

**Zwischen
der Präsidentin der Universität Koblenz-Landau
und
dem Gesamtpersonalrat der Universität Koblenz-Landau
wird folgende
Dienstvereinbarung
über Einführung, Anwendung, Erweiterung und Änderungen
einer Web-Content-Management-Software
(nachfolgend CMS-Software)
abgeschlossen:**

Vorbemerkung

Die Universität Koblenz-Landau und der Gesamtpersonalrat schließen diese Dienstvereinbarung um Einführung, Anwendung, Erweiterung und Änderungen einer CMS-Software durchzuführen. Hochschulleitung und Gesamtpersonalrat sind sich darin einig, dass ein zukunftsorientierter Einsatz von neuen IT-Technologien dem Wohle der Hochschule sowie seiner Mitarbeiter zu dienen hat. Ziel des Einsatzes von DV – Systemen, so auch bei der Einführung und dem Einsatz der CMS-Software, ist die Optimierung und Stärkung der Leistungsfähigkeit, die Steigerung der Dienstleistungsqualität, die Verbesserung der Arbeitsbedingungen der Beschäftigten sowie die Nutzung des Systems zur Qualitätssicherung. Um die gemeinsamen Ziele zu erreichen, werden die Universität Koblenz-Landau und der Gesamtpersonalrat eine kooperative Zusammenarbeit praktizieren, in der sich beide Parteien dafür einsetzen, praktikable Lösungen für die Einführung, Anwendung, Erweiterung und Änderungen zu finden.

§ 1

Zielsetzung

- (1) Die CMS-Software soll die Pflege von Webseiten grundlegend vereinfachen. Die Software stellt eine Oberfläche für gängige Webbrowser zur Verfügung, mittels derer die Erstellung und Pflege von Webseiten ohne spezifische Softwarekenntnisse gelingt, um damit eine dezentrale Wartung von Internetangeboten zu ermöglichen und die Qualität der von der Universität im Netz zu verantworteten Seiten insgesamt zu verbessern.
- (2) Durch diese Dienstvereinbarung bestimmt der Gesamtpersonalrat bei der Einführung, Anwendung, Änderung und Erweiterung der CMS-Software gemäß § 80 Abs. 2 Nr. 1-4 Landespersonalvertretungsgesetz mit.
- (3) Ziel dieser Dienstvereinbarung ist, in Ergänzung der einschlägigen gesetzlichen und tariflichen Bestimmungen
- den Schutz der personenbezogenen Daten der Beschäftigten zu gewährleisten,
 - unzulässige individuelle Verhaltens- und Leistungskontrollen im Zusammenhang mit dem Einsatz von CMS-Software auszuschließen,
 - der Universität, den Instituten und den zuständigen Personalvertretungen

einen Rahmen für die Schritte zu geben, die bei der Einführung, dem Einsatz und dem Ausbau und der Änderung der CMS-Software erfolgen müssen.

(4) Unter Berücksichtigung der Aufgaben der Universität Koblenz-Landau wird bei der Einführung, dem Einsatz, dem Ausbau oder der Änderung der CMS-Software insbesondere darauf geachtet, dass die bisherigen Handlungs- und Entscheidungsspielräume der Beschäftigten erhalten bleiben und Überlastungen und Unterforderungen der Beschäftigten vermieden werden.

§ 2

Geltungsbereich

(1) Die Dienstvereinbarung gilt für alle Beschäftigten im Sinne des LPersVG der Universität Koblenz-Landau, die eine Berechtigung zur Nutzung der CMS-Software erhalten.

(2) Die Dienstvereinbarung gilt für alle durch die CMS-Software unterstützten Arbeitsplätze und alle technischen und organisatorischen Entwicklungen bzw. Maßnahmen, die mit der Einführung, Anwendung, Erweiterung und Veränderung der CMS-Software verbunden sind. Dies gilt unabhängig davon, ob diese Maßnahmen betriebsintern oder durch externe Institutionen durchgeführt werden.

§ 3

Dokumentation des Systems

Als CMS-Software wird die in den Anlagen 1 bis 3 abschließend beschriebene Software betrieben.

§ 4

Verarbeitung personenbezogener Daten

(1) Personenbezogene Daten dürfen im Rahmen des Einsatzes der CMS-Software nur erhoben, gespeichert, übermittelt oder in sonstiger Weise verarbeitet werden, soweit diese zur Authentifizierung am System und zur Feststellung von Nutzungsberechtigungen oder für die Erstellung und Pflege von Webseiten erforderlich sind.

(2) Als personenbezogene Daten fallen solche Daten an, die Teil der Webseiten sind, also Content der CMS-Software darstellen („im CMS gespeichert“), und solche personenbezogenen Daten, die zum Betrieb und zur Nutzung der CMS-Software erhoben werden („für das CMS gespeichert“).

(3) Eine über Absatz 1 hinausgehende Erfassung, Verarbeitung, Übermittlung und Nutzung der für das CMS gespeicherten Daten ist unzulässig.

(4) Die Absätze 1 bis 3 finden auf nicht personenbezogene Daten im Sinne der datenschutzrechtlichen Bestimmungen keine Anwendung.

§ 5

Betrieb der CMS-Software, Protokollierung

(1) Die Systemverwaltung der CMS-Software erfolgt durch die in Anlage 4 festgelegten Beschäftigten der Universität Koblenz-Landau.

(2) Eine Protokollierung von User-Logins ist lediglich zum Zwecke der System-sicherheit zulässig.

(3) Eine nachträgliche Veränderung der Protokolldateien wird technisch ausgeschlossen. Die Dateien bzw. die Ausdrücke sind höchstens 6 Monate aufzubewahren.

(4) Die Dienststellenleitung verpflichtet sich, die Protokolldateien keinesfalls zu anderen als den in Abs. 2 beschriebenen Zwecken einzusetzen.

§ 6

Wartung der CMS-Software

(1) Die Betriebssicherheit der CMS-Software wird durch die/den in der Anlage 4 CMSBeauftragte/n sichergestellt.

(2) Bei Krankheit oder Urlaub vertreten die in Anlage 4 genannten Personen die/den CMSBeauftragten.

§ 7

Datensicherheit

(1) Die CMS-Software und die für ihren Betrieb betriebenen Server werden durch arbeitsorganisatorische, technische und räumliche Maßnahmen gegen eine unbefugte Benutzung gesichert. Zugang haben nur diejenigen Bediensteten, die in Anlage 4 mit den entsprechenden Befugnissen genannt werden.

(2) Die automatisch protokollierten Nutzungsdaten der Bediensteten werden spätestens nach Ablauf von 6 Monaten gelöscht. Eine über diesen Zeitpunkt hinausgehende Speicherung ist im Einzelfall zulässig, soweit und solange die Präsidentin und der Gesamtpersonalrat dies für erforderlich halten.

§ 8

Informationsrechte des Gesamtpersonalrats und der Datenschutzbeauftragten

(1) Die Dienststellenleitung stellt sicher, dass der Gesamtpersonalrat und die Datenschutzbeauftragten der Universität umfassend und rechtzeitig über wesentliche Änderungen und Erweiterungen der CMS-Software informiert werden.

(2) Der Gesamtpersonalrat kann verlangen, dass ihm die vollständigen Systemunterlagen der CMS-Software und der mit diesen durchführbaren Operationen und verwendeten Programme zur Verfügung gestellt werden.

§ 9

Information und Qualifikation der Bediensteten

(1) Die Beschäftigten werden von der Dienststellenleitung vor Beginn und bei Änderung ihrer Nutzungsmöglichkeiten der CMS-Software über Art (Rolle und Bereich gem. Anlage 3) und Bedienung der von ihnen nutzbaren Leistungsmerkmale schriftlich unterrichtet. Weiterhin werden sie darauf hingewiesen, dass die Zuweisung einer anderen Nutzungsart mit Schreib- oder Änderungsrechten nur durch die Personalabteilung erfolgen kann und dass die Weitergabe des eigenen Passwortes oder die Nutzung des Passwortes eines anderen Beschäftigten unzulässig ist und zu arbeitsrechtlichen Konsequenzen führen kann.

(2) Aufgabe der Universität Koblenz-Landau ist es, die Beschäftigten entsprechend den ihnen übertragenen Aufgaben im notwendigen Umfang aus-, fort- und weiterzubilden.

(3) Das Recht der Bediensteten, Auskunft über die zu ihrer Person gespeicherten Daten nach Maßgabe der geltenden datenschutzrechtlichen Bestimmungen zu verlangen, bleibt durch Regelungen dieser Dienstvereinbarung unberührt.

§ 10

Schlussbestimmungen

(1) Diese Dienstvereinbarung tritt mit der Unterzeichnung durch die Dienststellenleitung und den Personalrat in Kraft.

(2) Die Dienstvereinbarung kann einvernehmlich geändert werden; Änderungen bedürfen der Schriftform.

(3) Die Dienstvereinbarung kann von jeder Vertragspartei mit einer Frist von drei Monaten zum Ende eines Kalenderjahres gekündigt werden. Die Kündigung bedarf der Schriftform. Nach Kündigung der Dienstvereinbarung gelten ihre Regelungen weiter, bis sie durch eine andere Dienstvereinbarung ersetzt werden.

(4) Sollten einzelne Bestandteile der Dienstvereinbarung ungültig werden, so bleiben die anderen Bestandteile der Dienstvereinbarung davon unberührt.

(5) Die Anlagen sind Bestandteile dieser Dienstvereinbarung. Verändern sich die Anlagen, so bedarf es lediglich einer Zustimmung durch den Gesamtpersonalrat zu den geänderten/ergänzten Anlagen.

Mainz, den Universität Koblenz-Landau <i>May-Britt Kallenrode</i> Präsidentin Prof. Dr. May-Britt Kallenrode	Landau, den <i>02/05/18</i> Universität Koblenz-Landau <i>B. Wunder</i> Gesamtpersonalratsvorsitzender Bernd Wunder
--	---

Anlage 1: Leistungsbeschreibung der CMS-Software

Anlage 2: Liste der installierten Module

Anlage 3: Rollenkonzept

Anlage 4: Systemverwaltung

Anlage 1: Leistungsbeschreibung

Inhaltsverzeichnis

1	Zweck des Dokuments	3
2	Nicht-funktionale Anforderungen	3
2.1	Angenehme Benutzung	3
2.1.1	Erstellung und Pflege von Internetseiten	3
2.1.2	Unkompliziertheit	3
2.1.3	Komfortabler Texteditor	4
2.2	Barrierefreiheit, Kompatibilität	4
2.2.1	Barrierefreiheit	4
2.2.2	Browserkompatibilität	4
2.2.3	Responsive Darstellung.....	5
3	Funktionale Anforderungen.....	5
3.1	Benutzung.....	5
3.1.1	Druckansicht	5
3.1.2	Fehlerbehandlung in Formularen	5
3.1.3	Kontaktformular	6
3.2	Inhalte verschiedenen Typs	6
3.2.1	Contenttypen.....	6
3.2.2	Seite	6
3.2.3	Person.....	6
3.2.4	Ordner	7
3.2.5	Objekte aus Navigationsbaum ausblenden	7
3.2.6	Literaturlisten	7
3.2.7	Eventkalender.....	7
3.2.8	Dateien	8
3.2.9	Lehrveranstaltung.....	8
3.2.10	Meldungen	9
3.2.11	Termine	9
3.2.12	Newsletter	9
3.2.13	Sprechstundenzeiten.....	9
3.2.14	Funktion einer Person	10
3.2.15	Metadaten.....	10
3.2.16	Automatische Textersetzung.....	10
3.2.17	Erweiterung für Portalseiten	11
3.2.18	Vorlagen	11
3.3	Berechtigungen.....	12
3.3.1	Rollen.....	12
3.3.2	Zugriffsschutz.....	12
3.3.3	LDAP-Anbindung.....	12
3.3.4	Single-Sign-On.....	12
3.4	Workflow, Versionierung, Freigabe, Mehrsprachigkeit.....	13
3.4.1	Workflow	13
3.4.2	Mehrsprachigkeit.....	13
3.4.3	Zeitsteuerung: Veröffentlichungszeitraum.....	13

3.4.4	Versionskontrolle.....	14
3.5	Suche, Links	14
3.5.1	Suchfunktion.....	14
3.5.2	Link-Checker	14
3.5.3	Lesbare URL	14
3.5.4	Bezeichnung der URL.....	15
3.5.5	Sitemap.....	15
	Datenaustausch mit anderen Systemen	15
3.6	15
3.6.1	Externe Systeme	15
3.6.2	Trennung von Inhalt und Layout.....	15
3.6.3	XML-Export	16
3.7	Administration des Systems, Organisatorisches.....	16
3.7.1	Mehrere getrennte Internetseiten	16
3.7.2	Layout-Anpassung	16
3.7.3	Eigenentwicklungen.....	17
3.7.4	Zugriffsstatistik	17
3.7.5	Zuständige Personen	17
3.7.6	Umleitungen	17

1 Zweck des Dokuments

Dieses Dokument beschreibt den Leistungsumfang des zentralen Internetauftritts der Universität Koblenz · Landau (UniCMS). Es ist eine Anlage zur Dienstvereinbarung. Die Beschreibung geht von den *Anforderungen* aus, die zu Projektbeginn erhoben und im Verlauf fortgeschrieben wurden, und erfasst damit das gesamte Spektrum dessen, was das UniCMS an Funktionalitäten bietet.

Die einzelnen Anforderungen werden aus der Sichtweise eines Benutzers beschrieben, d.h. der Fokus liegt hier auf nicht-technischen Aspekten.

Die Anforderungen sind unterteilt in *funktionale* und *nicht-funktionale* Anforderungen. Funktionale Anforderungen beschreiben Dienste und Funktionen, die durch das UniCMS bereit gestellt werden sollen. Nicht-funktionale Anforderungen betreffen Aspekte wie die Benutzbarkeit. Es sind Anforderungen, die keine konkreten Funktionen oder Dienste betreffen, sondern allgemeiner und oft auch schwer messbarer Art sind.

2 Nicht-funktionale Anforderungen

2.1 Angenehme Benutzung

2.1.1 Erstellung und Pflege von Internetseiten

Anforderung: Das System ermöglicht den Benutzern die Erstellung und Pflege von Internetseiten.

Die zentrale Aufgabe des Systems ist die Unterstützung bei der Erstellung und der Pflege von Internetseiten sowie deren Veröffentlichung. Die Zuständigkeit für die Inhalte liegt bei den einzelnen Mitarbeiterinnen und Mitarbeitern, die das System benutzen ('Benutzer'). Das System ist daher auf ihre Bedürfnisse zugeschnitten.

Das System bietet Zugang zu den wesentlichen Funktionen der Inhaltserstellung und -pflege. Änderungen werden gemäß des eingestellten Workflows (siehe *Workflow*, Abschnitt 3.4.1) vom System verwaltet und auf den Internetseiten angezeigt.

2.1.2 Unkompliziertheit

Anforderung: Das Erstellen und Pflegen von Internetseiten ist unkompliziert und ohne Zusatzsoftware möglich.

Das Erstellen neuer Internetseiten und das Pflegen bestehender Internetseiten bezieht sich hier auf die Erstellung neuer und die Änderung bestehender *Inhalte*. Die Tätigkeit wird von der Rolle *Editor* vorgenommen.

Die Korrektur eines Tippfehlers, die Ergänzung eines Textes oder die Erstellung einer neuen Seite für eine Lehrveranstaltung sollten Tätigkeiten sein, die man 'mal gerade eben' erledigen kann. Der benötigte Zeitaufwand sollte nicht wesentlich höher sein, als würde man die gleichen Inhaltsänderungen in einem Textverarbeitungsprogramm vornehmen.

Bestimmte Routineaufgaben werden zusätzlich noch durch die Bereitstellung von Vorlagen unterstützt (siehe *Vorlagen*, Abschnitt 3.2.18).

Die Bedienung des Systems kann mit üblichen PC-/Office-Kenntnissen erfolgen. Für die Inhaltspflege sind keine zusätzlichen Programme auf dem PC notwendig. Der Zugang zum

System erfolgt durch den Browser. Eine Ausnahme bilden spezielle Arten von Inhalten, z.B. Grafiken, für deren Bearbeitung spezielle Programme verwendet werden. Der Hauptanteil der Inhalte jedoch besteht aus Text, der direkt im Browser über eine komfortable Benutzungsschnittstelle bearbeitet werden kann (*Komfortabler Texteditor*, Abschnitt 2.1.3).

2.1.3 Komfortabler Texteditor

Anforderung: Texte mit Formatierungen können mit einem komfortablen Editor bearbeitet werden.

Im UniCMS wird ein WYSIWYG-Editor verwendet. Dieser erlaubt es, Texte innerhalb des Webbrowsers in einer Umgebung zu editieren, die in der Art der Bedienung an ein Office-Programm erinnert. Der Benutzer benötigt keine HTML-Kenntnisse.

2.2 Barrierefreiheit, Kompatibilität

2.2.1 Barrierefreiheit

Anforderung: Die erzeugten Internetseiten genügen den Anforderungen zur Barrierefreiheit.

Die erzeugten Internetseiten des Systems sind auf Barrierefreiheit getestet.

Eine wichtige 'Schutzmaßnahme' gegen den Verstoß gegen Regeln zur Barrierefreiheit ist es, entsprechende Änderungen durch den Benutzer gar nicht erst zuzulassen. Der Benutzer hat beispielsweise keinen Einfluss auf die generelle Seitengestaltung, die durch das UniCMS fest vorgegeben wird.

Beim Erfassen von Texten wird der Benutzer zudem entsprechend unterstützt, indem der Texteditor z.B. automatisch die geforderten ALT-Attribute bei Bildern setzt.

2.2.2 Browserkompatibilität

Anforderung: Die erzeugten Internetseiten sind mit gängigen Webbrowsern darstellbar.

Die erzeugten Internetseiten des Systems sind mit gängigen Browser darstellbar. Konkret müssen die Browser (mindestens) HTML5 und CSS 3 unterstützen.

Umgekehrt bedeutet die Forderung nach der Unterstützung durch die Browser, dass die Internetseite mit den oben genannten Technologien auskommt, bzw. keine anderen benutzt, die hierzu inkompatibel sind. Zur Komfortsteigerung für die Benutzer darf JavaScript eingesetzt werden, die grundlegenden Funktionen der Internetseite sollen jedoch auch bei ausgeschaltetem JavaScript benutzbar sein.

Erfahrungsgemäß bereiten veraltete Webbrowser Probleme bei der Darstellung und führen zu unverhältnismäßig hohem Entwicklungsaufwand, um deren Schwachstellen auszugleichen und die Seiten trotzdem benutzbar zu halten. Daher werden nur Minimalanforderungen an die Funktionalität der Browser gestellt. Ältere Browser werden die Seiten trotzdem anzeigen, die Benutzer werden jedoch Kompromisse in Bezug auf die Darstellung und die Bedienungsfreundlichkeit machen müssen.

Da die Administration der Internetseiten ebenfalls mit einem Browser erfolgt (siehe *Unkompliziertheit*, Abschnitt 2.1.2), gelten die gleichen Anforderungen auch hierfür. Die Administrationsoberfläche lässt sich nur mit eingeschaltetem JavaScript komfortabel bedienen.

2.2.3 Responsive Darstellung

Anforderung: Die Internetseite muss von Endgeräten mit unterschiedlichen Bildschirmgrößen einfach bedienbar sein.

Durch das Konzept der responsiven Darstellung ermöglicht die Internetseite den einfachen Abruf auch von mobilen Endgeräten, wie Tablet oder Smartphone. Das Layout der Internetseite richtet sich dabei immer an die Auflösung und Orientierung des Endgerätes. Eine Bearbeitung der Seite von mobilen Endgeräten ist zwar möglich, aber durch die Bildschirmgröße nicht ganz unkompliziert.

3 Funktionale Anforderungen

3.1 Benutzung

3.1.1 Druckansicht

Anforderung: Internetseiten können ausgedruckt werden. Dazu gibt es jeweils eine separate Druckansicht.

Will ein Seitenbesucher eine Seite ausdrucken, so ist er i.d.R. an den Inhalten der Seite interessiert, nicht an Navigationselementen, Layoutgrafiken etc. Der Ausdruck wird um diese Elemente bereinigt. Grafische Elemente, die zum Inhalt der Seite gehören, also Fotos von Mitarbeitern, oder Lagepläne bei Anreisedaten, gehören natürlich zum Inhalt und damit in die Druckvariante.

Die Druckvariante wird automatisch durch den Webbrowser aktiviert, wenn der Benutzer dessen *Drucken*-Befehl benutzt. Das UniCMS verwendet hierzu die CSS-Medienvariante *print*, in der der Ausdruck entsprechend gesteuert wird. Wichtige Voraussetzung ist hierzu die Trennung von Inhalt und Design (siehe *Trennung von Inhalt und Layout*, Abschnitt 3.6.2).

3.1.2 Fehlerbehandlung in Formularen

Anforderung: Fehleingaben in einem Formular werden erkannt und können komfortabel korrigiert werden.

Einträge in Eingabefeldern in einem Formular werden nach dem Absenden des Formulars auf Korrektheit überprüft. Bei Fehlern erscheint eine entsprechende Meldung, und man muss die Eingabe korrigieren. Tauchen nun mehrere Fehler in einem Formular auf, erkennt das System alle Fehler gleichzeitig und präsentiert die Einträge übersichtlich zur Korrektur und nicht jeden Fehler einzeln (nacheinander).

3.1.3 Kontaktformular

| Anforderung: Auf allen Seiten wird auf ein Kontaktformular verwiesen. |

Gemäß Anforderung *Zuständige Personen*, Abschnitt 3.7.5, gibt es zuständige Personen für jeden Bereich der Internetseite. Ein Besucher einer Seite kann über einen Link ein Kontaktformular aufrufen und dort Kommentare, Fehlermeldungen oder allgemeine Fragen zur aktuellen Seite stellen. Über das Formular wird eine E-Mail erzeugt, die an die zuständige Person für den aktuellen Bereich geschickt wird.

Ist keine zuständige Person für den aktuellen Bereich definiert, dann gehen die Anfragen an den UniCMS-Support bzw. im englischen Bereich der Internetseite an das Welcome Center.

3.2 Inhalte verschiedenen Typs

3.2.1 Contenttypen

| Anforderung: Für die Speicherung von Inhalten gibt es passende Contenttypen. |

Um Inhalte weiterverarbeiten zu können, müssen sie in strukturierter Form gespeichert werden. Dies wird durch spezifische Contenttypen gewährleistet. Beispiele für Contenttypen sind *Person*, *Lehrveranstaltung* oder *Ordner*. Der Contenttyp definiert, welche Angaben zu dem jeweiligen Inhaltsobjekt gemacht werden können und wie diese jeweils genau beschaffen sein müssen. Bei einer Person ist dies beispielsweise die Festlegung, dass der Nachname und der Vorname getrennt voneinander erfasst und gespeichert werden, und dass zu einer Person ein Foto hinterlegt werden kann. Die Eingabe der Daten erfolgt über ein Formular.

Die Notwendigkeit für die zentralen Contenttypen ist durch entsprechende Anforderungen begründet, die in diesem Dokument aufgeführt sind.

3.2.2 Seite

| Anforderung: Es gibt einen Contenttypen *UniPage* zur Speicherung von Internetseiten. |

Eine Internetseite besteht aus einfachem Text mit Formatanweisungen. Das UniCMS speichert hierzu die Texte im HTML-Format. Die Texte können mit einem komfortablen Editor verändert werden (siehe *Komfortabler Texteditor*, Abschnitt 2.1.3).

3.2.3 Person

| Anforderung: Es gibt einen Contenttypen *Person* zur Speicherung von Angaben zu einer Person. |

An verschiedenen Stellen des UniCMS wird auf Angaben zu Personen verwiesen, zum Beispiel durch die Nennung von Kontaktpersonen. Es ist daher sinnvoll, die für diesen Zweck notwendigen Angaben zu speichern. Zu einer Person werden insbesondere die folgenden

Angaben gespeichert: Vorname, Nachname, Dargestellter Name, Raum, Telefon, Mailadresse, Foto, Sprechstunde, Privatadresse (falls gewünscht), Bemerkung.
Es sollte möglich sein aus einer zentralen Datenbank der Personen der Universität Koblenz · Landau die entsprechenden Informationen über Schnittstellen abzurufen.

3.2.4 Ordner

| Anforderung: Die Internetseite lässt sich in Bereiche (Ordner) unterteilen. |

Die Ordner definieren die primäre Navigationsstruktur. Sie werden durch den Contenttypen *UniFolder* implementiert.

3.2.5 Objekte aus Navigationsbaum ausblenden

| Anforderung: Objekte (Contenttypen) können bei Bedarf aus der Navigationsdarstellung verborgen werden. |

Standardmäßig zeigt das UniCMS Objekte (z.B. Ordner, Seiten, etc.) als Navigationspunkt. Allerdings sollen nicht alle Objekte, die man zur Strukturierung der Internetseite anlegt, auch als Navigationspunkte im Navigationsbaum angezeigt werden. Daher kann man einzelne Objekte gezielt verbergen. Ein Beispiel wäre ein Ordner in dem Bilder oder Icons liegen, die für Designzwecke notwendig sind. Dieser Ordner soll für Besucher der Internetseite nicht als eigener Navigationspunkt auftauchen.

Um zu erreichen, dass ein Objekt nie in der Navigation sichtbar ist, sondern nur in der Listenansicht (Inhaltsverzeichnis des übergeordneten Ordners), setzt man das Häkchen bei *Aus Navigation ausblenden* auf der Seite *Eigenschaften* des Objekts.

3.2.6 Literaturlisten

| Anforderung: Es können Literaturlisten angezeigt werden. |

Auf vielen Seiten sind Literaturlisten notwendig (z.B. Veröffentlichungen eines Wissenschaftlers), es sollte daher eine Möglichkeit geben, solche Listen erzeugen zu lassen. Idealerweise gibt es eine zentrale Datenbank, die alle Literaturhinweise beinhaltet und die für die Erzeugung der Listen benutzt werden kann.

Literaturlisten können im System mit Hilfe normaler Seiten (*UniPage*) und der *automatischen Textersetzung* (siehe Abschnitt 3.2.16), aus externen Quellen kommend, realisiert werden.

3.2.7 Eventkalender

| Anforderung: Benutzer können sich einen Überblick über Veranstaltungstermine verschaffen. |

Wünschenswert ist ein zentraler Eventkalender, damit Studierende und andere Interessierte (Mitarbeiter, Bewohner der Region, Presse) eine zentrale Internetseite haben, auf der sie sich über Veranstaltungstermine der Universität informieren können.

Der Eventkalender ist leicht zu pflegen. Schreib- und Löschrechte werden getrennt verwaltet, damit alle Redakteure Veranstaltungen eintragen können, ohne Einfluss auf andere Veranstaltungen zu haben.

Termine werden als Objekte des Contenttyps *Event* angelegt. Sie sind zu unterscheiden von *Ankündigungen* (*News*, siehe *Ankündigungen*, Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.**).

Aktuelle Termine werden in Listenform angezeigt. Es kann getrennte Listen für zukünftige und vergangene Termine geben.

Der explizite Austrag aus dem Eventkalender kann mittels Zeitsteuerung erfolgen. Bei der Eingabe des Events kann der Benutzer festlegen, zu welchem Datum der Event nicht mehr angezeigt werden soll (siehe *Zeitsteuerung: Veröffentlichungszeitraum*, Abschnitt 3.4.3).

Der Eventkalender kann als RSS-Feed abonniert werden (siehe *XML-Export*, Abschnitt 3.6.3).

3.2.8 Dateien

Anforderung: Es gibt einen Contenttypen zur Speicherung beliebiger Dateien (z.B. PDF, Word, Excel, Audio, Video).

Nicht nur für die Bereitstellung von Materialien zum Download (siehe *Download-Bereich/Materialiensammlung*, Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.**) wird ein Contenttyp gebraucht, mit dem man Dateien beliebigen Inhalts speichern kann. Gemeint sind herkömmliche Dateien verschiedenen Typs, z.B. PDF, DOC, Audio oder Video. Das System erkennt die Dateitypen i.d.R. und erlaubt teilweise den Zugriff auf Metadaten. Es gibt dazu einen generischen Contenttyp (*File*), mit dem eine Datei unbekanntem Typs gespeichert werden kann.

Da die Dateiinhalte aufgrund ihrer Größe die Datenbank nicht belasten sollen, werden sie im Dateisystem gespeichert. Für den Benutzer ist der Speicherort jedoch nicht relevant.

3.2.9 Lehrveranstaltung

Anforderung: Es können Zusatzinformationen zu Lehrveranstaltungen verwaltet werden.

Lehrveranstaltungen sind beispielsweise Vorlesungen, Übungen, Seminare oder Projektpraktika. Die Verwaltung von Lehrveranstaltungen erfolgt in der Universität Koblenz · Landau über KLIPS (<https://klips.uni-koblenz-landau.de>).

Lehrveranstaltungen können über Schnittstellen zu KLIPS im UniCMS angezeigt und verwaltet werden. Zu jeder Lehrveranstaltung, die im UniCMS über eine Schnittstelle angezeigt wird, können weitere Informationen in Form von beschreibenden Textelementen oder Dateien hinzugefügt werden.

Auf Lehrveranstaltungen, die im Lernmanagementsystem der Universität Koblenz · Landau verwaltet werden, kann im Beschreibungstext ein Link auf die entsprechende Kursinformation oder auf die Kursanmeldung gespeichert werden.

3.2.10 Meldungen

Anforderung: Es gibt einen Contenttypen *News (Meldungen)* zur Speicherung von Ankündigungen und (Presse-)Mitteilungen.

Eine Ankündigung ist ein (kurzer) Hinweistext für eine bestimmte Zielgruppe. Die Aktualität der Ankündigung ist dabei von kurzer zeitlicher Dauer.

Der Benutzer kann festlegen, in welchem Zeitraum eine *Meldung* als 'aktuell' gilt (Datum von/bis). Nach diesem Kriterium kann man dann die Anzeige steuern, um beispielsweise eine Liste aktueller *Meldungen* erzeugen zu lassen.

Eine *Meldung* wird standardmäßig aus der Navigation ausgeschlossen.

3.2.11 Termine

Anforderung: Es gibt einen Contenttypen *Event (Termin)* zur Ankündigung einer kommenden Veranstaltung.

Die meisten Termine (Contenttyp *Event*) sollen unter einer Rubrik 'Aktuelles' erscheinen.

Die Sperrfrist bzw. den Anzeigezeitraum der erzeugten Ankündigung könnte zudem automatisch gesetzt werden, abhängig von den Keywords: Bei der Eingabe eines Events soll der Eintrag bei 'Sperrfrist' automatisch einen Vorgabewert enthalten, der von vorher eingegebenen Keywords abhängt. Welches Keyword welche Sperrfrist bedingt, sollte ein Administrator einstellen können. Beispiele:

- Zeige Objekte vom Typ 'Semesterpolitikum' zwei Wochen vor dem Termin
- Zeige Objekte vom Typ 'Vortrag' eine Woche vor dem Termin

Termine werden nicht automatisch gelöscht sondern bleiben im System archiviert.

3.2.12 Newsletter

Anforderung: Es gibt E-Mail-Verteiler für Ankündigungen (Rundschreiben, Newsletter).

Alle Besucher der Internetseiten können schriftlich erreicht werden, um wichtige Ankündigungen zu kommunizieren. Ein Newsletter wird per E-Mail verschickt (Push-Dienst). Hierfür ist eine Einverständniserklärung der Besucher erforderlich. Dies geschieht durch ein Abonnement. Die Besucher können sich selbst ein- bzw. austragen.

Der Inhalt der Newsletter wird als Internetdokument erfasst und erscheint auch im Internet. Ein Newsletter besteht aus einem (einzigen) HTML-Dokument.

Das UniCMS benutzt eine Erweiterung, um den Newsletter zu implementieren.

3.2.13 Sprechstundenzeiten

Anforderung: Dozenten können Angaben zu ihrer Sprechstunde hinterlegen.

Der Contenttyp *Person* bietet ein Feld zur Speicherung der Sprechstundenzeiten. Zusätzlich zur Speicherung der Daten in den Personenangaben, bietet das UniCMS die Möglichkeit eine Anmeldung zu Sprechstunden zu realisieren. Hier kann für einen bestimmten Zeitpunkt eine Person mit Uni-Mail-Adresse sich zu einer Sprechstunde anmelden.

Das UniCMS benutzt eine Erweiterung, um die Sprechstundenverwaltung zu implementieren

3.2.14 Funktion einer Person

Anforderung: Es gibt einen Contenttypen *PersonFunction* zur Speicherung von Funktionen von Personen.

Der Contenttyp *PersonFunction* bildet Funktionen ab, die eine Person innerhalb der Universität innehaben kann. Beispiele für solche Funktionen sind *Dekanin des Fachbereichs 3* oder *Professor am Institut für Werkstoffkunde*. Für jede Funktion wird ein eigenes Objekt vom Typ *PersonFunction* angelegt, also beispielsweise acht Objekte für die Dekane der acht Fachbereiche.

Die Bezeichnung der Funktion einer Person ist dabei vom Benutzer frei definierbar.

3.2.15 Metadaten

Anforderung: Das System bietet die Möglichkeit, Meta-Daten zu definieren.

Meta-Daten im Sinne dieser Anforderung sind Zusatzangaben zu einer Internetseite. Meta-Daten werden im nicht-sichtbaren Bereich einer Internetseite hinterlegt und haben beschreibende Funktion. So können darin so genannte Keywords (Schlag- oder Schlüsselwörter), eine Inhaltsbeschreibung, Autorenschaft u.ä. festgehalten werden. Da diese Metadaten von Suchmaschinen ausgewertet werden und einen nicht unerheblichen Einfluss auf das Ranking, also die Anzeige innerhalb eines Suchergebnisses, haben, bietet das System dem Redakteur die Möglichkeit, Meta-Angaben zu machen.

Die Angaben erfolgen auf der Seite *Eigenschaften* eines Objekts und umfassen (dem Dublin Core-Standard folgend): die Beschreibung (*DC.description*), die Schlüsselwörter (*DC.subject* und *keywords*), das Erstellungsdatum (*DC.date.created*), das Änderungsdatum (*DC.date.modified*), den Typ (*DC.type*, z.B. *Seite*), die Sprache (*DC.language*) und ggf. Angaben zum Copyright (*DC.rights*).

3.2.16 Automatische Textersetzung

Anforderung: Dynamisch ermittelte Informationen (nicht-statische Inhalte) können einfach in Textseiten integriert werden.

Texte auf Internetseiten enthalten oft Informationsteile, die zum Zeitpunkt ihrer Anzeige 'berechnet' werden sollen, da sie aus Datenquellen stammen. Statt z.B. die Telefonnummer eines Mitarbeiters als Text in eine Seite zu schreiben, sollte sie vielmehr als 'Befehl' der Form 'setze an dieser Stelle die Telefonnummer des Mitarbeiters X ein' erfasst werden. Dadurch

bleibt die Telefonnummer automatisch aktuell. Die Funktionalität ist vergleichbar mit 'Feldfunktionen' in Microsoft Word.

Das Eintragen solcher berechneter Informationen darf nicht kompliziert sein (sonst würde gegen *Unkompliziertheit*, Abschnitt 2.1.2, verstoßen), d.h. es muss eine komfortable Möglichkeit zu deren Eingabe geben.

In der Praxis sind potentielle Anwendungen recht häufig. Eine kurze Recherche ergab folgende Beispiele:

- Telefonnummer (Durchwahl) von Angestellten (Beispiel auf Dekanat FB1, <http://www.uni-koblenz.de/fb1/dekanat.html>).
- E-Mail-Adresse oder Link zur Homepage von Personen (Beispiel auf AGPaulus, <http://www.uni-koblenz.de/FB4/Institutes/ICV/AGPaulus>)
- Name des aktuellen Dekans eines Fachbereichs (Beispiel auf Dekanat FB1, <http://www.uni-koblenz.de/fb1/dekanat.html>)

Die geforderte Funktionalität wird durch eine Erweiterung umgesetzt.

3.2.17 Erweiterung für Portalseiten

Anforderung: Es gibt spezielle Ansichten für Portalseiten.

Ein Benutzer des UniCMS kann selbst interaktiv eine Seite aus Einzelteilen zusammenstellen. Die Seiten bestehen aus so genannten *Panels*. In jedem Panel kann entweder ein Text (siehe Abschnitt 2.1.3) oder ein so genanntes *Viewlet* erscheinen. Ein Viewlet ist eine objektspezifische Darstellungsform.

Es gibt verschiedene vorgefertigte Viewlets. Eines der Viewlets kann den Inhalt eines Ordners auflisten. Das Viewlet wird so erweitert, dass man zusätzlich angeben kann, dass nur bestimmte Contenttypen dabei berücksichtigt werden sollen.

Z.B. definiert das Viewlet `viewlet_person_card.pt` das Layout einer 'Visitenkarte' für eine Person.

3.2.18 Vorlagen

Anforderung: Das System stellt Vorlagen mit häufig benötigten Inhaltsfragmenten bereit.

Bestimmte inhaltliche Ergänzungen fallen routinemäßig an. Hier macht es Sinn, vorbereitete Vorlagen bzw. Schablonen zur Verfügung zu stellen, die diese Inhalte bereits enthalten und nur noch angepasst werden brauchen. So kann beispielsweise die Schnittstelle zu einer Lehrveranstaltung einfach aus einem Katalog ausgewählt werden und der Benutzer muss lediglich die spezifischen Daten anpassen.

Es gibt im UniCMS einen Ordner, der die kopierbaren Objekte enthält ('Vorlagen-Sammlung'). Ein Benutzer kann die Objekte dann durchsuchen und per Knopfdruck an die gewünschte Stelle einfügen.

Man hat hierdurch gleichzeitig eine Sammlung aller Vorlagen an einer zentralen Stelle. Man sieht als Anwender also auch leicht, welche Möglichkeiten es überhaupt gibt.

Ein Link zu der Vorlagen-Sammlung ist in die Benutzungsoberfläche integriert. Es gibt dazu den Eintrag '... von Vorlage' in der Liste mit den normalerweise hinzufügbaren Objekten (*Artikel hinzufügen*).

3.3 Berechtigungen

3.3.1 Rollen

Anforderung: Das System unterstützt ein Rollenkonzept.

Das System hat ein Rollenkonzept, wonach Benutzer mit einer oder mehreren Rollen ausgestattet werden, denen wiederum Berechtigungen zugeordnet werden. Die Rollen und die mit ihnen verknüpften Berechtigungen sind im *Rollenkonzept* beschrieben.

Ein Benutzer darf keine seiner Rechte an andere Benutzer vergeben. Dies ist nur Administratoren erlaubt.

3.3.2 Zugriffsschutz

Anforderung: Der Zugriff auf Inhalte und Funktionen kann über Berechtigungen gesteuert werden.

Das UniCMS erlaubt es, bestimmte Inhalte bzw. Inhaltsbereiche vor unberechtigtem Zugriff zu schützen. Unter Zugriff ist hier lesender und schreibender Zugriff zu verstehen. Der Zugriff kann für jedes Inhaltsobjekt einzeln und auch für ganze Bereiche eingestellt werden.

Bereiche werden durch Ordner definiert (nach *Ordner*, Abschnitt 3.2.4). Für Ordner können lokale Rollen definiert werden. Die Vergabe von Berechtigungen für ein Objekt geschieht mit Hilfe von vorgegebenen Workflows. Der aktuelle Zustand innerhalb eines Workflows definiert die Berechtigungen. Es gilt der Workflow aus dem Rollenkonzept.

Die genaue Festlegung der Berechtigungsstruktur und des Workflows ist im *Rollenkonzept* beschrieben.

3.3.3 LDAP-Anbindung

Anforderung: Die Authentifizierung erfolgt über das LDAP-Verzeichnis des Rechenzentrums in Koblenz.

Die Benutzer des Systems können sich mit der bekannten Benutzerkennung des Rechenzentrums (und dem zugehörigen Passwort) anmelden. Das UniCMS greift dazu direkt auf den LDAP-Dienst zu.

Zum Anmelden gibt es einen eigenen Link (*Login*), der auf jeder Seite sichtbar ist.

Es muss sichergestellt sein, dass jeder Benutzer auch eine Kennung besitzt. Dies ist aber keine Aufgabe des UniCMS.

3.3.4 Single-Sign-On

Anforderung: Die Authentifizierung kann auch im Abgleich mit der Anmeldung an anderen Systemen erfolgen.

Die Benutzer können sich an anderen uni-internen Systemen, z.B. KLIPS anmelden und müssen sich bei einem Wechsel ins UniCMS nicht erneut authentifizieren. Dies geschieht über einen Single-Sign-On-Dienst innerhalb der Universität Koblenz · Landau.

3.4 Workflow, Versionierung, Freigabe, Mehrsprachigkeit

3.4.1 Workflow

Anforderung: Inhalte können einem Workflow unterworfen werden.

Es ist erwünscht, die Veröffentlichung von Inhalten in mehreren Stufen ablaufen zu lassen, so dass nach der Erstellung zunächst eine Freigabe durch eine andere Stelle erfolgen muss, bevor ein Inhalt tatsächlich veröffentlicht wird. Die einzelnen Entscheidungsschritte werden von unterschiedlichen Rollen vorgenommen, und der Gesamtprozess wird vom System überwacht.

Die Workflows werden im *Rollenkonzept* definiert.

3.4.2 Mehrsprachigkeit

Anforderung: Inhalte können in mehreren Sprachversionen verwaltet werden.

Alle Inhalte müssen prinzipiell in beliebig vielen Sprachversionen gespeichert werden können (W3C I18N, 'Internationalisation'). Das System bietet eine Möglichkeit, diese Versionen zu erfassen und zu pflegen. Die Pflege kann mindestens durch diejenigen erledigt werden, die auch die Originalinhalte pflegen können.

Bei der Darstellung der Inhalte berücksichtigt das System die Sprachwahl des Besuchers und verwendet die jeweils richtige Version ('Localisation').

Die Übersetzung von Inhalten bezieht sich immer auf ein vollständiges Objekt, nicht auf einzelne Attribute eines Objekts. Man kann also nicht Teile eines Objekts übersetzen, sondern immer nur das Objekt als Ganzes.

Die Mehrsprachigkeit wird durch eine Erweiterung zur Verfügung gestellt.

3.4.3 Zeitsteuerung: Veröffentlichungszeitraum

Anforderung: Inhalte können zeitgesteuert veröffentlicht werden.

Für Inhalte kann angegeben werden, in welchem Zeitraum sie für die Veröffentlichung freigegeben sind. Man kann also festlegen, dass ein bestimmter Inhalt erst zu einem bestimmten Termin sichtbar sein soll und ab einem bestimmten Termin nicht mehr sichtbar sein soll.

Die Einstellungen erfolgen dazu auf der Seite *Datum* eines Objekts in den Feldern *Freigabedatum* und *Ablaufdatum*.

3.4.4 Versionskontrolle

| Anforderung: Die Inhaltserstellung wird unter eine Versionskontrolle gestellt. |

Beim Bearbeiten von Seiten durch Benutzer sichert das System die bisherige Variante und erlaubt ein Rückgängig-Machen (*Undo*) der Änderung. Alte, überschriebene Varianten werden aufbewahrt. Der Benutzer kann auf jede alte Variante zugreifen und diese mit der aktuellen Variante vergleichen.

3.5 Suche, Links

3.5.1 Suchfunktion

| Anforderung: Das System bietet eine Inhalts-basierte Suchfunktion. |

Die Auffindbarkeit von Inhalten ist eine wesentliche Anforderung an eine Internetseite. Neben einer guten Strukturierung ist eine Suchfunktion unumgänglich. Das UniCMS stellt daher eine Suchfunktion bereit. Diese Suche berücksichtigt alle Inhalte der gesamten Internetseite.

Die Suchfunktion wird über eine externe Erweiterung realisiert.

3.5.2 Link-Checker

| Anforderung: Das System hat einen integrierten Link-Checker. |

Links sind das wichtigste Element des Internet, ihr Funktionieren gilt als selbstverständlich, ein Nicht-Funktionieren ist immer ein immenses Ärgernis und wirft ein schlechtes Licht auf den Seitenbetreiber.

Viele Links erstellt das UniCMS automatisch, z.B. zwischen einer *Funktion einer Person* und der *Person*. Diese Links sind daher stets korrekt. Von einem Benutzer erfasste Links, die auf Objekte innerhalb des UniCMS verweisen, werden durch die Verwendung von Referenzen gespeichert, was zur Folge hat, dass sie auch nach einem Verschieben des Zielobjektes noch funktionieren.

Von Hand eingegebene Links in Form von URLs überprüft das System *nicht* automatisch. Hier wird bei Bedarf eine Link-Check-Software installiert, mit der die Internetseite regelmäßig auf korrekte Links überprüft wird.

3.5.3 Lesbare URL

| Anforderung: Die verwendeten URLs sind für Menschen lesbar. |

URLs, also die Internetseitenadressen, können kryptisch aufgebaut sein (Bsp: <http://www.spiegel.de/politik/deutschland/0,1518,385032,00.html>) oder für Menschen lesbar und damit auch suchmaschinenoptimiert (<http://www.uni-kl.de/wcms/intclub.html>). Das UniCMS erlaubt den Benutzern, die URLs selbst einzugeben. Es werden also keine unlesbaren Varianten automatisch erzeugt. Gibt ein Benutzer für eine Seite keinen URL ein, so wird ein solcher aus dem Titel der Seite generiert.

3.5.4 Bezeichnung der URL

| Anforderung: Die Bezeichner in den URL sind in der jeweiligen Sprache verfasst. |

Die Bezeichner in den URLs entsprechen den IDs ('Kurznamen') der einzelnen Objekte (also Ordner und andere Inhalte). Die IDs können daher vom Benutzer selbst gewählt werden (siehe auch *Lesbare URL*, Abschnitt 3.5.3). Die Benutzer werden angehalten, Bezeichner in der jeweiligen Sprachvariante zu wählen. Es ist darauf zu achten, dass keine Umlaute, Leerzeichen oder andere Sonderzeichen in den Kurznamen benutzt werden.

3.5.5 Sitemap

| Anforderung: Das System generiert automatisch eine Sitemap. |

Eine Sitemap zeigt graphisch, als Baumstruktur, die Gesamtheit aller Seiten einer Internetseite an und bietet so eine nicht-textorientierte Suchunterstützung. Diese macht nur Sinn, wenn sie permanent aktuell ist. Dazu wird sie beim Aufruf dynamisch vom UniCMS erzeugt.

Das UniCMS nutzt die Standard-Sitemap von genutzten Redaktionssystem. Dazu gibt es auf jeder Seite einen Link *Sitemap*.

3.6 Datenaustausch mit anderen Systemen

3.6.1 Externe Systeme

| Anforderung: Externe Systeme können angebunden werden. |

Das System erlaubt den Zugriff auf externe Systeme. Dies umfasst eine geeignete Schnittstelle zur Anbindung eines Systems, z.B. mit Hilfe von SOAP, Rest oder JSON.

Zweck dieser Anforderung ist die dadurch gewonnene Möglichkeit, in externen Systemen vorhandene Daten auf den Internetseiten anzeigen zu können bzw. sie ändern zu können. Eine wesentliche Aufgabe des UniCMS ist die Integration von Datenbeständen und dem Verfügbarmachen derselben, und dies erfordert eine entsprechende Offenheit des Systems.

Ein Beispiel wäre der Zugriff auf die Liste der Telefonnummern, so dass zu einer Person automatisch die jeweils aktuelle Telefonnummer angezeigt werden kann. Man erspart sich bei einer Änderung der Telefonnummer somit die manuelle Suche nach allen Stellen, wo noch auf die alte Nummer verwiesen wird. Das UniCMS könnte die Daten einer Person automatisch aus einem Verzeichnis übernehmen (siehe z.B. *LDAP-Anbindung*, Abschnitt 3.3.3).

3.6.2 Trennung von Inhalt und Layout

| Anforderung: Inhalt und Layout sind voneinander getrennt. |

Die Inhalte sind in einer Form gespeichert, die ihre Wiederverwendung für verschiedene Ausgabemedien ermöglicht (siehe z.B. *Druckansicht*, Abschnitt 3.1.1). Dazu müssen die Inhalte von Layout- oder Design-spezifischen Angaben so weit wie möglich getrennt werden. Durch die Contenttypen ist eine Trennung der reinen Inhalte vom Layout ebenfalls gewährleistet.

3.6.3 XML-Export

Anforderung: Das System erzeugt, neben XHTML, Ausgaben beliebiger Formate, insbesondere XML.

Die primäre Ausgabesprache für die Internetseiten ist HTML (siehe *Browserkompatibilität*, Abschnitt 2.2.2). Daneben sind durch entsprechende Anpassung des Systems die Ausgabe in beliebigen weiteren Ausgabesprachen möglich. RSS Feeds für entsprechende Lesesoftware erstellt das UniCMS für jeden *intelligenten Ordner* (gespeicherte Suche) automatisch. Im Besonderen kann das System XML-Dokumente erzeugen, so dass daraus beliebige weitere Dokumentenformate generiert werden können, wie z.B. PDF, RTF oder Latex.

3.7 Administration des Systems, Organisatorisches

3.7.1 Mehrere getrennte Internetseiten

Anforderung: Es können mehrere getrennte Internetseiten verwaltet werden.

Das System erlaubt es, Bereiche zu definieren, die nach außen als eigenständige Internetseite erscheinen, die aber intern als Teile des Systems verankert sind und auf nur einen Datenbestand zugreifen.

Die Internetseiten sind bei Bedarf auch unter einer eigenen Domain erreichbar, diese dient allerdings lediglich als Weiterleitung.

3.7.2 Layout-Anpassung

Anforderung: Das Layout kann für Bereiche individualisiert werden.

Die Einhaltung der Designvorgaben wird durch die Trennung von Inhalt und Layout und die Verwendung eines Vorlagen-Mechanismus realisiert. Die Benutzer sollten sich um Fragen des Layouts möglichst nicht kümmern müssen. Anders ausgedrückt: das System bietet im normalen Betrieb erst gar keine Möglichkeit vom Layout abzuweichen.

Das Aussehen der Internetseiten des UniCMS kann für Bereiche jedoch in einem kleinen Rahmen angepasst werden.

3.7.3 Eigenentwicklungen

| Anforderung: Das System ist durch Eigenentwicklungen erweiterbar. |

Der Quelltext des Systems selbst ist zugänglich und die Lizenzbestimmungen erlauben es, diesen zu erweitern oder zu verändern.

3.7.4 Zugriffsstatistik

| Anforderung: Die Zugriffsstatistiken können abgefragt werden. |

Das System führt über eine Erweiterung ein Statistik über Zugriffe auf die einzelnen Internetseiten und erlaubt Berechtigten das Abrufen eben dieser.

Allein die Administratoren besitzen die Berechtigung zum Zugriff auf die Statistiken. Die Berechtigung zum Zugriff auf bereichsspezifische Teile der Statistiken kann Personen gewährt werden, wenn diese für den betroffenen Bereich zuständig sind. Hierbei ist zu beachten, dass die Belange des Datenschutzes jederzeit gewahrt bleiben.

3.7.5 Zuständige Personen

| Anforderung: Für jeden Bereich der Internetseite gibt es eine (inhaltlich) verantwortliche Person. |

Für jeden Bereich der Internetseite kann eine zuständige Person benannt werden, die als Ansprechpartner für Fragen von Benutzern zur Verfügung steht. An diese Person gehen die Kommentare, die über das Kontaktformular gestellt werden (siehe *Kontaktformular*, Abschnitt 3.1.3). Gleichzeitig ist sie Ansprechpartner für das UniCMS-Team.

3.7.6 Umleitungen

| Anforderung: Während einer Umstellung anderer Internetauftritte ins UniCMS müssen Umleitungen definiert werden. |

Die Umstellung der bestehenden Seiten auf das UniCMS muss auch graduell erfolgen können. Hierzu sind Umleitungen von alten URLs nötig. Teilweise müssen aber auch Fehlerseiten (404, 30x) erstellt werden, die auf neue Seiten aufmerksam machen oder dorthin umleiten. Diese Fehlerseiten müssen ggf. ordnerweise erstellt werden.

Anlage 2: Systemmodule

1 Zweck des Dokuments

Dieses Dokument listet die Systemmodule auf, aus denen das UniCMS aufgebaut ist. Es ist eine Anlage zur Dienstvereinbarung.

2 Software

Das UniCMS besteht im Wesentlichen aus einem Webserver (Apache) und einem Applikations- und Datenbankserver (Zope). Hinzu kommt noch ein Load-Balancer, der die Lastverteilung übernimmt und ein Caching-Server, der der Performancesteigerung dient. Auf dem Zope-System läuft Plone, das eigentliche Content-Management-System.

Der Suchserver basiert auf Apache Solr.

Die einzelnen Software-Bestandteile werden nachfolgend kurz genannt und mit Stichworten klassifiziert.

- Linux-Betriebssystem
- Webserver
- Load-Balancer
- Caching-Server
- Applikations- und Datenbankserver
 - Zope, <http://www.zope.org>. Applikations- und Datenbankserver
 - Python, <http://python.org>. Programmiersprache und Laufzeitumgebung
 - python-ldap. Bibliothek für den LDAP-Zugriff
 - PIL. Bildverarbeitungsbibliothek
- CMS-Software Plone
- Apache Solr
 - Solr, <http://lucene.apache.org/solr/>

Anlage 3: Rollen- und Berechtigungskonzept des UniCMS

1. Zweck des Dokuments

Diese Anlage legt fest, welche Rollen und Berechtigungen es innerhalb des zentralen Internetauftritts der Universität Koblenz · Landau (UniCMS) gibt. Er ist Bestandteil der Dienstvereinbarung über Einführung, Anwendung, Erweiterung und Änderungen einer Web-Content-Management-Software.

Rollen und Berechtigungen sind wichtig, um zu definieren, welche Personen welche Zugriffsmöglichkeiten auf welche Daten innerhalb des Systems haben. Die Einstellmöglichkeiten von Plone, der dem UniCMS zugrunde liegenden Software, sind hier sehr vielfältig. Es muss also darauf geachtet werden, dass das Konzept nicht zu kompliziert und dadurch unhandlich wird.

Der Text ist zweigeteilt. Zunächst werden in Abschnitt 3 die technischen Hintergründe dargestellt und es wird gezeigt, welche Einstellmöglichkeiten Plone bietet. Im zweiten Teil wird dann die konkrete Umsetzung des Konzepts für die Universität Koblenz · Landau beschrieben. Dazu beschreibt Abschnitt 4 insbesondere den benutzten Workflow, da dieser die Berechtigungen steuert.

Der folgende Abschnitt 2 gibt zunächst eine kurze Einleitung und dient als Zusammenfassung des Textes für eilige Leser.

2. Kurzfassung

Berechtigungen zur Ausführung bestimmter Aktionen mit dem UniCMS werden durch Rollen vergeben, die Benutzer einnehmen können. Beispiele für Rollen sind *Schlussredakteur* und *Administrator*. An eine Rolle sind Berechtigungen geknüpft. So hat ein Administrator beispielsweise andere – i.d.R. weiter reichende – Berechtigungen als ein Redakteur. Durch die Berechtigungen wird festgelegt, welche Aktionen eine Rolle ausführen darf, um die ihr zugeordnete Arbeit erledigen zu können. Berechtigungen werden nicht direkt an einzelne Benutzer, sondern immer an Rollen vergeben. Benutzer erhalten eine Berechtigung erst dadurch, dass sie eine Rolle innehaben, die die Berechtigung zugewiesen bekommen hat.

Ein Benutzer kann beliebig viele Rollen gleichzeitig innehaben. Die Berechtigungen des Benutzers ergeben sich dann aus der Summe aller Einzelberechtigungen der Rollen.

Die Rollen eines Benutzers können zudem an unterschiedlichen Stellen des UniCMS unterschiedlich sein. Die Rollen können also davon abhängen, für welchen *Bereich* ein Benutzer zuständig sein soll. Durch die Unterteilung in einzelne Bereiche kann man einem Benutzer eine Rolle ausdrücklich für einen bestimmten Bereich zuweisen. In anderen Bereichen hat er diese Rolle dann nicht, es sei denn, man definiert dies explizit.

Eine hilfreiche Metapher für den Begriff einer Rolle ist der Hut, den man aufsetzen kann. An unterschiedliche Hüte sind unterschiedliche Aufgaben und Berechtigungen geknüpft (man denke an die Kopfbedeckungen eines Kochs, einer Krankenschwester oder eines Polizisten). Im übertragenen Sinne bekommen Personen durch das Tragen eines Hutes eine Aufgabe, sie sind einmal Koch, einmal Krankenschwester und einmal Polizist. Hüte sind beliebig

austauschbar. Im UniCMS können Personen auch mehrere Hüte gleichzeitig tragen. Dies mag in der Vorstellung im Einzelfall komisch aussehen, aber es ist ja nur eine Metapher...

Das Berechtigungskonzept muss nun festlegen, welche Benutzer welche Rollen haben sollen (also: wer darf überhaupt welchen Hut tragen), aber auch, welche Rollen es überhaupt geben soll und welche Berechtigungen sie jeweils bekommen sollen (also: welche Hüte gibt es, und was stellen sie jeweils dar). Um diese Festlegung treffen zu können, ist ein Grundverständnis der technischen Hintergründe hilfreich, das im Folgenden vermittelt werden soll.

3. Technischer Hintergrund

Die Umsetzung des UniCMS erfolgt mit Plone. Daher wird nun der im Zusammenhang mit Rollen und Berechtigungen relevante technische Hintergrund dargestellt.

3.1. Begriffserklärung

Zunächst werden einige wichtige Begriffe definiert.

Benutzer: Ein Benutzer ist eine Person, die mit dem UniCMS arbeitet und dieses benutzt. Innerhalb des Systems wird ein angemeldeter Benutzer mit seiner Benutzerkennung identifiziert.

Rolle: Eine Rolle (Benutzerrolle, in Plone auch *Vollmacht* genannt) fasst im Allgemeinen Aufgaben, Eigenschaften und Berechtigungen eines Benutzers zusammen. Eine Rolle ist ein abstraktes Konzept, um von einzelnen, konkreten Benutzern und ihren Aufgaben abstrahieren zu können.

Ein Benutzer kann beliebig viele Rollen gleichzeitig inne haben. Die Rollen eines Benutzers können auch von Bereichen der Internetseite (siehe unten) abhängen, d.h. ein Benutzer kann in einem bestimmten Bereich eine bestimmte Rolle haben, in einem anderen Bereich hat er dieselbe Rolle jedoch nicht. Durch diese Möglichkeit erfolgt die eigentliche Vergabe von Berechtigungen: Man weist einem Benutzer in einem Bereich eine bestimmte Rolle zu, und an die Rolle wiederum sind die Berechtigungen geknüpft.

Eine Rolle umfasst eine Idee von den Aufgaben, die ein Benutzer erledigen muss, und den dazu benötigten Berechtigungen. Ein Beispiel ist die Rolle *Administrator*: Ein Benutzer, der diese Rolle hat, muss bestimmte Systemeinstellungen ändern dürfen, um seine Aufgaben als Administrator erfüllen zu können. Daher erhält die Rolle *Administrator* genau diese Berechtigungen. In der Beschreibung der Rolle muss also genau festgelegt sein, was ein Administrator überhaupt ist und welche Aufgaben er hat. Auf dieser Grundlage werden dann die Berechtigungen vergeben, die zu ihrer Erfüllung notwendig sind.

Rollen im UniCMS sind in der Regel nicht deckungsgleich mit Rollen, die im realen organisatorischen Umfeld der Universität existieren. So gibt es beispielsweise keine Rolle *Dekan*, aber z.B. *Schlussredakteur*. Die Rollen im UniCMS sind auf fachliche Aspekte fokussiert und beschreiben, welche Berechtigungen ein Benutzer innerhalb des UniCMS hat, und nicht, welche Stelle die Person innerhalb der Universität bekleidet. Zu den Rollen siehe auch Abschnitt 4.

Berechtigung: Eine Berechtigung ist die Erlaubnis zur Ausführung einer Aktion mit einem Objekt.

Aktion: Eine Aktion (Tätigkeit) ist ein definiertes Vorgehen mit einem Objekt innerhalb des UniCMS. Beispiele für Aktionen sind "Löschen", "Lesen" oder "Verändern". Die tatsächlich mit einem Objekt durchführbaren Aktionen hängen vom Typ dieses Objekts ab.

Objekt: Ein Objekt ist jedes einzelne, im UniCMS gespeicherte Datenelement. Die Objekte haben verschiedene Typen, wie z.B. Ordner, Internetseite oder Bild. Die Typen sind der wesentliche Inhalt des Datenmodells, das in einem gesonderten Dokument beschrieben wird.

Bereich: Das UniCMS gliedert die Internetseite in Bereiche, die durch ineinander geschachtelte Ordner definiert werden. Ein Bereich ist in diesem Zusammenhang daher der Inhalt eines Ordners, inklusive aller seiner Unterordner. Beispiele für Bereiche sind "Internetseite des Instituts für Sozialwissenschaft" und "Internetseite des Fachbereichs Erziehungswissenschaften".

3.2. Berechtigungskonzept in Plone

Plone unterstützt ein mächtiges Berechtigungskonzept, das auf Rollen, Berechtigungen, Objekten und Aktionen basiert. Das System bietet wesentlich mehr Möglichkeiten, als zur Umsetzung der Anforderungen der Universität Koblenz · Landau erforderlich sind. Daher können wir uns auf eine Teilmenge beschränken, was die Übersichtlichkeit erhöht und die Pflege vereinfacht.

Eine Berechtigungseinstellung in Plone ist die Zuordnung

- einer Rolle
- zur Ausführung einer Aktion
- mit einem Objekt.

	Anonymous	Authenticated	Manager	Member	Owner
Use mailhost services	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View Directory Index	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View Groups	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View History	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View management screens	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WebDAV Lock items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WebDAV Unlock items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WebDAV access	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zwiki: Add comments	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 1: Berechtigungsmatrix

Man kann für ein Objekt einstellen, dass eine bestimmte Rolle (also jeder Benutzer, der im gegebenen Bereich die gegebene Rolle innehat) eine bestimmte Aktion mit diesem Objekt ausführen darf oder nicht. Um diese Möglichkeiten zu realisieren, greifen verschiedene Konzepte ineinander: eine *Berechtigungsmatrix* pro Objekt, die so genannte *Akquisition* und so genannte *lokale Rollen*. Wichtig festzuhalten ist zunächst, dass Berechtigungen nicht direkt an einzelne Benutzer, sondern an Rollen vergeben werden. Die Zuordnung zu

Benutzern erfolgt dann implizit, nämlich dadurch, dass ein Benutzer eine Rolle innehaben kann.

3.2.1. Berechtigungsmatrix

Zunächst hat jedes einzelne Objekt eine Zuordnung von Aktionen zu Rollen. Sie wird in Form einer Matrix dargestellt und legt fest, ob Rolle R mit diesem Objekt die Aktion A ausführen darf oder nicht. Abbildung 1 zeigt einen Ausschnitt aus einer solchen Matrix, hier für einen Ordner. Links stehen einzelne Aktionen, rechts davon sind fünf Rollen in jeweils einer Spalte aufgeführt. Die Ankreuzfelder geben an, ob die Aktion in der betreffenden Zeile von der Rolle in der betreffenden Spalte ausgeführt werden darf. Das Bild zeigt, dass die Rolle *Anonymous* und die Rolle *Manager* die Aktion *View* mit dem Objekt ausführen darf; darüber hinaus darf niemand irgendetwas tun.

3.2.2. Akquisition (,Vererbung')

Durch die so genannte *Akquisition* können Berechtigungseinstellungen innerhalb der Bereiche 'vererbt' werden. Die Akquisition selbst ist ein umfassenderes Konzept, aber im Rahmen der Einstellungen von Berechtigungen genügt eine vereinfachte Sichtweise. Die Idee ist wie folgt: Man vergibt Berechtigungen nicht für jedes Objekt einzeln, sondern macht sich die hierarchische Ordnerstruktur insofern zu Nutze, als dass eine Einstellung automatisch auch in allen untergeordneten Bereichen (Ordnern) gilt, sofern dort nichts anderes definiert wird.

Um beispielsweise einen gesamten Bereich vor unbefugten Schreibzugriffen zu schützen, genügt eine einzige Einstellung bei dem den Bereich definierenden Ordner. Alle Unterordner 'erben' dann diese Einstellung. Ein untergeordneter Ordner darf die Akquisitionskette jedoch auch unterbrechen, indem dies dort explizit eingestellt wird und abweichende Einstellungen für die Berechtigungen vorgenommen werden.

Einstellungen für den in der Hierarchie ganz oben stehenden Ordner gelten somit als Standard für die gesamte Internetseite. Sinnvollerweise wählt man die Einstellungen so, dass Abweichungen in untergeordneten Bereichen nur selten notwendig sind.

Insgesamt erlaubt die Akquisition also ein komfortables Einstellen der Berechtigungen für ganze Bereiche der Internetseite. Es macht also Sinn, bei der Definition der Bereiche auf diesen Aspekt zu achten, so dass die Vergabe von Berechtigungen nicht unnötig kompliziert wird.

3.2.3. Lokale Rollen

Mit einer lokalen Rolle kann man einem Benutzer innerhalb eines Ordners weitere Rollen explizit zuzuweisen. Diese Rollen erhält der Benutzer zusätzlich zu denen, die er ohnehin in diesem Ordner hat. Man kann also beispielsweise festlegen, dass Benutzer A innerhalb einer Arbeitsgruppen-Internetseite auch die Rolle *Administrator* hat, obwohl er diese Rolle im übrigen System nicht hat.

Lokale Rollen gelten immer für den ganzen Bereich, also auch in allen Unterordnern des betreffenden Ordners. Für das UniCMS wird der überwiegende Teil der Einstellungen mit Hilfe von lokalen Rollen vorgenommen werden, da die Benutzer in der Regel nur auf

bestimmte Bereiche zugreifen können sollen. Eine Ausnahme bilden lediglich die Administratoren. Insofern gibt es also keine 'globalen' Rollen, sondern lokale Rollen sind der Normalfall.

3.3. Workflows in Plone

Plone unterstützt die Definition von Workflows, also den elektronischen Entsprechungen von Geschäftsprozessen. Der Standard-Workflow in Plone bildet ein "Vier-Augen-Prinzip" ab, nach dem ein Objekt erst dann freigeschaltet wird, wenn dies explizit erlaubt wurde. Nach der Erstellung eines Objekts muss dieses demnach zuerst zur Veröffentlichung eingereicht werden, und erst nach einer Prüfung durch einen Schlussredakteur erfolgt die eigentliche Freischaltung. Zuvor ist das Objekt nicht öffentlich sichtbar. Der Workflow, der im UniCMS genutzt wird, ist in Abschnitt beschrieben.

Der Zustand, den ein Objekt in einem Workflow einnimmt, hat auch Auswirkungen auf die geltenden Berechtigungseinstellungen für das Objekt. So darf ein Benutzer ein Objekt beispielsweise nicht sehen, wenn das Objekt im Zustand *Intern* ist, wohl aber, wenn es im Zustand *Veröffentlicht* ist. In der Definition eines Workflows kann daher für jeden Zustand angegeben werden, wie die Berechtigungseinstellungen des Objekts in diesem speziellen Zustand aussehen müssen.

3.4. Zusammenfassung: Vergabe von Berechtigungen an Benutzer

Auf Grund der technischen Gegebenheiten bieten sich für die Vergabe von Berechtigungen an Benutzer mehrere Möglichkeiten, die hier noch einmal zusammenfassend erläutert werden sollen. Dieser Abschnitt behandelt das Erlauben und das Verbieten von Aktionen.

3.4.1. Erlauben einer Aktion

Möchte man einem Benutzer B in einem gegebenen Ordner O erlauben, eine Aktion A auszuführen, muss man zunächst eine geeignete Rolle R finden, die Aktion A ausführen darf. Es kann mehrere Rollen geben, für die das zutrifft. Ist die Rolle gefunden, kann man eines der folgenden Dinge tun:

- Man weist dem Benutzer in dem Ordner O die lokale Rolle R zu.
- Man ändert den Workflow-Zustand des Ordners O bzw. des betroffenen Objekts so, dass der Benutzer die Aktion ausführen darf. Mit jedem Workflow-Zustand ist eine Berechtigungsmatrix verknüpft, die angewendet wird, wenn sich der Zustand ändert.

Die individuelle Zuordnung von lokalen Rollen an Benutzer ist der Normalfall.

Nur in absoluten Ausnahmefällen sollte man die Berechtigungsmatrix für eine existierende Rolle in einem Ordner ändern. Dies führt zu abweichenden Interpretationen einer gegebenen Rolle, die ja dann in unterschiedlichen Ordnern unterschiedliche Dinge tun darf. Der Fall, dass es überhaupt keine Rolle gibt, die die gewünschte Aktion ausführen darf, sollte in der Praxis nicht vorkommen, denn alle relevanten Fälle sollten in diesem Konzept bereits erfasst und beschrieben sein. Es wäre dann also dieses Konzept zu überarbeiten.

3.4.2. Verbiehen einer Aktion

Möchte man einem Benutzer B in einem gegebenen Ordner O verbieten, eine Aktion A auszuführen, muss man sicherstellen, dass der Benutzer keine Rolle hat, die Aktion A ausführen darf. Hier gibt es die folgende Möglichkeit: Es gibt eine lokale Rolle, über die der Benutzer die Aktion A ausführen darf. Die lokale Rolle kann für den Benutzer individuell gewährt worden sein. Um die Aktion A zu verbieten, ist die lokale Rolle zu.

Es ist möglich, einem Benutzer explizit eine Rolle in einem Ordner zu entziehen, obwohl er sie für einen übergeordneten Ordner bereits innehat.

4. Rollen und Berechtigungen im CMS

Dieser Abschnitt beschreibt, wie das Rollen- und Berechtigungskonzept für das CMS ausgestaltet werden soll. Grundlage für die Beschreibung sind die im vorigen Abschnitt erläuterten Begriffe und technischen Hintergründe.

Das Konzept wird durch drei Hauptsäulen getragen: Wir betrachten die Aktionen, die ein Benutzer mit dem CMS durchführen soll, die identifizierten Rollen von Benutzern und den *Workflow*, der die Zustände für die einzelnen Objekte und damit insbesondere die jeweils gültigen Berechtigungen definiert.

4.1. Aktionen (Tätigkeiten)

Hier werden kurz die Aktionen beschrieben, die mit einem Objekt durchgeführt werden können. Da die durchführbaren Aktionen auch von der Art des Objektes abhängen, werden hier nur diejenigen Aktionen aufgeführt, die mit allen Objekten durchführbar sind.

Neu anlegen: Ein Objekt, z.B. eine Internetseite, soll neu in einem Ordner erzeugt werden.

Inhalt ändern: Der eigentliche Inhalt eines Objekts, z.B. der Text einer Internetseite, soll verändert werden.

Zusatzangaben (Metadaten) ändern: Die Zusatzangaben (Metadaten) zu einem Objekt, z.B. Schlüsselwörter für die Verschlagwortung, sollen verändert werden. Die Festlegung, welche Metadaten überhaupt vorhanden sein sollen, wird aus der Anlage der Anforderungen getroffen.

Zur Veröffentlichung einreichen: Ein Objekt soll zur Veröffentlichung eingereicht werden, damit es begutachtet werden kann. (Dies ist eine Veränderung des *Workflow*-Zustandes eines Objekts. Weiteres hierzu wird bei der Beschreibung des *Workflows* erklärt.)

Veröffentlichen bzw. zurückweisen (nichtveröffentlichen): Ein zur Veröffentlichung eingereichtes Objekt wurde begutachtet und soll nun zur Veröffentlichung freigeschaltet oder zurückgewiesen werden.

Lesen: Der Inhalt eines Objekts soll gelesen werden.

Löschen: Ein Objekt soll gelöscht werden.

Ausschneiden, kopieren und einfügen: Ein Objekt soll in eine Zwischenablage kopiert oder verschoben (ausgeschnitten) und an anderer Stelle wieder eingefügt werden.

4.2. Rollen

Voreingestellt sind in Plone die folgenden Rollen: "Benutzer, Schlussredakteur und Manager".

Editor (Kann hinzufügen, Kann bearbeiten, Kann ansehen): Inhalte erfassen und ändern, aber nicht freischalten

Ein Editor ist eine Person, die Inhalte erstellen und ändern soll. Die Berechtigungen der Rolle Editor sind also so gestaltet, dass sie 'normale' Inhaltsobjekte wie Seiten, Ordner oder Bilder erzeugen kann. Erstellt ein Editor ein neues Inhaltsobjekt, so ist dies nicht sofort freigeschaltet, sondern hat den Status *intern*.

Schlussredakteur (Kann veröffentlichen): Nach Überprüfung Inhalte zur Veröffentlichung freischalten

Ein Schlussredakteur hat die Aufgabe, Objekte zu überprüfen und sie zu veröffentlichen, d.h., den Objekten den Zustand *veröffentlicht* zuzuweisen. Der Schlussredakteur prüft dazu die Objekte auf inhaltliche und redaktionelle Korrektheit und entscheidet daraufhin, ob sie veröffentlicht werden können oder nicht. Die Rolle *Editor* darf dies nicht. Durch die Trennung der beiden Rollen wird ein 'Vier-Augen-Prinzip' realisiert.

Die Rolle *Schlussredakteur* hat, abgesehen von der Berechtigung zur Freischaltung und zum Zurückweisen von Objekten, die gleichen Berechtigungen wie der *Editor*. Sie kann also beispielsweise auch neue Objekte anlegen oder diese verändern. Dies ist die Hauptaufgabe der meisten Personen, die mit dem UniCMS arbeiten.

Administrator (Manager): Volle Zugriffsmöglichkeiten

Ein Administrator hat unbeschränkten Zugriff auf die Objekte in seinem Bereich. Er kann damit Administrator-tätigkeiten ausführen.

Zusätzlich zu diesen Rollen gibt es noch die folgenden, System-internen Rollen:

Anonym (Anonymous): Die Rolle *Anonym* ist keine Rolle im eigentlichen Sinn. Jeder Zugriff, der von einem nicht am System angemeldeten Besucher erfolgt, gilt als 'anonym'.

Angemeldet (Authenticated): Die Rolle *Angemeldet* ist keine Rolle im eigentlichen Sinn. Jeder Zugriff, der von einem am System angemeldeten Besucher erfolgt, gilt als 'angemeldet'. Die Rolle dient also lediglich zur Unterscheidung von nicht-angemeldeten und angemeldeten Benutzern.

Besitzer (Owner): Der Besitzer eines Objekts ist derjenige Benutzer, der das Objekt angelegt hat. Mit der Rolle *Besitzer* kann man also Berechtigungen davon abhängig machen, wer ein Objekt besitzt.

	Anonym	Angemeldet	Editor	Schlussredakteur	Administrator
Neu anlegen			X	X	X
Inhalt ändern			X	X	X
Metadaten ändern			X	X	X
Einreichen			X	X	X
Veröffentlichen				X	X
Lesen			X	X	X
Löschen			X	X	X
Ausschneiden/Kopieren			X	X	X

Tabelle 1: Zuordnung von Aktionen zu Rollen im Zustand *intern*. Ein X bedeutet: Darf ausführen.

In Tabelle 1 sieht man die Zuordnung der Rollen zu den Aktionen, die sie normalerweise ausführen dürfen. Die Tabelle bezieht sich auf den Workflow-Zustand *intern*. Die Berechtigungen ändern sich, wenn der Workflow-Zustand ein anderer ist. Siehe dazu Abschnitt 4.3.

Die Einstellung, welcher Benutzer welche Aktion ausführen darf, geschieht durch die Zuweisung von Rollen.

4.3. Workflow

Der für das UniCMS benutzte Workflow bildet ein 'Vier-Augen-Prinzip' ab. Er basiert auf dem Intranet/Extranet-Workflow von Plone und lediglich der direkte Übergang von *intern* zu *veröffentlicht* wurde ergänzt. Abbildung 2 zeigt den UniCMS-Workflow. Jede Ellipse steht für einen Zustand, den ein Objekt einnehmen kann. Die Pfeile repräsentieren Zustandsübergänge. Ein Objekt kann immer nur in genau einem Zustand sein. Der Wechsel in einen anderen Zustand kann nur durch die angegebenen Zustandsübergänge erfolgen.

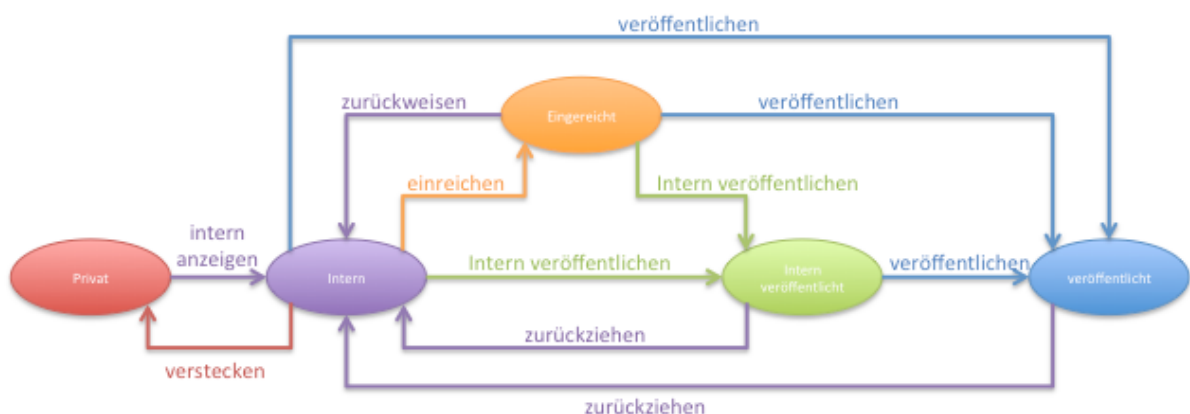


Abbildung 2: Darstellung des Workflow

Der Zustand „intern“ ist dabei der initiale Zustand der Objekte und nur für Personen mit Berechtigung im System verfügbar. Letztes gilt auch für Objekte mit dem Zustand „eingereicht“. Für den Status „intern veröffentlicht“ ist eine Kennung/E-Mail-Adresse der Universität Koblenz-Landau (uni-koblenz-landau.de, uni-koblenz.de, uni-landau.de) erforderlich. „Veröffentlicht“ bedeutet, dass die Seiten weltweit verfügbar sind. „Privat“ kann nur von dem jeweiligen Ersteller des Objektes angesehen werden.

Alle übrigen Berechtigungen bleiben gegenüber der Standardbelegung unverändert.

Kurzbeschreibung des Workflows. Die Zustände *intern*, *eingereicht* und *veröffentlicht* bilden den Kern des ‘Vier-Augen-Prinzips’. Unmittelbar nach dem Anlegen eines Objekts bekommt dieses initial den Zustand *Intern*. In diesem Zustand ist es nur für die Rolle Editor sichtbar und änderbar. Soll das Objekt auch für die Öffentlichkeit sichtbar werden, so muss es veröffentlicht werden. Dies dürfen nur die Rollen *Schlussredakteur* und *Administrator*. Nur diese können das Objekt direkt in den Zustand *Veröffentlicht* setzen (durchgezogener Pfeil *veröffentlichen*). Ist man kein Schlussredakteur, sondern Editor, so muss man das Objekt zur Freischaltung vorlegen. Dazu wählt man den Übergang *einreichen*, und das Objekt geht in den Zustand *eingereicht*. Ein Schlussredakteur findet das Objekt dann automatisch in seiner Revisionsliste und kann das Objekt dann entweder *veröffentlichen* oder *zurückweisen*. Im letzteren Fall geht das Objekt wieder zurück in den Zustand *Intern*, ansonsten in *Veröffentlicht*.

Berechtigungen in den Zuständen In jedem Zustand gelten unterschiedliche Berechtigungseinstellungen für die vorhandenen Rollen. Tabelle 2 zeigt die Zuordnungen. In jedem Zustand werden nur die folgenden Berechtigungen überhaupt betrachtet (in Klammern jeweils die System-interne Bezeichnung):

Lesen (view): Lesen (Betrachten) des Objektinhalts

Auflisten (list): Auflisten des Objekts mit seinem Namen und seiner Beschreibung in einem Ordner

Inhalt ändern (modify): Verändern des Objektinhalts

Zugreifen (access): Zugriff auf das Objekt

Alle übrigen Berechtigungen bleiben gegenüber der Standardbelegung unverändert.

	Anonym	Angemeldet	Editor	Schlussredakteur	Administrator
Zustand intern					
Lesen			X	X	X

Auflisten			X	X	X
Inhalt ändern			X	X	X
Zugreifen			X	X	X
Zustand eingereicht					
Lesen			X	X	X
Auflisten			X	X	X
Inhalt ändern				X	X
Zugreifen			X	X	X
Zustand veröffentlicht					
Lesen	X	X	X	X	X
Auflisten	X	X	X	X	X
Inhalt ändern				X	X
Zugreifen	X	X	X	X	X

Tabelle 2: Zuordnung von Aktionen zu Rollen für alle Workflow-Zustände. Ein 'X' bedeutet: Darf ausführen.

Anlage 4: Systemverwaltung

Technik

Herr Christian Schneider (CMS-Beauftragter)

Herr Matthias Barde

Herr Uwe Arndt (Rechenzentrum)

Herr Christoph Litauer (Rechenzentrum)

Online-Redaktion

Herr Nikolas Klauser (Online-Redakteur)