

Fachhochschule Wiesbaden
Fachbereich Design Informatik Medien
Studiengang Allgemeine Informatik

Diplomarbeit
Zur Erlangung des akademischen Grades
Diplom-Informatiker (FH)

Implementierung eines Hub & Spoke-Systems zum
effizienten und sicheren Austausch von
UN/EDIFACT-Daten via Internet mit Web-basierter
Benutzer-Selbstverwaltung und EDI Track & Trace-
Funktion.

Vorgelegt von David Böhm
Am 26.01.07

Referent: Prof. Dr. Werntges
Korreferent: Prof. Dr. Linn

Eidesstattliche Erklärung gemäß Prüfungsordnung Teil A § 6.4.2

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum

Unterschrift Diplomand

Hiermit erkläre ich mein Einverständnis mit den im Folgenden aufgeführten Verbreitungsformen dieser Diplomarbeit:

Verbreitungsform	ja	nein
Einstellung der Arbeit in die Bibliothek der FHW		
Veröffentlichung des Titels der Arbeit im Internet		
Veröffentlichung der Arbeit im Internet		

Ort, Datum

Unterschrift Diplomand

1.	<i>Einleitung</i>	6
1.1.	EDI elektronischer Datenaustausch	6
1.1.1.	UN/EDIFACT und Subsets	6
1.1.2.	Typisches Szenario mit Beispielen.....	7
1.2.	Aktuelle Situation	9
1.2.1.	Was ist ein VANS	11
1.2.2.	Was ist EDI-INT AS2	12
1.3.	Zielsetzung	13
1.3.1.	Besondere Merkmale.....	15
1.3.1.1.	Verteilte Anwendung.....	15
1.3.1.2.	Sicherheit.....	15
2.	<i>Standards und Werkzeuge</i>	17
2.1.	Entwicklungsumgebung	17
2.2.	Verwendete Standards	20
2.2.1.	HTTPS.....	20
2.2.2.	FTPS.....	20
2.3.	Verwendete Techniken	22
2.3.1.	Mailbox-Prinzip	22
2.3.2.	Hub & Spoke.....	22
2.4.	Ruby	23
2.4.1.	RubyGems	23
2.4.2.	RDoc und ri.....	23
2.4.3.	Die interaktive Ruby-Shell	24
2.5.	Ruby on Rails	25
2.5.1.	Die Verzeichnisstruktur.....	27
2.6.	AJAX	29
3.	<i>Das Design der Anwendung</i>	31
3.1.	Server	32
3.2.	Benutzerkonto	33
3.3.	Registrierung	34
3.4.	Teilnehmerselbstverwaltung	34
3.5.	Datenaustausch	34
3.5.1.	Senden von Daten.....	34
3.5.2.	Zustellen der Dateien.....	35
3.5.3.	Empfangen von Daten	35
3.6.	Track & Trace	35
3.7.	Entstehende Kosten	35
3.8.	Sicherheit	36
4.	<i>Implementierung</i>	37
4.1.	EDI-Hub-Server	39
4.1.1.	Klassenbeschreibung	41
4.1.2.	Sequenzdiagramm:	44
4.2.	EDI-Hub-Kontroller	46
4.2.1.	Kommandos	47

4.3.	Track & Trace-Recorder	48
4.4.	Track & Trace	49
4.4.1.	Ereignisse	49
4.5.	Datenbank	50
4.5.1.	Tabellenbeschreibung.....	51
4.6.	Webanwendung	55
4.6.1.	Registrierung	57
4.6.1.1.	Stammdaten	58
4.6.1.2.	E-Mail-Überprüfung.....	59
4.6.1.3.	Kontoaktivierung.....	59
4.6.2.	Eigene Einstellungen.....	60
4.6.2.1.	GLNs verwalten.....	60
4.6.2.2.	Geschäftspartner definieren	61
4.6.2.3.	Geschäftspartner mit Datenüberprüfung.....	62
4.6.3.	Firmen	63
4.6.3.1.	Suchen	63
4.6.3.2.	Akzeptieren Daten	63
4.6.4.	GLNs	63
4.6.4.1.	Suchen	64
4.6.4.2.	Akzeptieren Daten	64
4.6.5.	Track & Trace	64
4.6.5.1.	Filter	65
4.6.6.	Administration.....	67
4.6.7.	GLN Detailansicht.....	68
4.6.8.	Ajax	70
4.7.	Verwendete Programme	72
4.7.1.	Apache2.....	72
4.7.2.	Proftpd.....	72
4.7.3.	Mysql.....	72
5.	Test.....	73
5.1.	EDI-Hub-Server	73
5.2.	Webanwendung	75
6.	Diskussion	77
6.1.	Erweiterung um einen AS2 Gateway.....	78
6.2.	Komprimierung der eingehenden EDI-Dateien.....	78
6.3.	Archivierung der EDI-Dateien	79
6.4.	Statussätze für jede Mailbox	79
6.5.	Vereinfachung der Installation.....	80
6.6.	Versenden von Nicht-UN/EDIFACT-Daten.....	80
7.	Zusammenfassung und Ausblick	81
8.	Literaturverzeichnis	82
9.	Glossar	84
10.	Anhang	85
10.1.	Installationsanleitung.....	85
10.1.1.	Notwendige Programme installieren und einrichten.....	85
10.1.1.1.	Ruby	85
10.1.1.2.	RubyGems	85
10.1.1.3.	EDI Bibliotheken.....	85

10.1.1.4.	Ruby on Rails	86
10.1.1.5.	Datenbank.....	86
10.1.1.6.	Webserver.....	86
10.1.1.7.	FTPS-Server	86
10.1.2.	Hub-Server installieren und einrichten	87
10.1.3.	Webanwendung installieren und einrichten.....	88
10.1.4.	Hub-Server starten und bedienen.....	90
10.2.	Inhalt der CD	91

1. Einleitung

Ziel dieser Diplomarbeit ist eine Betreiber-Anwendung die den Austausch von standardisierten Nachrichten durchführt. Im Mittelpunkt steht dabei die Fragestellung nach einer kosteneffizienten Lösung zum automatisierten Datenaustausch von sensiblen Daten zwischen Geschäftspartnern.

Seit den 60er Jahren werden in der Industrie automatisierte Datenübertragungen zwischen Geschäftspartnern durchgeführt. In den späten 80er Jahren wurde dazu der Nachrichtenstandard UN/EDIFACT [Lit01][Lit02] durch die Vereinten Nationen verabschiedet.

Der automatisierte Datenaustausch gehört zu den Kernfunktionen von EDI.

1.1. EDI elektronischer Datenaustausch

1.1.1. UN/EDIFACT und Subsets

UN/EDIFACT [Lit01][Lit02] definiert ein allgemeines, branchen-, system- und länderunabhängiges Nachrichtenformat einer automatisierten Datenübertragung. Das Abstraktionsniveau bereitet jedoch im praktischen Einsatz Schwierigkeiten, da das Nachrichtenformat durch zu viel Funktionalität für viele Anwendungen überdimensioniert ist – konkretere Definitionen liefern Subsets.

Branchenspezifischer Einsatz von UN/EDIFACT kann durch Bildung von sogenannten Subsets erreicht werden. Ein Subset definiert eine Teilmenge mit den in UN/EDIFACT definierten Funktionen. Sehr spezielle Anforderung können dabei durch Wegfall von nicht benötigten Funktionen des allgemeinen Standards erreicht werden.

Es sind folgende genormte Subsets für spezifische Branchen bekannt:

Subset	Einsatzgebiet
CEFIC	Chemische Industrie
EANCOM	Konsumgüterindustrie
EDIBDB	Baustoffbranche
EDIFICE	High Tech Industrie
EDIFOR	Speditionsbranche

EDIFURN	Möbelbranche
EDIG@S	Erdgasbranche
EDILEKTRO	Elektroindustrie / Elektrogroßhandel
EDILIBE	Buchhandel
EDITEC	Sanitärbranche
EDITEX	Textilindustrie
EDITRANS	Transportwirtschaft
EDIwheel	Reifen- und Räderhersteller
ODETTE	Automobilindustrie
RINET	Versicherungswirtschaft
ODA/ODIF	Allgemeine Dokumentenformate
ETIS	Telekommunikation (nur für Rechnung)

1.1.2. Typisches Szenario mit Beispielen

Nachfolgend wird ein typisches Szenario eines Geschäftspartners beschrieben. Geschäftspartner können hierbei aus beliebigen Branchen, wie z.B. Konsumgüterindustrie oder Spedition, stammen.

Ein Geschäftspartner hat im heutigen Zeitalter zur Erfüllung seiner Prozesse im Workflow vielfältige Kommunikationen zu tätigen, sei es innerhalb seines Unternehmens, zum Beispiel mit der Rechts- oder Buchhaltungsabteilung oder mit einem externen Dienstleister, um seinen Kunden eine größere Produktpalette anbieten zu können.

Bestellabwicklung:

Firma X wickelt Bestellungen und Bestellungsänderungen mit seinem Großhändler über EDI ab, um Lagerbestände aufzufüllen. Der Großhändler reagiert auf Aufträge in der Art, dass Auftragsbestätigung, Lieferstatus und ggf. ein voraussichtlicher Liefertermin, wiederum in elektronischer Form, mitgeteilt werden. Dadurch kann Firma X zeitnah Lagerbestände anpassen und der Großhändler entsprechend schnell reagieren.

Mahnungsversand:

Firma X hat eine Rechnung von Firma Y erhalten, jedoch wurde diese bis zur Fälligkeit nicht beglichen. Firma Y möchte den Rechnungsbetrag nun anmahnen und verschickt die, durch das Buchhaltungssystem automatisch erzeugte,

elektronische Mahnung. Dadurch werden die effektiven Kosten des Mahnwesens stark gesenkt und der Versandt beschleunigt.

1.2. Aktuelle Situation

In der heutigen Zeit neigen häufig Firmen beliebiger Branchen dazu, viele ihrer Aufgaben an externe Dienstleister auszulagern, als Beispiel sei einmal das Mahnwesen genannt. Um einen reibungslosen Prozess auch nach einer Outsourcing Maßnahme zu gewährleisten, muss eine sehr gute Kommunikation zwischen den Beteiligten sichergestellt sein. Durch eine automatische Abwicklung der benötigten intra- bzw. inter-company Kommunikationen, kann der unterliegende Prozess zeitnah, weniger Fehler behaftet, kosten- und personaleffizient erfolgen. EDI übernimmt diese Aufgabe.

Insbesondere profitieren große Firmen von dem Einsatz von EDI. Ein schneller Datenaustausch mit Geschäftspartnern kann dafür sorgen, dass z.B. Rechnungen schneller ihren Adressaten erreichen können und schlussendlich schneller bezahlt werden können – ein Anstieg des Cash-Flow ist die positive Folge.

Eine Umstellung auf EDI erfordert jedoch auch Anpassungen der internen elektrischen Datenverarbeitung an ein bestehendes, auf genormte Datenaustauschformate ausgelegtes, System. Denn betriebsinterne Datenstrukturen, auch INHOUSE genannt, sind anwendungsspezifisch und spiegeln häufig betriebliche Abläufe in ihrem Format wieder. Um diese, im Betrieb etablierte, Datenstruktur auch für EDI Zwecke nutzen zu können, muss sie in das Zwischenformat UN/EDIFACT konvertiert werden. Im Anschluss kann die konvertierte Datei im Sinne des EDI an seinen Empfänger übermittelt werden, der seinerseits eine Konvertierung in ein für ihn betrieblich verwendbares Format vornehmen muss. (Abb. 1)

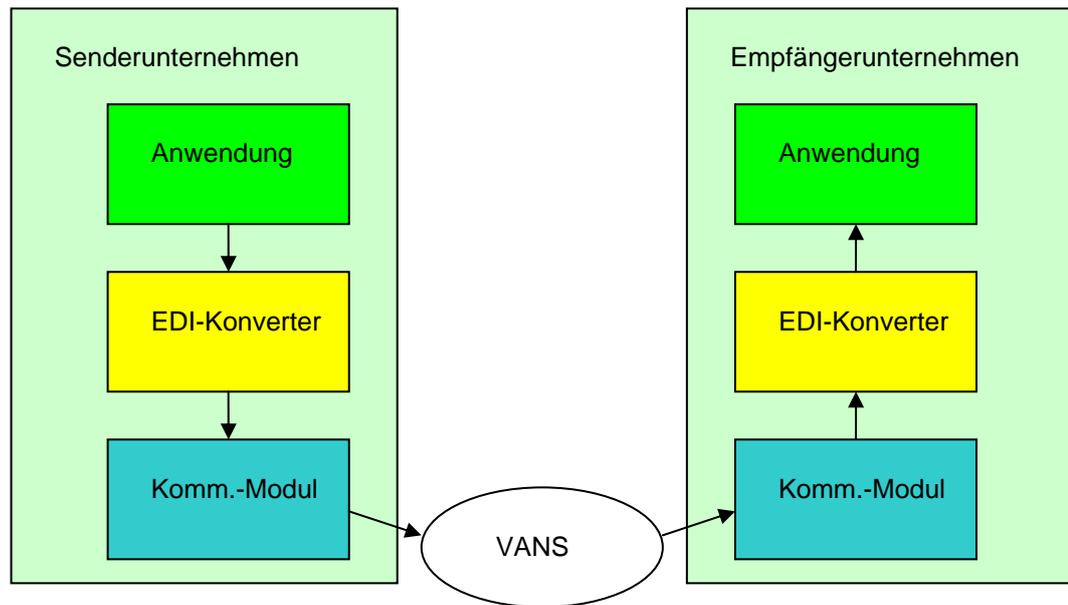


Abb. 1: Schritte des EDI-Versands.

Ohne den Einsatz des Zwischenformates UN/EDIFACT müsste ein Unternehmen die anwendungsspezifische Datenstruktur seiner Geschäftspartner kennen und einen entsprechenden Konverter je Geschäftspartner bereitstellen bzw. Konvertierungen in viele verschiedene Datenformate vornehmen, bevor eine Übermittlung auf sinnvolle Weise möglich wäre. Das Unternehmen müsste je neu gewonnenen Geschäftspartner einmalig einen Konverter für diesen Geschäftspartner bereitstellen. Neben diesem Einrichtungsaufwand kommt noch ein weiterer Nachteil zum Tragen. Bei jeder Änderung der Datenstrukturen müssten alle bestehenden Geschäftspartner ihre Konverter zeitnah anpassen, um einen störungsfreien EDI Verkehr zu gewährleisten. Der Wartungsaufwand einer solchen Struktur würde exponentiell zur Anzahl bestehender Geschäfts- und Kommunikationspartner ansteigen.

Einer der Hinderungsgründe für die Umstellung auf EDI für kleine und mittlere Unternehmen sind die entstehenden Fixkosten für die Anpassungen der EDV und die, unter Umständen, zusätzlich benötigte Hardware. Einen weiteren Grund stellen die laufenden Kosten einer Mitgliedschaft an einem Übertragungs-Netzwerk, dem sogenannten VANS, dar. Ein solches Übertragungs-Netzwerk ist für die Übertragung und Zustellung von EDI-Daten zwischen den Geschäftspartnern verantwortlich und rechnet nutzungsabhängig nach Anzahl der übermittelten Daten ab. Eine rege EDI-Kommunikation zwischen zwei Geschäftspartnern kann dadurch schnell die Kosten in die Höhe treiben. Zu erwähnen bleibt noch, dass Geschäftspartnern der

Datenaustausch grundsätzlich lediglich innerhalb eines Übermittlungs-Netzwerkes möglich ist. Möchten beispielsweise Geschäftspartner, die Mitglieder an unterschiedlichen VANS sind, miteinander kommunizieren, setzt dies voraus, dass alle beteiligten Geschäftspartner zusätzliche Mitgliedschaften bei den ihnen fremden VANS eingehen müssen. Es gibt jedoch auch Ausnahmen unter den VANS-Dienstleistern, die miteinander über ein sogenanntes Gateway kooperieren und intern nicht zustellbare EDI-Daten gegenseitig austauschen, so dass Mitglieder dieser Kooperationspartner lediglich eine Mitgliedschaft antreten müssen, um mit Mitgliedern im gesamten Kooperationsverbund zu kommunizieren.

Auf Grund der hohen laufenden Kosten von VANS steigen immer mehr große Unternehmen auf EDI-INT AS2 um. Der Vorteil von EDI-INT AS2 sind die sehr geringen Übertragungskosten. Da es sich um eine Ende-zu-Ende Anbindung der Teilnehmer selbst handelt, ist eine ständige Erreichbarkeit der Teilnehmer erforderlich und unterscheidet sich grundsätzlich vom Mailbox-Prinzip. Zusätzlich erfordert der Betrieb von EDI-INT AS2 dedizierte Hardware und eine komplexe Einrichtung, wodurch sehr hohe Einmalkosten entstehen. EDI-INT AS2 ist für große Unternehmen trotz der hohen Einmalkosten lukrativ, da diese durch die vernachlässigbaren Übertragungskosten, schnell amortisiert sind. EDI-INT AS2 etabliert sich als neuer Standard zur sicheren Übermittlung von EDI-Dateien zwischen zwei Teilnehmern im Internet. Bei kleinen und mittleren Unternehmen überwiegen jedoch die hohen Einmalkosten, die sich auf Grund des geringen Aufkommens an EDI-Daten nicht amortisieren.

Für ein Unternehmen, das auf EDI umgestellt hat, ist die profitabelste Kommunikation der Art, dass alle Geschäftspartner ebenfalls EDI verwenden. Es liegt somit im Sinne dieser EDI nutzenden Firmen, ihre Geschäftspartner für eine solche oben genannte Umstellung zu gewinnen. Ein Mittel, dieses Vorhaben zu erreichen, wäre z.B. die Reduzierung oder vollständige Aufhebung der laufenden Betriebskosten für nicht angebundene Geschäftspartner.

1.2.1. Was ist ein VANS

VANS [Lit03][Lit04] steht für Value Added Network System und ist ein kommerzielles Übertragungs-Netzwerk für EDI-Daten. Jeder EDI-Konsument benötigt dabei eine Mitgliedschaft bei mindestens einem Übertragungs-Netzwerk, um die Übermittlung seiner, für den Transport aufbereiteter, EDI-Daten durchführen zu können. Dabei sind neben pauschalen Fixkosten auch laufende

Nutzungsgebühren an die kommerziellen Betreiber dieser Netzwerke zu entrichten. Die Historie von VANS führt zurück bis auf eine Zeit, in der das Internet nicht in dem heutigen Maße etabliert war. Die Übermittlung basiert daher größtenteils noch auf rudimentären Telefontechnologien, wie zum Beispiel X.400, und verwendet eine Hub & Spoke Netzwerktopologie als Ende-zu-Ende Anbindung.

Neben der Übermittlung, werden auch zusätzliche Dienstleistungen von VANS Betreibern angeboten, auf die jedoch in diesem Dokument nicht weiter eingegangen wird:

- Transformierung von Daten zwischen unterschiedlichen EDI-Standards
- Automatische Datensicherung
- Archivierung der Mailboxinhalte
- Fehlererkennung und Korrekturdienstleistungen

1.2.2. Was ist EDI-INT AS2

EDI-INT AS2 [Lit05] ist ein Standard zur sicheren Übertragung von Dateien zwischen zwei Teilnehmern im Internet unter Verwendung des HTTP Protokolls. Die übertragenen Daten werden mit einer authentifizierten Quittung als MDN bestätigt, so dass der Empfänger der Daten nicht leugnen kann, diese erhalten zu haben.

Die Übertragung findet über eine sichere Ende-zu-Ende Verbindung statt, womit eine ständige Erreichbarkeit des Empfängers über das Internet zwingend erforderlich ist. Dieses Verfahren unterscheidet sich grundsätzlich vom Mailbox-Prinzip, wie es bei VANS der Fall ist.

Durch die ständige Erreichbarkeit der Firma entstehen hohe Einrichtungskosten und aufwendige Firewall-Einstellungen. Außerdem ist die Installation und Einrichtung einer EDI-INT AS2 Software, zum Empfangen und Senden von Daten, erforderlich.

1.3. Zielsetzung

Es soll ein Anwendungssystem entstehen, das, analog zu einem VANS, die Verteilung von EDI-Dateien nach dem Mailboxing-Prinzip in einer Hub-and-Spoke Topologie durchführt. Im Gegensatz zum kommerziellen VANS soll eine leichtgewichtige, auf OpenSource Komponenten stützende Software implementiert werden. Das heißt eine Software, die so viel Funktionalität wie nur möglich von vorhandenen Softwarelösungen unter GPL Lizenz nutzt, statt sie eigen zu implementieren.

Folgende funktionale Eigenschaften sollen dabei erfüllt werden:

- EDI Datenübertragung über das Internet
Die Datenübertragung der EDI-Daten soll mit vorhandenen Internetstandards wie FTPS durchgeführt werden.
- Nachweisbarkeit der Zustellung mittels Track & Trace
Der gesamte Lebenszyklus einer EDI-Datei soll protokolliert werden, anhand von festgelegten Zeitpunkten, wie zum Beispiel die erfolgreiche Zustellung einer EDI-Datei.
- Direkter Zugriff auf Track & Trace-Informationen durch betreffende Handelspartner
Den Teilnehmern soll die Möglichkeit gegeben werden eigenständig den Status ihrer gesendeten und empfangenen EDI-Daten abzufragen. Um die Suche zu erleichtern sollen zusätzliche Suchfilter zur Verfügung stehen.
- Eigene GLN-Verwaltung
Den Teilnehmer soll ein direkter Zugriff für die Verwaltung eigener GLNs und Geschäftspartner ermöglicht werden.

Folgende Vorteile ergeben sich dadurch:

- Freie Wahl der Einzelkomponenten
Durch die Verwendung von Standards, kann oft auf vorhandene Komponenten zurückgegriffen werden, die bereits mehrere Jahre im produktiven Betrieb anderer Umfelder fehlerfrei absolviert haben. Eine durch die Praxis bewiesene Reife solcher Komponenten ist damit gewährleistet.
- Geringe absolute Betriebskosten (TCO) bestehend aus:
 - einmalige Anschaffungskosten

Lizenzgebühren für Software entfallen vollständig durch die Verwendung von lizenzfreier Software. Kosten entstehen lediglich für die Einrichtung eines dedizierten Serversystems und die Anwendungsinstallation.

o Laufende Kosten

Die laufenden Kosten beschränken sich auf Miete für dedizierte Serverhardware, Wartung und Verwaltung

▪ Verwaltung

Die Verwaltungskosten fallen sehr gering aus, da die Verwaltung der Benutzerdaten und aktiven Partnerschaften von den Handelspartnern eigenständig durchgeführt werden können. Lediglich der manuelle Prozess durch den Administrator bei einer Registrierung eines Teilnehmers erfordert Verwaltungsarbeiten.

▪ Wartung

Administrative Eingriffe zwecks Wartung sind für die Anwendung nicht nötig. Lediglich für Sicherheitsrelevante Updates des Serversystems sind anfallende Wartungsarbeiten denkbar.

Ferner ist zusätzlich der Fokus auf die ausschließliche Nutzung standardisierter Schnittstellen von hoher Bedeutung. Mit Hilfe von standardisierten Schnittstellen wird einem Unternehmen, das diese Anwendung verwenden möchte, der Zugang mit kostengünstiger Standard-Software ermöglicht.

Ziel dieser Vorsätze ist zum Einen eine schnelle Integration der hier neu entstandenen VANS Alternative in das bestehende Zielsystem und zum Anderen die Steigerung der Attraktivität für nicht EDI-Anwender, Geschäftspartner, die bereits EDI verwenden, auf selbigem Weg zu erreichen.

1.3.1. Besondere Merkmale

1.3.1.1. Verteilte Anwendung

Aus Gründen der Leichtgewichtigkeit soll die Anwendung aus mehreren Teilen bestehen. Dadurch wird Austauschbarkeit bzw. Erweiterbarkeit der Einzelteile zum Einen und resultierende Skalierbarkeit zum Anderen angestrebt. Folgende Aufteilung ist zu verwenden:

- Verteiler Anwendung, der sogenannte EDI-Hub-Server
- Daten Ein-/Ausgang
- Steuerungs- und Verwaltungswerkzeuge

1.3.1.2. Sicherheit

Die angestrebte Verteiler-Anwendung ist für die Bearbeitung geschäftssensibler Daten vorgesehen. Nicht ungewöhnlich wäre auch die Abwicklung der Kernkompetenzen des Kunden mit Derselbigen. Aus diesen Gründen ist beim Umgang mit den Nutzdaten (Geschäftsdaten des Kunden) auf hohe Sicherheit zu achten. Nachfolgend sind Sicherheitskriterien sog. IT Schutzziele [Lit06] im Einzelnen aufgeführt:

- **Datenintegrität**
Nutzdaten dürfen während dem gesamten Vorgang des automatisierten Datenaustausches nicht unbefugt und unbemerkt modifiziert werden.
Priorität: Hoch
- **Vertraulichkeit der Informationen**
Nutzdaten dürfen nicht von unautorisierten Nutzern eingesehen werden.
Priorität: Hoch
- **Verfügbarkeit**
Das System sollte widerstandsfähig gegenüber vorsätzliche Manipulationsversuchen oder anderer Fehlbedienung der Schnittstelle sein.
Priorität: Mittel
- **Verbindlichkeit**
Unabstreitbare Nachweisbarkeit von durchgeführten Handlungen einzelner Nutzer, z.B. getätigter Datenaustausch oder Änderungen an Benutzerinformationen

Priorität: Gering

- Authentizität

Glaubwürdiger Identitätsnachweis aller Akteure.

Priorität: Mittel

- Privatheit

Ausschließliche Verwendung von unbedingt benötigten personenbezogenen Daten, im Sinne des informellen Selbstbestimmungsrechtes.

Priorität: Nicht relevant für Zielanwendung

2. Standards und Werkzeuge

2.1. *Entwicklungsumgebung*

Durch die Verwendung der Interpretersprache Ruby sind der EDI-Hub-Server und die Webanwendung auf jeder Plattform, für die Ruby verfügbar ist, einsetzbar. Entwickelt wurde der EDI-Hub-Server unter der Linux Distribution Debian 3.1 Sarge mit Kernelversion 2.6.8-1-686. Für die Entwicklung und Tests wurde eine Pentium 2 CPU mit 500 MHz und 256 MB Arbeitsspeicher verwendet. Der Server lief im Serverbetrieb, ohne grafische Oberfläche und wurde mittels `ssh` bedient.

Ruby in der Version 1.8.2 und die benötigte Datenbank-Anbindungsbibliothek `libmysql-ruby` in der Version 2.4.5 wurden für die Implementierung des EDI-Hub-Servers eingesetzt. Für die Webanwendung wurde das Web-Framework Ruby on Rails in der Version 1.1.6 verwendet. Der Webserver Apache, in der Version 2.0.54, wurde um das `libapache2-mod-fastcgi` Modul erweitert, das für eine schneller Ausführung der Webanwendung sorgt. Die Datenhaltung der EDI-Hub-Teilnehmer wurde von MySQL, der für Webanwendungen sehr verbreitete Datenbank, in der Version 4.0.24 übernommen. Der Datenaustausch der EDI-Daten über FTPS wurde mit dem FTP-Server Proftpd in der Version 1.2.10 und der MySQL-Erweiterung für die MySQL-Datenbank gestützte Authentifizierung realisiert.

Alle verwendeten Programme und Komponenten stehen bereits vorkompiliert als Paket zum Herunterladen und Installieren auf den Seiten des Distributors zur Verfügung. Die Installation kann dadurch sehr einfach und unkompliziert durchgeführt werden. Alle notwendigen Abhängigkeiten werden selbstständig aufgelöst und die nicht auf dem System installierten Pakete auf Wunsch installiert. Eine paketorientierte Installation erspart Administratoren das manuelle Kompilieren der einzelnen Programme und Komponenten, und damit sehr viel Zeit. Lediglich die Ruby Zusatzbibliotheken, das Webdevelopment Framework Ruby on Rails und die dafür notwendigen Abhängigkeiten werden von der paketorientierten Installation ausgespart und über das Ruby eigene Paketverwaltungsprogramm `RubyGems` installiert.

Der wesentliche Nachteil der paketorientierten Installation ist, dass unter Umständen Programme nicht in der neusten Version verfügbar sind. Bei den Paketen, die von Debian zur Verfügung gestellt werden, ist die Wahrscheinlichkeit

der Inaktualität höher als bei anderen Distributoren, da Debian sein Focus auf Stabilität und nicht auf Aktualität setzt. Die Zeiträume zwischen Folgeversionen fallen daher größer aus.

Die Implementierung wurde mit der integrierten Entwicklungsumgebung Radrails durchgeführt (Abb. 2). Radrails ist eine sehr empfehlenswerte IDE¹ zur Entwicklung von Ruby on Rails Webanwendungen und Ruby Programmen. Radrails wurde aus technischer Sicht als Erweiterung für Eclipse, eine modular aufgebaute interaktive Entwicklungsumgebung für Java, realisiert, das als eigenes Programm ausgekoppelt wurde. Durch die Tatsache, dass Radrails in Java programmiert wurde, ist es Plattform übergreifend einsetzbar. Die verwendete Versionsnummer 0.7.2 bietet folgende Funktionen:

- Texteditor
- Debugger
- Quelltextformatierungsfunktionen
- Direkter Datenbankzugriff
- Starten/stoppen des Webservers
- Konsole für Debugausgaben

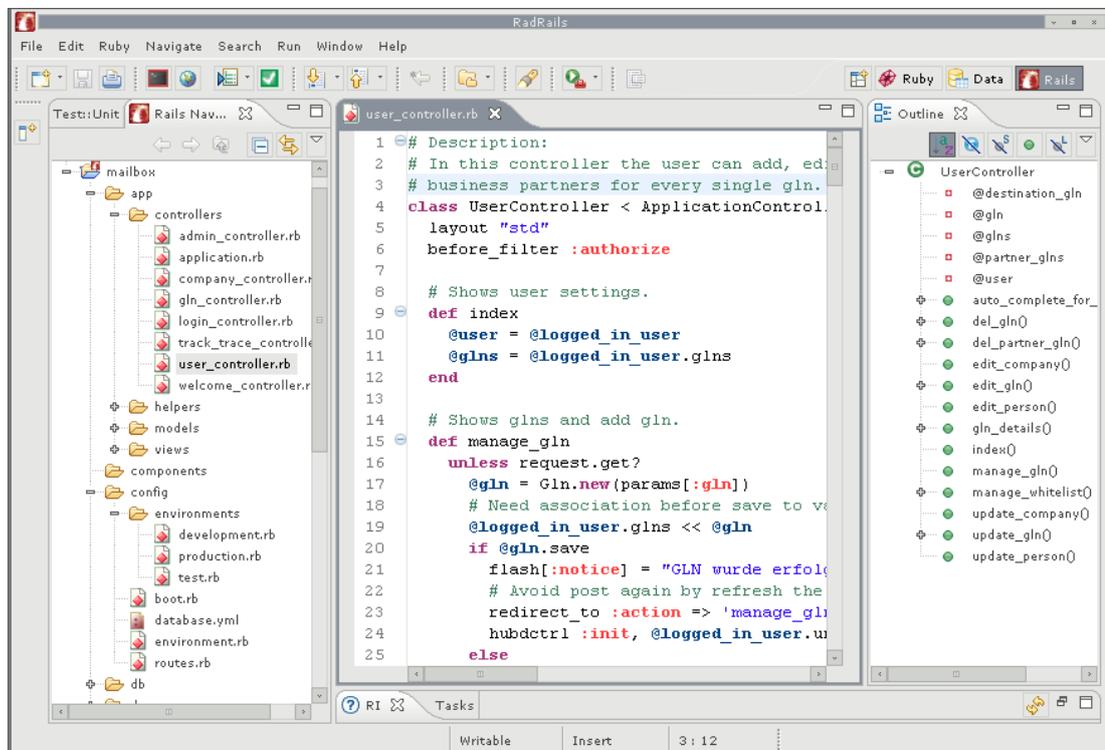


Abb. 2: Ruby on Rails Entwicklungsumgebung Radrails.

¹ Eine integrierte Entwicklungsumgebung, s. <http://www.radrails.org/>

Um alle Funktionen von Radrails verwenden zu können müssen Ruby, Ruby on Rails und MYSQL auf dem Klienten installiert sein. Aber auch als reiner Quelltexteditor mit Projektverwaltung ist Radrails sehr gut einsetzbar. So ist es möglich RadRails eingeschränkt zu benutzen, wenn die Webanwendung als auch der EDI-Hub-Server auf einem entfernten Rechner, wie zum Beispiel auf einem Linux-Server, liegen und die Verzeichnisse mit Samba freigeben werden. Das Ausführen und Testen der Anwendung muss dann über die Konsole auf dem Linux-Server durchgeführt werden.

2.2. Verwendete Standards

Für den Austausch von UN/EDIFACT-Daten wurde auf bestehende Internetprotokolle zurückgegriffen, um für die Anwender den Aufwand so gering wie möglich zu halten. So findet die Benutzerselbstverwaltung über das Protokoll HTTPS statt, um die sensiblen Daten, wie die Geschäftspartner oder die Track & Trace-Daten, zu verschlüsseln. Hier steht es dem EDI-Hub-Betreiber frei lediglich die Registrierung und Passwortänderung zu verschlüsseln, um Performance zu sparen. Es ist aber nicht empfehlenswert, da es sich um sensible Daten handelt und die Performanceeinbußen nicht ins Gewicht fallen würden. Die Frequentierung des Webinterfaces und die Übertragene Datenmenge werden sehr gering ausfallen. Hauptsächlich werden die Anwender das Track & Trace verwenden, um eventuell fehlgeschlagene Zustellungen zu erfragen oder zur Kontrolle einer erfolgreichen Zustellung. Eine Änderung der Einstellungen oder Geschäftspartner wird voraussichtlich nach der ersten Konfiguration nicht so schnell vorgenommen.

2.2.1. HTTPS

HTTPS ist der Standard für die verschlüsselte Übertragung von Daten zwischen Webserver und Klienten. Für die Verschlüsselung der Verbindung wird ein SSL-Zertifikat benötigt. Mit Hilfe der verschlüsselten Verbindung sind die sensiblen Daten vor Unbefugten geschützt und können nicht mitgelesen werden. Um eine absolute Sicherheit zu garantieren wird ein SSL-Zertifikat benötigt, das von einer Zertifizierungsstelle ausgestellt wurde um die Authentizität der Domain zu garantieren. Durch die Sicherstellung der Authentizität entfällt die Gefahr vor einem Man-In-The-Middle Angriff, da das Abfangen der Verbindung durch Dritte entdeckt werden würde.

2.2.2. FTPS

FTPS ist eine SSL/TLS-Erweiterung für das File Transfer Protocol FTP. Es werden sowohl der Datenkanal als auch der Kontrollkanal mit einem SSL-Zertifikat verschlüsselt. Um Performance zu steigern, könnte lediglich der Kontrollkanal verschlüsselt werden. Dadurch würden Benutzername und Passwort weiterhin nicht im Klartext ausgetauscht werden. Die Übertragung der Daten über den Datenkanal fände allerdings unverschlüsselt statt und wäre für Dritte sicht- beziehungsweise mitlesbar. Da die Verschlüsselung von großen Datenmengen viel Rechenleistung erfordert, würde eine Beschleunigung der Übertragung erzielt werden. Davon wird

allerdings abgeraten, da es sich in der Regel um sensible Daten, wie zum Beispiel Rechnungen oder Bestellungen, handelt.

Neben FTPS existiert SFTP, das ein SSH-Subsystem mit einem FTP ähnlichen Service ist und auf das SSH-Protokoll aufsetzt.

Allerdings besitzt SFTP einige Nachteile [Lit07]:

- SSH/SFTP bietet keine eingebaute Funktionalität zur chroot-Umgebung
- Manuell erstellte chroot-Umgebungen sind nur umständlich und mit viel Aufwand zu pflegen
- Manuelle chroot -Umgebungen benötigen viel Plattenplatz
- Eine Authentifikation über SQL oder LDAP benötigt ein zusätzliches PAM-Modul und ist damit Systemabhängig

2.3. **Verwendete Techniken**

Der EDI-Hub-Server verwendet verschieden Techniken für die Verwaltung der Teilnehmer und die Zustellung der EDI-Daten.

2.3.1. **Mailbox-Prinzip**

Das Mailbox-Prinzip beschreibt, dass Nachrichten jeglicher Art in elektronische Postfächer abgelegt werden, aus denen sie - im sogenannten store-and-forward-Verfahren - wieder entnommen und weiterverarbeitet werden können. Dies verschafft räumliche und zeitliche Freiheiten und Vorteile.

2.3.2. **Hub & Spoke**

Hub & Spoke beziehungsweise Nabe-Speiche beschreibt die Form der Integration von Endknoten. Alle Endknoten sind mit einem Zentralknoten verbunden und kommunizieren ausschließlich über diesen. Der Zentralknoten wird Hub (Nabe) und die Verbindung zu den einzelnen Endknoten wird Spoke (Speiche) genannt. Möchte der Endknoten A mit dem Endknoten B in Verbindung treten, so geschieht das stets über den Zentralknoten Z. (Abb. 3)

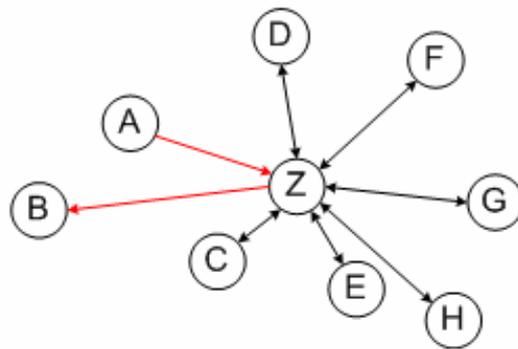


Abb. 3: Hub & Spoke Netzwerktopologie.

2.4. Ruby

Sowohl der EDI-Hub-Server als auch die Webanwendung sind in Ruby geschrieben und können damit auf eine Vielzahl von Plattformen eingesetzt werden.

Ruby ist eine Interpretersprache und wurde von 1993 bis 1995 von Yukihiro „Matz“ Matsumoto in Japan entworfen und wird heute als OpenSource-Projekt weiter gepflegt. In Japan ist die Programmiersprache sehr bekannt und weit verbreitet, wie die Programmiersprachen Python oder Perl im Westen. Erst nach dem Jahr 2000 wurde Ruby auch in Europa und Amerika durch erste nicht japanische Literatur bekannt. Mittlerweile ist Ruby für die meisten Plattformen erhältlich.

Ruby ist eine interpretierte, objektorientierte Programmiersprache, die einige Vorteile anderer Programmiersprachen in sich vereint. Sie besitzt eine einfach zu lesende Syntax, die sich intuitiv nutzen lässt.

2.4.1. RubyGems

RubyGems ist, ähnlich wie apt-get unter Debian, ein Paketmanager. RubyGems bietet eine einfache Möglichkeit Ruby-Programme oder Bibliotheken mit allen ihren Abhängigkeiten zu verwalten. Neben den Abhängigkeiten werden auch Konflikte zwischen Packten überwacht. Die, in Pakete gepackten, Ruby Programme beziehungsweise Bibliotheken werden Gems genannt. RubyGems erleichtert die Installation für den Anwender erheblich, da die Installation automatisiert stattfindet. Außerdem können die Programme schnell und einfach aktualisiert werden. Neben dem Installieren kann RubyGems installierte Pakete auch wieder restlos vom System entfernen. Den größten Nutzen bringt RubyGems erst bei der Installation von Programmpaketen über das Internet. Die aus dem Internet bezogenen Pakete werden von Rubyforge² verwaltet.

2.4.2. RDoc und ri

RDoc ist ein Software-Dokumentationswerkzeug, vergleichbar mit Javadoc für Java. Dabei verwendet RDoc die im Programmcode stehenden Kommentare, die mit simple markup³ formatiert werden können. Dabei extrahiert RDoc Klassendefinitionen, Moduldefinitionen und Funktionsdefinitionen. Zum Generieren

² Ein Software-Entwicklungsmanagementsystem speziell für Ruby, s. <http://rubyforge.org/>

³ Definiert die Formatierung von lesbaren Texten.

der Dokumentation sollen zukünftig verschiedene Ausgabeformate genutzt werden können. Bisher wird lediglich HTML als Ausgabeformat angeboten.

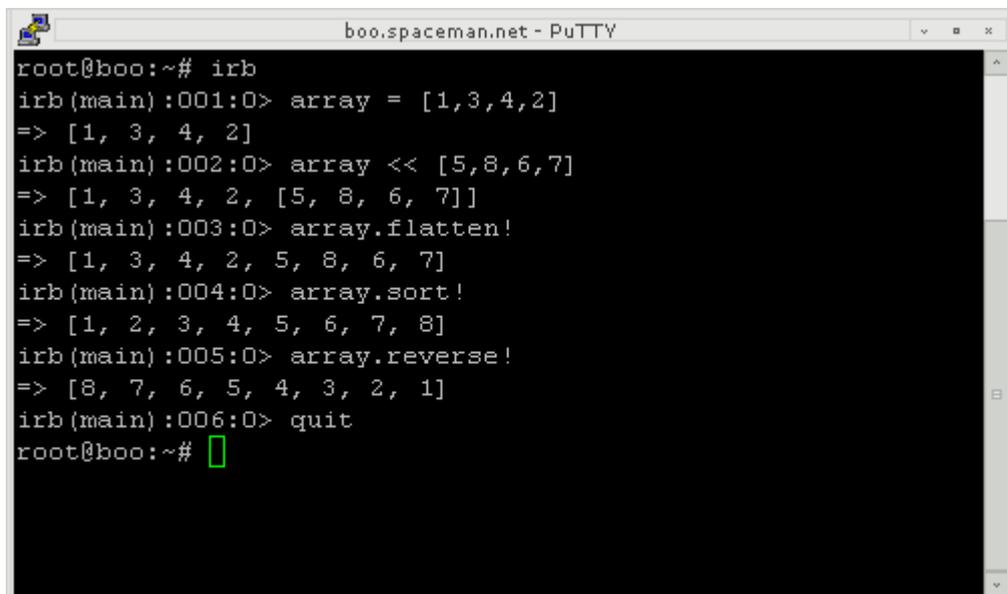
Neben dem Generieren von Dokumentationen im HTML-Format werden auch interne Datenbanken aufgebaut, die mit dem Tool `ri` durchsucht werden können. Wenn zum Beispiel Informationen über die Klasse `String` benötigt werden, genügt folgender Aufruf:

```
ri String
```

Sowohl `RDoc` als auch `ri` werden zusammen mit dem Ruby-Interpreter ausgeliefert.

2.4.3. Die interaktive Ruby-Shell

Die `irb` ermöglicht es, Ruby Code auszuführen und dessen Verhalten zu beobachten und analysieren. `irb` steht für Interactive Ruby, zu deutsch interaktive Ruby-Shell. Nach dem Starten der interaktiven Ruby-Shell gelangt man zu einem Zeileninterpreter, der sich ähnlich wie eine Shell verhält und Ruby-Code Zeile für Zeile interpretiert (Abb. 4). Die Rückgabewerte der einzelnen Befehle werden sofort sichtbar. Die interaktive Ruby-Shell ist zum Analysieren und Testen von Funktionen und kleineren Funktionsblöcken sehr hilfreich und erspart das Anlegen einer Datei.



```
root@boo:~# irb
irb(main):001:0> array = [1,3,4,2]
=> [1, 3, 4, 2]
irb(main):002:0> array << [5,8,6,7]
=> [1, 3, 4, 2, [5, 8, 6, 7]]
irb(main):003:0> array.flatten!
=> [1, 3, 4, 2, 5, 8, 6, 7]
irb(main):004:0> array.sort!
=> [1, 2, 3, 4, 5, 6, 7, 8]
irb(main):005:0> array.reverse!
=> [8, 7, 6, 5, 4, 3, 2, 1]
irb(main):006:0> quit
root@boo:~#
```

Abb. 4: Interaktive Ruby-Shell im Einsatz.

2.5. *Ruby on Rails*

Ruby on Rails [Lit08] ist ein Webdevelopment Framework, das auch kurz Rails genannt wird. Es ist komplett in Ruby geschrieben, basiert auf dem MVC⁴-Schema und wurde für die Entwicklung der Webanwendung verwendet.

1979 wurde MVC für das Entwickeln von interaktiven Programmen konzipiert. Dabei teilt MVC die Anwendung in drei Komponenten: Model, View und Controller

Das Model ist im Allgemeinen für die Modellierung der Geschäfts- bzw. Anwendungslogik zuständig. Im Fall der entwickelten Webanwendung, besteht die Anwendungslogik im Abstrakten darin, verwaltungstechnische Daten einzulesen, ggf. zu verändern und schließlich auf die Festplatte zurückzuschreiben. Dabei können diverse automatische Prüfungen zwischen den einzelnen Phasen dieser Logik durchgeführt werden. Als Beispiele seien Folgende erwähnt:

- Validierung von Eingabewerten nach Plausibilität und numerischen Schranken
- Validierung von Abhängigkeiten zwischen Daten, z.B. $A = 2/3 B$ muss stets erfüllt bleiben, auch nach Änderung von A oder B
- Validierung der Änderungsrechten
- Validierung der GLN-Nummer durch Überprüfung der Checksumme.

Auch legt das Model Speichertyp und Speicherort fest. Als Speichertyp kommen üblicher Weise nur Datenbanken in Frage, da diese in performanter Art mit größeren Datenmengen umgehen können. Der Speicherort bestimmt – im Fall der Datenbank – die Zugangsdaten für den Zieldatenbank-Server.

Die View regelt die Darstellung der Anwendung. Modellerte Anwendungsdaten werden hierfür vom Model zur Verfügung gestellt und von der View aktiv entgegengenommen. Üblicher Weise wird eine View von Extern aufgerufen und antwortet sozusagen nur auf Abfrage. Im GUI Umfeld ist das GUI-Framework der Auslöser für eine View-Aktion, im Webumfeld hingegen ist es der Browser bzw. der Webserver, der aktive Kommandos an die View übermittelt.

Die dritte Komponente, der Controller, steuert den Eingriff in die Anwendung und stellt praktisch eine Schnittstelle nach außen dar, für das ansonsten geschlossene

⁴ *Model View Controller Pattern*

System Model-View. Benutzereingaben und Befehle werden vom Controller entgegengenommen, interpretiert und durchgeführt. Für diese Aufgabe ist eine Interaktion mit View und Model unabdinglich, da der Controller weder die Anwendungslogik, noch die Darstellung kennt.

Ruby on Rails wurde entwickelt, um dem Entwickler die Erstellung einer interaktiven datenbankgestützten Webanwendung zu erleichtern. Es basiert auf dem Prinzip „Don't repeat Yourself“ und verfolgt das Ziel mit möglichst wenig manuell geschriebenen Quellcode viel Funktionalität für die Internetseite zur Verfügung zu stellen. Mit Rails ist es möglich Anforderungen rasch umzusetzen. Es unterstützt damit die agile Softwareentwicklung.

2.5.1. Die Verzeichnisstruktur

Das Webframework Ruby on Rails [Lit08] gibt eine einheitliche Verzeichnisstruktur vor (Abb. 5):

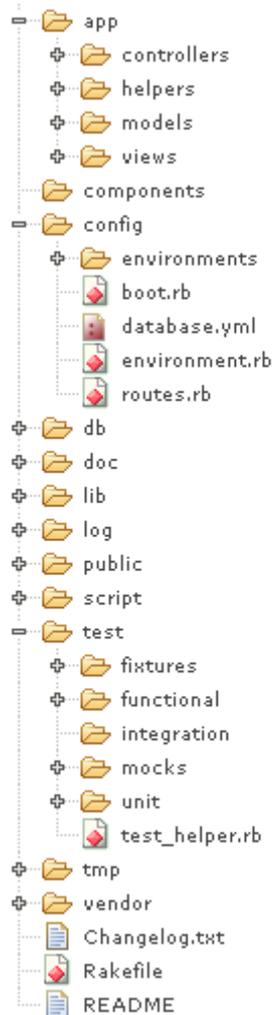


Abb. 5: Verzeichnisstruktur einer Rails-Anwendung.

Nachfolgend eine Beschreibung der wichtigsten Verzeichnissen und deren sematischen Funktionen:

- `app`
In `app` befindet sich die Anwendung inklusive den Controllern, Views und Models aus dem oben geschilderten MVC-Schema. Zudem befinden sich hier auch die sogenannten Helper-Funktionen, die den Views zusätzliche

Funktionen zur Verfügung stellen, damit diese, durch das Ausgliedern der Funktionen, übersichtlich bleiben.

- `config`

In `config` befinden sich die Konfigurationsdateien von Ruby on Rails. Die Datei `environment.rb` definiert verschiedene Umgebungsvariablen wie zum Beispiel die aktuell verwendete Umgebung oder Einstellungen für den ActionMailer. In `database.yml` werden die Datenbankeinstellungen für die Umgebungen `development`, `test` und `production` definiert. Die einzelnen Umgebungen werden spezialisiert in den Dateien `development.rb`, `test.rb` und `production.rb`, die sich im Ordner `environments` befinden. Durch diese Aufteilung in unterschiedlichen von einander entkoppelten Umgebungen, kann die Anwendung schnell in ein anderes Einstellungsprofil gebracht werden. So wird zum Beispiel beim Testen der Anwendung die Test-Umgebung verwendet, um unter anderem das Verschicken von E-Mails zu unterdrücken.

- `public`

Die statischen Seiten, sowie die Scripte und Bilder, befinden sich im Ordner `public`, der auf dem Webserver als Documentroot eingetragen wird und somit von außen erreichbar ist. Zusätzlich befindet sich dort auch die Datei `.htaccess` mit Apache Optionen zum Umschreiben der HTTP Requests für Ruby on Rails.

- `test`

Der Ordner `test` beinhaltet alle Tests die für Ruby on Rails verwendet werden. Die Tests werden in Ruby on Rails für die Models, Controllers und Mailer unterteilt. Die Unit-Tests für die Models testen die Funktionen eines Models wie zum Beispiel das Anlegen, Löschen oder Editieren und befinden sich im Ordner `units`. Die Funktions-Tests testen die Funktionalität der Controller und befinden sich mit den Tests für den ActionMailer im Ordner `functional`.

2.6. AJAX

AJAX ist kein Standard im klassischen Sinne sondern beschreibt vielmehr die Verwendung von verschiedenen Technologien wie (X)HTML, CSS, Javascript, DOM, XML, XSLT und das XMLHttpRequest Objekt für die Erstellung einer dynamischen, eine für den Benutzer interaktionsfreudigen, Webseite. AJAX ermöglicht es einer Webanwendungen schnell und schrittweise Änderungen an der Webseite vorzunehmen. Bei Änderungen ist es nicht notwendig die gesamte Seite erneut zu laden, es genügt den Teil der Webseite mit den veränderten Daten zu erneuern. Dadurch kann die Webanwendung schneller auf Benutzereingaben reagieren und durch die dynamische Gestaltung der Seite wirkt sie wie eine normale Rich-Client-Anwendung.

Das Framework Ruby on Rails unterstützt die Verwendung von AJAX Elementen und bietet bereits einen breiten Einsatzbereich mit Hilfe von hauseigenen Funktionen aus den Bibliotheken Prototype und Scriptaculous. Neben AJAX-Methoden werden auch visuelle Effekte, Drag & Drop-Integration und neue Controls angeboten.

Die Bibliotheken gewährleisten einen reibungslosen Zugriff von Ruby on Rails heraus und bieten dem Programmierer alle notwendigen Methoden. Damit wird die Entwicklung einer von AJAX gestützte Webanwendung erleichtert, jedoch auch in ihrer Flexibilität eingeschränkt.

Um Dynamische Webseiten zu erzeugen ist es nicht notwendig die Funktionsweise von AJAX zu kennen, da Ruby on Rails die Aufrufe dazu direkt erstellt.

Einige mögliche AJAX Funktionen, die Rails zur Verfügung stellt, sind:

- Autocomplete
Eingabefeldern werden mit automatischer Vervollständigung ausstatten.
- Autocomplete mit Template
Die automatische Vervollständigung von Eingabefeldern unter Verwendung von Template für eine angepasste Darstellung mit eventuellen Zusatzinformationen.
- Effekte
Diverse Effekte zum ein- /ausblenden von HTML-Elementen.
- Observe

Periodisches auswerten von einzelnen Formularelementen oder ganzen Formularen und Aktualisierung von HTML-Elementen.

- link_to_remote

Setzt eine AJAX-Anfrage ab und füllt ein HTML-Element mit der Antwort.

[Lit09]

3. Das Design der Anwendung

Das EDI-Hub-System ist ein Verbund aus miteinander arbeitenden Programmen. Jedes Programm läuft eigenständig und ist für seine eigenen Aufgaben zuständig, sowie für die einwandfreie Funktion des EDI-Hub-Systems. (Abb. 6)

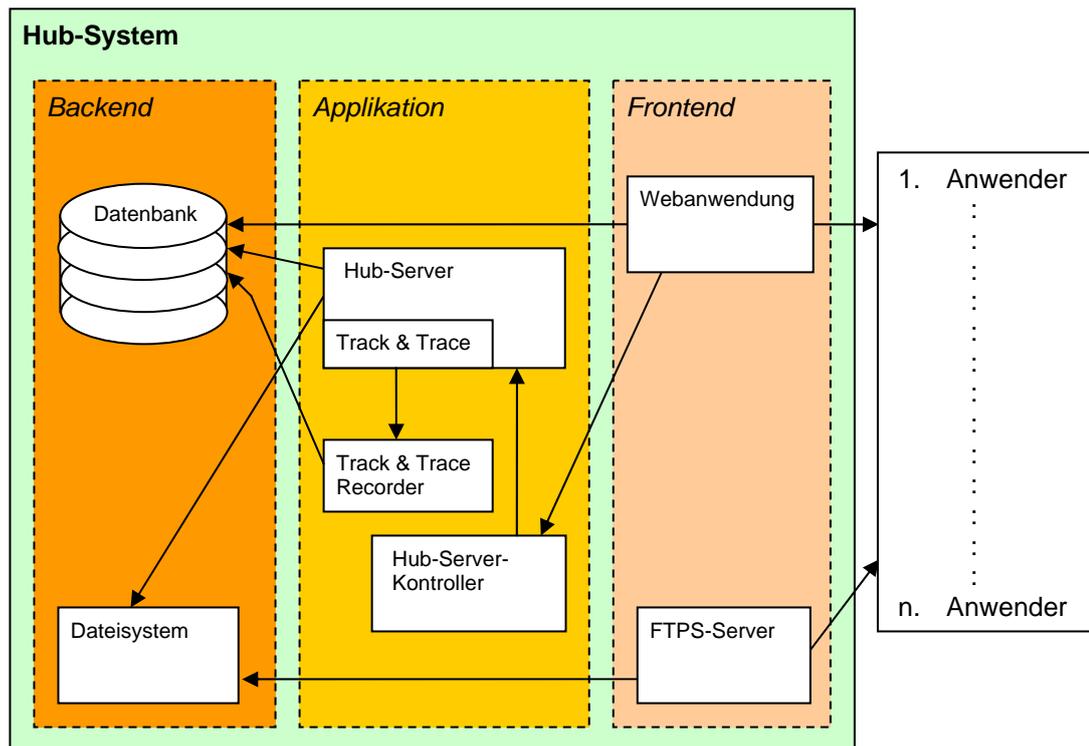


Abb. 6: Aufbau des EDI-Hub-Systems.

Die Hauptaufgabe im EDI-Hub-System übernimmt der EDI-Hub-Server. Dieser ist für die Zustellung der neu eingetroffenen Daten zuständig, stößt Überprüfungen an und protokolliert alle seine Aktivitäten in die Track & Trace-Datenbank. Damit der EDI-Hub-Server während dem regulären Betrieb auch ohne die Datenbank arbeiten kann, wird ein ständiger Datenbankzugriff vermieden. Datenbankzugriffe seitens des EDI-Hub-Servers sind nur beim Starten erforderlich, dann werden die vorhandenen Teilnehmer inklusive deren Informationen geladen und intern gespeichert. Auch die Track & Trace-Daten werden nicht direkt in die Datenbank geschrieben, sondern von einem eigenständigen Programm übernommen, dem Track & Trace-Recorder. Durch die Trennung wird der Betrieb des EDI-Hub-Servers gewährleistet, sollte die Datenbank zeitweise nicht erreichbar sein.

Die Mailbox-Verzeichnisse der Teilnehmer befinden sich im Dateisystem und müssen jedem Teilnehmer eindeutig zugeordnet werden. Um eine Einschränkung der Benutzernamen durch das Dateisystem zu vermeiden, werden eindeutige Namen verwendet, die nicht mit der Namensgebung aus der Webanwendung übereinstimmen.

Das EDI-Hub-System bietet für die Teilnehmer zwei Schnittstellen an. Eine ist die Webanwendung zum Konfigurieren der GLNs und Geschäftspartnern und die Andere ist der FTPS-Server für den Austausch von EDI-Daten.

Für die Bedienung der Webanwendung wird ein Browser, im optimalen Fall mit Javaunterstützung, benötigt. Die Webanwendung erlaubt es den Teilnehmern sich am EDI-Hub-System zu registrieren, sich anzumelden, eigene GLNs und deren Geschäftspartner zu verwalten und das Track & Trace zu benutzen.

Änderungen der Teilnehmer oder ihrer GLNs, die für den Betrieb des EDI-Hub-Servers relevant sind, müssen übernommen werden. Dazu werden die Einstellungen der betroffenen Teilnehmer vom EDI-Hub-Server gezielt neu geladen.

Über den FTPS-Server laden die Teilnehmer Daten vom EDI-Hub-System herunter oder hoch. Die Teilnehmer erhalten ausschließlich Zugriff auf ihre Mailbox und sind nicht in der Lage die Mailbox zu wechseln.

3.1. Server

Für das EDI-Hub-System inklusive Webinterface ist ein so genannter „managed root server“ mit dem Betriebssystem Linux ausreichend.

Ein „managed root server“ steht ein 24 Stunden Support zur Seite, wird in einigen Fällen durch den Anbieter in Betrieb gehalten und bietet eine Hardwareüberwachung an. Damit kann eine sehr geringe Ausfallrate gewährleistet werden.

Der benötigte Festplatten-Platz hängt sehr vom Datenaufkommen und der Anzahl der Mitglieder ab. Bei einer Mitgliederanzahl < 100 und kleinen Datenvolumina, wie Bestellungen oder Rechnungen nach EANCOM, sollte praktisch jedes Root-Server-Angebot ausreichend sein. Sollte jedoch die Mitgliederzahl oder das Datenvolumen steigen, wächst der Festplattenbedarf linear an. Als zusätzlicher Faktor ist noch die Verweildauer der EDI-Dateien in den Mailboxverzeichnissen zu nennen.

Mit folgender Formel kann der benötigte Festplatten-Platz ermittelt werden:

$$c = m * d * s * t$$

m	Anzahl der Teilnehmer
d	Anzahl der Dateien die von jedem Teilnehmer durchschnittlich pro Tag verschickt werden
s	durchschnittliche Größe der Dateien
t	die durchschnittliche Verweildauer der Dateien in Tagen

Beispiel für eine geringe Teilnehmerzahl und geringem Datenvolumen:

100 Teilnehmer (m=100) verschicken täglich 20 Dateien (d=20) mit einer durchschnittlichen Größe von 100 KB (s=100KB). Die Dateien verweilen durchschnittlich 5 Tage (t=5) in der Mailbox.

$$c = 100 * 20 * 100KB * 5$$

Dann wären 977 MB Festplattenplatz notwendig.

Wenn sich dagegen die Teilnehmerzahl erhöht und das Datenvolumen steigt, z.B. m = 1000 und s = 20 MB:

$$c = 1000 * 20 * 20MB * 5$$

Dann steigt der erforderliche Festplatten-Platz auf ca. c = 1,9 TB.

Anhand dieser Rechnung ist leicht ersichtlich, dass der Festplattenplatz zur Einführung kein Problem darstellen sollte. Sobald jedoch die Teilnehmerzahl steigt dominiert der Festplattenplatz die entstehenden Kosten.

3.2. Benutzerkonto

Das Konto wird vom Webinterface erstellt und in die Datenbank geschrieben. Um dem Benutzerkonto ein eindeutiges Mailbox-Verzeichnis zuzuordnen erhält das Konto einen eindeutigen Namen, der mit dem Teilnehmer assoziiert wird. Im Mailbox-Verzeichnis eines jeden Benutzers befinden sich folgende Unterverzeichnisse:

- in Hier werden die zugestellten Dateien von Geschäftspartnern gespeichert.
- out Alle in diesem Verzeichnis befindlichen Dateien werden vom EDI-Hub-Server zugestellt.
- err Dateien die vom EDI-Hub-Server nicht zugestellt werden konnten, werden in dieses Verzeichnis verschoben.

3.3. Registrierung

Die Teilnehmer können sich selbstständig über das Webinterface am EDI-Hub-System registrieren. Während der Registrierung müssen folgende Schritte durchgeführt werden:

- E-Mail-Überprüfung
- Akzeptierung / Ablehnung durch den Administrator
- Konto-Aktivierung

3.4. Teilnehmer selbstverwaltung

Die Teilnehmer haben die Möglichkeit ihre Einstellungen selbstständig mit Hilfe der Webanwendung durchführen. So können die Teilnehmer unter anderem weitere GLNs hinzufügen und deren Geschäftspartner auswählen. Dabei können Sie für jede GLN die Geschäftspartner entweder aus einer Liste mit den registrierten GLNs auswählen oder gezielt durch die Eingabe einer GLN-Nummer bestimmen.

3.5. Datenaustausch

Mitglieder können sich über FTPS am EDI-Hub-System mit ihren Login-Daten, die sie sich bei der Registrierung ausgesucht haben, anmelden. Die Login-Daten für die Webanwendung als auch FTPS sind identisch. Nach erfolgreicher Anmeldung befinden sich die Mitglieder in ihrem Mailbox-Verzeichnis und können dieses nicht verlassen.

3.5.1. Senden von Daten

Das Versenden von Dateien an andere Mitglieder erfolgt mit dem Hochladen der Dateien in das „out“ Verzeichnis. Alle Dateien die sich im „out“ Verzeichnis befinden und die Endung „edi“ besitzen werden vom EDI-Hub-Server zugestellt. Der EDI-Hub-Server berücksichtigt, beim Scannen der Mailbox-Verzeichnisse, nur EDI-Dateien, die vollständig hochgeladen wurden. Dafür sorgt der FTPS-Server in dem

Dateien während des Uploads eine temporäre Namensänderung erhalten. Erst nach Beendigung des Hochladens erhalten die Dateien ihren ursprünglichen Namen.

Die Namen der Dateien sind bis auf die Endung frei wählbar, es ist aber eine eindeutige Namensgebung zu bevorzugen, um das Überschreiben von bereits vorhandenen Dateien zu vermeiden.

3.5.2. Zustellen der Dateien

Der EDI-Hub-Server scannt periodisch – in der Konfigurationsdatei frei wählbare Zeit – die Mailbox-Verzeichnisse der Mitglieder und versucht alle EDI-Dateien zu verschicken, die sich im „out“-Verzeichnis befinden. Die Informationen über Sender- und Empfänger-GLN werden aus der EDI-Datei extrahiert. Wenn die EDI-Datei alle Kriterien, wie zum Beispiel korrekte Absender GLN, erfüllt, wird sie in das „in“-Verzeichnis des Empfängers verschoben.

Der Empfänger erhält keine E-Mail-Benachrichtigung über das Eintreffen der Datei, vielmehr sollte das „in“-Verzeichnis von den Teilnehmern regelmäßig auf Vorhandensein neuer EDI-Dateien kontrolliert werden.

3.5.3. Empfangen von Daten

Das Empfangen von Dateien erfolgt mit dem Herunterladen der in den „in“-Verzeichnis befindlichen Dateien über FTPS. Nachdem Herunterladen können diese Dateien gelöscht werden, worauf sie vom EDI-Hub-Server als „received“ (abgeholt) protokolliert werden.

3.6. Track & Trace

Die vom EDI-Hub-Server erzeugten Track & Trace-Daten [Lit10] werden in der Datenbank gespeichert. Die Teilnehmer können über die Webanwendung ihre Track & Trace-Daten ansehen und flexibel recherchieren. Für die vereinfachte Suche können verschiedene Filter aktiviert werden. So können zum Beispiel nur die noch nicht abgeholten Dateien oder Daten spezieller Geschäftspartner angezeigt werden.

3.7. Entstehende Kosten

Da die Tätigkeiten des EDI-Hub-Betreibers durch die Selbstverwaltung der Handelspartner sehr gering ausfällt, reduzieren sich die Kosten auf die Mietkosten für den root - Server. Mit einem root-Server ohne Volumenbegrenzung fallen keine weiteren Kosten an.

3.8. Sicherheit

Der Datenaustausch findet ausschließlich über das verschlüsselte File Transfer Protokoll over SSL statt. Damit sind die Daten vor Unbefugten geschützt. Das Gleiche gilt auch für das Webinterface durch den Einsatz von HTTPS. Damit sind alle vertraulichen Informationen wie die Geschäftspartner und die Track & Trace-Daten geschützt.

Die Registrierung erfordert eine Überprüfung der E-Mail-Adresse, um eine Erreichbarkeit des Teilnehmers zu garantieren. Zusätzlich muss jeder neu registrierte Teilnehmer von dem Administrator akzeptiert werden, bevor dieser sein Konto aktivieren kann. Damit behält der EDI-Hub-Betreiber Überblick und Kontrolle über die neuen Teilnehmer am EDI-Hub-System.

Alle Teilnehmer sind in ihren Mailbox-Verzeichnissen grundsätzlich eingesperrt und können nicht in Mailbox-Verzeichnisse anderer Teilnehmer gelangen.

4. Implementierung

Das EDI-Hub-System läuft auf dem Betriebssystem Linux von Debian und ist ein Zusammenspiel von mehreren Programmen. Jedes für sich übernimmt eine Aufgabe.

Die Hauptaufgabe des EDI-Hub-Systems übernimmt der EDI-Hub-Server. Er ist für die Zustellung und die Protokollierung der EDI-Dateien zuständig. Die Schnittstelle für die Teilnehmer, zum Verwalten ihrer eigenen GLNs und Geschäftspartnern, bildet die Webanwendung. Beide Programme sind in Ruby geschrieben, wodurch eine schnelle und effiziente Implementierung möglich war.

Die übrigen Aufgaben, wie zum Beispiel das Hoch- und Runterladen von EDI-Dateien, werden von bereits vorhandenen Programmen durchgeführt. Das Hoch- und Runterladen von EDI-Daten wird von dem FTPS-Server proftpd, der unter anderem eine datenbankgestützte Authentifizierung unterstützt, übernommen. Die Benutzerdaten werden in einer MySQL-Datenbank gespeichert und die Webanwendung läuft auf dem Apache Webserver.

Das Parsen der EDI-Daten übernimmt die Ruby Bibliothek `edi4r` von Herrn Prof. Dr. Werntges⁵. Diese Bibliothek wird zum Parsen und Erstellen von EDI-Dateien verwendet und findet zum Beispiel im EDI-Hub-Server ihren Einsatz um Metadaten wie Sender-GLN, Empfänger-GLN, usw. zu extrahieren. Da mit dieser Bibliothek auch EDI-Dateien erstellt werden können, kann sie von den Teilnehmern als Voraussetzung für die Implementierung eines Konverters für das hausinterne EDV-System verwendet werden.

Durch die sehr guten Komprimierungseigenschaften der EDI-Dateien bietet es sich an diese alternativ in komprimierter Form zu verschicken, was der EDI-Hub-Server bereits unterstützt. Das dekomprimieren der EDI-Dateien wird von der Bibliothek `edi4r` durchgeführt und unterstützt die Formate `gz` und `bzip2`. Gleichgültig welches Komprimierungsformat verwendet wird, die Datei-Endung der EDI-Datei `edi` darf nicht geändert werden. Das verwendete Format wird automatisch erkannt.

Die notwendigen Teilnehmerinformationen werden in einer MySQL-Datenbank gespeichert, die alle notwendigen Daten über die Teilnehmer beinhaltet, die von

⁵ *Professor an der FH Wiesbaden Fachbereich Design Informatik Medien.*

verschiedenen Programmen benötigt werden wie der Webanwendung, dem EDI-Hub-Server und dem FTPS-Server für die Authentifizierung. Durch die Verwendung der DBI-Bibliothek können der EDI-Hub-Server, der Track & Trace-Recorder und die Webanwendung auch andere Datenbankprogramme als MySQL verwenden, wie zum Beispiel PostgreSQL⁶. Es muss nur darauf geachtet werden, dass auch Proftpd und eventuell zusätzliche verwendete Programme eine Schnittstelle für diese Datenbank besitzen.

Alle verwendeten Programme und Komponenten sind Lizenzkostenfrei und für den kommerziellen Einsatz ausdrücklich freigegeben. Damit fallen keine Lizenzkosten an wodurch die Kosten für das EDI-Hub-System nicht unnötig ansteigen.

Die Verwendung von verteilten Anwendungen macht das System sehr gut skalierbar.

⁶ *Eine objektrelationale Datenbank, s. <http://www.postgres.de/>*

4.1. EDI-Hub-Server

Der EDI-Hub-Server `hubd.rb` wird mit einem einfachen Aufruf gestartet. Nach dem Starten läuft das Programm als Dämon im Hintergrund und alle notwendigen Ausgaben werden in die Log-Datei geschrieben. Gesteuert werden kann der EDI-Hub-Server mit Hilfe des EDI-Hub-Kontrollers `hubdctrl.rb`.

Der EDI-Hub-Server wurde so entwickelt, dass er ohne die Datenbank arbeiten kann und keine ständigen Zugriffe darauf vornimmt. Datenbankzugriffe werden ausschließlich beim Starten, und bei Änderungen von Teilnehmerdaten durchgeführt. Damit eventuell vorgenommen Änderungen von den Teilnehmern nicht ignoriert werden, kann der EDI-Hub-Server zu einer Neu-Initialisierung eines oder aller Teilnehmern gebracht werden. Mehr dazu wird im Kapitel 4.2 EDI-Hub-Kontroller beschrieben.

Da der EDI-Hub-Server während des Betriebs keinen Zugriff auf die Datenbank hat, werden die Track & Trace Daten in eine Datei geschrieben und werden regelmäßig vom Track & Trace-Recorder in die Datenbank geschrieben, siehe Kapitel 4.3 Track & Trace-Recorder.

Der EDI-Hub-Server holt sich die notwendigen Datenbank- und Programm-Konfigurationen aus der Konfigurationsdatei `hubd-conf.yml`. Darüber hinaus können einzelne Parameter beim Programmstart übergeben werden. Die Konfigurationsdatei wird unter folgendem Pfad erwartet: `/usr/local/etc/hubd-conf.yml`. Mit der Option `-c` kann ein alternativer Pfad angegeben werden.

Alle verfügbaren Parameter inklusive Beschreibung werden mit dem Aufruf `hubd.rb -h` angezeigt.

Die Konfigurationsdatei `hubd-conf.yml` wird nicht nur vom den EDI-Hub-Server, sondern auch vom Track & Trace Recorder verwendet. Hauptsächlich, weil beide Programme die Datenbankeinstellungen benötigen und durch die Verwendung einer zentralen Konfigurationsdatei diese nicht in verschiedenen Konfigurationsdateien stehen müssen.

Mit dem verwendeten Format YAML⁷ lassen sich Daten strukturiert und lesbar abbilden. YAML bot sich vor allem an, weil es für Ruby bereits eine Bibliothek zum

⁷ Eine leicht lesbare Datenserialisierungssprache, s. <http://www.yaml.org/>

Auswerten von YAML-Dateien gibt und eine eigene Implementierung überflüssig macht. Eine Konfigurationsdatei könnte folgendermaßen aussehen:

```
config:
  tack_trace_data: /var/cache/hubd/trace.csv
  tack_trace_cache: /var/cache/hubd/tt.pstore
hubd:
  host: localhost
  port: 9876
  log_file: /var/log/hubd.log
  pid_file: /var/run/hubd.pid
  interval_time: 10 #seconds
  debug: true
  web_user: nobody
tt_record:
  pid_file: /var/run/tt_record.pid
database:
  adapter: mysql
  database: mailbox_development
  username: hubd
  password: secret
  host: localhost
```

In YAML werden der Bezeichner und der Wert mit einem Doppelpunkt getrennt. So bedeutet zum Beispiel `pid_file: /var/run/tt_record.pid`, dass `pid_file` den String `/var/run/tt_record.pid` besitzt.

Die Konfigurationsdatei ist in verschiedene Unterpunkte aufgeteilt. Im YAML wird das mit entsprechendem Einrücken gekennzeichnet.

Unter `config` werden allgemein gültige Konfigurationen, die für alle Programme gelten, geschrieben wie zum Beispiel die Pfade für Track & Trace die vom EDI-Hub-Server als auch von dem Track & Trace-Recorder benötigt werden. Daneben werden auch die Datenbankeinstellungen die unter `database:` stehen von beiden Programmen ausgewertet.

Konfigurationen die nur für den EDI-Hub-Server gelten stehen unter `hubd` und die nur für den Track & Trace-Recorder gelten stehen unter `tt_record`.

4.1.1. Klassenbeschreibung

Der EDI-Hub-Server verwendet die folgenden Klassen: Hub, Mailbox, OutboxEntry, GLN, Logger und NumberGenerator (Abb. 7):

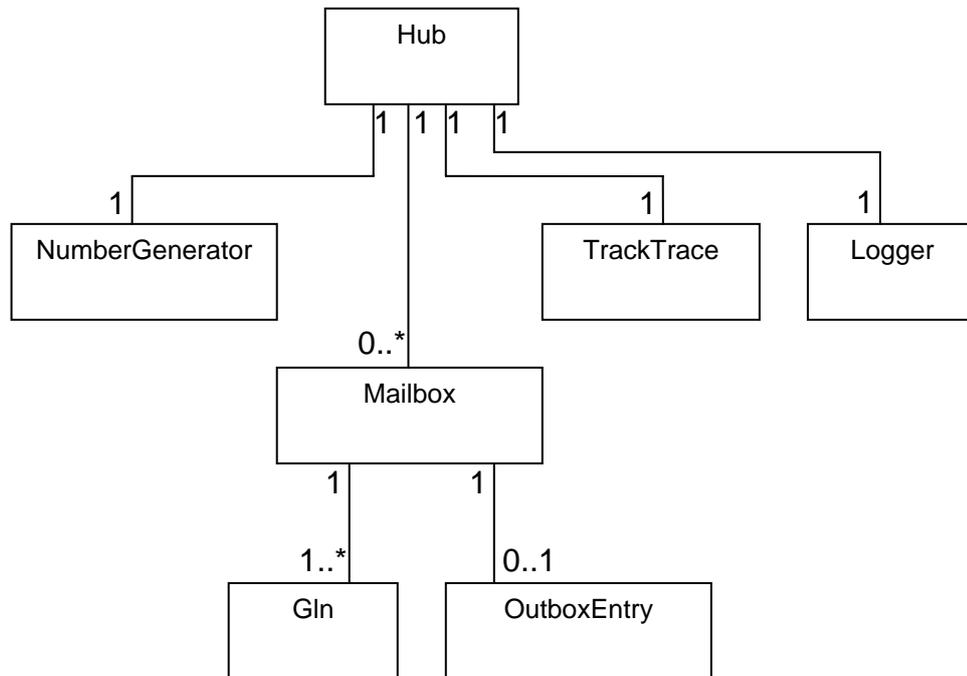


Abb. 7 Klassendiagramm des EDI-Hub-Servers.

Eine genaue Klassenbeschreibung befindet sich im Verzeichnis `doc` auf der Installations-CD. Die Dokumentation wurde mit RDoc erstellt und verwendet dafür die Kommentare im Quellcode. Diese werden ausgewertet, um daraus eine Dokumentation zu generieren.

Nachfolgend eine kurze Beschreibung der Klassen:

Hub:

Der `Hub` führt das Track & Trace der einzelnen Ereignisse aus und protokolliert alles in der Log-Datei. Außerdem übernimmt der `Hub` die Vergabe der Identifikationsnummern, die mit `NumberGenerator` erzeugt werden.

Für jeden Teilnehmer werden die Benutzerinformationen aus der Datenbank gelesen und ein `Mailbox`-Objekt erstellt. Die weitere Initialisierungen der GLN-

Nummern, Geschäftspartnern usw. werden von den Mailbox-Objekten übernommen.

Zusätzlich werden die Benutzerverzeichnisse vom Hub repariert oder gegebenenfalls neu erstellt, falls das Mailbox-Verzeichnis nicht existiert, wie es nach der Neuanmeldung eines Teilnehmers der Fall ist.

Mailbox:

Die Mailbox-Klasse bildet die Mailbox eines Teilnehmers ab. Sie übernimmt die Initialisierung der GLNs, deren Partner-GLNs und die GLN-Nummern, die für eine ausführliche Validierung markiert sind.

Daneben scannt jedes Mailbox-Objekt das „out“-Verzeichnis auf neue EDI-Dateien, die zugestellt werden sollen. Die gefundenen Dateien werden als OutboxEntry-Objekt abgebildet.

OutboxEntry:

Die OutboxEntry-Klasse bildet eine ausgehende EDI-Nachricht ab und übernimmt das Auswerten der EDI-Datei unter Verwendung der edi4r Bibliothek. Beim Erstellen eines Objekts erhält es eine eindeutige Identifikationsnummer vom Hub. Nach dem Auswerten einer gültigen EDI-Datei, kennt die Klasse den Sender und Absender. Sobald die Sender-GLN und die Empfänger-GLN dem Hub bekannt sind, wird sie weiterverarbeitet, anderenfalls wird sie abgelehnt.

Das Umbenennen und das Verschieben der abgebildeten EDI-Datei wird von dieser Klasse übernommen.

Gln:

Das Gln-Objekt kennt alle Ihre Partner-GLNs, von denen EDI-Dateien akzeptiert werden. Zusätzlich besitzt jedes Gln-Objekt eine Liste von GLN-Nummern für eine ausführliche Validierung.

NumberGenerator:

Diese Klasse wird zum Erzeugen von eindeutigen Verarbeitungsnummern verwendet. Die neu eintreffenden EDI-Dateien bekommen eine Verarbeitungsnummer zugeteilt, mit der sie im Hub eindeutig identifizierbar sind. Diese Verarbeitungsnummer wird ebenfalls zur Umbenennung der EDI-Dateien verwendet.

TrackTrack:

Die Klasse `TrackTrace` protokolliert die einzelnen Zustände, die eine EDI-Datei während ihres Lebenszyklus erreicht. Sie übernimmt das Extrahieren der, für die Protokollierung erforderlichen, Daten aus dem `OutboxEntry`-Objekt und schreibt diese in die Track & Trace-Datei `trace.csv`. Die Daten werden im kommaseparierten Format gespeichert auch CSV⁸ genannt.

Logger:

Hierbei handelt es sich um eine Klasse, die mit Ruby ausgeliefert wird. Mit ihr wird eine Log-Datei erstellt, in die das `Hub`-Objekt die Bearbeitungsschritte protokolliert.

⁸ *Kommasepariertes Format, s. <http://de.wikipedia.org/wiki/CSV-Datei>*

4.1.2. Sequenzdiagramm:

Nachfolgend ein Sequenzdiagramm einer erfolgreich zugestellten EDI-Datei inklusive Track & Trace: (Abb. 8)

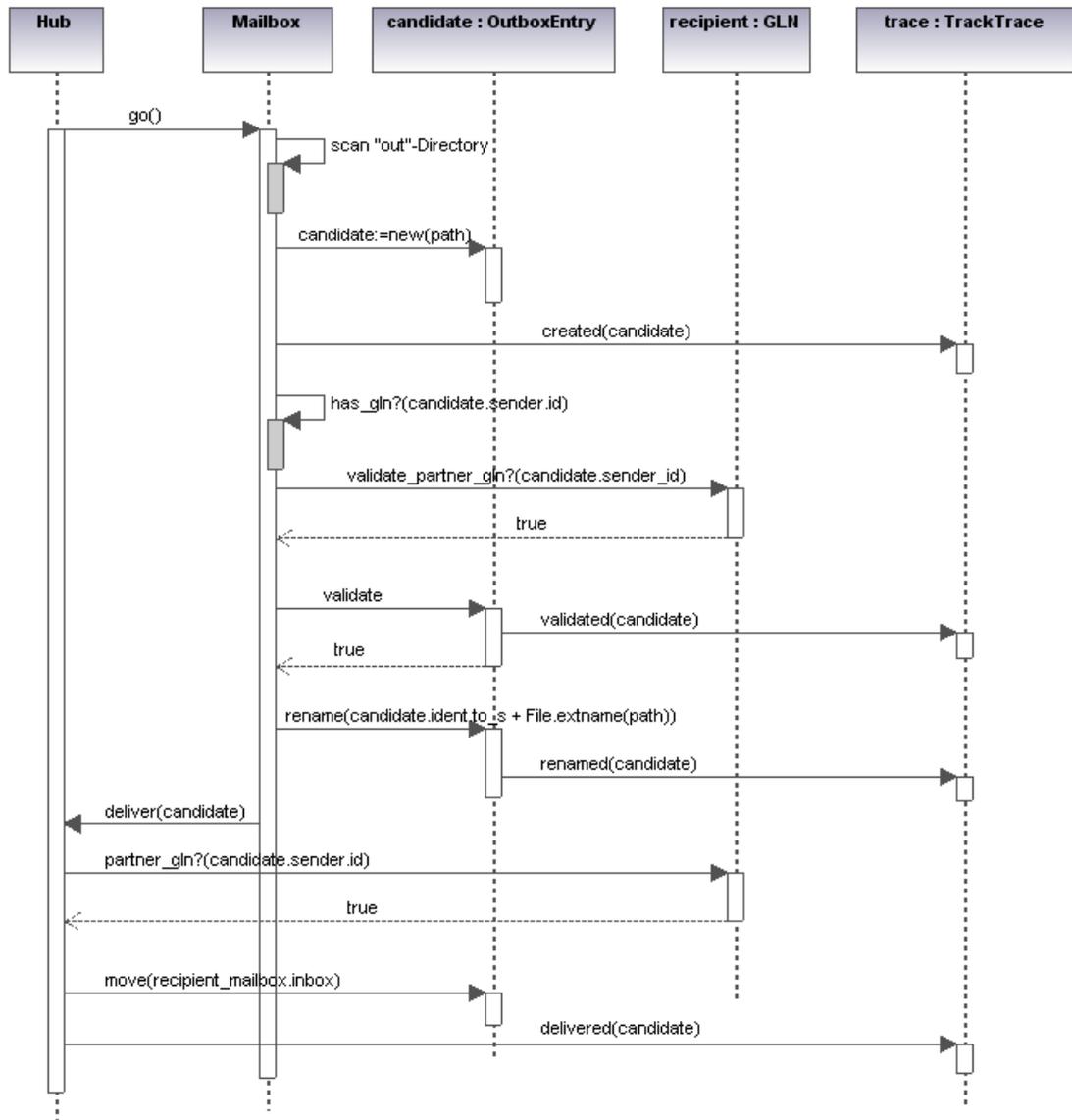


Abb. 8: Sequenzdiagramm: Erfolgreiche Zustellung einer EDI-Datei inklusive Track & Trace.

Eine Beschreibung des Sequenzdiagramms (Abb. 8):

Der Hub besitzt eine Liste mit Mailbox-Objekten, die beim Initialisieren der Teilnehmer erstellt wurden. Für jedes Mailbox-Objekt wird die Funktion go aufgerufen, worauf das Mailbox-Objekt beginnt in seinem „out“-Verzeichnis nach EDI-Dateien zu suchen. Anschließend versucht das Mailbox-Objekt die gefundenen EDI-Dateien nach einander zu zustellen. Für jede gefundene EDI-Datei

wird ein `OutboxEntry`-Objekt erstellt, welches die EDI-Datei parst und alle erforderlichen Informationen wie Sender-GLN und Empfänger-GLN extrahiert. Das Erstellen des `OutboxEntry`-Objekts wird sogleich von der `TrackTrace`-Klasse mit dem Ereignis „created“ protokolliert. Um die Integrität der EDI-Datei zu garantieren, wird überprüft, ob die Sender-GLN zum `Mailbox`-Objekt gehört. Nach positiver Prüfung wird mit Hilfe der Funktion `validate_partner_gln` des `GLN`-Objektes überprüft, ob die EDI-Datei einer ausführlichen Validierung unterzogen werden soll. Die Validierung wird daraufhin durchgeführt und vom `TrackTrace`-Objekt mit dem Ereignis „validated“ protokolliert. Anschließend wird die EDI-Datei umbenannt und vom `TrackTrace`-Objekt mit dem Ereignis „renamed“ protokolliert. Der `Hub` stellt das `OutboxEntry`-Objekt mit der Funktion `deliver` der Empfänger `Mailbox` zu. Davor wird abschließend geprüft, ob die Sender-GLN Geschäftspartner der Empfänger-GLN ist und nach erfolgreicher Überprüfung in das „in“-Verzeichnis der Empfänger `Mailbox` verschoben. Das Zustellen wird vom `TrackTrace`-Objekt mit dem Ereignis „delivered“ protokolliert. Damit ist die Zustellung der EDI-Datei abgeschlossen.

4.2. EDI-Hub-Kontroller

Um den EDI-Hub-Server im Betrieb zu steuern verwendet das Webinterface `hubdctrl.rb`, wobei im Wesentlichen drei Kommandos verwendet werden: `init <mailbox>`, `create <mailbox>` und `delete <mailbox>`.

Die Kommunikation mit dem EDI-Hub-Server findet über das Protokoll TCP/IP statt. Die IP-Adresse und der Port auf den der EDI-Hub-Server auf Kommandos lauscht, ist in der Konfigurationsdatei des EDI-Hub-Servers `/usr/local/etc/hubdconf.yml` frei wählbar. Die Standardeinstellungen für die IP-Adresse lautet `localhost` und der Port `9876`. Mit diesen Einstellungen werden nur Kommandos auf `localhost` entgegengenommen, wodurch eine Erreichbarkeit von Programmen ermöglicht wird, die ausschließlich auf dem gleichen Rechner laufen. Für einen externen Betrieb der Webanwendung ist es notwendig, dass der EDI-Hub-Server Kommandos auch von außen entgegennimmt. Für diesen Zweck wird die IP-Adresse, die von außen erreichbar ist, beim EDI-Hub-Server als IP-Adresse eingetragen. In der Regel ist es die IP-Adresse von der Netzwerkkarte und kann mit dem Konsolenbefehl `ipconfig` ermittelt werden. Dabei muss beachtet werden, dass der EDI-Hub-Server auch von einer nicht autorisierten Person gesteuert werden kann, da sie nicht unterschieden werden können. Um dem entgegen zu wirken müsste die Firewall des Betriebssystems so eingerichtet werden, dass ausschließlich TCP/IP-Pakete von der IP-Adresse, des Server auf dem die Webanwendung läuft, über PORT 9876 akzeptiert werden. Alternativ könnte der EDI-Hub-Server um eine ACL⁹ Steuerung Erweitert werden, über die nur Kommandos von frei wählbaren IP-Adressen entgegengenommen werden.

Der Grund für die Verwendung von TCP/IP ist neben der Möglichkeit den EDI-Hub-Server von einem entfernten Server kontrollieren zu können, die Möglichkeit, dass ein nicht privilegierter Prozess Kontrolle über den mit Systemadministratorrechten laufenden EDI-Hub-Server bekommt. Anders als der EDI-Hub-Server läuft die Webanwendung, aus Sicherheitsgründen, mit eingeschränkte Rechten und muss trotzdem in der Lage sein den EDI-Hub-Server zu Kontrollieren, um zum Beispiel Teilnehmer, nach Änderung ihrer Einstellungen, neu zu initialisieren.

⁹ Eine Zugriffskontrollliste, s. <http://www.networkliquidators.com/definition-acl.asp>.

Der EDI-Hub-Kontroller verwendet für die Kommunikation keine Klassen, da das Protokoll zum Übermitteln der Kommandos rudimentär ist. Es wird eine TCP/IP Verbindung hergestellt, worauf zugleich das entsprechende Kommando abgesetzt wird. Nach Annahme des Kommandos wird die Verbindung vom EDI-Hub-Server beendet. Für Rückmeldungen müsste das Protokoll erweitert werden, um zum Beispiel Statussätze für eine Fernüberwachung abzufragen.

4.2.1. Kommandos

Nachfolgende Kommandos werden vom EDI-Hub-Server verarbeitet.

Kommando:	Beschreibung
scan	Stößt ein Abarbeiten der Benutzer-Verzeichnisse an.
interval <seconds>	Ändert die Wartezeit bis zum erneuten Abarbeiten der Mailboxverzeichnisse.
init <mailbox>	Initialisiert <mailbox> beim nächsten Abarbeiten der Mailboxverzeichnisse.
create <mailbox>	Initialisiert <mailbox> sofort und erstellt die Verzeichnisstruktur.
delete <mailbox>	Entfernt <mailbox> aus dem Hub und markiert es mit einem anhängen von <code>_deleted</code> als gelöscht. So wird aus <code>hu000014</code> zum Beispiel <code>hu000014_deleted</code> .
reset	Stoppt die Zustellung der EDI-Dateien und initialisiert alle Benutzer neu.
stop	Beendet den Hub.

Für <mailbox> steht der Name der Mailbox, zum Beispiel `hu00001` für die Mailbox `/home/hu000001`.

Die meisten Kommandos werden nicht von der Webanwendung verwendet und sind hauptsächlich für den Administrator bestimmt, um von der Konsole aus Einfluss auf den Hub-Server zu nehmen.

4.3. Track & Trace-Recorder

Die vom EDI-Hub-Server geschriebenen Track & Trace-Daten befinden sich in der Datei `trace.csv`. Die Track & Trace-Daten werden kommasepariert gespeichert.

Der Track & Trace-Recorder `tt_record.rb` versucht die Datei `trace.csv` zu öffnen und in die Datenbank zu schreiben. Bevor die Daten in die Datenbank geschrieben werden, wird die Datei umbenannt. Dadurch wird ein gleichzeitiges Schreiben von dem Track & Trace-Recorder und dem EDI-Hub-Server verhindert und die Datei bleibt konsistent. Der EDI-Hub-Server erstellt automatisch eine neue Track & Trace-Datei für die folgenden Track & Trace-Daten.

Sollte die Datenbank nicht erreichbar sein, bricht `tt_record.rb` mit einer Fehlermeldung ab.

Aus der Konfigurationsdatei des EDI-Hub-Servers werden neben den Datenbankeinstellungen auch der Pfad für die Track & Trace-Datei ausgelesen. Der Standardpfad für die Konfigurationsdatei lautet `/usr/local/etc/hubd-conf.yml`. Ein alternativer Pfad kann mit der Option `-c` beim Starten übergeben werden. Zusätzlich kann mit der Option `-t` der Pfad zu den Track & Trace-Daten angegeben werden, falls dieser nicht mit den in der Konfigurationsdatei übereinstimmen sollte. Eine genaue Beschreibung des Track & Trace-Recorders wird mit `tt_record.rb -h` angezeigt.

Das Programm `tt_record.rb` verwendet zum Schreiben der Track & Trace-Daten die Klasse `TrackTraceRecorder`.

TrackTraceRecorder
- file - database - count_entities - dbh
+ connect + record + disconnect

4.4. Track & Trace

Die Zustellung einer EDI-Datei wird mit Hilfe des Track & Trace [Lit10] durch den EDI-Hub-Server an fest definierten Zeitpunkten protokolliert. Die Protokollierung wird von der `TrackTrace` Klasse durchgeführt.

Bis auf „received“ lassen sich alle Ereignisse während der Zustellung einer EDI-Datei protokollieren. Um das Ereignis „received“ protokollieren zu können, muss vor dem Zustellen der EDI-Dateien nach gelöschten EDI-Dateien gesucht werden. Dafür wird eine Liste mit den Namen der zugestellten und bisher nicht als „received“ protokollierten EDI-Dateien pro Mailbox geführt. Damit diese Liste nach einem Neustart des EDI-Hub-Servers nicht verloren geht, wird diese Liste nicht im Arbeitsspeicher, sondern mit Hilfe der Klasse `PStore`¹⁰ in der Datei `/var/cache/tt.pstore` gehalten.

4.4.1. Ereignisse

Ergebnis	S	Beschreibung:
* created	Yellow	Der Anfangszustand wenn eine neue Datei im „out“-Ordner erscheint.
validated	Yellow	Der Zeitpunkt und das Ergebnis der Überprüfung. Die Überprüfung der Dateien kann jeder Teilnehmer im Webinterface selbst bestimmen.
* renamed	Yellow	Der Zeitpunkt, wenn die Datei vom Hub umbenannt wird.
* rejected	Red	Ablehnung der Zustellung mit Fehlercode und Zeitpunkt der Ablehnung. Ende-Zustand.
* delivered	Green	Zeitpunkt der Zustellung in das Verzeichnis „in“ beim Empfänger. Ende-Zustand.
* received	Green	Zeitpunkt der Abholung vom Empfänger, wenn sich die Datei nicht mehr im „in“ – Ordner befindet.

*) Pflichtwerte

Die Spalte S stellt den Status des Ereignisses farblich dar, Fehler (rot), Zwischenschritte (gelb) und erfolgreiche Beendigungen (grün) an.

¹⁰ `PStore` bildet ein Hash-Array in einer Datei ab, s. <http://www.ruby-doc.org/core/classes/PStore.html>

4.5. Datenbank

In der Datenbank werden die Benutzerdaten gespeichert, auf die sowohl die Webanwendung als auch der EDI-Hub-Server und der FTPS-Server zur Authentifizierung zugreifen. Änderungen an den Daten werden ausschließlich über die Webanwendung gemacht, mit deren Hilfe Teilnehmer hinzugefügt, gelöscht und geändert werden. Der EDI-Hub-Server liest einmalig beim Initialisieren der Mailboxen alle notwendigen Benutzerinformationen aus der Datenbank.

Der FTPS-Server proftpd benutzt die Datenbank ausschließlich bei der Anmeldung eines Teilnehmers. Die Teilnehmer werden mit Hilfe der Datenbank authentifiziert. Außerdem wird der Pfad der Mailbox aus der Datenbank ausgelesen, der gleichzeitig das Wurzelverzeichnis des Teilnehmers ist.

Das folgende Entity Relationship mit Pfeilnotation soll zur Veranschaulichung dienen (Abb. 9):

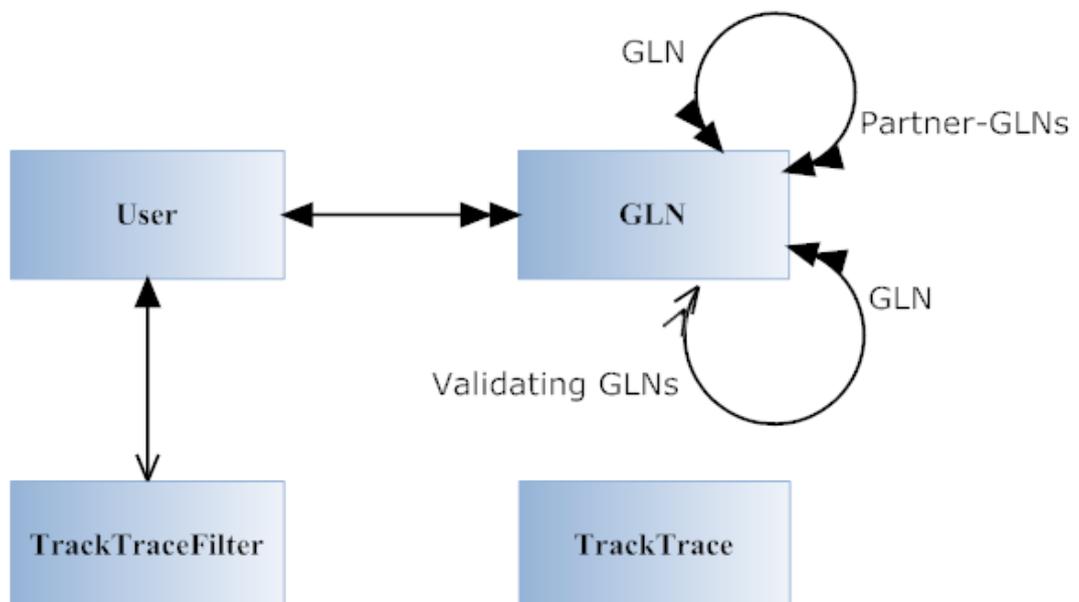


Abb. 9: Entity Relationship mit Pfeilnotation.

4.5.1. Tabellenbeschreibung

Die registrierten Teilnehmer werden in der Tabelle `users` gespeichert:

Spalte	Beschreibung
<code>id</code>	Primärschlüssel
<code>company</code>	Name der Firma
<code>hashed_password</code>	Passwort mit crypt Verschlüsselung
<code>unix_user</code>	Eindeutiger Name des Benutzers unter Linux, der den Namen der Mailbox entspricht.
<code>home_dir</code>	Pfad zum Mailboxverzeichnis
<code>firstname</code>	Vorname
<code>lastname</code>	Nachname
<code>street</code>	Strasse
<code>postcode</code>	Postleitzahl
<code>place</code>	Wohnort
<code>telefon</code>	Telefonnummer
<code>fax</code>	Faxnummer
<code>email</code>	E-Mail-Adresse
<code>registered_at</code>	Zeitpunkt der Registrierung mit Datum und Uhrzeit
<code>last_login</code>	Zeitpunkt der letzten Anmeldung mit Datum und Uhrzeit
<code>admin</code>	Kennzeichnung für Administrationsrechte

Vorname, Nachname, Strasse, Postleitzahl, Wohnort, Telefon, Fax und E-Mail beziehen sich auf die Kontaktperson der Firma. Mit `home_dir` ist das Wurzelverzeichnis des Benutzers gemeint, das gleichzeitig das Mailboxverzeichnis darstellt.

Die registrierten GLNs werden in der Tabelle `glns` gespeichert und bestehen aus der GLN-Numer und einem Kurzname:

Spalte	Beschreibung
<code>id</code>	Primärschlüssel
<code>GLN</code>	GLN-Nummer
<code>Name</code>	GLN-Kurzname

Die Track & Trace-Daten werden in der Tabelle `track_traces` gespeichert:

Spalte	Beschreibung
<code>id</code>	Primärschlüssel
<code>ident</code>	Eindeutige Identifizierungsnummer der EDI-Datei
<code>event_date</code>	Datum und Zeit des Ereignisses
<code>status</code>	Der protokollierte Status
<code>mailbox</code>	Die versendete Mailbox
<code>sender_id</code>	Absender-GLN
<code>recipient_id</code>	Empfänger-GLN
<code>serial_no</code>	Interchange Reference Number
<code>msg_type</code>	EDI-Standard
<code>doc_type</code>	Nachrichtentyp
<code>error_code</code>	Fehlercode
<code>filename</code>	Dateiname
<code>size</code>	Größe der Datei in Bytes
<code>md5</code>	MD5 Prüfsumme der Datei

Die Suchmasken für das Track & Trace werden in der Tabelle `track_trace_filters` gespeichert und bilden alle Auswahlfelder ab:

Spalte	Beschreibung
<code>id</code>	Primärschlüssel
<code>user_id</code>	Verweis auf <code>user</code>
<code>gln</code>	GLN des Benutzers, die angezeigt werden sollen
<code>period</code>	Track & Trace-Daten anzeigen, die älter sind als X Sekunden.
<code>direction</code>	Richtung die EDI-Daten: Ausgehend, Eingehend oder Alle
<code>doc_type</code>	EDI-Standard
<code>msg_type</code>	Nachrichtentyp
<code>partner_gln</code>	Track & Trace-Daten mit Partner-GLN
<code>order</code>	Aufsteigend oder Absteigend sortiert
<code>event_type</code>	Art der Ereignisse: Normale, Alle, Nur Fehler, Noch nicht beendet.

Die Beziehungen zwischen den GLNs und der Partner-GLNs werden in der Tabelle `business_partners` abgebildet:

Spalte	Beschreibung
<code>gln_id</code>	Verweis auf GLN
<code>partner_gln_id</code>	Verweis auf GLN, die die Partner-GLN von <code>gln_id</code> darstellt.

Hier bei handelt es sich um eine N:M Tabelle.

Die Geschäftsbeziehungen mit aktivierter Validierung werden in der Tabelle `validate_data` abgebildet:

Spalte	Beschreibung
<code>gln_id</code>	Verweis auf GLN
<code>validate_gln_id</code>	Verweis auf GLN, die von <code>gln_id</code> einer ausführlichen Validierung unterzogen werden.

Hier bei handelt es sich um eine N:M Tabelle.

Während einer Neuregistrierung werden die Benutzerdaten in der Tabelle `accounts` gehalten und beinhaltet die meisten Felder der Tabellen `users` und `glns` und zusätzliche für die Registrierung.

Spalte	Beschreibung
<code>id</code>	Primärschlüssel
<code>company</code>	Name der Firma
<code>hashed_password</code>	Passwort mit crypt Verschlüsselung
<code>firstname</code>	Vorname
<code>lastname</code>	Nachname
<code>street</code>	Strasse
<code>postcode</code>	Postleitzahl
<code>place</code>	Wohnort
<code>telefon</code>	Telefonnummer
<code>fax</code>	Faxnummer
<code>email</code>	E-Mail-Adresse
<code>registered_at</code>	Zeitpunkt der Registrierung mit Datum und Uhrzeit
<code>gln_gln</code>	GLN-Nummer
<code>gln_name</code>	GLN-Kurzname
<code>valdiate_email_code</code>	Code zum bestätigen der E-Mail-Adresse

acceptation_code | Code zum akzeptieren oder ablehnen durch den
Administrator

Felder der Tabelle `users`, die während der Registrierung nicht relevant sind,
wurden nicht übernommen.

4.6. Webanwendung

Die Webanwendung ist die Schnittstelle für die Benutzer und wurde mit dem Web-Framework¹¹ Ruby on Rails programmiert. Über die Webanwendung wird die komplette Benutzerverwaltung abgewickelt, wobei sich jeder Teilnehmer selbst verwaltet. Ein Eingreifen des Hub-Betreibers ist lediglich bei einer Neuregistrierung notwendig, wodurch die Kosten gering gehalten werden.

Für eine korrekte Darstellung der Webanwendung wird ein XHTML [Lit11] fähiger Browser benötigt, da sie konsequent nach dem XHTML 1.1 Standard entwickelt wurde.

XHTML ist eine XML Neuformulierung von HTML 4.01 und entspricht den Syntaxregeln von XML. XHTML ist eine Auszeichnungssprache und beschreibt ausschließlich den Strukturierten Inhalt der Seite. Aus diesem Grund sind Elemente und Attribute, die direkt die Darstellung beeinflussen, in XHTML nicht übernommen worden. Es hat sich vor allem im modernen Webdesign durchgesetzt, wo der Schwerpunkt auf Struktur und Barrierefreiheit [Lit11] gerichtet ist. Die Formatierung des Dokuments findet ausschließlich mit CSS¹² statt.

Alle aktuellen Webbrowser wie der Internet Explorer ab Version 6, Mozilla Firefox ab Version 1.x, und einige mehr unterstützen dieses Format. Nicht-XHTML-fähige Webbrowser sind ebenfalls in der Lage XHTML-Dokumente darzustellen. Dazu werden die fehlertoleranten HTML-Parser der Webbrowser ausgenutzt und das XHTML-Dokument als HTML-Dokument geparkt.

Die Webanwendung wurde auf einem Linux-Server mit einem Pentium 2 mit 500 MHz Taktrate entwickelt und getestet. Während den Tests war kein Einbruch der Performance zu verzeichnen. Eine Ausnahme bildete allerdings die Track & Trace Übersicht. Die dynamische Zusammenfassung der zahlreichen Track & Trace-Daten benötigt viel Rechenleistung. Durch das Einschränken der Suche mit Hilfe der Suchoptionen wird die Performance gesteigert. Die Aufbauzeit der Track & Trace-Übersicht steigt mit der Anzahl der für die Firma zutreffenden Track & Trace-Daten. Wie man aus folgendem Diagramm ersehen kann, benötigt die Darstellung der Track & Trace-Übersicht von ca. 235 Dateien die aus 692 Datensätzen bestehen 13 Sekunden Bearbeitungszeit: (Abb. 10)

¹¹ *Entwicklungsumgebung für Internetanwendungen.*

¹² *Cascade Style Sheet, s. <http://de.selfhtml.org/css/>*

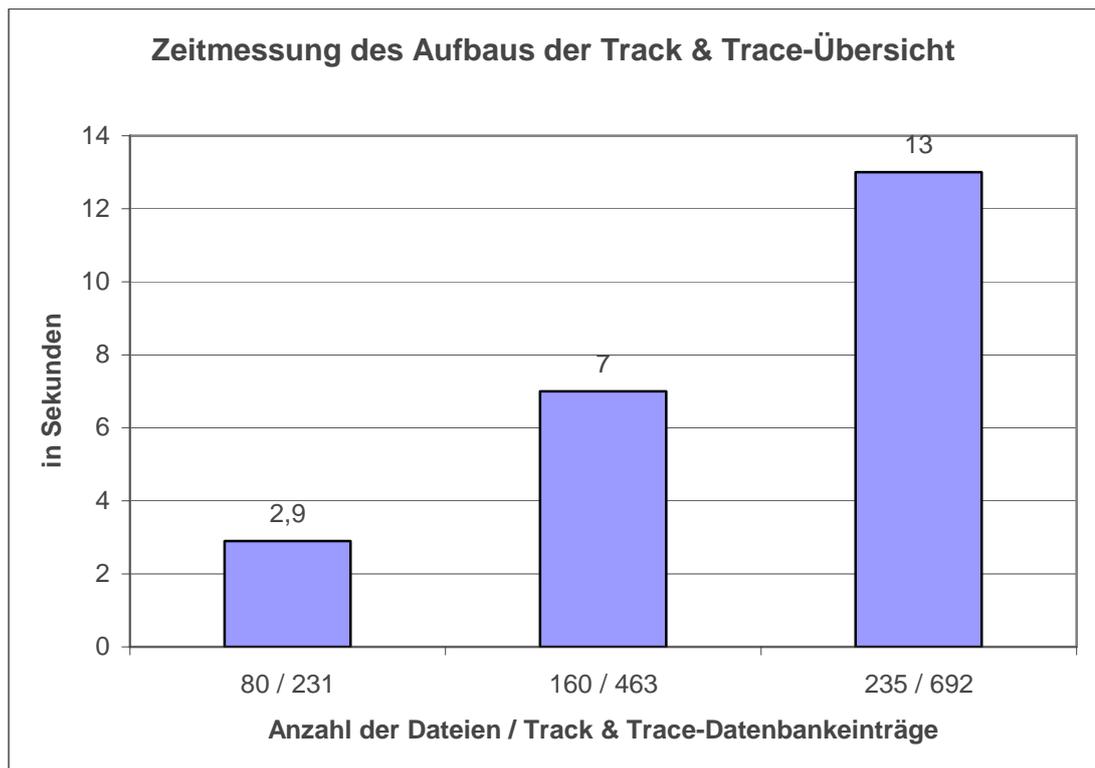


Abb. 10: Diagrammdarstellung der Darstellungszeit der Track & Trace-Übersicht.

Die gesamte Oberfläche der Webanwendung ist in deutscher Sprache gehalten. Leider konnte das nicht konsequent durchgeführt werden.

Dazu gehört die von Ruby on Rails verwendete Darstellung bei fehlgeschlagenen Gültigkeitsprüfungen eines Models innerhalb der View. Es besteht zwar die Möglichkeit für jede Gültigkeitsprüfung einen eigenen Fehlertext zu definieren, allerdings mit Einschränkungen. So wird den Fehlertexten der Datenbankfeldname des Attributes vorangestellt und die Fehlertexte für die einzelnen Gültigkeitsprüfungen bekommen eine englischsprachige Überschrift. Die Attribute eines Models sind in der Datenbank abgebildet und besitzen englische Bezeichnungen, daher passen sie nicht zu deutschen Fehlertexten. Da sich durch die Vermischung der Sprachen unverständliche Fehlertexte ergeben würden, wurden englische Fehlertexte verwendet. (Abb. 11)

Um die Fehlertexte komplett ins Deutsche zu übersetzen, wäre es notwendig die Funktion, die für die Darstellung verantwortlich ist, innerhalb der Ruby on Rails Bibliotheken anzupassen. Dann müsste aber bei jedem Update des Ruby on Rails Frameworks darauf geachtet werden, dass die Änderungen noch fehlerfrei funktionieren. Es ist daher sinnvoller auf eine angepasste Version zu warten, mit der Möglichkeit die Fehlertexte vollständig frei zu definieren. Da Ruby on Rails noch

sehr jung ist, ist mit einer Weiterentwicklung beziehungsweise Ausbau des Frameworks zu rechnen.

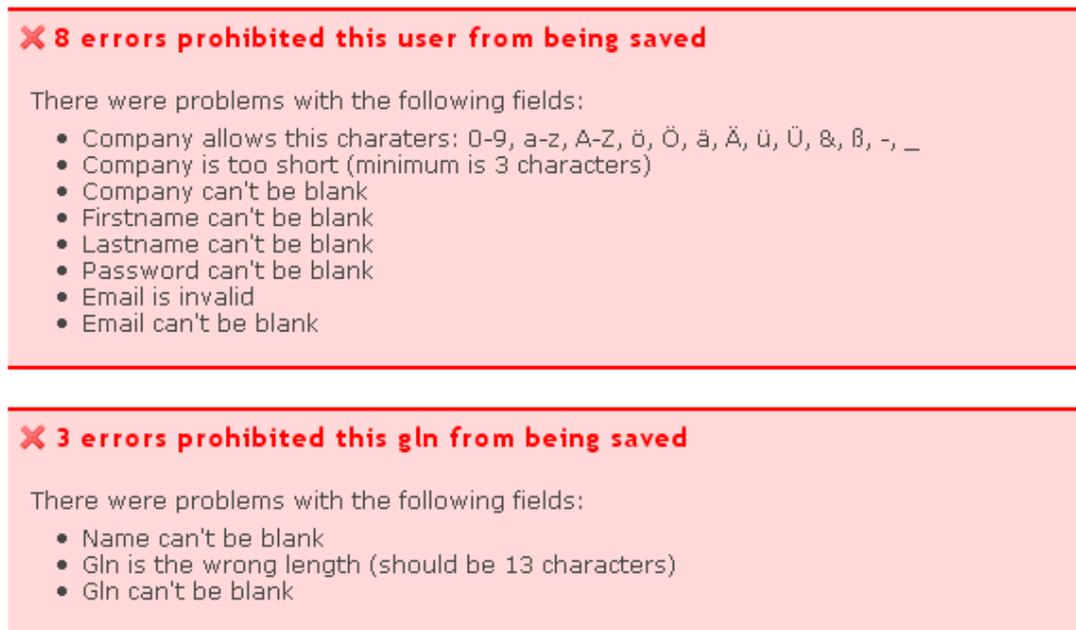


Abb. 11: Ruby on Rails generierte Fehlermeldungen.

Dem Besucher wird die Möglichkeit geboten, Informationen über das Hub-System auf der Startseite zu lesen, sich zu registrieren oder wenn bereits registriert, sich anzumelden.

4.6.1. Registrierung

Über das Webinterface registrieren sich alle Handelspartner, die am EDI-Hub-System teilnehmen wollen. (Abb. 12)

Abb. 12: Formular um sich am EDI-Hub zu registrieren.

Die Registrierung verläuft in drei Schritten: Stammdaten eingeben, E-Mail Überprüfung und Kontoaktivierung.

4.6.1.1. Stammdaten

Um am EDI-Hub-System teilnehmen zu können, müssen zunächst einige Stammdaten eingetragen werden (Abb. 12). Dazu gehören die folgenden Pflichtfelder:

Eingabefelder	Beschreibung
Der Firmenname	Der Name der Firma wird auch zum anmelden am EDI-Hub-System verwendet.
GLN	Eine GLN ist mindest erforderlich.
Kurzname	Eine Kurzname der GLN.
Ansprechspartner	Vorname, Nachname, E-Mail-Adresse, usw. des Ansprechpartners der Firma für eventuelle Rückbeziehungsweise Anfragen.

Die einzelnen Felder werden auf Gültigkeit geprüft, um fehlerhafte oder fehlende Angaben zu vermeiden. So muss zum Beispiel die E-Mail-Adresse folgendes Format entsprechen: Mindestens ein Zeichen vor und mindestens ein Zeichen nach dem @ mit einem abschließenden Punkt gefolgt vom Länder-Code. Diese Überprüfung wird von dem Model durchgeführt.

4.6.1.2. E-Mail-Überprüfung

Um die E-Mail-Adresse des neuen Teilnehmers zu überprüfen wird das „double-opt-in“ Verfahren angewandt. Dazu wird ein generierter Bestätigungscode an die angegebene E-Mail-Adresse geschickt. Dem Teilnehmer steht es frei die E-Mail-Überprüfung mit dem, in der E-Mail stehenden HTTP Link, direkt abzuschließen oder alternativ das Formular auf der Webseite zu benutzen, um dem ihm bekannten Bestätigungscode einzugeben. Da die E-Mail-Überprüfung nur mit dem passenden Code abgeschlossen werden kann und dieser nur dem Besitzer der E-Mail-Adresse bekannt gemacht wurde, wird sichergestellt, dass die E-Mail-Adresse gültig ist. Der Link zum Bestätigungsformular der Webseite steht ebenfalls in der E-Mail.

4.6.1.3. Kontoaktivierung

Ein Teilnehmer kann sein Konto aktivieren, sobald seine Registrierung von dem Administrator akzeptiert wurde. Damit behält der EDI-Hub-Betreiber beziehungsweise Administrator die Kontrolle über die am EDI-Hub registrierten Teilnehmer. Der Administrator erhält eine E-Mail mit den Anmeldeinformationen des Teilnehmers und einem Link, zum Akzeptieren des Teilnehmers und einen Link um den Teilnehmer abzulehnen. Damit der Administrator nicht mit Benachrichtigungen von neu registrierten Teilnehmern überschwemmt wird, werden die Akzeptierungsanfragen erst nach Überprüfung der E-Mail-Adresse verschickt. Dadurch ist sichergestellt, dass ernsthaftes Interesse an der Teilnahme am EDI-Hub-System besteht und ein Kontakt durch die verifizierte E-Mail Adresse mit dem Interessenten für Rückfragen möglich ist.

Mit dem Akzeptieren wird verhindert, dass sich Teilnehmer anmelden, die keine Geschäftsbeziehung mit dem EDI-Hub-Betreiber besitzen und das EDI-Hub-System als Plattform für Ihre Geschäftspartner nutzen wollen.

Es wäre vorstellbar, dass das EDI-Hub-System von einem Interessensverbund betrieben wird und sich mit monatlichen Teilnehmergebühren finanziert. Bereits bei einer Teilnehmerzahl von 50 würden die monatlichen Beiträge sehr gering ausfallen. Wenn man davon ausgeht, dass ein Root-Server mit unbegrenztem Traffic 70 Euro kostet, würden sich die monatlichen Beiträge auf 1,40 Euro belaufen.

Nachdem der Administrator den neuen Teilnehmer akzeptiert hat wird der Teilnehmer per E-Mail darüber informiert und herzlich Willkommen geheißen. Um die Registrierung zu beenden muss der Teilnehmer das Konto noch Aktivieren. Zum aktivieren wird kein Code benötigt, es genügt den Link in der E-Mail zu folgen. Mit Eingabe des Firmenamens erfolgt die Aktivierung. Alternativ erreicht man das Aktivierungsformular auch über die Anmeldeseite.

Sollte der Administrator den neuen Teilnehmer ablehnen, wird der Teilnehmer darüber mit einer E-Mail in Kenntnis gesetzt und für eventuelle Rückfragen auf die E-Mail-Adresse des Administrators verwiesen.

Bis das Konto vom Teilnehmer aktiviert wurde, werden die Benutzerdaten in einer temporären Tabelle gespeichert. Erst nach einer erfolgreichen Registrierung, die mit der Aktivierung des Kontos beendet ist, wird der Teilnehmer in das System übernommen. Um das System konsistent zu halten, werden die Benutzerinformationen beim aktivieren des Kontos nochmals auf Gültigkeit geprüft. Damit wird zum Beispiel ausgeschlossen, dass sich in der Zeit der Registrierung bereits ein anderer Teilnehmer mit überschneidenden Stammdaten registriert hat.

4.6.2. Eigene Einstellungen

Nach erfolgreicher Registrierung kann sich der Teilnehmer an dem EDI-Hub-System anmelden und gelangt in dem Menüpunkt „Ihre Einstellungen“. Hier bekommt der Teilnehmer einen Überblick über seine Einstellungen und kann diese bei Bedarf bearbeiten. Das sind im Einzelnen seine Stammdaten, seine GLNs und die Geschäftspartner seiner GLNs.

4.6.2.1. GLNs verwalten

Die Teilnehmer benötigen mindestens eine GLN, die bereits bei der Registrierung angegeben werden muss. Die Global Location Number, kurz GLN, wird jeder Firma, die sich am EAN-UCC¹³ System anmeldet, zugeteilt und ist Basis der Artikel-Kennzeichnung. Im deutschen Sprachraum wird auch die Bezeichnung Internationale Lokalisierungsnummer, kurz ILN, verwendet. Die GLN ist 13-stellig und besteht aus einer fest vergebenen Basisnummer sowie einem frei wählbarer Bereich mit abschließender Prüfungsziffer. Die Basisnummer ist inklusive des

¹³ *Europäische Artikel-Nummer*

Ländercodes 7,8 oder 9 Stellen lang. Abzüglich der Prüfziffer bleiben dem Besitzer zur Eigengenerierung 3-5 Stellen. Diese können zum Beispiel für verschiedene Filialen, Betriebszweige oder Ähnliches verwendet werden.

Dem registrierten Teilnehmer ist es möglich, über die Webanwendung GLNs hinzufügen, zu editieren oder zu löschen (Abb. 13). Mindestens eine GLN muss jeder Teilnehmer besitzen. Für weitere Filialen oder Geschäftszweige können weitere GLNs hinzugefügt werden. Damit die einzelnen GLNs am System besser zugeordnet werden können besitzt jede GLN eine Bezeichnung, auch Kurzname genannt. Die Teilnehmer können über den Kurznamen frei verfügen, so lange dieser nicht bereits von einem anderen Teilnehmer verwendet wird. Innerhalb eines Teilnehmers darf der Kurzname mehrfach vorkommen.



Abb. 13: Es können GLNs hinzugefügt, entfernt und geändert werden.

4.6.2.2. Geschäftspartner definieren

Der Teilnehmer kann für jede GLN die Geschäftspartner definieren. Daten von Nicht-Geschäftspartnern werden von dem EDI-Hub-Server nicht zugestellt und erhalten im Track & Trace den Status „rejected“ (verworfen) mit entsprechender Fehlermeldung.

Die Geschäftspartner können über die Webanwendung auf verschiedenen Wegen definiert werden. Zum Einen können die Geschäftspartner zentral für einzelne GLNs oder für alle GLNs über das Eingabefeld bestimmt werden. Dafür wird die GLN-Nummer des Geschäftspartners in ein Formular eingegeben und über ein Auswahlfeld

die eigene GLN oder „Alle“ ausgewählt für die der Geschäftspartner hinzugefügt werden sollen. Die Eingabe der GLN-Nummer wird mit AJAX unterstützt, siehe Kapitel 4.6.8. Zum Anderen ist es möglich die gewünschte GLN über die Detailansicht der GLN, siehe Kapitel 4.6.7, als Geschäftspartner der eigenen GLNs zu definieren (Abb. 18).

Ferner können existierende Geschäftspartner entfernt werden. Das kann gezielt für eine einzelne GLN oder für alle GLNs durchgeführt werden. (Abb. 14)



Abb. 14: Geschäftspartner einer GLN können hinzugefügt und entfernt werden.

4.6.2.3. Geschäftspartner mit Datenüberprüfung

Für jede Geschäftsbeziehung kann eine ausführliche Validierung aktiviert werden. Diese Funktion ist im EDI-Hub-Server noch nicht implementiert, aber bereits vorgesehen und wird im Track & Trace protokolliert.

Das Definieren von Geschäftsbeziehungen, deren EDI-Daten einer ausführlichen Validierung unterzogen werden, wird in der Detail-Ansicht der GLN durchgeführt. Dort können Geschäftspartner für die gründliche Gültigkeitsprüfung markiert werden.

4.6.3. Firmen

Um sich einen Überblick über die registrierten Firmen zu verschaffen, können diese aufgelistet werden. Neben dem Firmennamen werden in der Liste weitere Informationen, wie der Ansprechpartner mit E-Mail Adresse und die Anzahl der GLNs, angezeigt.

Mit einem Klick auf den Firmennamen werden weitere Details angezeigt. So werden unter anderem die GLNs der Firma angezeigt und welche der eigenen GLNs als Geschäftspartner eingetragen ist. So kann im Vorfeld kontrolliert werden ob die eigene GLN als Geschäftspartner verwendet wird.

4.6.3.1. Suchen

Es ist nicht notwendig, sich alle registrierten Teilnehmer anzeigen zu lassen. Das Webinterface bietet eine Suchfunktion um nach einem Teilnehmer gezielt suchen zu können. Gesucht wird nach dem Firmennamen mit dem sich der gesuchte Teilnehmer registriert hat. Der Suchbegriff muss mindestens drei Zeichen lang sein, bevor die Suche beginnen kann.

Unterstützt der Browser Java, wird das Sucheergebnis mit Hilfe von AJAX bereits während der Eingabe angezeigt, wodurch eine schnellere und bequemere Suche möglich ist.

4.6.3.2. Akzeptieren Daten

Hier erhält der angemeldete Teilnehmer einen Überblick über die Teilnehmer, die Daten von einer seiner GLNs akzeptieren. Es werden alle Teilnehmer aufgelistet, bei denen mindestens eine GLN des angemeldeten Benutzers als Geschäftspartner eingetragen ist. Diese Funktion dient nur zur Übersicht, da nicht ersichtlich ist welche GLN Geschäftspartner welcher GLN ist. Eine genauere Auflistung erhält der angemeldete Benutzer unter: „GLNs -> akzeptieren Daten“.

4.6.4. GLNs

Analog zu der Auflistung der Firmen, werden hier die einzelnen GLNs aufgelistet, die am EDI-Hub-System registriert sind. Hier ist der Teilnehmer in der Lage gezielt nach GLNs zu suchen und sich einen Überblick zu verschaffen. Über jede GLN werden folgende Informationen angezeigt: GLN-Nummer, Kurzname, Firmennamen und der Ansprechpartner.

Detaillierte Informationen über eine GLN werden in der Detailansicht angezeigt. Zu dieser gelangt man mit einem Klick auf die GLN-Nummer. Mehr zur Detailansicht der GLN findet sich im Kapitel 4.6.7.

4.6.4.1. Suchen

Um die Suche zu vereinfachen kann nach GLN-Nummern gesucht werden. Alternative kann auch nach Kurznamen gesucht werden. Der Suchbegriff muss mindestens drei Zeichen lang sein bevor die Suche beginnen kann.

Unterstützt der Browser Java, wird das Sucheergebnis mit Hilfe von AJAX bereits während der Eingabe angezeigt, wodurch eine schnellere und bequemere Suche möglich ist.

4.6.4.2. Akzeptieren Daten

Hier erhält der angemeldete Teilnehmer alle GLNs aufgelistet, die Daten von wenigstens einer seiner GLNs akzeptieren. Diese Auflistung ist analog zu Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** mit dem Unterschied, dass die Auflistung einen genaueren Überblick darüber bietet, welche GLNs Daten, von wenigstens einer GLN des angemeldeten Benutzers akzeptieren. Genauere Details über vorhandene Geschäftsbeziehungen werden in der Detailansicht der GLN angezeigt, die über die GLN-Nummer erreichbar ist.

4.6.5. Track & Trace

Im Track & Trace können alle ein- und ausgehenden Daten kontrolliert werden. Für die Teilnehmer bietet das die Möglichkeit jederzeit den Status über die versendeten Daten abzufragen. Der Teilnehmer kann überprüfen ob die Daten ordnungsgemäß beim Adressat angekommen sind und vielleicht bereits abgeholt oder vom Hub verworfen wurden (Abb. 15). Um die Track & Trace Daten überschaubar zu halten kann man die Suche stark eingrenzen (Abb. 16). Aus Datenschutzgründen können Teilnehmer nur Track & Trace-Daten abrufen, in denen der angemeldete Teilnehmer selbst involviert ist. Dazu muss der Teilnehmer Sender oder Empfänger der EDI-Datei sein.

EDI-Hub Webinterface

Track & Trace

Parameter	Werte			
Datum	2006-12-18	2006-12-18	2006-12-18	2006-12-18
Uhrzeit	12:12:35	12:12:46	12:12:46	12:40:52
Dateiname	20061218_121234.edi	11.edi	11.edi	11.edi
Status	created	renamed	delivered	received
Mailbox	hu000069	hu000069	hu000069	
Sender	44444444444444	44444444444444	44444444444444	
Empfänger	1111111111109	1111111111109	1111111111109	
Seriennummer	1166440354.614	1166440354.614	1166440354.614	
Standard	UN/EDIFACT	UN/EDIFACT	UN/EDIFACT	
Nachrichtentyp				
Richtung	Ausgehend	Ausgehend	Ausgehend	
Fehlercode				
Fehlertext				
#Bytes	1924	1924	1924	
MD5-Prüfsumme	c1522ee5106a6e7e 45770b603f023898	c1522ee5106a6e7e 45770b603f023898	c1522ee5106a6e7e 45770b603f023898	

Startseite | XHTML 1.1 | CSS2 | Design | Impressum

Abb. 15: Track & Trace Detailansicht einer Versendeten EDI-Datei.

4.6.5.1. Filter

Mit Hilfe der Filterfunktion lassen sich die Track & Trace-Daten eingrenzen. Das Eingrenzen der Track & Trace-Daten verschafft nicht nur einen besseren Überblick, sondern verringert auch die benötigte Rechenzeit der Darstellung. Zum Eingrenzen stehen verschiedene Filtermöglichkeiten zur Verfügung:

Filteroption	Beschreibung
Eigene GLN	Grundsätzlich werden nur Track & Trace-Daten mit den eigenen GLNs angezeigt. Die Auflistung kann auf eine einzige GLN begrenzt werden. Mögliche Optionen: Alle, eine eigene GLN.
Zeitraum	Der Zeitraum der angezeigten Track & Trace-Daten. Mögliche Optionen: Letzte Stunde, die letzten 6 Stunden, die letzten 12 Stunden, letzter Tag, letzte Woche, letzter Monat, letztes Quartal
Richtung	Hier können Track & Trace-Daten mit einer bestimmten Richtung gefiltert werden. Mögliche Optionen: Eingehend, Ausgehend, Alle
EDI Standard	Hier können die verschiedenen EDI-Standards gefiltert werden. Mögliche Optionen: UN/EDIFACT, Alle
Nachrichtentyp	Nur Track & Trace-Daten mit dem ausgewählten Nachrichtentyp

	werden angezeigt. Bisher sind keine Nachrichtentypen implementiert. Mögliche Optionen: Alle
Partner-GLN	Um Track & Trace-Daten von einem bestimmten Geschäftspartner zu finden kann diese als Partner-GLN eingegeben werden. Dann werden nur die Track & Trace-Daten angezeigt, in denen die eingegebene Partner-GLN vorkommen.
Sortierung	Das Suchergebnis chronologisch aufsteigend oder absteigend sortieren.
Ereignisse	Nach Ereignissen filtern. Mögliche Optionen: Alle, Normale, Noch nicht beendete, Nur Fehler

Mit den oben genannten Filteroptionen ist es möglich gezielt nach Track & Trace-Daten zu suchen (Abb. 16).

Ein Beispiel:

Ein Teilnehmer möchte wissen ob die Daten, die er vor drei Tagen an die Karstadt AG mit der GLN-Nummer 1212121212121 geschickt hat, zugestellt und von dem Empfänger abgeholt wurde. Um die Suche zu vereinfachen, werden die in der Abbildung dargestellten Filteroptionen verwendet. (Abb. 16)

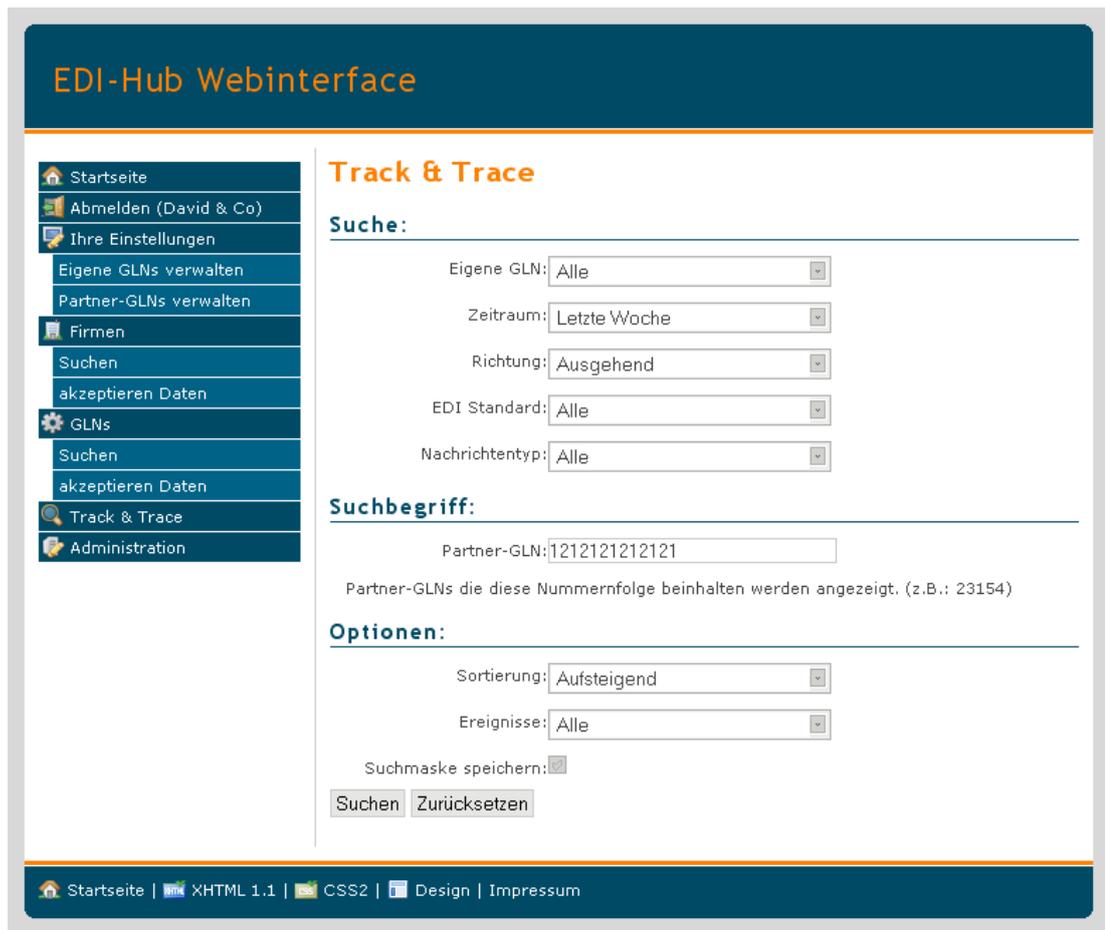


Abb. 16: Das Track & Trace bietet verschieden Filtermöglichkeiten.

Es besteht die Möglichkeit die eingestellten Suchoptionen als Standardsuchmaske abzuspeichern. Damit kann ein Teilnehmer die am häufigsten benutzen Filteroptionen mit seinen Standardwerten füllen und muss diese nicht bei jeder Suche erneut eingeben.

4.6.6. Administration

Nur der Administrator kann am Webinterface die Administration durchführen. Die Administration ist bisher sehr spartanisch ausgestattet. Bisher ist es dem Administrator nur möglich registrierte Teilnehmer aus EDI-Hub-System entfernen (Abb. 17). Das Entfernen eines Teilnehmers ist endgültig und kann nicht rückgängig gemacht werden. Der Teilnehmer wird aus der Datenbank entfernt und kann sich damit nicht mehr am Webinterface und am FTP-Server anmelden. Sein Mailbox-Verzeichnis wird nicht gelöscht sondern als gelöscht markiert. Das Löschen des Benutzerverzeichnisses muss vom Systemadministrator manuell durchgeführt werden. Damit wird verhindert, dass der EDI-Hub-server aktiv am Dateisystem

löscht und das Dateisystem eventuell beschädigt. Außerdem sind dadurch die Daten des Teilnehmers nicht verloren, falls die Löschung unbeabsichtigt war.

Es ist denkbar den Administrationsbereich funktional zu erweitern. Zum Beispiel dass einzelne Teilnehmer gesperrt werden können.



Abb. 17: Administrationsbereich zum entfernen von Teilnehmern.

4.6.7. GLN Detailansicht

Zur detaillierten Ansicht einer GLN gelangt man von fast allen Seiten. In der Regel besitzt eine GLN-Nummer einen Link zur Detailansicht. So zum Beispiel bei der Auflistung der registrierten GLNs, die über den Menüeintrag GLNs dargestellt wird. Die GLN-Nummern in der Liste besitzen einen Link zur Detailansicht.

Die Webanwendung unterscheidet in der Detailansicht zwischen fremde und eigene GLNs.

Fremde GLN:

Über die Detailansicht einer GLN kann diese als Geschäftspartner der eigenen GLNs hinzugefügt werden. Dazu werden die eigenen GLNs aufgelistet, aus denen die GLNs ausgewählt werden können, von denen die angezeigte GLN der Geschäftspartner ist.

Ferner werden die Geschäftspartner der GLN angezeigt. Damit ist auf einem Blick ersichtlich, welche eigenen GLNs als Geschäftspartner eingetragen sind. Aus

Datenschutzgründen werden alle GLNs ausgeblendet, die nicht dem angemeldeten Benutzer gehören. (Abb. 18)

EDI-Hub Webinterface

GLN: 4334400000008 (Karstadt)

GLN

Firmenname: [Karstadt AG](#)
GLN: 4334400000008
Kurzname: Karstadt

Partner-GLNs

Folgende Ihrer GLNs sind berechtigt an 4334400000008 (Karstadt) Daten zu schicken.

GLN	Kurzname
4444444444444	David & Co Italy

Zu Ihren GLNs als Partner-GLN hinzufügen/entfernen

Hier können Sie die GLN 4334400000008 (Karstadt) bei Ihren GLNs als Geschäftspartner hinzufügen oder entfernen.

GLN	Kurzname
<input type="checkbox"/> 3333333333338	David & Co Germany
<input type="checkbox"/> 4444444444444	David & Co Italy

Startseite | XHTML 1.1 | CSS2 | Design | Impressum

Abb. 18: Detailansicht einer GLN mit Informationen über Geschäftspartner und die Möglichkeit die GLN zu den eigenen GLNs als Geschäftspartner hinzuzufügen/zu entfernen.

Eigene GLN:

In der Detailansicht einer eigenen GLN werden deren Geschäftspartner aufgelistet. Hier können einzelne Geschäftsbeziehung für die ausführliche Validierung der EDI-Daten ausgewählt werden. Daneben werden die GLNs aufgelistet, die Daten von dieser GLN akzeptieren. (Abb. 19)

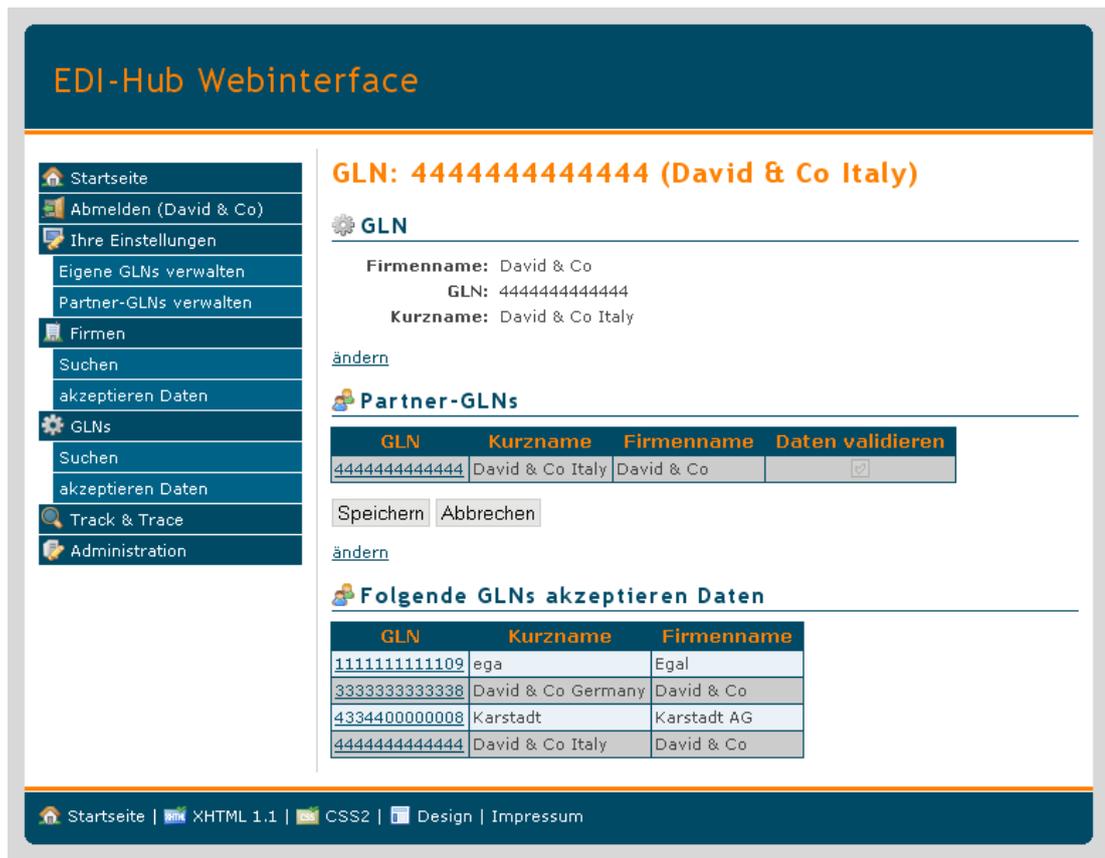


Abb. 19: Detailansicht einer eigenen GLN mit erweiterten Informationen der Geschäftspartner und der Möglichkeit die erweiterte Validierung pro Geschäftspartner zu aktivieren/entfernen.

4.6.8. Ajax

Um die Bedienbarkeit der Webanwendung zu erhöhen, sind einige AJAX-Elemente in die Webanwendung eingeflossen. Dabei ist darauf geachtet worden, dass diese Elemente die Bedienung der Webanwendung erleichtert, aber nicht zwingend erforderlich machen.

So findet AJAX zum Beispiel als Eingabehilfe, auch als Autovervollständigung bekannt, Verwendung. Es ist zum Beispiel ausreichend bei dem hinzufügen eines Geschäftspartners einen Teil der gewünschten GLN-Nummer oder sogar des GLN-Kurznamens einzugeben, worauf eine Auswahl der infrage kommenden GLNs erscheint, unter denen ausgewählt werden kann. (Abb. 20)

Partner-GLNs verwalten

Partner-GLN hinzufügen

Hier können weitere Partner-GLNs hinzugefügt werden.

Partner-GLN: für

 **4334400000008**
Karstadt (Karstadt AG)

 **4444444444444**
David & Co Italy (David & Co)

Eine GLN ist in der Partner-Liste vorhanden und wird grau dargestellt.

Abb. 20: Eingabefeld mit AJAX gestützter Autovervollständigung.

Weiterhin wurde AJAX verwendet um die Reaktionszeit der Webseite zu erhöhen. So werden bei Suchmasken die Suchergebnisse bereits während der Eingabe des Suchbegriffs aktualisiert, wodurch ein schnelleres und effizienteres Arbeiten ermöglicht wird. Damit die Suchfunktion auch funktioniert falls AJAX deaktiviert ist, ist es weiterhin möglich die Suche mit dem Button „Suchen“ zu starten. (Abb. 21)

Firma suchen

Sich suchen einen bestimmte Firma? Geben sie einfach den Firmennamen ein

Hinweis: Es werden maximal 20 Suchergebnisse angezeigt.

Firma:

Gefunden: (1)

Firmenname	Kurznamen	Ansprechpartner	#GLN
Karstadt AG	Karstadt	David Böhm	1

Abb. 21: Aktualisierung des Suchergebnisses während der Eingabe.

Für eine optimale und komfortable Bedienung ist die Verwendung eines Java-fähigen Browsers empfehlenswert.

4.7. Verwendete Programme

Neben dem EDI-Hub-Server und der Webanwendung sind für den Betrieb des EDI-Hub-Systems weitere Programme notwendig.

4.7.1. Apache2

Apache2 wird als Webserver verwendet auf dem die Webanwendung läuft. Apache2 ist bei Debian bereits seit einigen Versionen als Paket verfügbar und gut in das System integriert. Außerdem existiert für Apache2 bereits eine Vielzahl von Erweiterung, darunter auch FCGLI, womit die Ausführung und Reaktionszeit der Webanwendung gesteigert wird.

Es ist nicht zwingend notwendig Apache2 als Webserver zu verwenden. Es bleibt dem Betreiber überlassen. Voraussetzung ist die Integrationsmöglichkeit der Ruby on Rails Webanwendung.

4.7.2. Proftpd

Für die Verwendung von Proftpd als FTPS-Server, zum Hoch- und Runterladen der EDI-Daten, gibt es folgende Gründe:

- Unterstützung von FTPS
Durch die Verwendung von FTPS können sowohl die Steuerdaten sowie Nutzdaten verschlüsselt werden.
- Eingebaute chroot-Umgebung
Die Teilnehmer werden in Ihrer Mailbox eingesperrt und können nicht EDI-Daten andere Teilnehmer herunterladen. Es ist nicht notwendig einen chroot-Umgebung per Hand herzustellen wie es bei SFTP über SSH der Fall wäre.
- Zugriffsvermeidung des EDI-Hub-Severs während Datei-Upload
Während dem Datei-Upload werden die Dateien umbenannt und erhalten ihren Ursprungsnamen nach Beendigung des Uploads. Dadurch werden die Dateien vom EDI-Hub-Server ignoriert. Die Namensänderung sieht wie folgt aus: „.in.<Dateiname>.“

4.7.3. Mysql

MySQL ist de facto Standard für datenbankgestützte Webanwendungen und ist unter nicht kommerziellen Datenbanken sehr performant. Zusätzlich existieren für MySQL in fast jeder Programmiersprache native Anbindungsmöglichkeiten für eine eventuelle Anbindung von externen Programmen.

5. Test

5.1. EDI-Hub-Server

Um den EDI-Hub-Server zu testen wurde eine Test-Suite programmiert, die EDI-Dateien mit unterschiedlichen Inhalten generiert und in die einzelnen Mailbox-Verzeichnisse zum Versenden ablegt. Neben dem Generieren von EDI-Dateien werden Störfälle, zum Beispiel ungültige EDI-Dateien, nachgebildet.

Zum Generieren von EDI-Dateien wird das Programm `gen_edifact_orders.rb` das mit Hilfe der Bibliothek `edi4r` UN/EDIFACT-Dateien erstellt. Sowohl die Bibliothek als auch das Programm `gen_edifact_order.rb` wurden von Prof. Dr. Werntges geschrieben und zur Verfügung gestellt. Zum Generieren der ENACOM'02 Bestellung können folgende Parameter definiert werden:

- Sender-GLN
- Empfänger-GLN
- Interchange Reference Number
- Anzahl Bestellungen
- Anzahl Bestellpositionen

Durch die Steuerung der Häufigkeit, der Dateigröße und durch variieren der Bestellungen und Bestellpositionen der ENACOM'02 Bestellung, ist ein Massentest leicht skalierbar.

Die Testsuite `hub_test_suite.rb` benötigt zum Erstellen der Test-Dateien mittels `gen_edifact_order.rb`, Informationen über die im System vorhandenen Benutzer, die zum Testen verwendet werden. Diese Informationen werden innerhalb des Programms definiert:

- Mailbox-Bezeichnung
Eine Liste der Mailboxen, die in den Test mit einbezogen werden.
- Sender-GLNs
Für jede Mailbox gibt es eine Liste von GLNs, aus der zufällig eine GLN als Sender-GLN ausgewählt wird.
- Empfänger-GLNs
Für jede Mailbox gibt es eine Liste von GLNs, aus der zufällig eine GLN als Empfänger-GLN ausgewählt wird.

Mit den obengenannten Informationen generiert die Test-Suite eine bestimmbare Anzahl an EDI-Dateien und stellt diese zum Verschicken in die „out“-Ordner der Mailboxen bereit.

Um die Robustheit des EDI-Hub-Servers zu testen, werden auch ungültige und fehlerhafte EDI-Dateien erzeugt:

- Leere Dateien
Leere Dateien sollen verarbeitet werden und im Track & Trace erscheinen.
- Syntaxfehler in EDI-Dateien
Ungültige EDI-Dateien dürfen nicht verschickt werden.
- Ungültige Sender-GLN verwenden
Hiermit wird die Integrität der EDI-Dateien getestet. Es dürfen nur EDI-Dateien verschickt werden, deren Sender-GLN zur sendenden Mailbox gehören.
- Ungültige Empfänger-GLN verwenden
Es werden solche EDI-Dateien abgelehnt, deren Empfänger nicht existiert oder deren Absender nicht als Geschäftspartner beim Empfänger eingetragen ist und daher nicht berechtigt ist EDI-Dateien zu schicken.
- EDI-Datei doppelt erstellen
EDI-Dateien die doppelt verschickt werden, müssen erneut zugestellt werden und entsprechend im Track & Trace erscheinen. Für eine bessere Übersicht werden diese Dateien im Track & Trace zusammengefügt.

Durch die Test-Suite lassen sich sehr schnell eine große Anzahl EDI-Dateien generieren und in die „out“-Verzeichnisse der Teilnehmer verschieben. Damit erreicht man eine hohe Vielfalt an Möglichkeiten, um den Hub-Server zu testen. Dieser Massentest bringt Fehler sehr schnell zum Vorschein. Vor allem die Track & Trace-Funktion kann, durch die große Menge an gültigen und ungültigen EDI-Daten, sehr gut getestet werden.

Für das Erstellen von 20 EDI-Dateien genügt folgender Aufruf:

```
hub_test_suite.rb 20
```

5.2. Webanwendung

Durch den Einsatz von Ruby on Rails kann die Webanwendung mit vorgesehenen Mitteln getestet werden. So können die Models, die die Datensätze abbilden und die Controller, die die Funktionalitäten beinhalten, auf deren korrekte Funktionalität getestet werden. Bei den Controller werden die einzelnen Anfragen in Funktionen aufgeteilt. Nach Änderungen oder Ergänzungen an der Webanwendung können alle Funktionen und Modelle in wenigen Sekunden getestet werden. Umso ausführlicher die Tests geschrieben werden und umso mehr Varianten durchgespielt werden, desto höher ist die Wahrscheinlichkeit, dass Fehler ausgeschlossen werden können. Diese Tests könnten auch per Hand durchgeführt werden, würden aber wesentlich mehr Zeit in Anspruch nehmen und wären nicht so ausführlich. Da für alle Models und Controller sorgfältige Tests angelegt wurden, kann, bei einem fehlerfreien Durchlauf der Tests, davon ausgegangen werden, dass die Webanwendung korrekt funktioniert.

Dank Ruby on Rails gibt es mehrere Möglichkeiten die Webanwendung zu testen:

- Unit-Tests
Testet die Models auf Funktionalität
- Functional-Tests
Testet den Ablauf der Controller

Ruby on Rails verwendet zum Testen der Anwendung eine eigene Datenbank und ist im Ruhezustand stets leer. Sie wird für den Test verwendet und vor jedem Test mit Daten gefüllt. Um das Befüllen von definierten Daten nicht bei jeder Testanweisung zu wiederholen gibt es die so genannten „fixtures“. Die „fixtures“ bilden für jedes Model einzeln einen oder mehrere Datensätze ab, die dann separat geladen werden können.

Nachfolgend einige konkrete Testbeispiele der Webanwendung:

- Sicherstellen, dass der Firmenname aus mindestens drei Zeichen besteht und keine Sonderzeichen enthält.
- Sicherstellen, dass die einzelnen Registrierungsschritte fehlerfrei funktionieren.

- Sicherstellen, dass beim Löschen einer Firma auch alle GLNs und Geschäftsbeziehungen der Firma gelöscht werden.
- Sicherstellen, dass eine GLN stets sein eigener Geschäftspartner ist.

Es ist möglich die Tests einzeln oder alle auf einmal zu starten. Um einzelne Tests zu starten, genügt es die Datei direkt auszuführen. Mit folgendem Aufruf im Verzeichnis der Webanwendung wird zum Beispiel das Model `users` getestet:

```
ruby test/units/user_test.rb
```

Es lassen sich auch Controller testen:

```
ruby test/functional/user_controller_test.rb
```

Um alle Tests zu starten, dazu gehören alle Tests, die sich im Verzeichnis `test` befinden, genügt folgender Aufruf:

```
rake test
```

6. Diskussion

Zu Beginn war es geplant die EDI-Hub-Teilnehmer als Betriebssystembenutzer abzubilden. Allerdings wäre das EDI-Hub-System dadurch sehr betriebssystemabhängig geworden und hätte spezieller Anpassungen bedurft. Da für Ruby noch keine PAM¹⁴-Bibliothek verfügbar war, die die Authentifizierung von Betriebssystembenutzern ermöglicht hätte, wurden die Teilnehmer nicht als Betriebssystembenutzer abgebildet. Dadurch können den EDI-Hub-Teilnehmern nur Dienste angeboten werden, die eine datenbankgestützte Authentifizierung anbieten. Alternative wäre es möglich gewesen das Betriebssystem, mit Hilfe des PAM-MySQL-Modules, auf eine datenbankgestützte Authentifizierung umzustellen. Dann wären die Teilnehmer in der Lage gewesen Dienste in Anspruch zu nehmen, die die Authentifizierung über das Betriebssystem vornehmen, wie zum Beispiel SSH. Das würde die Installation des EDI-Hub-Systems zusätzlich erschweren und wurde daher nicht mit einbezogen.

Die Authentifizierung der EDI-Hub-Teilnehmer findet nicht auf Betriebssystem-Ebene statt, sondern erfolgt durch die Webanwendung und dem FTPS-Server über die Datenbank.

Ein Mehrfachbetrieb ist durch die Verwendung einer datenbankgestützter Benutzerverwaltung möglich, wobei jeder EDI-Hub-Server eine eigene Datenbank zugewiesen bekommt. So lassen sich auf einem Server mehrere EDI-Hub-Server betreiben. Allerdings treten dadurch bei der Webanwendung Schwierigkeiten auf.

Jeder EDI-Hub-Server benötigt jeweils eine eigene Webanwendung über die sich die Teilnehmer verwalten können. Damit die einzelnen Webanwendungen über eine IP-Adresse aber unterschiedlichen Domänen-Namen erreichbar sind, ist der Betrieb von virtuellen Servern, etwa bei dem Apache 2 Web-Server über den „virtual host“-Mechanismus, notwendig. Mit virtuellen Servern können auf einem Webserver mehrere Domänen-Namen abgebildet werden. Der Webserver ruft je nach angefordertem Domänen-Namen die entsprechende Webanwendung auf. Allerdings ist das nur mit einer unverschlüsselten Verbindung möglich, da dem Webserver der Domänen-Namen erst nach dem Verbindungsaufbau bekannt ist. Die Zertifikate sind

¹⁴ *Pluggable Authentication Modules zur Authentifizierung von Betriebssystembenutzer.*

nur für einen Domänen-Name gültig und der Webserver müsste für die angeforderte Domäne das entsprechende Zertifikat verwenden. Das Zertifikat wird aber bereits für den Verbindungsaufbau benötigt. Zu diesem Zeitpunkt ist dem Webserver der Domänen-Name aber noch nicht bekannt und kann daher nicht das Zertifikat für die angeforderte Domäne verwenden. Mit der in Planung befindlichen Spezifikation TLS 1.2 soll dies mittels „Server Name Indication“ ermöglicht werden.

Als Lösung für diese Problem könnte zum Einem das parallele Betreiben von mehreren Webserver auf unterschiedlichen TCP Ports sein. Und zum Anderen eine Anpassung der Webanwendung für die Bedienung mehrerer EDI-Hub-Server.

Für einen Mehrfachbetrieb des EDI-Hub-Servers wird dagegen nur ein FTPS-Server benötigt, da die Verschlüsselte Authentifizierung implizit nach der Verbindungsherstellung geschieht. Proftpd ist in der Lage für die angesprochene Domäne das entsprechende Zertifikat zu verwenden. Zusätzlich kann für jeden virtuellen Server eine separate Datenbank für die Authentifizierung definiert werden. Durch die Verwendung von separaten Datenbanken entsteht eine Trennung der Benutzergruppen.

Als Fazit bleibt zu sagen, dass der Mehrfachbetrieb trotz Datenbankgestützter Benutzerverwaltung nicht einfach umzusetzen ist, beziehungsweise die Akzeptanz des EDI-Hub-Systems herabsetzen würden, solange der Webserver keine virtuelle Server mit SSH-Verschlüsselung erlaubt. Durch die niedrigen Server-Kosten ist es ratsamer für jedes weitere EDI-Hub-System einen eigenen Server zu mieten.

In den folgenden Punkten werden mögliche Erweiterungen des EDI-Hub-System erläutert.

6.1. Erweiterung um einen AS2 Gateway

Viele große Unternehmen haben ihre EDI bereits auf EDI-INT AS2 umgestellt. Daher bietet es sich an das EDI-Hub-System um einen EDI-INT AS2 Gateway zu erweitern. Dadurch könnte sich der EDI-Hub-Betreiber, ohne größere Anpassungen ihrer EDI-Software, an das EDI-Hub-System anbinden und die Kosten für den Betrieb weiter senken.

6.2. Komprimierung der eingehenden EDI-Dateien

Komprimierte EDI-Dateien können vom EDI-Hub-Server bereits verarbeitet werden. Eine Erweiterung könnte sein, dass jeder Teilnehmer für sich bestimmt, in welchem

Komprimierungsformat die EDI-Dateien in seinem „out“-Ordner liegen. Der Teilnehmer könnte dann zwischen folgenden Optionen in der Webanwendung wählen:

- Unberührt
Die EDI-Dateien werden nicht verändert und behalten das Komprimierungsformat des Versenders.
- Unkomprimiert.
Die EDI-Dateien werden unkomprimiert gespeichert, auch wenn sie vom Versender komprimiert verschickt wurde.
- GZip
Die EDI-Dateien werden unabhängig vom ursprünglichen Komprimierungsformat in das GZip-Komprimierungsformat gebracht.
- BZip2
Die EDI-Dateien werden unabhängig vom ursprünglichen Komprimierungsformat in das BZip2-Komprimierungsformat gebracht.

6.3. Archivierung der EDI-Dateien

Das EDI-Hub-System könnte um eine Archivierungsfunktion für EDI-Dateien im „out“-Ordner und „in“-Ordner erweitert werden. So könnten die Dateien nach dem Zustellen nach „out-sent“ verschoben werden. Damit erhielte der Teilnehmer die Möglichkeit, falls notwendig, eine EDI-Datei erneut zu verschicken. Für das Archivieren von Dateien des „in“-Ordners wäre der Eingriff des Teilnehmers gefragt, da der EDI-Hub-Server nicht feststellen kann welche Dateien vom Teilnehmer bereits gelesen wurden. So könnte der Teilnehmer, die in dem „in“-Ordner befindlichen Dateien nach dem Herunterladen in den „in-receive“-Ordner verschieben, anstatt diese Dateien zu löschen.

Auch eine Langzeitarchivierung der Dateien, für einen einstellbaren Zeitraum, durch Komprimierung wäre denkbar.

Allerdings muss man bedenken, dass durch die Archivierungsfunktion der benötigte Festplatten-Platz sehr stark ansteigen würde.

6.4. Statussätze für jede Mailbox

Als Erweiterung des EDI-Hub-Systems wäre eine Datei mit Statusätzen für jede Mailbox denkbar, die alle Track & Trace-Informationen der jeweiligen Mailbox beinhalten würde. Dort ständen alle Informationen über aus- und eingehende Dateien, ähnlich wie im Track & Trace. Die Teilnehmer hätten dadurch die

Möglichkeit diese Datei periodisch herunter zu laden und in die interne EDV einzubinden. Dadurch wäre eine webinterface-unabhängige Track & Trace-Funktion möglich. Weiterhin könnten diese Informationen für Statistiken, zu Datenvolumen oder Anzahl der verschickten Daten, herangezogen werden.

Es wäre zum Beispiel eine Anbindung an eine SAP-Anwendung denkbar. Dafür müsste aus den Track & Trace-Informationen IDOC-Statussätze generiert werden, die eine Track & Trace-Funktion innerhalb der SAP-Anwendung ermöglichen würde.

6.5. Vereinfachung der Installation

Bei hoher Akzeptanz des EDI-Hub-System könnte über eine einfachere Installationsvariante nachgedacht werden. Denkbar wäre die Verwendung von einem Installationsscript welches, alle benötigten Programm-Pakete installiert und die Konfigurationen der einzelnen Programme nach den Angaben des Anwenders automatisiert vornimmt. Dadurch wäre eine manuelle Anpassung der Konfigurationsdateien durch den Benutzer überflüssig.

Bei Verwendung der Linux Distribution Debian könnte diese Aufgaben durch ein Debian Paket übernommen werden.

6.6. Versenden von Nicht-UN/EDIFACT-Daten

Bisher können nur UN/EDIFACT-Daten verschickt werden, da die Adressierung mit Hilfe der edi4r Bibliothek ausgelesen wird. Zukünftig könnte der Versand von Nicht-UN/EDIFACT-Daten durch die Adressierung mittels des Dateinamens, inklusive der Interchange Reference Number, erfolgen. Der Dateiname könnte dann folgendes Format haben: <s>_<r>_<n>.edi. Wobei <s> die Sender GLN, <r> die Empfänger GLN und <n> die Interchange Referenze Number darstellt.

7. Zusammenfassung und Ausblick

In dieser Ausarbeitung wurde ein homogenes EDI-Hub-System zum Verteilen von EDI-Dateien entwickelt. Dabei wurde viel Wert auf die Einfachheit, Sicherheit und die Verwendung von Internet-Standards gelegt.

Ein großes Unternehmen ist mit diesem EDI-Hub-System in der Lage, seinen Geschäftspartnern, die aus Kostengründen bisher kein EDI verwenden, eine Plattform anzubieten. Da den Teilnehmern, Dank der Verwendung von vorhandenen Internetstandards, nur geringe Kosten entstehen, wird ihnen der Einstieg in den automatisierten Datenaustausch erleichtert.

Die Übertragung der EDI-Dateien als auch die Benutzerverwaltung benötigen keine dedizierte Hardware. Die Teilnehmer müssen ihre EDV-Struktur nur geringfügig anpassen, um am EDI-Hub-System teilnehmen zu können.

Die Kosten für den Betrieb des EDI-Hub-Systems sind gering und sollten sich, durch die Umstellung der Geschäftspartner auf EDI, die bisher ihre Geschäftsabwicklungen manuell durchführten, schnell amortisiert haben.

Das EDI-Hub-System könnte eine Brücke zwischen den großen Unternehmen und den kleinen und mittleren Unternehmen schlagen.

8. Literaturverzeichnis

- ❖ [Lit01]
 - Titel: UN/EDIFACT
 - Url: http://www.gs1-germany.de/internet/content/produkte/ean/ecommerce_edi/un_edifact/index_ger.html
 - Datum: 20.12.2006

- ❖ [Lit02]
 - Titel: UN/EDIFACT
 - Url: <http://de.wikipedia.org/wiki/EDIFACT>
 - Datum: 20.12.2006

- ❖ [Lit03]
 - Titel: Value-Added Networks VAN
 - Url: <http://www.123edi.com/edi-van.asp>
 - Datum: 28.12.2006

- ❖ [Lit04]
 - Titel: Mehrwertnetzwerk
 - Url: http://de.wikipedia.org/wiki/Value_Added_Network
 - Datum: 28.12.2006

- ❖ [Lit05]
 - Titel: AS2
 - Autor: D. Moberg, R. Drummond
 - Url: <http://tools.ietf.org/html/rfc4130>
 - Datum: 30.12.2006

- ❖ [Lit06]
 - Titel: IT-Sicherheit. Konzepte - Verfahren - Protokolle
 - Autor: Claudia Eckert
 - Verlag: Oldenbourg, 2006

- ❖ [Lit07]
 - Titel: Burn, FTP, burn!
 - Autor: Sven Höxter, Stephan Rottmann
 - Url: http://www.pro-linux.de/t_netzwerk/burn-ftp-burn.html
 - Datum: 3.1.2007

- ❖ [Lit08]
 - Titel: Agile Web Development with Rails
 - Autor: Dave Thomas, David Heinemeier Hansson
 - Verlag: Pragmatic Programmers, 2005

- ❖ [Lit09]
 - Titel: Zugmaschine
 - Untertitel: Web 2.0: Ajax mit Ruby on Rails
 - Autor: Till Vollmer, Andreas Gehret
 - Verlag: Heise Zeitschriften Verlag GmbH
 - Ausgabe: iX 4/2006, S. 143-146

- ❖ [Lit10]
 - Titel: EDI Tracking & Tracing
 - Autor: Prof. Dr. H. Werntges
 - Verlag: Beuth-Verlag
 - Ausgabe: edi-change 03/2001, S. 26-34

- ❖ [Lit11]
 - Titel: Professionelles Webdesign mit (X)HTML und CSS
 - Autor: Björn Seibert, Manuela Hoffmann
 - Verlag: Galileo Computing, 2006

9. Glossar

ACL	Access Control List
AS2	Applicability Statement 2
CSV	Comma Separated Values
EDI	Electronic Data Interchange
FTPS	File Transfer Protocol Secure
GLN	Global Location Number
HTTPS	HyperText Transport Protocol over SSL/TLS
IDE	Integrated development environment
KMU	Kleine und mittlere Unternehmen
MVC	Model Controller View
UN/EDIFACT	United Nations Electronic Data Interchange For Administration, Commerce and Transport.
VANS	Value Added Network System
YAML	YAML Ain't Markup Language

10. Anhang

10.1. Installationsanleitung

Diese Installationsanleitung orientiert sich an der Linux Distribution Debian 3.1 Sarge.

10.1.1. Notwendige Programme installieren und einrichten

Das Hub-System besteht neben dem Hub-Server und der Webanwendung aus diversen zusätzlichen Programmen. Die meisten zusätzlich benötigten Programme sind bei Debian als Paket verfügbar und benötigen nicht viel Arbeit für deren Installation. Als Erstes wird das Paketverwaltungssystem von Debian mit folgendem Befehl aktualisiert:

```
apt-get update
```

Anschließend können die Benötigten Programme installiert werden. Eventuell auftretende Abhängigkeiten müssen mit „J“ bestätigt werden.

10.1.1.1. Ruby

Da der Hub und die Webanwendung in Ruby programmiert sind, muss der Ruby Interpreter mit einigen Bibliotheken installiert werden:

```
apt-get install ruby rdoc eruby libdbi-ruby1.8 libzip-ruby1.8 \  
libdbm-ruby1.8 liberuby libfcgi-ruby1.8 libmysql-ruby1.8 \  
libwebrick-ruby1.8
```

10.1.1.2. RubyGems

RubyGems ist leider nicht als Paket verfügbar und muss von Hand installiert werden. RubyGem liegt in der Version 0.9 im Installations-Ordner bereit. Die neueste Version findet sich unter: http://rubyforge.org/frs/?group_id=126

Nach dem Extrahieren wird es mit folgendem Aufruf installiert:

```
ruby setup.rb
```

10.1.1.3. EDI Bibliotheken

Die EDI Bibliotheken von Dr. Werntges liegen der Installations-CD bei und werden wie folgt installiert:

```
gem install edi4r-0.9.4.1.gem edi4r-tdid-0.6.3.1.gem
```

10.1.1.4. Ruby on Rails

Ruby on Rails und alle seine Abhängigen Pakete werden mit RubyGems installiert:

```
gem install rails
```

10.1.1.5. Datenbank

Die Mysql Datenbank in der Version 4.0 wird wie folgt installiert:

```
apt-get install mysql-server
```

Nach der Installation kann die Datenbank gestartet werden mit:

```
/etc/init.d/mysql start
```

Die notwendige Datenbankstruktur befindet sich in der Datei hubd.sql und wird folgendermaßen übernommen:

```
mysql -uroot < hubd.sql
```

Für den Datenbankzugriff vom EDI-Hub-Server und der Webanwendung wird ein Datenbankbenutzer angelegt. In der Datei hubd-user.sql stehen die SQL-Anweisung zum erstellen des Datenbankbenutzers mit den notwendigen Rechten. Es ist jedoch empfehlenswert in dieser Datei das Passwort zu ändern. In den weiteren Schritten ist darauf zu achten, dass das neu gewählte Passwort verwendet wird. Das neue Passwort kann in der Datei hubd-user.sql geändert werden und anschließend wie folgt installiert werden:

```
mysql -uroot < hubd-user.sql
```

Damit ist die Datenbank für das EDI-Hub-System vorbereitet.

10.1.1.6. Webserver

Wurde der Apache2-Webserver nicht bei der Installation von Debian installiert, kann er mit folgendem Kommando installiert werden:

```
apt-get install apache2 libapache2-mod-fastcgi
```

Für eine akzeptable Geschwindigkeit muss das Fast-CGI-Modul aktiviert werden mit:

```
a2enmod fastcgi
```

Die Standardkonfigurationen stehen in `/etc/apache2/sites-available/default`. Damit Ruby on Rails funktioniert muss `DocumentRoot` auf `/var/www/webinterface/public` gesetzt werden und `AllowOverride` auf `All` stehen.

10.1.1.7. FTPS-Server

Für das EDI-Hub-System wird Proftpd als FTPS-Server mit Mysql-Erweiterung eingesetzt und wie folgt installiert:

```
apt-get install proftpd-mysql
```

Nach der Installation muss Proftpd für die Authentifizierung mittels der Datenbank und der Benutzung eines Zertifikats konfiguriert werden. Eine entsprechend angepasste Konfigurationsdatei liegt dem Paket bei und lautet `proftpd.conf`. Bevor die angepasste Konfigurationsdatei nach `/etc/proftpd.conf` kopiert wird ist es ratsam die ursprüngliche Konfigurationsdatei zu sichern. Wenn das Passwort für den Datenbankbenutzer bei der Datenbankeinrichtung geändert wurde muss es in `proftpd.conf` angepasst werden.

Für eine sichere Verbindung über FTPS wird ein Zertifikat benötigt. Proftpd erwartet das Zertifikat in `/etc/ssl/ftpcert/` und müssen `host.cert` und `host.key` heißen.

Mit `/etc/init.d/proftpd restart` wird Proftpd mit der neuen Konfiguration gestartet.

10.1.2. Hub-Server installieren und einrichten

Zunächst werden die einzelnen Programme im Betriebssystem integriert. Debian sieht dafür `/usr/local` vor. Es bietet sich daher an folgende Dateistruktur zu verwenden:

`/usr/local/bin:`

- `hubd.rb`
- `hubdctrl.rb`
- `tt_record.rb`

`/usr/local/etc:`

- `hubd-conf.yml`

Beim Hub-Kontroller `hubdctrl.rb` ist darauf zu achten, dass dieser vom Webserver gestartet werden darf, da ansonsten Benutzeränderungen über das Webinterface nicht vom EDI-Hub registriert werden können. Damit nur der Webserver die Datei ausführen darf wird am besten der Benutzer mit

```
chown www-data:www-data /usr/local/bin/hubdctrl.rb
```

geändert.

In der Konfigurationsdatei `hubd-conf.yml` muss gegebenenfalls das Passwort für den Zugriff auf die Datenbank geändert werden.

Damit die Track & Trace Daten dem Web-Interface zur Verfügung stehen, müssen

diese in die Datenbank geladen werden. Das Programm `tt_record.rb` schreibt die vom EDI-Hub gespeicherte Track & Trace Daten in die Datenbank und greift dafür auf die Konfigurationsdatei `hubd-conf.yml` zurück.

Für ein regelmäßiges Schreiben der Track & Trace Daten in die Datenbank empfiehlt es sich einen Cronjob anzulegen, der alle 5 Minuten ausgeführt wird. Eventuell auftretende Fehlermeldungen werden an `root` per Mail geschickt. Dafür wird die Datei `tt_record` aus der Installations-CD nach `/etc/cron.d` kopiert.

Detaillierte Installationshinweise befinden sich auf der Installations-CD unter `INSTALL.TXT`.

10.1.3. Webanwendung installieren und einrichten

Die Webanwendung wird nach `/var/www` kopiert. Für den produktiven Einsatz wird Apache2 empfohlen. Damit die Webanwendung unter Apache2 läuft, sollte die Konfigurationsdatei `/etc/apache2/sites-available/default` wie folgt aussehen:

```
ServerAdmin admin@webinterface.domain.de
ServerName webinterface.domain.de
DocumentRoot /var/www/webinterface/public
ErrorLog /var/log/apache2/webinterface/error.log
# Possible values include: debug, info, notice, warn, error,
crit,
# alert, emerg.
LogLevel warn
CustomLog /var/log/apache2/webinterface/access.log combined
ServerSignature On
```

Der Name der Domäne muss natürlich angepasst werden.

Da im Webinterface sensible Daten ausgetauscht werden, wird eine verschlüsselte Verbindung verwendet. Für einen `Https`-Betrieb ist folgender Eintrag in der Konfigurationsdatei `/etc/apache2/sites-available/default` notwendig:

```
<IfDefine SSL>
  Listen 443
  AddType application/x-x509-ca-cert .crt
  SSLPassPhraseDialog builtin
```

```

    SSLSessionCache          dbm:/var/run/ssl_scache
    SSLSessionCacheTimeout  300
    SSLMutex file:/var/run/ssl_mutex
    SSLEngine on

    SSLCipherSuite
    ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNUL
L
    SSLCertificateFile /etc/apache2/ssl/apache2.cert
    SSLCertificateKeyFile /etc/apache2/ssl/apache2.key
    SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
</IfDefine>

```

Damit auch die Webanwendung auf die Datenbank zugreifen kann, muss in der Konfigurationsdatei `config/database.yml` das Passwort für die Datenbank angepasst werden.

Die Webanwendung versickt für die Registrierung E-Mails an die Teilnehmer und benötigt daher einen E-Mail-Server. Die Zugangsdaten für einen vorhandenen E-Mail-Server werden in `config/environment.rb` beim ActionMailer vorgenommen. Das Vorhanden sein eines E-Mail-Servers ist zwingend erforderlich, da ansonsten keine E-Mail-Überprüfung durchgeführt werden kann.

Nach erfolgreicher Installation kann die Webanwendung mit folgendem Befehl unter `/var/www/webinterface` einem Testdurchlauf unterzogen werden:

```
rake test
```

Ein erfolgreicher Test muss vollständig durchlaufen und darf keine „failures“ oder „errors“ anzeigen.

Jetzt kann der Apache2 Server mit folgendem Befehl gestartet werden:

```
/etc/init.d/apache2 start
```

Bevor die Webanwendung für andere zugänglich gemacht wird, sollte ein Benutzer registriert werden. Die Registrierung erfolgt über das Webinterface der Webanwendung. Das als erstes aktivierte Konto erhält die Administrationsrechte zum Verwalten der Webanwendung. Der Administrator wird über neue Registrierungen per E-Mail informiert und muss diese akzeptieren oder ablehnen.

Daher ist es wichtig, dass die E-Mail-Adresse vom Administrator gültig ist und die Aufgabe erfüllt wird.

Detaillierte Installationshinweise befinden sich auf der Installations-CD unter `INSTALL.TXT`.

10.1.4. Hub-Server starten und bedienen

Nach erfolgreicher Installation kann der Hub-Server das erste Mal gestartet werden. Um zu kontrollieren ob der EDI-Hub erfolgreich startet, sollte der Debug-Modus mit `debug: true` in der Konfigurationsdatei des EDI-Hub-Server `hubd-conf.yml` aktiviert werden. Alle notwendigen Ausgaben werden in die Log `/var/log/hubd.log` geschrieben. Nach dem Starten des EDI-Hubs mit `hubd.rb` sollten in der Log keinen Fehlermeldungen oder Errors erscheinen. Nach erfolgreichem Start des Hub-Servers kann der Debug-Modus wieder deaktiviert werden, damit die Logs nicht so schnell wachsen. Die Änderungen in der Konfigurationsdatei werden erst nach einem Neustart des EDI-Hubs übernommen.

Im Betrieb benötigt der Hub-Server keine weiteren Eingriffe. Es ist jedoch möglich mit Hilfe des Hub-Kontrollers Einfluss auf den Hub-Server zu nehmen. Der Hub-Server kann zum Beispiel mit `hubd-ctrl.rb stop` beendet oder mit `hubd-ctrl.rb intervall 60` die Zeit bis zum erneuten Scannen der Mailboxverzeichnisse verändert werden.

10.2. Inhalt der CD

Auf der CD befindet sich:

- Hub-Server inklusive Quellcode
- Webanwendung inklusive Quellcode
- Ausführliche Installationsanleitung
- die Diplomarbeit als Word-Dokument
- die Diplomarbeit als PDF-Dokument
- Quellcode-Dokumentation des EDI-Hub-Servers und der Webanwendung