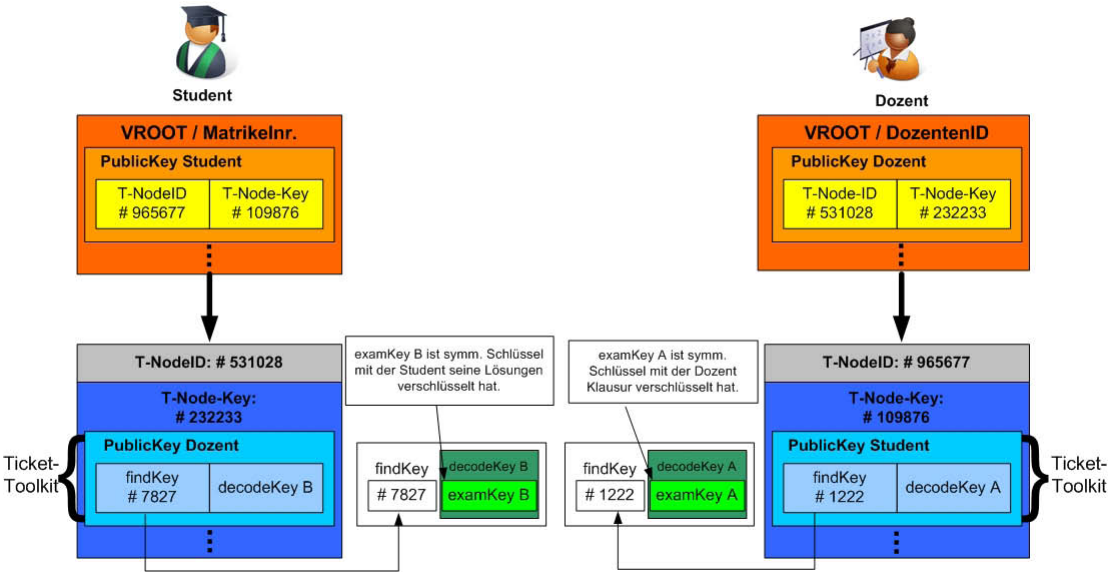

Abbildung 7.3: Ticketsystemstruktur (aus [HWB09])

7.3 Ticketserver und virtuelles, ticketbasiertes Dateisystem

Die Implementierung des Ticketservers (TS) beschränkte sich in [Bod09] darauf, nicht die gesamte Verzeichnisstruktur des Ticketsystems zu implementieren, sondern nur die Hierarchie vRoot → tnode-PrüfungsID → Datensatz. Des Weiteren wurde auf das Abspeichern der signierten Prüfungsangabe, bzw. der signierten Prüfungslösung eines jeden Teilnehmers verzichtet. Stattdessen wurde im Ticketsystem nur der symmetrische Schlüssel (examKey) hinterlegt, mit dem die Prüfungsangaben im PS verschlüsselt sind. Die Teilnehmer hinterlegen wiederum nicht ihre signierten Prüfungslösungen, sondern nur den symmetrischen Schlüssel mit dem ihre Lösungen im PS verschlüsselt sind (siehe Abbildung 7.3). Die Begründung liegt darin, dass das Ticketkonzept auf ein bestehendes Prüfungssystem angepasst werden sollte und dabei so wenig wie möglich Anpassungen am Prüfungssystem erfolgen sollten.

Des Weiteren werden in dem vRoot des Dozenten die Zugriffsinformationen für die TicketToolkits gespeichert, die der Dozent von den Studenten erhalten hat, die an der Prüfung teilgenommen haben. Die Teilnehmer wiederum haben die Zugriffsinformationen für das TicketToolkit gespeichert, das der Dozent dem Teilnehmer zur Durchführung ausgestellt hat. Das bedeutet,

dass in [Bod09] der Referenz-Dienst der Zugangs- und Integrationsschicht nicht implementiert wurde.

Das realisierte virtuelle, ticketbasierte Dateisystem in [Bod09] wurde in der Arbeit von [Dar09] erweitert. Die Erweiterungen betreffen vor allem die Umsetzung der kompletten Hierarchie des Dateisystems. Des Weiteren wurde neben dem in [Bod09] realisierten Client-Proxy (Konnektor) durch [Dar09] ein Server-Proxy sowie ein MySQL-Proxy erstellt (siehe Abbildung 7.4). In [Dar09] wurde ein Konzept entwickelt, um die Änderungen die im Client-Proxy durch [Bod09] dargelegt wurden, serverseitig durchzuführen. In Verbindung mit dem MySQL-Proxy kann somit die feingranulare Zugriffskontrolle serverseitig durchgeführt werden. Das bedeutet, die sensiblen Prüfungsangaben werden bei einer fehlenden Zugriffsberechtigung erst gar nicht aus der Datenbank ausgelesen und somit auch nicht übertragen.
Der MySQL-Proxy ist eine Software-Komponente, die zwischen einem MySQL-Client bzw. Webserver und einer oder mehrerer MySQL-Datenbanken angesiedelt ist[1]. Durch die integrierte Scriptsprache „Lua“ können individuelle Skripte erstellt werden, die den ein- und ausgehenden Datenverkehr überwachen und ggf. auch verändern können (siehe [Max07]).

7.4 Protokollierung und Ausfallsicherheit

Die Protokollierung des Prüfungsverlaufes wird in [Bod09] durch einen Protokollserver realisiert. Der Protokollserver erhält vom Konnektor (Web-Proxy) Informationen darüber, welche Daten zwischen Client und dem Web-Proxy geflossen sind. Der Proxy modifiziert die HTTP-Anfragen und HTTP-Antworten, so dass nachvollzogen werden kann, welche Daten der Client selbst verschickt und welche er empfangen hat. Nur diese Daten hat zum Beispiel der Student im Hinblick auf eine Prüfung und der rechtlichen Absicherung zu verantworten („What you see is what you sign“). Daher werden die HTTP-Anfragen durch den Protokollserver aufgezeichnet.

[Bre08] verbindet die Realisierung der Anforderungen Protokollierung und Ausfallsicherheit durch den Einsatz von USB-Sticks. In dem Konzept zur Sicherstellung der Verbindlichkeit computergestützter Prüfungen mit hoher Verfügbarkeit wurde ein vorgegebenes Szenario (siehe Abbildung 7.5)

[1] ähnliche Ansätze für relationale Datenbanken finden sich z.B. auch für *Postgres* mit dem PL/Proxy (siehe http://plproxy.projects.postgresql.org/doc/tutorial.html, aufgerufen am 28.04.2010)

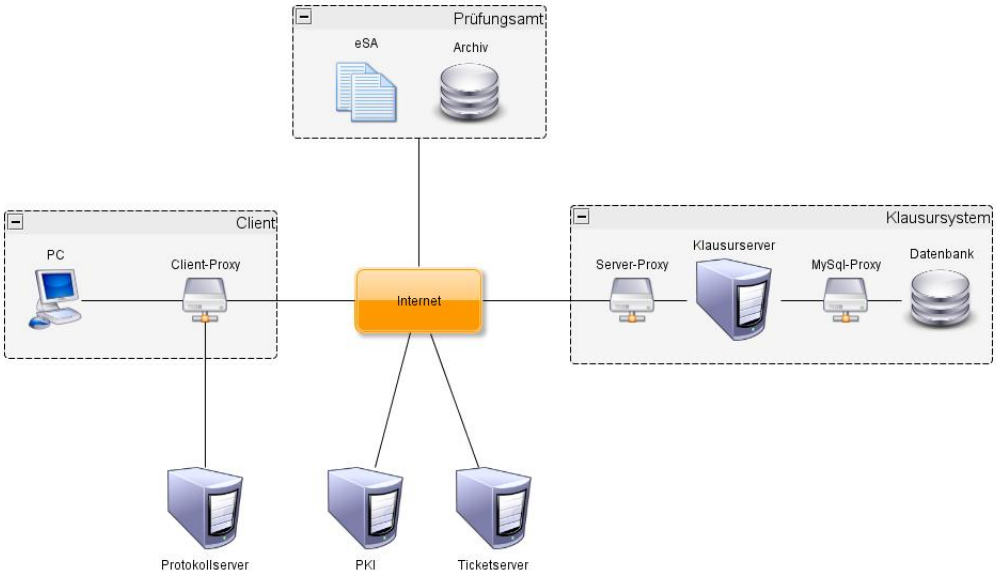

Abbildung 7.4: Erweitertes Sicherheitskonzept [Dar09]

wiederverwendet und durch den Einsatz von USB-Sticks zur Ausfallsicherheit und Protokollierung ergänzt. Die Protokolldaten werden somit nicht auf einen Protokollserver übertragen, sondern werden nur lokal vorgehalten durch Speicherung auf dem USB-Stick. Die Umsetzung dessen erfolgt mit Hilfe eines Java-Applets.

In Abbildung 7.5 sind die einzelnen Schritte des Szenarios beschrieben. Zuerst generiert der Dozent einen symmetrischen Schlüssel Sek.data und stellt den Schlüssel den Studierenden zur Verfügung, die sich zur Prüfung anmelden. Serverseitig wird zur Anmeldung für jeden Teilnehmer ein „Ticket“ (Authentisierungstoken) generiert, bestehend aus TeilnehmerID, PrüfungsID und dem SeK.data. Jeder Teilnehmer erhält ein solches Ticket nach der erfolgreichen Anmeldung zur Prüfung auf seine JavaCard.

Der Dozent verschlüsselt die Prüfung mit dem SeK.data und stellt die verschlüsselte Prüfung auf dem Server bereit. Die Teilnehmer authentifizieren sich an den Prüfungsrechnern und das Java-Applet lädt die verschlüsselte Prüfung vom Server auf den Client. Zur Entschlüsselung der Prüfung wird der SeK.data aus dem Ticket verwendet. Die Prüfung liegt nun vollständig auf dem Client. Die Prüfungsdurchführung erfolgt durch Speicherung der Lösungen sowohl in der Prüfungsdatenbank als auch auf dem lokalen USB-Stick. Die Daten werden auf dem USB-Stick in einem extra Ordner abgespeichert. Zusätzlich werden in einer Log-Datei auf dem USB-Stick auch Server- und

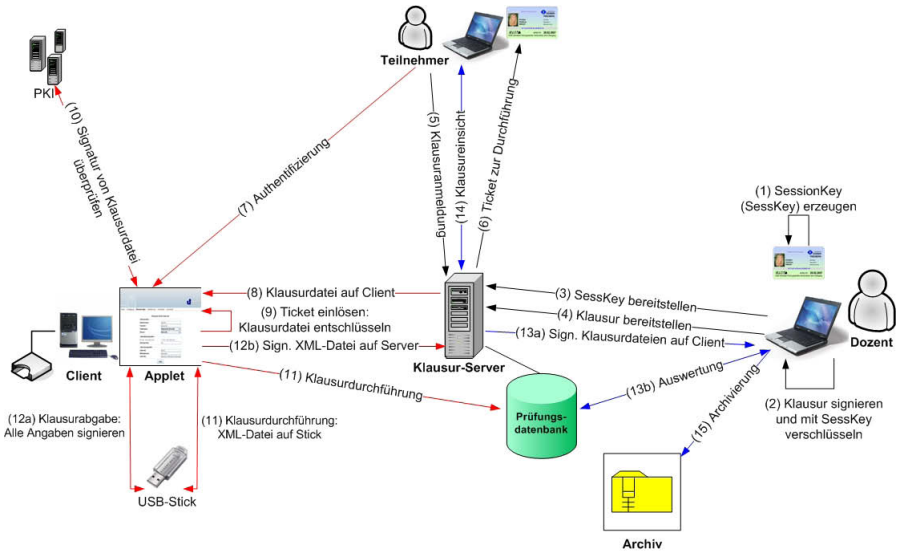

Abbildung 7.5: Szenario Prüfungsdurchführung [Hof07, Bre08]

Netzwerkinformationen abgespeichert.
Fällt der Server oder aber das Netzwerk während der Durchführung aus, so registriert das Applet dies und speichert die Lösungen nur auf dem USB-Stick ab. Der Teilnehmer wird aber darüber nicht informiert, damit er sich weiterhin nur auf die Beantwortung der Fragen konzentrieren kann. Gleichwohl wird im Log-File auf dem USB-Stick ein Eintrag gesetzt, der den Zeitpunkt und den Art des Problems festhält. Das Applet versucht dann nach der Prüfungsdurchführung den Server anzusprechen. Wenn dies dann möglich ist, werden die Lösungen des Teilnehmers vom USB-Stick auf den Server übertragen. Wenn nicht, müssen zur Auswertung die Daten von den Sticks auf den Server geladen werden. Dies erfolgt durch den Dozenten bzw. einer zentrale eAssessment-Einheit.
Fällt der USB-Stick aus, so kann er durch einen anderen USB-Stick ersetzt werden. Das Applet registriert, dass eine Speicherung auf dem USB-Stick nicht möglich ist und gibt eine Warnmeldung aus, dass der USB-Stick durch eine Aufsicht zu wechseln ist. Der neue USB-Stick synchronisiert sich dann über das Applet mit der Datenbank und erhält so den aktuellen Stand.
Nach der Beendigung erhält der Teilnehmer eine Übersicht über seine getätigten Lösungsangaben. Diese werden werden von dem Teilnehmer signiert und auf einen FTP-Server übertragen. Falls der Teilnehmer das Ergebnis der Auswertung anzweifelt, wird die signierte Datei herangezogen.

7.5 Integration in ein bestehendes Prüfungssystem

7.5.1 Prüfungssystem *KLAUSIE*

[Bod09] hat das Ticketsystem auf ein existierendes Prüfungssystem an der Universität Siegen angepasst. Das Prüfungssystem *KLAUSIE* (Klausursystem Universität Siegen) wird seit dem Sommersemester 2008 am Fachbereich Wirtschaftswissenschaften, Wirtschaftsrecht und Wirtschaftsinformatik der Universität Siegen eingesetzt und wurde seit dem in über 45 Prüfungen mit mehr als 4000 Teilnehmern eingesetzt[2]. *KLAUSIE* ist ein auf PHP und MySQL basiertes System, mit dem elektronische Prüfungen auf Basis von Multiple- /Single- /Matrix-Choice, Lückentext und Freitext umgesetzt und durchgeführt werden. *KLAUSIE* verwendet bislang nur die üblichen Sicherheitsmechanismen wie SSL und die verschlüsselte Speicherung der Daten in der Datenbank. Zur Authentifizierung der Studierenden werden deren Studierendenkennungen verwendet. Alle anderen Nutzer (Dozenten, Administrator, etc.) erhalten eine eigene *KLAUSIE*-Kennung. Die Software wird durch die Forschergruppe Online-Testen der Universität Siegen bereitgestellt und administriert. Die Durchführung der Prüfungen erfolgt in 5 PC-Laboren mit insgesamt 100 Arbeitsplätzen.

7.5.2 Anpassungen

Beim Login wird das Passwort mittels der Smartcard signiert übertragen. Damit eine Verifizierung der Authentifizierungsdaten stattfinden kann, muss das Prüfungssystem minimal angepasst werden. Hier wird bisher das Passwort überprüft. Nun müssen zusätzlich die Signatur des Passworts, als auch der korrekte Zusammenhang zwischen Signatur und der hinzukommenden Benutzerkennung überprüft werden. Dies ist über eine Schnittstelle zum PKI-Servers möglich.
Die Prüfungsangaben werden komplett verschlüsselt in der Datenbank gespeichert, so dass ein Angreifer keine vertraulichen Daten auslesen kann. Damit *KLAUSIE* nun die verschlüsselten Daten einlesen kann, wurde die Schnittstelle zur Datenbank angepasst. Hier werden die Ergebnisse lesender Operationen, wie zum Beispiel ein SELECT-Aufruf, automatisch entschlüsselt und Schreiboperationen, wie INSERT und UPDATE verschlüsselt. Bei einem SELECT stehen die aktuell gelesenen (verschlüsselten) Daten in ei-

[2]http://www.online-testen.com, aufgerufen am 21.05.2010

nem Array zur Verfügung, die automatisch entschlüsselt werden. Vorhandene Signaturen werden dabei ignoriert. Die zur Entschlüsselung notwendigen tnodeIDs werden festgehalten, da mit diesen später eine Verschlüsselung der kompletten HTML-Ausgabe stattfindet.
Beim INSERT und UPDATE von Daten wird der vorhandene „unverschlüsselte“ SQL-Query durch einen Parser verschlüsselt. Dazu kennt *KLAUSIE* die vom Benutzer verwendete tnodeID und überprüft, ob im SQL-Query Daten vorkommen, die nur verschlüsselt in der Datenbank gehalten werden sollen. Diese Zuordnung wird über die Spaltennamen der Datenbank getroffen. Damit die Schlüsseltexte auch in die Datenbank geschrieben werden können, müssen Datentypen der Tabellenspalten ggf. geändert werden. So wurde die korrekte Antwort einer Single-Choice Aufgabe bisher als Integer gespeichert. Da diese Zahl nun aber verschlüsselt in der Datenbank steht, reicht der Integer-Datentyp nicht mehr, so dass eine Änderung in einen „Text“-Datentyp notwendig ist. Da nach dem Auslesen der Datenbank, die Daten in *KLAUSIE* unverschlüsselt zur Verarbeitung vorliegen, müssen diese vor dem Senden einer Antwort zum Client wieder verschlüsselt werden. Dazu verschlüsselt *KLAUSIE* seine kompletten Ausgaben zusammen mit den o.g. tnodes.
Die Prüfungsangaben werden durch den Studenten ebenso verschlüsselt in der Datenbank abgespeichert wie auch die Prüfungslösungen der Teilnehmer. Allerdings müssen die Prüfungsangaben jeweils einzeln signiert werden. Die Prüfungslösungen werden dem Teilnehmer nach der Durchführung zusammen mit den Prüfungsangaben und den persönlichen Daten des Teilnehmers in Form eines Klausurbogens angezeigt und werden dann durch den Teilnehmer signiert und verschlüsselt. Dieser Klausurbogen wird in einem Blob-Feld innerhalb der Datenbank zusätzlich gespeichert. Der Klausurbogen dient jedoch nur im Streitfall dazu, als Beweisquelle verwendet zu werden.

7.6 Fazit und kritische Betrachtung

Die in diesem Kapitel beschriebene Arbeiten realisieren die wichtigsten Sicherheitsanforderungen und das virtuelle, ticketbasierte Dateisystem. Bei den Realisierungen besteht jedoch das Problem, dass z.B. in [Bod09] der Aufbau des Ticketservers stark vereinfacht wurde. Dazu gehört auch, dass eine eigene Schnittstellenbeschreibung für die Kommunikation zwischen Konnektor und Ticketserver verwendet wurde. Ebenso wurde weitestgehend auf die Realisierung der Zugangs- und Integrationsschicht (ZIS) verzichtet und durch eigene Methoden ersetzt. Dennoch bleibt der Charakter des virtuellen, ticketbasierten Dateisystems erhalten, so dass die Sicherheitsanforderungen umgesetzt

werden konnten.
Die in [Dar09] durchgeführten Erweiterungen des Ticketkonzeptes aus [Bod09] konnten jedoch nur in kleinem Umfang realisiert werden. So wurde nur der Einsatz bzw. die Machbarkeit eines MySQL-Proxy in [Dar09] realisiert.
Was die Fallback- und Protokollierungsmechanismen betrifft, die in [Bod09] und [Bre08] implementiert wurden, so sind diese vollständig anwendbar. Ebenfalls anwendbar sind die Anbindung der Smartcards über einen Konnektor, mitsamt sämtlichen Funktionalitäten wie Signierung und Verschlüsselung.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

Diese Arbeit hat sich mit der Sicherheit von elektronischen Prüfungen an Hochschulen beschäftigt. Dazu wurde ein Sicherheitskonzept entwickelt, das alle technischen Anforderungen umsetzt, aber auch die formalen sowie administrativen Anforderungen berücksichtigt. Der Schwerpunkt der Arbeit liegt dabei auf der Umsetzung eines virtuellen, ticketbasierten Dateisystems (vtD), das zusammen mit den qualifizierenden digitalen Signaturen auf bestehende Prüfungssysteme angepasst werden kann und sogar einen multifunktionalen Nutzen im Sinne von weiteren Anwendungen wie z.B. der elektronischen Studierendenakte besitzt.
Der Bedarf eines solchen Sicherheitskonzeptes wurde in dieser Arbeit u.a. damit begründet, dass die Sicherheitsanforderungen, die an eine papierbasierte Prüfung gestellt werden, auch für die elektronischen Prüfungen gelten müssen. Daraus ergibt sich die Notwendigkeit, dass die Durchführung einer elektronischen Prüfung der Formvorschrift genügen muss. Wenn eine handschriftliche, papierbasierte Prüfung durch die elektronische Form ersetzt werden soll, dann ist der Einsatz von qualifizierenden, digitalen Signaturen nötig, um die Rechtssicherheit zu gewährleisten. Allerdings werden die qualifizierenden Signaturen in nahezu keinem der im deutschsprachigen Raum eingesetzten Prüfungssysteme verwendet (siehe Kapitel 4). Stattdessen wird durch Medienbrüche - wie Ausdruck und Unterschrift der elektronischen Lösungen - versucht, die Rechtssicherheit zu gewährleisten.
In dieser Arbeit wurde die Notwendigkeit der qualifizierenden digitalen Signaturen begründet und wie die Signaturen zur Umsetzung weiterer Sicherheitsanforderungen verwendet werden können. Die Signaturen sind aber nicht nur eine Notwendigkeit, sondern sie dienen mit dem vtD auch einem mul-

tifunktionalen Zweck, der über die reine Verwendung für die elektronischen Prüfungen hinausgeht. Dazu zählen u.a. die bereits erwähnte elektronische Studierendenakte und die formativen Prüfungen (siehe auch Abschnitt 8.3). Für alle Erweiterungen gilt, dass der Anwender dank des vtD „Herr seiner Daten“ bleibt.

Das in dieser Arbeit dargestellte Sicherheitskonzept betrachtet aber nicht nur die technische Sicht, sondern sieht sich vor allem als ganzheitlicher Ansatz aus technischen, administrativen und formalen Maßnahmen, die unabhängig vom verwendeten Prüfungssystem sind. Dies liegt darin begründet, dass das Konzept in einzelnen Bausteinen realisiert ist, in denen vor allem die kryptografischen Funktionen des Sicherheitskonzeptes umgesetzt wurden. So kann mit Hilfe des clientseitigen Konnektors und der USB-Speicherlösung die Prüfungsumgebung auf der Clientseite definiert werden ohne das zusätzlicher Installationsaufwand entsteht. Dazu werden die Prüfungsclients vom USB-Stick gebootet und die Ausführung des Konnektors erfolgt ebenfalls vom USB-Stick.
Das Sicherheitskonzept wurde in Kapitel 7 auf ein existierendes Prüfungssystem adaptiert. Es konnte am Beispiel des webbasierten Prüfungssystems *KLAUSIE* gezeigt werden, dass der Aufwand der Anpassungen auf eine Minimum beschränkt werden kann.

8.2 Fazit

Das dargestellte Konzept schafft die Anforderungen bei elektronischen Prüfungen in Sachen Datenschutz und Datensicherheit vollständig und praktikabel umzusetzen. Die Schwierigkeit bei der Umsetzung war, die sich teilweise widersprechenden Anforderungen der Datensicherheit und des Datenschutzes in ein Konzept zu integrieren (siehe Tabelle 8.1).
Denn aus Datenschutzgründen sollen so wenig persönliche Daten wie nur möglich (bzw. am besten gar keine) erhoben werden. Aus Gründen der Datensicherheit bzw. Rechtssicherheit sollten bei den elektronischen Prüfungen aber so viele Daten wie nur möglich zu einem Teilnehmer gesammelt und gespeichert werden, um u.a. die Nachvollziehbarkeit des Prüfungsablaufs zu gewährleisten. Des Weiteren ist die Anonymität eine weitere Anforderung des Datenschutzes. Bei den Prüfungen muss aber eindeutig klar sein, dass derjenige, der die Prüfung durchführt, auch derjenige ist, der zur Prüfung zugelassen wurde. Außerdem muss die erbrachte Prüfungsleistung dem Studenten zugeordnet werden können. Des Weiteren sind die signierten Prüfungsangaben

Datenschutz	**Datensicherheit**
Datensparsamkeit/-vermeidung	Erhebung von Prüfungsdaten und Protokollierung des Prüfungsablaufes
Anonymität	Eindeutige Zuordnung Teilnehmer - Prüfung und Authentifizierung der Teilnehmer
Löschung der Spuren	Nichtabstreitbarkeit der Prüfungslösungen und Archivierung

Tabelle 8.1: Widerspruch zwischen Datenschutz und Datensicherheit

bzw. -lösungen über einen längeren Zeitraum zu speichern, um die Nichtabstreitbarkeit der Daten sicherzustellen.
Das virtuelle, ticketbasierte Dateisystem des Ticketkonzeptes löst genau diese Widersprüche auf, indem es z.B. die Anonymität trotz Authentizität gewährleistet. D.h., dass z.B. bei der Bewertung der Prüfungslösungen durch den Dozenten bzw. Korrekteur der Student anonym bleibt und trotzdem die Rechtssicherheit der Lösung erhalten bleibt.
Ein weiterer wichtiger Aspekt ist die Anforderung der Verfügbarkeit. In den existierenden Sicherheitsmodellen werden zwar Maßnahmen wie Load-Balancing und Replikation eingesetzt, aber bei einem Ausfall der Netzverbindung ist oftmals eine weitere Durchführung nicht möglich. Das in dieser Arbeit präsentierte Sicherheitskonzept realisiert sowohl eine serverseitge Ausfallsicherheit über Load-Balancing etc. als auch einen clientseitigen Fallback per USB. Somit kann die Prüfung auch bei einem Server- oder Netzausfall weiter lokal durchgeführt werden.

Allerdings konnte das Sicherheitskonzept noch nicht vollständig realisiert werden. Es wurden in Kapitel 7 speziell bei der Umsetzung des virtuellen, ticketbasierten Dateisystems nur die Kernelemente implementiert. Eine vollständige Realisierung steht noch aus. Des Weiteren wurde speziell die Spezifikation des Ticketservices der eGK im Rahmen von mehreren Releases modifiziert, so dass das in dieser Arbeit beschriebene und verwendete Ticketkonzept der eGK nicht dem aktuellen Konzept der Gesundheitskarte entspricht.
Dennoch konnte die Umsetzbarkeit der zentralen Elemente des Ticketkonzepts in Kapitel 6 und Kapitel 7 gezeigt werden. Dazu gehören das Anlegen und Verwalten von TicketToolkits sowie der gesamte Vorgang der Erstellung und Einlösung eines Tickets. Des Weiteren wird die Rechtssicherheit und die Nichtabstreitbarkeit durch den Einsatz der qualifizierenden digitalen Signa-

turen gezeigt und wie die Anpassung an ein existierendes Prüfungssystem erfolgen kann.
Trotz aller technischen Realisierungen bleibt die Notwendigkeit einer zentralen vertrauenswürdigen Instanz an einer Hochschule (eAssessment-Dienst), die sich für den kompletten Prüfungsprozess verantwortlich zeigt.

8.3 Ausblick

Das in Kapitel 5 und Kapitel 6 beschriebenen Konzept gilt es vollständig zu realisieren und somit die in Kapitel 7 beschriebenen Umsetzungen praktisch zusammenzuführen. Des Weiteren ist das virtuelle ticketbasierte Dateisystem (vtD) auf einen multifunktionalen Einsatz hin zu erweitern. Eine Erweiterung wäre für die folgenden Anwendungen denkbar:

- elektronische Studierendenakte (eSA)
- formative und diagnostische Prüfungen (Übungen, Selbsttests, etc.)
- Evaluationen (Lehrevaluationen, Akkreditierungen, etc.)
- hochschulweites, verteiltes Dateisystem für persönliche Zwecke (MyData)

Die eSA wurde bereits in Kapitel 5 dargestellt. Sie ist ein eigenes Verzeichnis im Studentenbaum, in dem das Prüfungsamt alle Leistungen des Studenten einträgt. D.h. das Prüfungsamt besitzt create-Recht (11), read-Recht (11), delete-Recht (11) und ein list-Recht innerhalb des eSA Verzeichnisses. Der Student besitzt ein read-Recht (11) und list-Recht um sich den Inhalt seiner Studierendenakte anschauen zu können (vgl. Abschnitt 5.4).
Neben den summativen Prüfungen kann das vtD auch für die Durchführung und Auswertung von Übungen verwendet werden. Dazu muss der Dateibaum unterhalb des jeweiligen virtuellen Root-Verzeichnisses (vRoot) um ein Verzeichnis „Uebungen“ erweitert werden (siehe Abbildung 8.1). Dieses Verzeichnis kann dann wiederum auf Semesterebene unterteilt werden. Die Prüfungsangabe entspricht dann dem jeweiligen Übungsblatt und durch die Datumswerte des jeweiligen tnodes kann der Bearbeitungszeitraum eines Übungsblattes angegeben werden. Die Studenten legen ihre Lösungen innerhalb ihres Verzeichnisses ab und stellen den Korrekteuren ein Personal-Ticket Toolkit aus, um die Lösungen auszuwerten. Somit kann das Sicherheitskonzept auch für elektronische Übungssysteme angewendet werden.

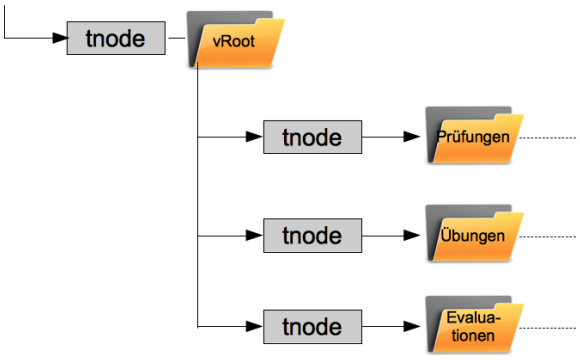

Abbildung 8.1: Erweiterter Dateibaum

Ein solches Übungssystem ist *DUESIE*, das für den Übungsbetrieb zur Einführung in die Informatik an der Universität Siegen eingesetzt wird (siehe [HWB08, HBH+08]). Das System basiert auf der grundlegenden Funktionalität des Prüfungssystem *KLAUSIE*, das in Kapitel 6 kurz beschrieben wurde. Die notwendigen Anpassungen, die an dem *KLAUSIE*-System durchgeführt wurden (siehe Unterabschnitt 7.5.2), gelten auch für das *DUESIE*-System.

Ebenfalls wäre der Einsatz des vtD im Rahmen von Evaluierungen (wie z.B. Lehrevaluierungen) denkbar. Denn durch die Anonymität trotz Authentizität könnte der Dozent nur den Teilnehmern seiner Veranstaltung ein Ticket-Toolkit zur Evaluierung ausstellen, wüsste aber nicht, welcher Student welche Angaben gemacht hat. Aber der Dozent kann sich sicher sein, dass nur die Studenten seine Veranstaltung bewerten, die auch an dieser teilgenommen haben. Das Ticket, das der Student einlöst, ist nur einmalig gültig. Somit kann der Student keine Mehrfachbewertungen durchführen.

Das virtuelle, ticketbasierte Dateisystem könnte auch als einfaches aber hochschulweites verteiltes Dateisystem agieren, das den Studenten, aber auch den Dozenten, die Möglichkeit gibt, Dateien und Verzeichnisse anzulegen und anderen zur Verfügung zu stellen.

Literaturverzeichnis

[ASS07] Rose-Mharie Ahlfeldt, Paolo Spagnoletti, and Guttorm Sindre. Improving the Information Security Model by using TFI. In Hein Venter, Jan Eloff, Mariki Eloff, Les Labuschagne, and Rossouw Solms, editors, *New Approaches for Security, Privacy and Trust in Complex Environments*, volume 232 of *Springer-11645 /Dig. Serial]*, pages 73–85. International Federation for Information Processing, Boston, MA, 2007.

[BÖ8a] Jens Bücking. Organisation elektronischer Prüfungen an der Universität Bremen. `http://www.his.de/publikation/seminar/Workshop_E-Pruefung/TOP11.pdf`, 2008. 29.09.2009.

[BÖ8b] Jens Bücking. Rechtlich relevante Merkmale von eKlausuren and der Universität Bremen. `http://www.eassessment.uni-bremen.de/vortraege.php`, 2008. 21.01.2010.

[BÖ09] Jens Bücking. Testcenter Universität Bremen. `http://www.elc.uzh.ch/veranstaltungen/GMW-Workshop2009/Referat_Buecking_UZH.pdf`, 18.06.2009. 24.11.2009.

[BB04] Peter Biltzinger and Herbert Bunz. Erarbeitung einer Strategie zur Einführung der Gesundheitskarte: Sicherheitsarchitektur, 2004.

[Beh08] Agnieszka Behrens. Untersuchung und prototypische Entwicklung eines Java-basierten Systems für elektronsiche Prüfungen an Hochschulen auf Basis von XML-Standards. Master's thesis, Universität Siegen, 2008.

[Ber08] Klaus Bernshausen. Erstellung einer Sicherheitsarchitektur für Online Klausuren an Hochschulen auf Basis der Lösungsarchitektur der elektronischen Gesundheitskarte. Master's thesis, Universität, Siegen, 2008.

[Beu05] Andre Beunink. Sicherheitsmodelle, Autorisierung und Zugriffskontrolle: Seminarvortrag WS04/05. http://www.bs.informatik.uni-siegen.de/www/lehre/ws0405/sec/8_Folien.ppt, 2005. 18.12.2009.

[BG09] Patrick Brunner and Christian Guretzki. OLAT Scalability. http://www.olat.org/website/en/download/coco/mi/cg_skalierbarkeit.pdf, 2009. 25.05.2010.

[BH04] Torsten Brinda and Andreas Hoffmann. Entwicklung von Software zur Exploration im Bildungskontext. In Gregor Engels and Silke Seehusen, editors, *DeLFI 2004*, volume 52 of *GI-EditionProceedings*, pages 343–354, Bonn, 2004. Ges. für Informatik.

[Blo08] Egon Bloh. E-/Online-Assessment - Einführung und Überblick. http://www.vcrp.de/fileadmin/pdf/LMSDesignerKonferenz2008/bloh.pdf, 08.10.2008. 14.07.2009.

[BMB+05] Roland Bless, Stefan Mink, Erik-Oliver Blaß, Michael Conrad, Hans-Joachim Hof, Kendy Kutzner, and Marcus Schöller. *Sichere Netzwerkkommunikation: Grundlagen, Protokolle und Architekturen*. X.Systems.press. Springer, Berlin, 2005.

[Bod09] Markus Bode. Verschlüsselung und Signierung von HTTP-Requests für elektronische Prüfungen mittels Software-Konnektor. Master's thesis, Universität, Siegen, 03.03.2009.

[Bor08] Christian M. Borchers. *Die Einführung der elektronischen Gesundheitskarte in das deutsche Gesundheitswesen: Datenschutzrechtliche Risiken und potentielle Gefahren strafrechtlich relevanten Missbrauchs*, volume 12 of *Das Strafrecht vor neuen Herausforderungen*. Logos-Verl., Berlin, 2008.

[Bre08] Jens Brennscheidt. Erstellung eines Sicherheitskonzepts für computergestützte Prüfungen mittels mobiler Ausführungsplattformen und digitaler Signaturen. Master's thesis, Universität, Siegen, 30.01.2008.

[BSfW08] Forschung und Kunst Bayerisches Staatsministerium für Wissenschaft. Zielvereinbarung zwischen der Universität Augsburg und dem Bayerischen Staatsministerium für Wissenschaft, Forschung und Kunst. 06.04.2010, 2008. http://www.stmwfk.bayern.de/Hochschule/pdf/zv09_uni_augsburg.pdf.

[BSW04] Albrecht Beutelspacher, Jörg Schwenk, and Klaus-Dieter Wolfenstetter. *Moderne Verfahren der Kryptographie: Von RSA zu Zero-Knowledge*. Vieweg, Wiesbaden, 5., verb. aufl. edition, 2004.

[BW07] Dieter Bartmann and Martin Wimmer. Kein Problem mehr mit vergessenen Passwörtern: Web-basiertes Password Reset mit dem psychometrischen Merkmal Tippverhalten. *Datenschutz und Datensicherheit (DuD)*, 31(3):199–202, 2007.

[BWB+02] Marion Bultmann, Rita Wellbrock, Heinz Biermann, Jürgen Engels, Walter Ernestus, Udo Höhn, Rüdiger Wehrmann, and Andreas Schurig. Datenschutz und Telemedizin: Anforderungen an Medizinnetze. `http://www.datenschutz-bayern.de/verwaltung/DatenschutzTelemedizin.pdf`, 2002. 24.02.2010.

[Cau05] Jörg Caumanns. Management von Zugriffen auf medizinische Daten: Tickets und Berechtigungen: Arbeitspapier, 2005.

[Cau06] Jörg Caumanns. Der Patient bleibt Herr seiner Daten: Realisierung des eGK-Berechtigungskonzepts über ein ticketbasiertes, virtuelles Dateisystem. *Informatik-Spektrum*, 29(5):323–331, 2006.

[Cri07] Geoffrey Crisp. *The e-Assessment Handbook*. Continuum International Puiblishing Group, New York, 2007.

[CWF+06] Jörg Caumanns, Herbert Weber, Arne Fellin, Holger Kurrek, Oliver Boehm, Jan Neuhaus, Jörg Kunsmann, and Bruno Struif. Die eGK-Lösungsarchitektur: Architektur zur Unterstützung der Anwendungen der elektronischen Gesundheitskarte. *Informatik-Spektrum*, 29(5):341–348, 2006.

[Dar09] Alexander Daraban. Erweitertes Sicherheitskonzept für elektronische Prüfungen mitels serverseitigen Software-Proxys. Master's thesis, Universität Siegen, Fachbereich elektrotechnik und Informatik, 2009.

[Eck06] Claudia Eckert. *IT-Sicherheit: Konzepte, Verfahren, Protokolle*. Oldenbourg, München, 4., überarb. aufl. edition, 2006.

[Eck09] Claudia Eckert. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. Oldenbourg, München, 6., überarb. und erw. aufl. edition, 2009.

[EGK08] Björn Eilers, Susanne Gruttmann, and Herbert Kuchen. Konzeption eines integrierbaren Systems zur computergestützten Lernfortschrittskontrolle. In Heinz Lothar Grob, Jan vom Brocke, and Christian Biddendick, editors, *E-Learning-Management*, pages 215–234. Franz Vahlen, München, 2008.

[Eib08a] Christian J. Eibl. Entwicklung von E-Learning-Designkriterien und Implikation für die Informationssicherheit. In Silke Seehusen, Ulrike Lucke, and Stefan Fischer, editors, *DeLFI 2008*, pages 377–388, Bonn, 2008. Ges. für Informatik.

[Eib08b] Christian J. Eibl. Vertraulichkeit persönlicher Daten in Lern-Management-Systemen. In Silke Seehusen, Ulrike Lucke, and Stefan Fischer, editors, *DeLFI 2008*, pages 317–328, Bonn, 2008. Ges. für Informatik.

[Eib10] Christian J. Eibl. *Discussion of Information Security in E-Learning*. PhD thesis, Universität Siegen, Siegen, 2010.

[EvSS07] Christian J. Eibl, H.J. von Solms, and Sigrid Schubert. Development and Application of a Proxy Server for Transparently, Digitally Signing E-Learning Content. In Hein Venter, Jan Eloff, Mariki Eloff, Les Labuschagne, and Rossouw Solms, editors, *New Approaches for Security, Privacy and Trust in Complex Environments*, volume 232 of *Springer-11645 /Dig. Serial]*, pages 181–192. International Federation for Information Processing, Boston, MA, 2007.

[Fox09] Dirk Fox. Hardware Security Module. *Datenschutz und Datensicherheit (DuD)*, 33(9):564, 2009.

[Fra05] Spezifikation der Lösungsarchitektur zur Umsetzung der Anwendungen der elektronischen Gesundheitskarte, 2005.

[FS03] Niels Ferguson and Bruce Schneier. *Practical cryptography*, volume / Bruce Schneier... of *Schneier's cryptography classics library*. Wiley, Indianapolis, Ind., 2003.

[Gem08] Gematik. Spezifikation der SMC–K: Version 1.1.1. `http://www.bmg.bund.de/cln_151/nn_1168248/SharedDocs/Downloads/DE/GV/GT/Gesundheitskarte/SMC/SMC-Spezifikation-2,templateId=raw,property=publicationFile.pdf/SMC-Spezifikation-2.pdf`, 2008. 03.03.2010.

[Gem09] Gematik. Konnektorspezifikation: Stand: 15.09.2009. `http://www.bmg.bund.de/nn_1210508/SharedDocs/Downloads/DE/GV/GT/Gesundheitskarte/Technische_20Festlegungen_20fuer_20die_20Testverfahren/Dezentrale_20Komponenten/Konnektorspezifikation,templateId=raw,property=publicationFile.pdf/Konnektorspezifikation.pdf`, 2009. 04.07.2010.

[GHK+07] Volker Gruhn, Christian Haase, André Köhler, Torsten Kresse, and Vincent Wolff-Marting. *Elektronische Signaturen in modernen Geschäftsprozessen: Schlanke und effiziente Prozesse mit der eigenhändigen elektronischen Unterschrift realisieren.* Springer-11774 /Dig. Serial]. Friedr. Vieweg & Sohn Verlag | GWV Fachverlage GmbH Wiesbaden, Wiesbaden, 2007.

[Gra03] Frank Graf. *Lernspezifische Sicherheitsmechanismen in Lernumgebungen mit modularem Lernmaterial.* PhD thesis, Technische Universität, Darmstadt, 2003.

[Gro06] Malte Gronau. *Analyse und prototypischer Entwurf einer Java-basierten clientseitigen Sicherheitskomponente zur Authentifizierung und Verbindlichkeit von verteilten Online-Klausuren.* PhD thesis, Universität Siegen, 2006.

[GRSF09] Florian Gnägi, Sandra Roth, Renata Sevcikova, and Joël Fisler. OLAT 6 - Funktionsübersicht. `http://www.olat.org/website/en/download/OLAT_6_0_Funktionsuebersicht.pdf`, 03.02.2009. 05.04.2010.

[GSS+05] Ulrich Glowalla, Stefan Schneider, Maria Siegert, Martin Gotthardt, and Jan Koolmann. Einsatz wissensdiagnostischer Module in elektronischen Prüfungen. In Jörg M Haake, Ulrike Lucke, and Djamshid Tavangarian, editors, *DELFI 2005*, pages 283–294, Bonn, 2005. Ges. für Informatik.

[HBH+08] Andreas Hoffmann, Markus Bode, Christoph Hellweg, Michael Garbas, Alexander Quast, and Marco Nichau. DUESIE - Ein Online-Übungssystem zur Informatik-Ausbildung. In Silke Seehusen, Ulrike Lucke, and Stefan Fischer, editors, *DeLFI 2008*, page 416, Bonn, 2008. Ges. für Informatik.

[Hei08] Peter Heinrich. *Design und Implementierung einer Software für online-Prüfungssysteme.* PhD thesis, Eidgenössische Technische Hochschule, Zürich, 17.08.2008.

[HH08] Jan Hansen and Nadine Hatteh. Datenschutz beim E-Learning - Zum Verhältnis von Kontrolle und Vertrauen in der Informationsgesellschaft. In Silke Seehusen, Ulrike Lucke, and Stefan Fischer, editors, *DeLFI 2008*, pages 329–340, Bonn, 2008. Ges. für Informatik.

[HHS99] Thomas Hoeren and Hoeren-Schüngel. *Rechtsfragen der digitalen Signatur: Eine Einführung in Recht und Praxis der Zertifizierungsstellen*, volume 1 of *Electronic commerce und Recht*. E. Schmidt, Berlin, 1999.

[HK03] Eckehard Hermann and Dieter Keßler. XML-Signaturen in Datenbanken: Archivierung signierter Dokumente und die Unterstützung von Signaturen durch Datenbanken. *Datenschutz und Datensicherheit (DuD)*, 27(12):753–756, 2003.

[HK06] Detlef Hühnlein and Ulrike Korte. *Grundlagen der elektronischen Signatur: Recht - Technik - Anwendung*. SecuMedia, Ingelheim, 2006.

[HNP+09] Ding Huaiyu, Joachim Nink, Michael Pulfrich, Martin Reimer, Michael Schmidt, Joel Tsannang Sokeng, and Rony Wolf. Entwicklung eines Proxy-Servers fuer eine transparente, digitale Signierung von E-Learning-Inhalten - Dokumentation: Projektgruppenbericht. `http://www.die.informatik.uni-siegen.de/pgproxy/files/Dokumentation.pdf`, 2009. 26.01.2010.

[Hof07] Andreas Hoffmann. Ein prozessorientiertes und dienstbasiertes Sicherheitsmodell für elektronische Prüfungen an Hochschulen. Kurzbeitrag. In Christian Eibl, Johannes Magenheim, Sigrid Schubert, and Martin Wessner, editors, *DeLFI 2007*, volume 111 of *GI-EditionProceedings*, pages 297–298, Bonn, 2007. Ges. für Informatik.

[Hop03] Achim Hopbach. *Qualitätssicherung im Zuge des Bologna-Prozesses: Deutschland ein Jahr vor Berlin 2003 ; Dokumentation zur gleichnamigen Tagung am 7./8. November 2002 in Bonn*. Forum der Hochschulpolitik. Bertelsmann, Bielefeld, 2003.

[HQC01] Andreas Hoffmann and Klaus Quibeldey-Cirkel. Zentnerlast im Netz. *e-commerce Magazin*, (06/01):54–57, 2001.

[HS07] Andreas Hoffmann and Michael Schuhen. Online-Testen: Eine alternative Form der Leistungsabfrage. Posterbeitrag. `http://www.bs.informatik.uni-siegen.de/www/mitarbeiter/hoffmann/Poster_Didaktik2007.pdf`, 2007. 29.06.2010.

[HW08] Andreas Hoffmann and Roland Wismüller. Sicherheitskonzept für elektronische Prüfungen an Hochschulen auf Basis eines ticketbasierten, virtuellen Dateisystems. In Silke Seehusen, Ulrike Lucke, and Stefan Fischer, editors, *DeLFI 2008*, pages 197–208, Bonn, 2008. Ges. für Informatik.

[HWB08] Andreas Hoffmann, Roland Wismüller, and Markus Bode. Online-Übungssystem zur Programmierausbildung zur Einführung in die Informatik. In Silke Seehusen, Ulrike Lucke, and Stefan Fischer, editors, *DeLFI 2008*, pages 173–184, Bonn, 2008. Ges. für Informatik.

[HWB09] Andreas Hoffmann, Roland Wismüller, and Markus Bode. Realisierung eines Sicherheits- und Rechtemanagements für elektronische Prüfungen an Hochschulen mittels Software-Proxy. In Andreas Schwill and Nicolas Apostolopoulos, editors, *Lernen im digitalen Zeitalter*, volume 153 of *GI-EditionProceedings*, pages 271–282, Bonn, 2009. Ges. für Informatik.

[Kal08] Nadine Kalberg. Die Verwertung von E-Learning-Produkten aus urheberrechtlicher Sicht. In Heinz Lothar Grob, Jan vom Brocke, and Christian Biddendick, editors, *E-Learning-Management*, pages 63–84. Franz Vahlen, München, 2008.

[KF08] Iris Kirchner-Freis. *Rechtliche Aspekte des eLearning und eAssessment: Ein Praxisleitfaden.* Kirchner Andree Prof. Dr., Bremen, version 2.0 edition, 2008.

[Kno06] Michael Knorr. Datenschutzkonforme Protokollierung. *Datenschutz und Datensicherheit (DuD)*, 30(5):268–269, 2006.

[Kri05] Wolfgang Krischke. Der elektronische Prüfer. *Die Zeit*, (32), 4. August 2005.

[Lan07] Frank Christoph Langer. *Erprobung eines internetbasierten Prüfungssystems in der zahnmedizinischen Ausbildung und die Beurteilung der Grenzen, Chancen und Nutzen.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität, Bonn, 2007.

[LH09] Kai-Uwe Loser and Thomas Herrmann. Ansätze zur Entwicklung datenschutzkonformer E-Learning-Plattformen. In Andreas Schwill and Nicolas Apostolopoulos, editors, *Lernen im digitalen Zeitalter*, volume 153 of *GI-EditionProceedings*, pages 79–90, Bonn, 2009. Ges. für Informatik.

[MÖ9] Daniel Möbs. Rechtssicherheit bei e-Prüfungen: Es wird Zeit für einen Standard! `http://www.codiplan.de/download.html?file=tl_files/Download/Rechtssicherheit+bei+e-Pruefungen+%E2%80%93+Es+wird+Zeit+fuer+einen+Standard%21.pdf`, 2009. 29.05.2009.

[Max07] Giuseppe Maxia. Getting Started with MySQL Proxy. `http://www.oreillynet.com/lpt/a/7098`, 2007. 27.04.2010.

[Ö06] Salih Örtlek. Referenzimplementierung der gegenseitigen Kartenauthentisierung von elektronischen Gesundheitskarten, Heilberufsausweisen und Sicherheitsmodulkarten. Master's thesis, Technische Universität, Darmstadt, 2006.

[PF07] Michael Piotrowski and W. Fenske. Interoperabilität von elektronischen Tests. In Christian Eibl, Johannes Magenheim, Sigrid Schubert, and Martin Wessner, editors, *DeLFI 2007*, volume 111 of *GI-EditionProceedings*, pages 185–196, Bonn, 2007. Ges. für Informatik.

[Por03] Ulrich Pordesch. *Die elektronische Form und das Präsentationsproblem: Techn. Univ., Diss.–Ilmenau, 2002.*, volume 7 of *Der elektronische Rechtsverkehr*. Nomos Verl.-Ges, Baden-Baden, 1. aufl. edition, 2003.

[PR05] Michael Piotrowski and Dietmar Rösner. Integration von E-Assessment und Content-Management. In Jörg M Haake, Ulrike Lucke, and Djamshid Tavangarian, editors, *DELFI 2005*, pages 129–140, Bonn, 2005. Ges. für Informatik.

[Ree06] Jan-Armin Reepmeyer. Entwicklung eines Rahmens für den Einsatz eines computergestützten Prüfungssystems. Münster, 2006.

[Ree08a] Jan-Armin Reepmeyer. Onlinklausuren. In Heinz Lothar Grob, Jan vom Brocke, and Christian Biddendick, editors, *E-Learning-Management*, pages 257–274. Franz Vahlen, München, 2008.

[Ree08b] Jan-Armin Reepmeyer. Rechtssichere E-Prüfungen. `http://www.his.de/publikation/seminar/Workshop_E-Pruefung/TOP03.pdf`, 2008. 06.04.2010.

[RMP06] Jim Ridgway, Sean McCusker, and Daniel Pead. Literature Review of E-assessment. `http://www.futurelab.org.uk/resources/documents/lit_reviews/Assessment_Review.pdf`, 12.09.2006. 14.07.2009.

[Roß09] Alexander Roßnagel. *Rechtssichere Transformation signierter Dokumente*, volume 21 of *«Der» elektronische Rechtsverkehr*. Nomos, Baden-Baden, 1. aufl. edition, 2009.

[RSNSS07] Cornelia Ruedel, Mandy Schiefner, Caspar Noetzli, and Eva Seiler Schiedt. Risikomanagement für eAssessment. In Marianne Merkt, Kerstin Mayrberger, Rolf Schulmeister, Angela Sommer, and Ivo van den Berg, editors, *Studieren neu erfinden - Hochschule neu entdecken*, volume 44, pages 180–190. Waxmann, Münster, 2007.

[Rue09] Cornelia Ruedel. Einführung E-Assessment & Organisation von E-Assessment. `http://www.elc.uzh.ch/veranstaltungen/GMW-Workshop2009/Referat_Ruedel_UZH.pdf`, 18.06.2009. 13.04.2010.

[RZ06] Heiko Roßnagel and Jan Zibuschka. Single Sign On mit Signaturen: Integration von elektronischen Signaturen und Passwortsysremen. *Datenschutz und Datensicherheit (DuD)*, 30(12):773–777, 2006.

[SBBH08] William Stallings, Lawrie Brown, Mick Bauer, and Michael Howard. *Computer security: Principles and practice*. Pearson Education/Prentice Hall, Upper Saddle River, NJ, internat. ed. edition, 2008.

[Sch04] Sandra Schaffert. *Einsatz von Online-Prüfungen in der beruflichen Weiterbildung: Gegenwart und Zukunft*. PhD thesis, Bonn, 13.10.2004. 29.05.2009.

[Sch06a] Jürgen Schmidt. Sichere mobile Bearbeitung und Aufbewahrung vertraulicher Daten. In Patrick Horster, editor, *D-A-CH Mobility 2006*, IT Security & IT Management, pages 307–312. syssec, Klagenfurt, 2006.

[Sch06b] Stefan Schneider. Das Secure-Browser-System "WinKeyox" zur sicheren Durchführung von E-Klausuren. In Max Mühlhäuser, Guido Rößling, and Ralf Steinmetz, editors, *DeLFI 2006*, volume 87 of *GI-EditionProceedings*, pages 399–400, Bonn, 2006. Ges. für Informatik.

[Sch06c] Bruce Schneier. *Angewandte Kryptographie: Protokolle, Algorithmen und Sourcecode in C ; [der Klassiker]*. InformatikKryptographie. Pearson Studium, München, [2. aufl.] edition, 2006.

[Sch08] Kai Schwedes. Ressourceneinsatz bei elektronischen Prüfungen an der Universität Bremen. `http://www.his.de/publikation/seminar/Workshop_E-Pruefung/TOP12.pdf`, 2008. 05.10.2009.

[Sch09a] Klaus Schmeh. Countdown: Anwendungen mit dem elektronischen Personalausweis. *iX*, (9/2009):96–99, 2009.

[Sch09b] Klaus Schmeh. *Elektronische Ausweisdokumente: Grundlagen und Praxisbeispiele*. Hanser, München, 2009.

[Sch09c] Klaus Schmeh. *Kryptografie: Verfahren, Protokolle, Infrastrukturen*. iX Edition. dpunkt-Verl., Heidelberg, 4., aktualisierte und erw. aufl. edition, 2009.

[Sch09d] Baron Schwartz. *High Performance MySQL: Optimierung, Backups, Replikation und Lastverteilung ; [fortgeschrittene Techniken für MySQL-Administratoren]*. O'Reilly, Beijing, 2. aufl., dt. ausg. edition, 2009.

[Ste06] Monika Steinberg. Organisatorisches Konzept für Online-Prüfungsverfahren: Ein Stufenmodell für die Realisierung von Online-Assessment. `http://www.sra.uni-hannover.de/fileadmin/uploads/Forschung/Publikationen/2006/PerU06_E-Lehre_Brehm_Steinberg.pdf`, 2006. 07.07.2009.

[Str08] Jörg Stratmann. Das Testcenter der Universität Duisburg-Essen als Teil einer gesamt-universitären E-Strategie. `http://www.his.de/publikation/seminar/Workshop_E-Pruefung/TOP13.pdf`, 2008. 05.10.2009.

[TEM+08] Ulf Troppens, Rainer Erkens, Wolfgang Müller, Nils Haustein, and Rainer Wolafka. *Speichernetze: Grundlagen und Einsatz von Fibre Channel SAN, NAS, iSCSI und InfiniBand*. iX-Edition.

dpunkt-Verl., Heidelberg, 2., aktualisierte und erw. aufl. edition, 2008.

[TH08] Jürgen Taeger and Janine Horn. *Rechtsfragen der Nutzung neuer Medien und des Internets an Hochschulen*, volume 2 of *Schriften zum Zivil- und Wirtschaftsrecht*. OlWIR Oldenburger Verl. für Wirtschaft Informatik und Recht, Edewecht, 2008.

[Vad03] Derek Vadala. *Managing RAID on Linux*. O'Reilly, Beijing, 1. ed. edition, 2003.

[Vog05] Marko Vogel. Single-Sign-On in Unternehmen. In Patrick Horster, editor, *D-A-CH Security 2005*, IT Security & IT Management, pages 52–63. syssec, Klagenfurt, 2005.

[VS09] Michael Vogt and Stefan Schneider. E-Klausuren an Hochschulen: Didaktik - Technik - Systeme - Recht - Praxis. `http://cms.uni-kassel.de/unicms/fileadmin/groups/w_430000/Download/E-Klausuren-an-Hochschulen.pdf`, 2009. 28.05.2009.

[Wan06] Klaus Wannemacher. Computerbasierte Prüfungen – Zwischen Self-Assessment und Abschlussklausuren. In Eva Seiler Schiedt, Siglinde Kälin, and Christian Sengstag, editors, *E-Learning - alltagstaugliche Innovation?*, volume 38 of *Medien in der Wissenschaft*, pages 163–172. Waxmann, Münster, 2006.

[Wei05] Edgar R Weippl. *Security in e-learning*, volume 16 of *Advances in information security*. Springer, New York, NY, 2005.

[Wet08a] Günter Wetter. E-Klausuren mit ILIAS an der Uni Mainz. `http://www.his.de/publikation/seminar/Workshop_E-Pruefung/TOP05.pdf`, 2008. 12.10.2009.

[Wet08b] Michael Wettern. Zur Einhaltung des Datenschutzes an Hochschulen. *Datenschutz und Datensicherheit (DuD)*, (7):466–468, 2008.

[WKD09] Klaus Wannemacher, Bernd Kleimann, and Lars Degenhardt. Vor einem Kulturwandel? Über elektronische Prüfungen an Hochschulen. *Forschung & Lehre*, 16(07/09):5002–5503, 2009.

[ZB07] Wolfgang Zimmerling and Robert G. Brehm. *Prüfungsrecht: [Verfahren, vermeidbare Fehler, Rechtsschutz]*. Heymann, Köln, 3., überarb. und erw. aufl. edition, 2007.

[ZIM09] ZIM. Das neue Klausurenzentrum an der UDE. Das Projekt "Kompetenzzentrum PC-gestützte Prüfungen". http://www.uni-due.de/imperia/md/content/zim/projekte/info-workshop_pc-klausuren_16072009.pdf, 2009. 15.01.2010.

[ZMM09] ZMML. eKlausuren: Rollenkonzept, Arbeitspakete und Ablaufplanung. http://www.eassessment.uni-bremen.de/documents/Organisation_eKlausuren2009.pdf, 2009. 29.09.2009.