

Lösungen zum Aufgabenblatt 2

Begriffe

Servlets/JSP Servlets, zu Deutsch etwa „Server-chen“, stellen Anwendungen für Server zur Verfügung oder erweitern diese. Alle Servlets implementieren die Java-Klasse `javax.servlet` oder eine davon abgeleitete Klasse, um anschließend innerhalb eines J2EE Applicationsservers zu laufen. Servlets arbeiten auf dem request-respond Prinzip und können auf zahlreiche requests antworten, hauptsächlich werden sie jedoch verwendet um mittels HTTP die Anwendungsmöglichkeiten eines Webservers zu erweitern. Zu diesem Zweck existiert auch die von `javax.servlet` abgeleitete Klasse `javax.servlet.http.HttpServlet`. Dies ermöglicht das Erstellen von dynamischen Web-Inhalten. Die JavaServer Pages erweitern die Servlet-Technologie, indem sie einfaches, dynamisches Erstellen von HTML und XML-Ausgaben ermöglichen. Damit wird zum einen wiederum die Darstellung von der Logik getrennt und zum anderen einfaches Erstellen von dynamischen Websites ermöglicht. OLAT als Webanwendung basiert auf Servlets und den JSPs, weshalb diese Begriffe eine zentrale Rolle erhalten.

Velocity Velocity ist ein Open-Source-Projekt und entstammt der Apache Jakarta-Projektgruppe. Es handelt sich hierbei um ein Java-basiertes Template, das es ermöglicht Java-Objekte in verschiedenste andere Formate, wie Text, XML, SQL, Post Script oder HTML einzubinden. Da es streng nach dem MVC-Konzept konzipiert wurde und nicht sonderlich schwer zu erlernen ist, erlaubt Velocity quasi paralleles Arbeiten an funktionalen Code und einem ansprechenden Äußeren. Die Hauptanwendungen von Velocity sind XML-Transformationen, das Erstellen von dynamischen Webseiten und Webanwendungen. OLAT nutzt Velocity aufbauend auf der Präsentationslogik, so dass der Client die erhaltenen Daten nur noch darstellen muss. Es wird deshalb ein wichtiges Hilfsmittel im Projekt sein.

J2EE Die Java (2) Platform, Enterprise Edition ist eine Entwicklungs- und Laufzeitumgebung von verteilten Anwendungen, die auf einer n-Schichten Architektur beruhen. Hierfür stellt die Plattform eine Vielzahl von APIs zur Verfügung, um etwa Servlets, EJBs oder RMI's etc. erzeugen zu können. Desweiteren beinhaltet die J(2)EE Spezifikationen und Standards, damit die entwickelten Programme auf allen möglichen Applicationsservern ausführbar bleiben. J2EE stellt also auch die Standards und die meisten Werkzeuge für OLAT und somit für dieses Projekt zur Verfügung.

Tomcat Tomcat bezeichnet einen Open-Source Servlet-Container, also eine Laufzeitumgebung für Servlets, die bei Apache innerhalb des Jakarta-Projektes entwickelt wird. Er wurde vor allem für JavaServer Pages konzipiert und nutzt hierzu den JSP-Compiler Jasper. Tomcat bietet neben dem Servlet-Container Catalina und einem HTTP-Server auch den Connector Coyote an, um die Kommunikation mit anderen Servern zu ermöglichen. Da Tomcat Open-Source ist, also nix kostet und relativ benutzerfreundlich ist, ist es einer der am häufigsten benutzten Applicationserver.

MVC Das MVC-Paradigma ist ein grundlegendes Architekturkonzept, das Modell, Controller und View (Darstellung) klar voneinander trennt. Dabei können die einzelnen Komponenten nicht beliebig miteinander kommunizieren, sondern gehorchen dem MVC-Modell, so dass die einzelnen Teile möglichst unabhängig voneinander sind. Diese Trennung macht zum einen nahezu paralleles Arbeiten an diesen Teilen möglich, untertützt eine hohe Wiederverwendbarkeit sowie Wartbarkeit und macht es möglich, relativ leicht Änderungen und Erweiterungen vor zu

nehmen. Die Architektur von OLAT richtet sich mit ihren Schichten stark am MVC-Modell aus und so wird auch dieses Projekt nicht auf dieses Muster verzichten können.

XML Bei XML handelt es sich um eine Auszeichnungssprache („Extensible Markup Language“) für Datenrepräsentation in Textform. Dabei stellt XML nur eine Metasprache dar, aus der mittels Schemasprachen wie DTD die eigentliche Sprache gewonnen wird. Zu den auf XML basierenden Sprachen gehören u.a. XHTML, WML und RSS. Die repräsentierten Daten sind relativ leicht durch Maschinen zu verarbeiten, bleiben aber auch für Menschen les- und änderbar. Die Anwendungen von XML sind entsprechend vielfältig, am häufigsten wird XML jedoch zur Speicherung Daten oder zur intermaschinellen Kommunikation genutzt. Im Projekt OLAT taucht XML u.a. für die Konfiguration und Kommunikation auf.

Ajax Ajax steht für Asynchrones JavaScript und XML und bezeichnet das Zusammenspiel zwischen Javascript, DOM und (X)HTML, das es mittels eines XMLHttpRequests möglich macht, nur bestimmte Teile einer HTML-Seite während einer HTTP-Anfrage nachzuladen. Auf diese Weise muss nicht die gesamte Seite nachgeladen werden, was Desktop-ähnliche Webanwendungen wie Flickr oder Google Maps erst ermöglicht. Das GUI-Framework von OLAT nutzt Ajax, um nur die benötigten Komponenten der Oberfläche neu zu laden und so einen schnelleren und flüssigeren Seitenaufbau zu ermöglichen. Ajax ist also vor vor allem für die Benutzeroberflächengestaltung in diesem Projekt interessant.

Extension / ExtensionPoint Eine Extension soll eine Erweiterung der vorliegenden Rahmenapplikation OLAT beschreiben, mittels derer neue Funktionalitäten hinzugefügt werden können. Dabei sollte es nicht vonnöten sein, den bereits bestehenden Quellcode zu verändern, sondern die bereits von den OLAT-Entwicklern zur Verfügung gestellten Schnittstellen, die sogenannten ExtensionPoints oder Erweiterungspunkte zu adaptieren. Es existieren verschiedene Erweiterungspunkte für unterschiedliche Zwecke, so bspw. einer für das Erstellen neuer Seiten. Da beabsichtigt wird eine computergestützte Lösung für bislang noch nicht durch die bestehende Rahmenapplikation unterstützte Arbeitsabläufe zu entwerfen, ohne jedoch bei letztllicher Implementierung bestehenden Quellcode zu ändern, kann das Ziel dieses Projekts aus technischer Sicht als eine Erweiterung von OLAT, basierend auf dessen Extension-Concept, respektive entprechenden ExtensionPoints angesehen werden.

Spring-Framework Spring ist ein Open-Source Java/J2EE Application-Framework, oder besser eine Sammlung von Frameworks, die die Entwicklung und Ausführung von Java-Programmen vereinfachen sollen. Dafür stellt das Framework Methoden und Möglichkeiten zur Verfügung und übernimmt selber viele der anfallenden Aufgaben (Inversion of Control). Zu den Werkzeugen des Frameworks zählen u.a. Frameworks für das MVC-Konzept, für den Datenzugriff per JDBC, für Transaktionen und einem Testframework. Bei all dieser Hilfe für den Entwickler, sind die im Framework entstehenden Programme nicht abhängig von den Spring-APIs. Aufgrund seiner zahlreichen Vorteile kam Spring auch bei der Entwicklung von OLAT zum Einsatz und wird auch in diesem Projekt eine große Hilfe sein.

LMS Ein Learning Management System (Lernplattform) ist ein Programm, das die Betreuung von Lehrinhalten ermöglicht. Dabei bietet das LMS eine Benutzerverwaltung, sowie Rollen- und Rechtevergaben an, stellt Arbeitswerkzeuge und Kommunikationsmethoden wie Foren, Wikis etc. zur Verfügung und bietet Möglichkeiten zur Kursverwaltung, für Onlineevaluationen und -bewertungen. Natürlich sorgt das LMS außerdem für die Verfügbarkeit und die Präsentation der Lehrinhalte in einem normalen Browser. Das Open-Source LMS OLAT ist Gegenstand dieses Projektes, weshalb zumindest der Begriff geklärt sein muss.

Konzepte

Multi-Tier-Architektur

Die Multi-Tier-Architektur (mehrschichtige Architektur) ist eine bestimmte Form einer Anwendungsarchitektur. Dabei wird die gesamte Anwendung modularisiert. Häufig wird eine 3-Tier-Architektur für Webanwendungen eingesetzt, die allerdings auch in n-Tier-Architekturen verfeinert werden kann.

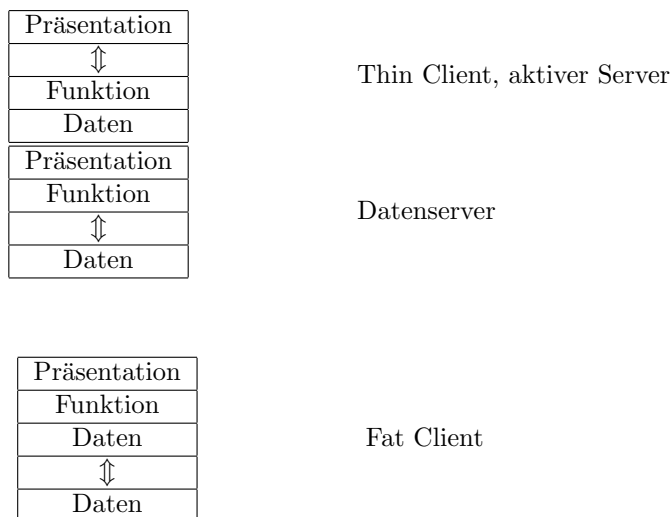
Es wird in folgende Schichten unterteilt:

- Präsentation
- Steuerung
- Anwendung
- Datenverwaltung
- Datenhaltung

Eine typische 3-Tier-Architektur besteht aus:

- Präsentation
- Funktion
- Daten

Es ist möglich, die Last der Applikation zu verlagern, so dass der Schwerpunkt zum einen beim Server oder beim Client liegen kann. Folgende Aufteilungen sind möglich:



Datenbankunabhängigkeits-Konzept

Zur Persistierung von Objekten wird auf den Object-Relational Mapper Hibernate zurückgegriffen. Hibernate abstrahiert dabei die konkrete Datenbank, so dass OLAT mit jedem relevanten Datenbankmanagementsystem zusammenarbeiten kann, sei es MySQL, Postgres oder Oracle. Dies gibt die Möglichkeit die Installation von Anforderungen anzupassen.

Komponentenbasiertes-Systemarchitektur-Konzept

Die Systemarchitektur von OLAT ist sorgfältig und basierend auf den allgemein gültigen und anerkannten Blue-Prints aufgebaut. Durch die komponentenbasierte Bauweise ist eine das System durchdringende Modularität erreicht worden. Wo immer möglich werden generische Komponenten eingesetzt. Dadurch wird in OLAT eine enorm hohe Wiederverwendbarkeit erreicht, was sich wiederum in stabilem, mehrfach getestetem und bewährtem Code niederschlägt.

Webapplikation

Eine Webanwendung oder auch Webapplikation ist ein Computer-Programm, welches auf einem Webserver ausgeführt wird, wobei eine Interaktion mit dem Benutzer ausschließlich über einen Webbrowser erfolgt. Die Hauptfunktion besteht in der gegenseitigen Kommunikation zwischen Client und Server. Angenommen ein Benutzer ruft eine Webanwendung über eine URL des Webserver auf und startet somit einen HTTP-Request an den Webserver und dieser übergibt die Anfrage an die eigentliche Webanwendung. Die Webanwendung generiert HTML-Quellcode einer Webseite, und der Webserver sendet per HTTP-Response diesen zurück an den Browser (Client).

In der Regel läuft eine Webanwendung auf dem Webserver, aber kann auch im professionellen Bereich auf einen Application-Server ausgelagert werden, welche dann mit Benutzer-Anfragen von mehreren Webservern bedient werden können. Man unterscheidet dabei zwei verschiedene Architekturen:

- Standalone:
 - Die Webanwendung ist ein eigenständiges Binär-Programm, welches für jede Anfrage neu gestartet wird.
- Integriert:
 - Die Webanwendung ist ein Teil vom Webserver. Es muss nicht für jeden neuen Request ein Programm gestartet werden.

Pro:

Plattformunabhängigkeit Die Webanwendungen setzen auf dem System des Benutzers nur Webbrowser voraus.

Wartungsfreudig Muss in einer Webanwendung etwas geändert werden, dann sind die Änderungen nur an einer zentralen Stelle, nämlich auf dem Webserver notwendig.

Contra:

ständige TCP/IP-Verbindung Zwischen Client und Server muss eine dauerhafte Verbindung stehen und die Bandbreite der Verbindung muss außerdem auf die Anforderungen der Webanwendung ausgelegt sein.

Webbrowserunabhängigkeit Im Idealfall sollten die Webanwendungen in allen Webbrowsern richtig funktionieren und dargestellt werden, jedoch ist dies nicht selbstverständlich denn von manchen Browsern wird trotz bestehender Standards der HTML-Quellcode unterschiedlich interpretiert.

Servlets

Dabei handelt es sich um Server-Applications (“Server“ + “Applets“), sprich meist kleinere Java-Programme die im Gegensatz zu den bekannten Applets nicht auf der Clientseite laufen sondern direkt auf dem Server. Sie dienen dazu, auf Java basierte, dynamische Webseiten anbieten zu können und stehen damit in Konkurrenz zu ASP, PHP und CGI.

In Hinblick auf die Kommunikation mit dem User beschreibt das Servlet-Konzept keine neuen Pfade: das HTTP-Protokoll, insbesondere GET und POST, dienen zur Übertragung der Nutzeranfragen, auf die i.a. mit einem HTML-Text geantwortet wird. Möglich ist jedoch jede beliebige Rückgabe, wie XML, Bilder, Binärdaten, Zusätzlich lassen sich Sessions via Cookie, URL oder Variablen verfolgen.

Ein zentraler Bestandteil des Servlet-Konzepts ist der Webcontainer, der im Webserver läuft und die Servlets ausführt. Er managt den vollständigen Lebenszyklus des Servlets, erstellt das Servlet-Objekt, initialisiert es einmal über einen parameterlosen Konstruktor und ruft die `init(ServletConfig config)` Methode auf, gibt es wieder aus dem Speicher frei indem es `destroy()` des Servlets aufruft. Der Container überwacht die Zugriffsrechte auf die Servlets und verknüpft eine URL mit dem entsprechendem Servlet.

Eine Besonderheit der Servlet-Technologie ist dabei die Skalierbarkeit, die bei anderen teils nicht gegeben ist (z.B. CGI). Anstatt für jede Nutzeranfrage einen eigenen Prozess zu starten läuft ein Servlet meist nur einmal im Speicher (Ausnahmen sind Servlet-Pools). Aus diesem Grund sind alle Methoden threadsicher zu bauen oder falls nötig das Interface `SingleThreadModel` zu implementieren.

Auf Nutzerinteraktion reagiert ein Servlet mit der `service()` Methode die je nach Anfrage die Bearbeitung auf `doPost()` oder `doGet()` weiterleitet. Um Zugriff auf die Benutzer-Eingaben zu erhalten erhalten, bzw den Output zu erzeugen erhalten `doGet()` und `doPost()` je ein Objekt vom Typ `ServletRequest` sowie `ServletResponse`. Über `ServletRequest` können die mitgesendeten Cookies ausgelesen, die mit GET oder POST geschickten Werte und sonstige Nutzerinformationen. Mit dem Response wird die HTML-Seite gebaut, die dem Nutzer dann als Ausgabe zurückgesendet wird.

Rechte- und Rollenkonzept

Dieses Konzept von OLAT lehnt sich stark an das Policy-Konzept von JAVA an. Es gibt Identities (User), Gruppen mit Identities, Rechte, Ressourcen und Policies. User sind in einer oder mehreren Gruppen. Eine Policy ist die Kombination von einer Gruppe, einem Recht und einer Resource. Ein Benutzer hat dann ein gewisses Recht (z.B. lesen) auf eine gewisse Resource (z.B. `RepositoryEntry`), wenn er in mindestens einer Gruppe ist, die von einer Policy referenziert wird, welche das Recht “lesen“ auf die Resource hat, d.h. die Policy hat einen Fremdschlüssel auf die Gruppe, einen Fremdschlüssel auf die OLAT-Resource und einen Text für das Recht. Die Systemrollen sind also nichts anderes als Gruppen mit bestimmten Rechten.

Beschreibung der Rahmenapplikation

- OLAT ist ein Lernmanagement-System, das mit dem Ziel entwickelt wurde, Vorlesungen und Übungen über das Internet anbieten zu können, wobei der Fokus auf dem Aufbau und auf

der Durchführung von Lehrveranstaltungen liegt.

- Quelle: Ist ein Open-Source-Projekt, das seit 1999 an der Universität Zürich entwickelt wird
- Leistungsmerkmale:
 - Unterstützung etablierter E-Learning-Standards für Tests und Fragebögen sowie für sonstige Lehrinhalte
 - Einbindung externer Ressourcen mittels Inline-Rendering oder eingebetteter Frames
 - Komfortable (Lern-)Ressourcenverwaltung
 - Verwendung von AJAX zur Anreicherung des LMS mit Funktionen wie Drag and Drop, automatischer Vervollständigung bei Eingabefeldern usw.
 - Übersichtliche Benutzeroberfläche und AdministrationAuf Serverseite wird JAVA benötigt, was in vielen Webhost-Angeboten nicht enthalten ist. Ein Hochschul-RZ kann dies aber sicher bereitstellen.
 - Übersichtliche Benutzeroberfläche und Administration
- Softwarekomponenten:
 - OLAT-Distribution: enthält die JAVA-Sources und Bytecode, die Übersetzungsdateien, und die statische Ressourcen (Icons etc ...)
 - JAVA Servlet Engine: Version ab 2.3; Empfehlung: Apache Tomcat 5.0.x oder 5.5.x
 - JAVA 1.4 oder höher. Wird die optionale Instant Messaging und Chat Komponente eingesetzt wird Java 1.5 vorausgesetzt
 - Datenbank: Folgende Datenbanken werden durch Hibernate unterstützt: DB2, MySQL, SAP DB ...
- OLAT Systemgruppen (Rollen):
 - OLAT ist ein rechtebasiertes System mit mehreren Hierarchiestufen. Der Benutzer hat jederzeit diejenigen Funktionen zur Verfügung, zu denen er im gegebenen Kontext berechtigt ist. In OLAT werden die folgenden Systemgruppen unterschieden:
 - * Gäste: Sind anonyme Benutzer mit eingeschränkten Rechten auf Funktionen. Gäste können ihre Benutzeroberfläche nicht anpassen und weder Tests absolvieren noch Forumsbeiträge verfassen.
 - * Benutzer: OLAT-Benutzer (Studenten) können ihre Benutzeroberfläche gemäss ihren persönlichen Präferenzen gestalten und einen Kurs als Teilnehmer starten.
 - * Autoren: Zusätzlich zu den Berechtigungen eines OLAT-Benutzers können Autoren Lernressourcen hinzufügen, herstellen sowie administrieren.
 - * Gruppenverwalter: Mitglieder dieser Systemgruppe können kursübergreifend Lern- und Rechtegruppen verwalten - dies zusätzlich zu den Rechten eines Benutzers.
 - * Benutzerverwalter: Die Personen dieser Rollen können neue Benutzer erstellen oder importieren und diesen Rollen zuweisen.
 - * Administratoren: Berechtigt die Benutzer dieser Gruppe zu administrativen Tätigkeiten im gesamten OLAT-System. Beinhaltet ebenfalls die Rechte der Gruppenverwalter, Autoren und Benutzer.
- Anwendungsfälle:
 - Anwendungsfall: Anmeldung am OLAT

- * Akteur: Benutzer, Autor, Gruppenverwalter, Benutzerverwalter und Admin
- * Beschreibung: Der Akteur gibt auf einer Weseite Kennung und Passwort ein. (Gäste benötigen keine Anmeldung!)
- * Ergebnis: Wenn Kennung und Passwort stimmt, wird der Zugang mit den entsprechenden Zugriffsrechten gewährt.
- Anwendungsfall: Einschreibung in Lehrveranstaltungen
 - * Akteur: Benutzer (Student)
 - * Beschreibung: Nach erfolgreicher Anmeldung wählt der Nutzer eine Lehrveranstaltung bzw. eine Prüfung aus und füllt ein Formular aus, um sich für die Lehrveranstaltung / Prüfung anzumelden.
 - * Ergebnis: Wenn die Einschreibung erfolgreich ist, erhält der Nutzer sofort eine Bestätigung der Einschreibung auf der Webseite.
- Anwendungsfall: Anlegen von Lehrveranstaltungen
 - * Akteur: Autor (Dozent)
 - * Beschreibung: Nach erfolgreicher Anmeldung kann ein Autor eine neue Lehrveranstaltung anlegen.
 - * Ergebnis: Wenn das Anlegen der Lehrveranstaltung (mit allen Parametern) erfolgreich war, steht diese im Veranstaltungsangebot zur Verfügung.
- Anwendungsfall: Vergeben von Rechten
 - * Akteur: Admin
 - * Beschreibung: Admin kann an einen Benutzer Autor-, Gruppenverwalter-, Benutzerverwalter- und Adminrechte zuweisen.
 - * Ergebnis: Benutzer erhält erweiterte Rechte
- Anwendungsfall: Erstellen und Bearbeiten von News
 - * Akteur: Administrator, Autoren
 - * Beschreibung: Administratoren bzw. Autoren können das System betreffende bzw. eine Lehrveranstaltung betreffende News (z.B.: Klausurergebnisse) erstellen und ältere News bearbeiten.
 - * Ergebnis: Neu erstellte bzw. veränderte News sind auf der Seite für alle berechtigten Nutzer sichtbar.
- Anwendungsfall: Informationen einsehen
 - * Akteur: Benutzer, Autor
 - * Beschreibung: Falls ein Benutzer über die entsprechenden Rechte verfügt hat er Zugriff auf News, wie Klausurergebnisse usw. im OLAT.
 - * Ergebnis: Der Benutzer erhält für ihn relevante Informationen.
- Anwendungsfall: Bereitstellen von Material zu Lehrveranstaltungen
 - * Akteur: Autoren
 - * Beschreibung: Autoren von Lehrveranstaltungen können zu Ihren zuvor angelegten Veranstaltungen Materialien zur Verfügung stellen.
 - * Ergebnis: Die entsprechenden Dokumente stehen den mit entsprechenden Rechten ausgestatteten Benutzern zur Verfügung.

- Anwendungsfall: Download von Material zu Lehrveranstaltungen
 - * Akteur: Benutzer (Student), Autoren
 - * Beschreibung: Der Nutzer kann sich Material (Folien o.ä.) vom Server downloaden, wenn er über die entsprechenden Leserechte verfügt.
 - * Ergebnis: Download wird gestartet
- Anwendungsfall: Neue Lerngruppe anlegen
 - * Akteur: Gruppenverwalter
 - * Beschreibung: Ein Gruppenverwalter kann eine kursübergreifende oder kursinterne Lerngruppen anlegen.
 - * Ergebnis: Die (neue) Lerngruppe steht den Benutzern im Veranstaltungsangebot zur Verfügung.
- Anwendungsfall: Beitritt zu einer Lerngruppe
 - * Akteur: Benutzer (Student)
 - * Beschreibung: Ein Student kann einer angebotenen Lerngruppe beitreten, falls die Voraussetzungen erfüllt sind. (max Teilnehmerzahl usw. ...)
 - * Ergebnis: Ist die Anmeldung erfolgreich, so erscheint ein Link zu der entsprechenden Lerngruppe.
- Anwendungsfall: Sperren/Löschen von Accounts
 - * Akteur: Admin, Benutzerverwalter
 - * Beschreibung: Admin bzw. Benutzerverwalter haben die Möglichkeit Accounts zu sperren bzw. zu löschen. Benutzer können gesperrt werden, falls sie die Voraussetzungen für das OLAT-System nicht mehr erfüllen (Exmatrikulation usw. ...).
 - * Ergebnis: Benutzer ist gesperrt/gelöscht
- Anwendungsfall: Registrieren
 - * Akteur: Gast
 - * Beschreibung: Jeder Gast kann sich bei OLAT registrieren.
 - * Ergebnis: Gast kann sich als Benutzer mit allen Rechten eines Benutzers einloggen.
- Anwendungsfall: Anlegen von Accounts
 - * Akteur: Benutzerverwalter
 - * Beschreibung: Neue Benutzer werden einzeln erfasst, oder durch Excel-Dateien importiert und mit den entsprechenden Rechten ausgestattet.
 - * Ergebnis: Die neuen Benutzer erhalten Zugang zu OLAT mit den entsprechenden Rechten.

