

Seminararbeit

Apache Commons E-Mail

von

Stefan Donninger

h0953364

eingereicht am

19.06.2013

Betreuer:

Univ.Prof. Mag. Dr. Rony G. Flatscher

Lehrveranstaltung:

SBWL Management Information Systems

Kurs V – IS Projektseminar

LV Nummer:

4387

Inhaltsverzeichnis

1 Problemstellung und Ziele.....	1
2 E-Mail.....	2
2.1 Die Anfänge.....	2
2.2 Schattenseiten der E-Mail.....	3
2.3 Komponenten einer E-Mail.....	4
2.4 Protokolle.....	6
2.4.1 SMTP.....	6
2.4.2 POP.....	6
2.4.3 IMAP.....	7
2.4.4 POP oder IMAP.....	7
3 Vom Sender zum Empfänger.....	8
3.1 Simple Darstellung der Übermittlung.....	8
3.2 MUA, MTA, MDA.....	10
3.3 Komplexe Darstellung der Übermittlung.....	12
4 Apache Software Foundation.....	14
4.1 Unternehmensphilosophie.....	14
4.2 Apache Commons.....	15
5 Apache Commons E-Mail.....	16
5.1 SimpleEmail.....	16
5.2 MultiPartEmail.....	19
5.3 Resume.....	20
Zusammenfassung.....	21
Ausblick.....	22
Literaturverzeichnis.....	23
Quellenverzeichnis.....	23

Abbildungsverzeichnis

Abbildung 1: Einfaches E-Mail Versand Schema.....	9
Abbildung 2: Agents und Protokolle.....	11
Abbildung 3: Komplexeres E-Mail Versand Schema.....	12
Abbildung 4: ASF Logo.....	14

Tabellenverzeichnis

Tabelle 1: Aufbau einer E-Mail.....	5
Tabelle 2: Nutshell Textnachricht mit SimpleEmail.....	16
Tabelle 3: Nutshell Textnachricht mit Java Mail.....	18
Tabelle 4: Nutshell Nachricht und Anhang mit MultiPartEmail.....	19

1 Problemstellung und Ziele

Ziel der Arbeit ist es einen Überblick über das Thema E-Mail zu bieten. Dieser Überblick reicht von der Entstehungsgeschichte über Probleme, bis zu den Techniken und Protokollen, welche versteckt im Hintergrund arbeiten.

Ein weiterer Punkt der Arbeit ist die Vorstellung der Apache Software Foundation. Besonderes Augenmerk liegt dabei auf dem Apache Commons Projekt mit Fokus auf die Commons E-Mail Klassenbibliothek. Dieses Projekt bietet eine vereinfachte Programmierschnittstelle welche weniger Zeilen benötigt und mit menschenfreundlichem Code arbeitet. Dies wird anhand von kleinen Programmierbeispielen veranschaulicht.

2 E-Mail

Dieses Kapitel gewährt einen grundlegenden Überblick über die Entstehung und die technischen Aspekte des großen Themengebietes E-Mail.

Im deutschen Sprachgebrauch ist sowohl die weibliche als auch die sächliche Form des englischen Wortes „electronic mail“ gebräuchlich. Im Zuge dieser Seminararbeit wird die weibliche Form („die E-Mail“) verwendet. Die ebenfalls gebräuchliche Kurzform „Mail“ wird synonym zu „E-Mail“ verwendet.

2.1 Die Anfänge

Die Erfolgsgeschichte der E-Mail begann Anfang der 1970er-Jahre im Arpanet, dem Vorgänger des Internet. Als Erfinder der E-Mail gilt der US-amerikanische Computertechniker Raymond Samuel Tomlinson. Tomlinson arbeitete bei dem privaten Forschungsunternehmen „Bolt Beranek and Newman“ kurz BBN. Dieses erhielt 1968 vom US-amerikanischen Verteidigungsministerium den Auftrag das Arpanet aufzubauen. Arpanet steht für „Advanced Research Projects Agency Network“. Es wurde entwickelt um verschiedene Universitäten, welche für das Verteidigungsministerium forschten, miteinander zu verbinden.

Im Rahmen dieses Auftrages entstand CPYNET. Dies ist ein Protokoll zur Datenübertragung zwischen verbundenen Computersystemen. Die Grundlagen eines Mailbox System waren zu dieser Zeit mit dem Programm SENDMSG bereits vorhanden. Jedoch konnte man diese Nachrichten nur auf dem gleichen lokalen Gerät hinterlassen. Dieses Programm hatte den Zweck Nachrichten zwischen zwei Programmierern auszutauschen welche zu unterschiedlichen Zeiten am gleichen Gerät arbeiteten.

Tomlinson kombinierte CPYNET und SENDMSG und konnte somit Nachrichten nicht mehr nur lokal sondern auch an andere Geräte versenden. Um eine Datei senden zu können benötigte man eine Adress-Kombination aus dem Adressaten und dem Rechnernamen. Dazu benötigte er ein eindeutiges Zeichen zur Trennung dieser beiden Namen. Es musste also ein Zeichen sein, welches nicht innerhalb der Zeichenfolge eines Namens von

Adressat oder Rechner vorkommt. In einem Sonderzeichensatz entdeckte er das @-Zeichen, welches seitdem diesem Zweck dient.

Die erste E-Mail zu Testzwecken versendete Tomlinson 1971. Nach einer Demonstration seines Programms vor seinen Kollegen sagte er: „Don't tell anyone! This isn't what we're supposed to be working on.“¹

Anfangs wurde seine Arbeit von ihm selbst als Spielerei gesehen. Zu dieser Zeit hatte er noch keine Ahnung, dass er damit Geschichte schreiben sollte.

Aufbauend auf Tomlinsons Grundstruktur wurde die Programmierung der E-Mail stetig weiterentwickelt. Allerdings war Tomlinson selbst an der weiteren Entwicklung nicht mehr beteiligt.

2.2 Schattenseiten der E-Mail

Mit zunehmender Verbreitung des Internets setzte sich die Erfolgsgeschichte der E-Mail fort. Die schnelle Übermittlung von Nachrichten - wobei sich Sender und Empfänger an beliebigen Orten der Erde befinden können vorausgesetzt man verfügt über eine Internetverbindung, ist der größte Vorteil der E-Mail. Als weitere positive Entwicklungen sind Free-Mail-Anbieter, verschlüsselte E-Mail Systeme sowie das mögliche Versenden von immer größer werdenden Anhängen zu nennen.

Bedingt durch diese rasant ansteigende Popularität kam es neben den positiven Entwicklungen zu zunehmend mehr Negativen. Hierbei ist vor allem Spam-Mail ein Problem. Die erste bekannte erfolgreiche kommerzielle Spam-Mail wurde 1994 verschickt. Darin wurde für eine Green-Card-Lotterie geworben. Seitdem nimmt der Anteil an Spam-Mails stetig zu. Im April 2013 beträgt ihr Anteil am gesamten E-Mail Verkehr weltweit rund 72 %². Das Sichten und Löschen von Spam kostet den Unternehmen wertvolle Arbeitsstunden. Hinzu kommt die verbrauchte Energie durch das Senden der Spam-Mails. Somit entstehen weltweit jährlich Schäden in Milliardenhöhe.

1 <http://www.amacando.de/geschichte-der-email.php> Zugriff am 15.06.2013

2 <http://www.viruslist.com/de/spam/analysis?pubid=200883818> Zugriff am 15.06.2013

Eine weitere negative Entwicklung ist das Aufkommen von Phishing-Mails, mit denen versucht wird an sensible Daten des Users zu kommen. Dabei bekommt der User eine Nachricht, in der sich die Angreifer als eine vertrauenswürdige Stelle ausgeben indem sie deren Nachrichten- oder Webdesign nachahmen. In dieser wird der User aufgefordert die jeweiligen Account-Daten einzugeben. Oft handelt es sich um Kontodaten mit denen das Konto des Users leergeräumt wird oder um Account-Daten mit denen die Angreifer Identitätsdiebstahl begehen. Diese Angriffe können zusätzlich durch den Einsatz von Malware wie beispielsweise Trojaner unterstützt werden.

2.3 Komponenten einer E-Mail

Eine E-Mail besteht grundsätzlich aus 3 Teilen. Diese sind „envelope“, „header“ und „message body“ der Nachricht.

Der „envelope“ enthält mehrere Zeilen, welche Transportinformationen beinhalten. Ein „envelope“ beginnt mit der Zeile „From“. Jeder Server den die E-Mail auf dem Weg zum Ziel passiert verändert diese Zeile. Dadurch wird es möglich den zurückgelegten Weg der E-Mail zurück zu verfolgen.

Im „header“ einer E-Mail findet man diverse Daten, welche die Eigenschaften der E-Mail bestimmen. Somit sind für den Nutzer einige „header“ Felder nicht relevant. Bei Bedarf kann man sich jedoch auch den ganzen „header“ anzeigen lassen.

Der „body“ beinhaltet den Text der E-Mail Nachricht und ist durch ein Leerzeichen vom Rest der E-Mail getrennt

Die nachfolgende tabellarische Darstellung soll den grundsätzlichen Aufbau veranschaulichen:

Mehrere Zeilen die dem Transport der E-Mail dienen, vergleichbar mit einem Umschlag bei der Brief-Post. Sollte die Nachricht unzustellbar sein, wird der Absender informiert.		
E N V E L O P E	H Recieved:	Mehrere Received Zeilen, jeder Server der die Mail weiterleitet fügt seine Kennung und das Datum hinzu
	H Date:	Sendedatum
	E Subject:	Betreff
	A Message-ID:	Eindeutige Zeichenfolge zur Identifikation der Nachricht
	A From:	Absender
	D To:	Empfänger
	E Cc:	Empfänger einer Kopie der E-Mail
	R Content-Type:	Zeichensatz des Body Textes
	R Reply-To:	Eine oder mehrere Adressen an die eine Antwort auf diese E-Mail geschickt werden soll. (falls nicht identisch mit From)
		Leerzeile – trennt Header und Body
B O D Y	Inhalt der E-Mail: Text, Attachments, ... „message body“	

Tabelle 1: Aufbau einer E-Mail

Die Reihenfolge der Header Zeilen ist beliebig und abhängig von der verwendeten Software. Felder die im Normalfall nicht vom E-Mail Programm angezeigt werden sind grau unterlegt. Hellblaue Felder sind die Informationen die üblicherweise sichtbar sind, da diese auch für die beteiligten Personen interessant sind. Das Orange-Hinterlegte Feld ist optional.

2.4 Protokolle

Damit nun eine E-Mail vom Computer des Senders zu dem des Empfängers gelangen kann, sind mehrere Protokolle nötig. Jedes dieser Protokolle erfüllt einen Teil der Aufgabe ohne den die E-Mail nicht zugestellt werden kann.

Die wichtigsten Protokolle für die Zustellung und das Abrufen werden sind nachfolgend beschrieben.³

2.4.1 SMTP

Das „Simple Mail Transfer Protocol“ dient hauptsächlich dem Senden und Weiterleiten einer E-Mail. Es ist ein textbasiertes Protokoll, welches für den User unsichtbar im Hintergrund arbeitet. Die Kommunikation mit dem SMTP Server wird automatisch von dem sogenannten „Mail User Agent“ -MUA für den User ausgeführt. Was ein MUA ist, wird in einem späteren Kapitel erläutert. Der MUA nimmt die zu sendende Nachricht entgegen und übergibt sie dem SMTP-Server des Senders. Dieser leitet die Nachricht entweder direkt oder über einen oder mehrere andere SMTP-Server an den SMTP-Server des Empfängers weiter.

2.4.2 POP

Hinter dem Namen „Post Office Protocol“ verbirgt sich ein Übertragungsprotokoll mit dem Zweck E-Mails von einem E-Mail-Server abzurufen. Somit kann man es als Gegenstück zu SMTP sehen. Die aktuellste Version dieses Protokolls ist die Version 3. Aus diesem Grund wird auch häufig von POP3 gesprochen. POP3 ermöglicht das Auflisten, Abholen und Löschen der E-Mails welche sich auf dem E-Mail-Server befinden. Die E-Mails welche abgerufen werden, werden lokal auf dem Computer des Users gespeichert und anschließend vom Server gelöscht. Der Abruf von E-Mails setzt eine Authentifizierung des Users voraus. Diese Authentifizierungsdaten werden anschließend im Klartext an den Server übertragen was ein Sicherheitsrisiko darstellt. Dies wurde in der Vergangenheit

³ <http://de.kioskea.net/contents/38-die-mail-protokolle-smtp-pop3-imap4> Zugriff am 12.06.2013

immer wieder bemängelt. Somit wurde durch mehrere Erweiterungen Abhilfe geschafft. In das Protokoll selbst wurde solch ein Mechanismus nicht implementiert.

2.4.3 IMAP

Die Abkürzung steht für „Internet Message Access Protocol“. Es erlaubt den Zugriff auf E-Mails die sich auf dem Mailserver des Users befinden. Anders als bei POP3 werden diese Mails nicht lokal gespeichert, sondern werden stattdessen online verwaltet. Somit ist es möglich von mehreren Geräten aus, wenn notwendig sogar simultan, auf die E-Mails zuzugreifen. Somit bietet sich IMAP an, wenn man von mehreren Arbeitsplätzen Zugriff auf seine E-Mails benötigt oder wenn der Account mit anderen Personen geteilt wird. Allerdings ist es notwendig, dass man permanent mit dem Internet verbunden ist.

2.4.4 POP oder IMAP

Da POP3 und IMAP über ähnliche Funktionen verfügen, ergeben sich einige Vor- und Nachteile für beide Protokolle.

Vorteile POP:

- es ist keine ständige Verbindung zum Mailserver notwendig
- die Verbindung wird bei Bedarf vom Client aufgebaut und beendet
- nach der Anmeldung werden alle E-Mails vom Mailserver heruntergeladen

Vorteile IMAP

- erlaubt das Verwalten von mehreren Zugriffen gleichzeitig
- verwaltet mehrere Mailboxen
- kann die Post nach mehreren Kriterien sortieren

Aufgrund der zunehmenden Verbreitung von mobilen Devices, wie Smartphones oder Tablet PCs welche ständig mit dem Internet verbunden sind („always on“) baut das IMAP Protokoll derzeit seinen Vorteil immer weiter aus.

3 Vom Sender zum Empfänger

Um die Frage zu beantworten, wie eine E-Mail von Absender zum Empfänger kommt, kann man sich einer Analogie zur Brief-Post bedienen. Nimmt man als Beispiel den Mail-Account „alice@aon.at“ so steht at für das Land, in dem Fall Österreich. Aon steht für das zuständige Postamt und alice lautet der Name des Postfaches.

Jeder Brief der an die Adresse alice@aon.at geht, landet in dem alice Postfach auf dem aon Postamt in Österreich. Der Besitzer dieses Postfaches kann jederzeit auf das Postamt gehen und sämtliche Nachrichten, die seit dem letzten Entleeren hinzugekommen sind abholen. Umgekehrt funktioniert das ebenso einfach. Nachdem ein Brief geschrieben und der Empfänger richtig angegeben wurde, übergibt man den Brief dem Briefkasten woraufhin dieser zugestellt wird.

3.1 Simple Darstellung der Übermittlung

Um die eingangs ausgeführte Analogie nun auf technischer Ebene zu betrachten dient folgende Abbildung:

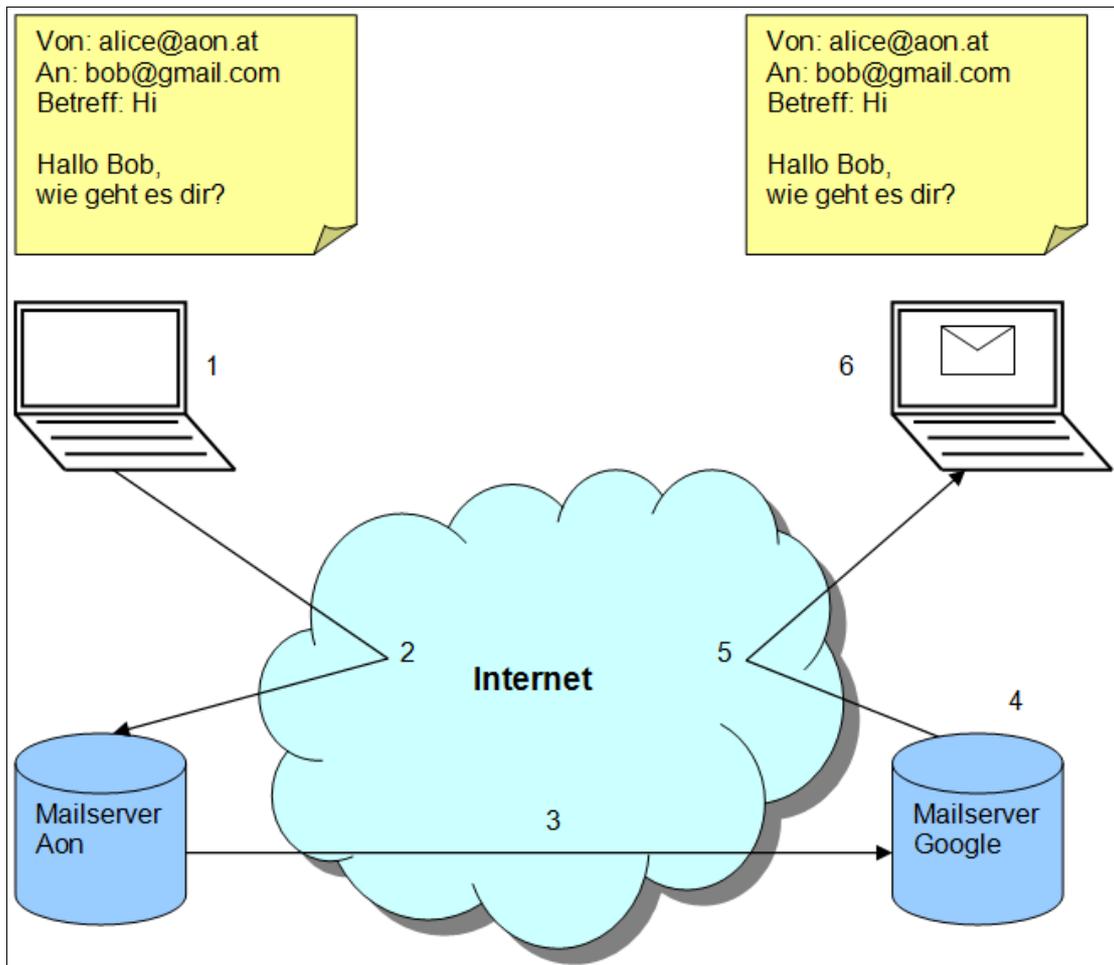


Abbildung 1: Einfaches E-Mail Versand Schema

Abbildung1 zeigt eine vereinfachte Form des E-Mail Versandes einer kurzen Nachricht von Alice an Bob.

1. Der Weg beginnt auf dem PC von Alice. Dort wird in Alices E-Mail-Programm die Nachricht für Bob verfasst.
2. Nachdem der „Senden“-Button geklickt wird, wird die Nachricht von Alices E-Mail-Programm über die Internetanbindung von Alice an ihren Mailserver geschickt. In diesem Fall ist das der Mailserver von Aon.
3. Der Aon-Mailserver macht über das Internet den Mailserver der Zieladresse ausfindig, in diesem Fall den Mailserver von Google. Auf dem Weg zwischen Aon Mailserver und Google Mailserver kann die Nachricht unter Umständen viele weitere Server passieren.
4. Der Mail-Server von Bob nimmt die Nachricht entgegen und speichert sie in Bobs Postfach. Dort verweilt die Nachricht, bis Bob sein Postfach vom Server abrufen.

5. Wenn Bob das nächste mal sein E-Mail-Programm öffnet um seine E-Mails abzurufen, werden die E-Mails auf seinem PC gespeichert.
6. Bob kann nun auf alle neuen E-Mails in seinem Postfach zugreifen.

Diese technische Herangehensweise zeigt, dass man sich aufgrund der geringen Komplexität des Modells noch nahe an der gewählten Analogie befindet.

3.2 MUA, MTA, MDA

In Literatur zum Thema E-Mail stößt man häufig auf diese 3 Abkürzungen. MUA steht für „Mail User Agent“ und ist nichts anderes als das E-Mail Programm (MS Outlook, Mozilla Thunderbird, etc.), welches der User verwendet.

MTA steht für „Mail Transport Agent“. Ein MTA ist ein SMTP-Server, der die zu sendenden Nachrichten entgegennimmt und sie entsprechend weiterleitet.

MDA steht für „Mail Delivery Agent“. Die Aufgabe eines MDA ist das Entgegennehmen der für ihn bestimmten Nachrichten von seinem MTA. Dort werden die Nachrichten zumindest bis zum Abrufen den Users gespeichert. Was nach dem Abrufen mit den Nachrichten passiert hängt vom verwendeten Protokoll ab. Ein MDA ist ein POP3 oder ein IMAP Server.

Das Zusammenspiel der Protokolle sowie der „Agents“ ist in der nachfolgenden Grafik ersichtlich.

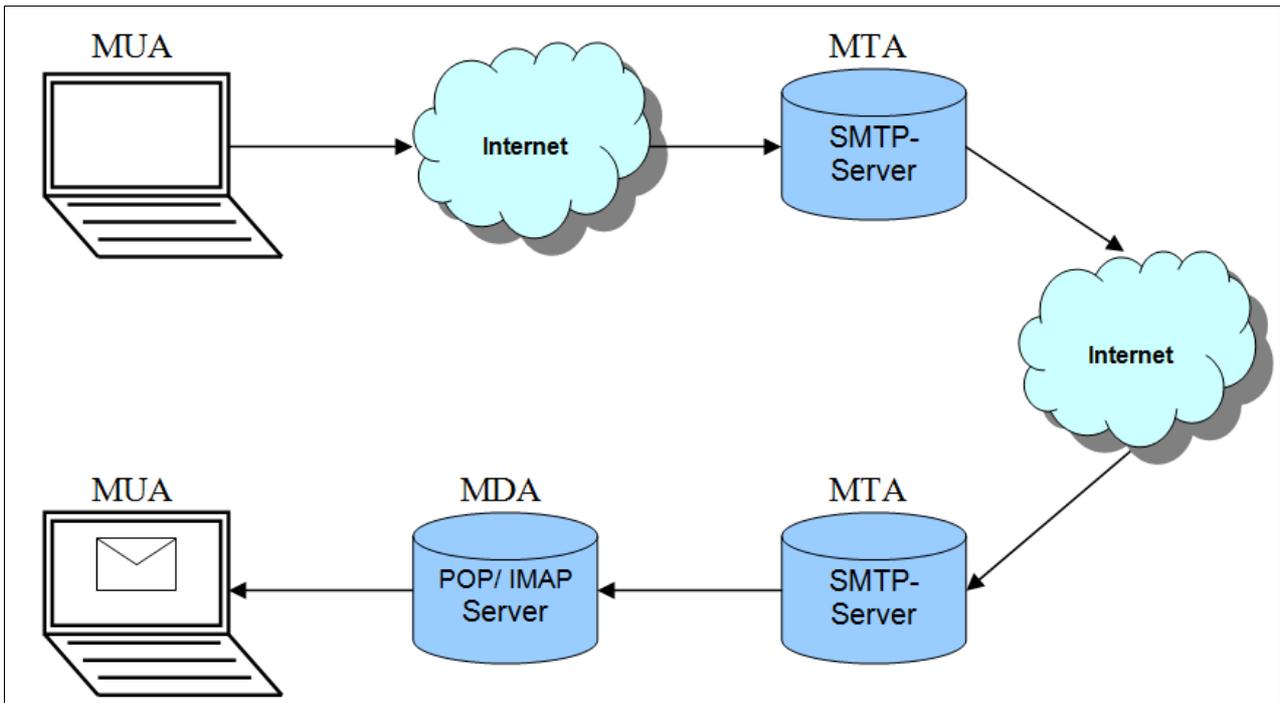


Abbildung 2: Agents und Protokolle

In Abbildung 2 beginnt der Weg einer E-Mail Nachricht in der linken oberen Ecke, dem MUA des Senders. Entlang den Pfeilen durchquert die Nachricht das Internet auf dem Weg zum MTA des Senders. Wie bereits erwähnt handelt es sich bei einem MTA um einen SMTP-Server, welcher die Nachricht entsprechend weiterleitet bis sie den MTA des Empfängers erreicht. Dieser legt die Nachricht in das entsprechende E-Mail-Postfach des Empfängers. Dort wartet die Nachricht, bis sie vom MUA des Empfängers mithilfe der Protokolle POP oder IMAP abgerufen wird.

3.3 Komplexe Darstellung der Übermittlung

Werden die beiden vorhergehenden Modelle kombiniert, erhält man ein komplexes Modell das alle wichtigen Zusammenhänge in sich aufnimmt und veranschaulicht. Das Modell zeigt den Weg den die E-Mail durch das Internet nimmt mit den am Sendevorgang beteiligten Protokollen.

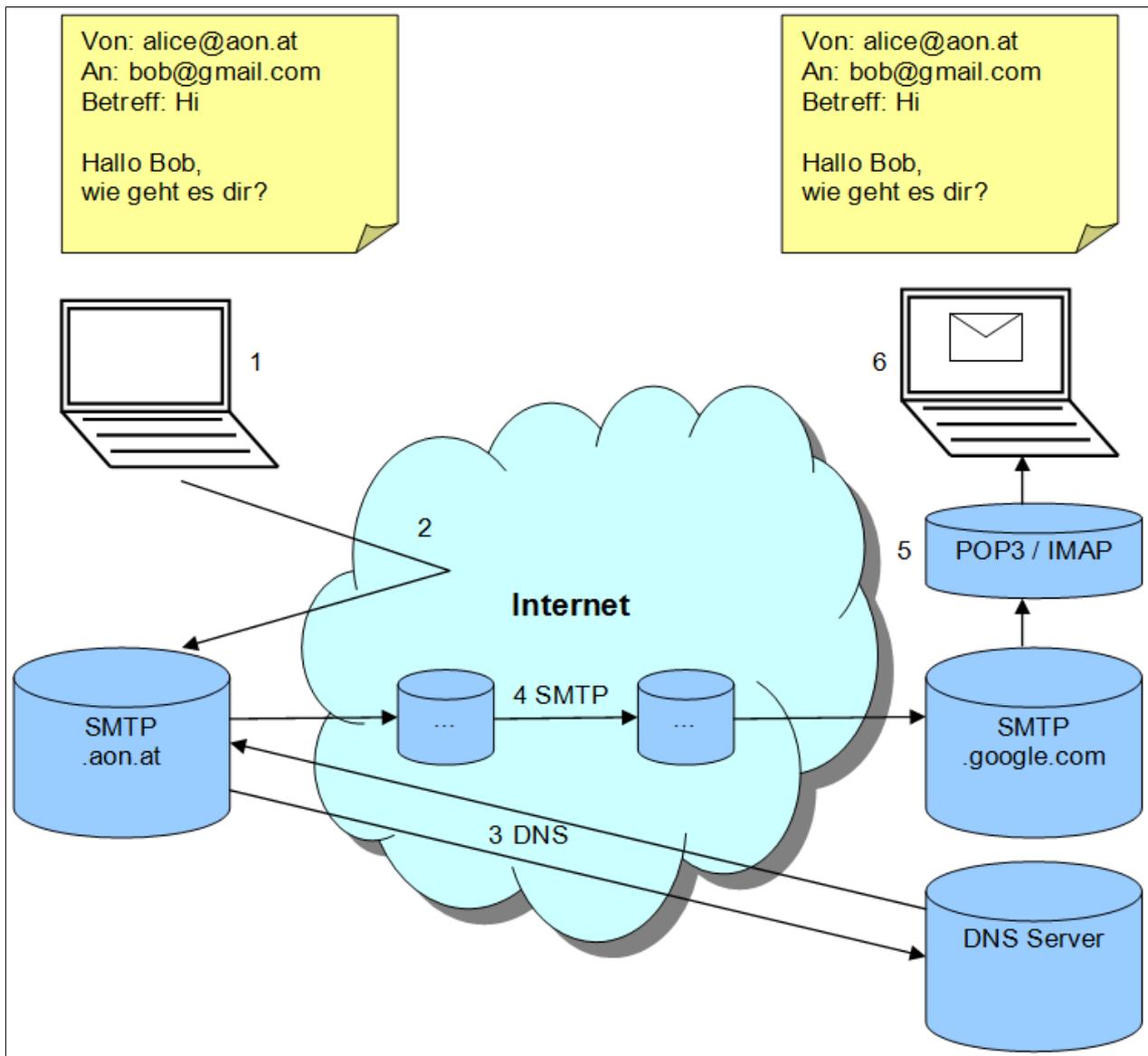


Abbildung 3: Komplexeres E-Mail Versand Schema

1. Der Weg einer E-Mail Nachricht beginnt beim MUA von Alice.
2. Der MUA schickt eine SMTP-Anfrage an den E-Mail-Server des Senders.

3. Der SMTP Server von aon.at (MTA) nimmt Kontakt mit einem DNS Server auf um den sogenannten „Mail Exchange Resource Record (MX-RR)“ der Empfängeradresse zu erfragen. Dafür wird der Teil der Empfängeradresse nach dem „@“ Zeichen verwendet (google.com). Dieser Eintrag gibt Auskunft darüber wie der E-Mail-Server des Empfängers erreichbar ist.
4. Nach der erfolgreichen Anfrage sendet der SMTP-Server von aon.at (MTA Sender) die E-Mail an den SMTP-Server von google.com (MTA Empfänger).
5. Der SMTP-Server von google.com übergibt die neue Nachricht an den MDA wo sie in das richtige Postfach gelegt wird.
6. Wenn Bob das nächste mal sein E-Mail-Programm (MUA) öffnet wird dieses, je nach Protokoll, die notwendigen Aufgaben durchführen um seine E-Mails abzurufen.

4 Apache Software Foundation

Die Apache Software Foundation (ASF) ist eine Non-Profit Organisation, welche sich die Förderung diverser Software Projekte zum Ziel gesetzt hat. Diese Projekte werden, nach Fertigstellung, der Allgemeinheit kostenlos zur Verfügung gestellt. Nach aktuellem Stand (2013) sind es bereits über 140 Projekte welche von Mitgliedern der ASF entwickelt oder unterstützt werden. Es handelt sich um eine verteilte Gemeinschaft von Entwicklern welche gemeinsam an Open-Source-Softwareprojekten arbeiten. Der Name Apache⁴ wurde als Zeichen des Respekts dem Stamm der Apachen gegenüber gewählt. Diese amerikanischen Ureinwohner waren vor allem für ihr strategisches Geschick im Krieg und ihre unerschöpfliche Ausdauer bekannt. Daraus lässt sich die Wahl des Logos begründen, welches eine bunte Feder zeigt.

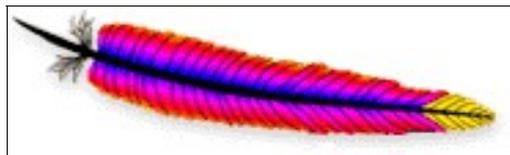


Abbildung 4: ASF Logo

Der Aufgabenbereich der ASF besteht aus dem rechtlichen Schutz ihrer Mitglieder sowie finanzielle und organisatorische Unterstützung der Projekte.

Die ASF wurde 1999 gegründet und entstand aus der Apache Group. Das Gründungsprojekt und gleichzeitig eines der erfolgreichsten Projekte der ASF war der Apache HTTP Server. Dieser quelloffene und freie Webserver ist derzeit (2013) der Meistbenutzte im Internet. Er unterstützt rund 65 % aller Webseiten weltweit.

4.1 Unternehmensphilosophie

Es existieren keine allgemein gültigen Regeln innerhalb der Foundation, allerdings können die folgenden 6 Grundsätze als Unternehmensleitbild gesehen werden. Die Leitsätze werden in englischer Sprache angeführt, um Verzerrungen bei der Übersetzung zu

⁴ <http://httpd.apache.org/docs/1.3/misc/FAQ#name> Zugriff am 15.06.2013

vermeiden.

- collaborative software development
- commercial-friendly standard license
- consistently high quality software
- respectful, honest, technical-based interaction
- faithful implementation of standards
- security as a mandatory feature

Diese Philosophie ist auch bekannt als „The Apache Way“⁵. Alle Projekte der ASF folgen diesen 6 Grundüberzeugungen um deren Erfolg zu gewährleisten.

4.2 Apache Commons

Eines der ASF Projekte ist Apache Commons. Das Commons Projekt befasst sich mit der Erstellung und Wartung von wiederverwendbaren Java Komponenten und Klassenbibliotheken. Das Projekt besteht grundsätzlich aus 3 unterschiedlichen Teilen⁶:

- The Commons Proper – ist eine Sammlung aus wiederverwendbaren Java Komponenten. Es handelt sich sozusagen um das Archiv der bereits fertiggestellten Commons Projekte.
- The Commons Sandbox – ist eine Arbeitsumgebung für die Entwicklung der Java Komponenten. Die Commons Sandbox ist dazu da um gemeinsam an Projekten zu arbeiten und den Entwicklern zu ermöglichen sich untereinander auszutauschen. Sie dient als Fundament auf der alle Commons Projekte erstellt werden.
- The Commons Dormant – ist eine Sammlung von Komponenten, welche nicht fertiggestellt wurden und an denen derzeit niemand arbeitet. Sie dient als Ablage für derzeit unfertige und vorzeitig abgebrochene Projekte.

Apache Commons Projekte sind soweit es geht unabhängig von anderen Bibliotheken, um

5 <http://www.apache.org/foundation/how-it-works.html> Zugriff am 15.06.2013

6 <http://commons.apache.org/> Zugriff am 15.06.2013

sicherzustellen, dass sie schnell und einfach zu implementieren und einzusetzen sind.

5 Apache Commons E-Mail

Eines der zahlreichen Commons Projekte ist Apache Commons E-Mail (ACE) mit der aktuellen Version 1.3.1 (Stand 2013). Es wurde auf der Basis der Java Mail Programmierschnittstelle erstellt. Das Ziel des Projektes war, den Code, welcher zum versenden einer E-Mail notwendig ist, zu vereinfachen und verkürzen. Weiters sollte der Code User-freundlicher werden. Konkret bedeutet das, dass der Code für Menschen leichter zu lesen und zu verstehen sein soll. ACE beinhaltet mehrere Klassen, welche unterschiedliche Aufgaben übernehmen.

5.1 SimpleEmail

Mit Hilfe von SimpleEmail⁷ ist es möglich, einfache E-Mail Nachrichten, die nur Text beinhalten, zu versenden. Nachfolgend ist der Code dargestellt, der für eine Nachricht notwendig ist.

```
1 import org.apache.commons.mail.*;
2 Email email = new SimpleEmail();
3 email.setHostName("smtp.googlemail.com");
4 email.setSmtpPort(465);
5 email.setAuthenticator(new DefaultAuthenticator("username", "password"));
6 email.setSSLonConnect(true);
7 email.setFrom("alice@gmail.com");
8 email.setSubject("TestMail");
9 email.setMsg("This is a test mail ...");
10 email.addTo("bob@gmx.at");
11 email.send();
```

Tabelle 2: Nutshell Textnachricht mit SimpleEmail

⁷ <http://commons.apache.org/proper/commons-email/userguide.html> Zugriff am 18.06.2013

In Zeile 1 wird die Commons Mail Klassenbibliothek importiert um Zugriff auf die dort zur Verfügung gestellten Ressourcen zu bekommen. Nun wird in Zeile 2 ein Objekt der SimpleEmail Klasse erstellt. Es ist notwendig, dass man die Adresse sowie Port des SMTP Servers des Senders definiert, wie man in Zeile 3 und 4 sieht. In Zeile 5 gibt der Absender seine Account Daten bekannt. Ohne diese Maßnahme könnte man die E-Mail nicht verschicken, da der Server eine Authentifizierung des Users verlangt. Zeile 6 bewirkt, dass die Daten mittels SSL sicher übertragen werden. In den Zeilen 7 bis 10 werden Absender, Betreff, Nachrichtentext und Zieladresse bestimmt. In der letzten Zeile wird die Nachricht schließlich verschickt. Eine einfache Text Nachricht kann also dank SimpleEmail mit nur 11 Zeilen einfachen Code verschickt werden.

Im Gegensatz dazu zeigt die nachfolgende Tabelle den Java Code ohne SimpleEmail.

```

1 import java.util.Properties;
2 import javax.mail.Address;
3 import javax.mail.Message;
4 import javax.mail.Session;
5 import javax.mail.Transport;
6 import javax.mail.internet.InternetAddress;
7 import javax.mail.internet.MimeMessage;
8 public class SendMailTLS
9 {
10 public static void main(String[] args) throws Exception
11 {
12     String host="smtp.gmail.com";
13     int port=587;
14     String user="alice@gmail.com";
15     String pass="password";
16
17     Properties props=new Properties();
18     props.put("mail.smtp.auth", "true");
19     props.put("mail.smtp.starttls.enable", "true");
20
21     Session session=Session.getInstance(props);
22     Transport transport=session.getTransport("smtp");
23     transport.connect(host, port, user, pass);
24
25     Address[] addresses=InternetAddress.parse("bob@gmx.at");
26
27     Message message=new MimeMessage(session);
28     message.setFrom(new InternetAddress(user));
29     message.setRecipients(Message.RecipientType.TO, addresses);
30     message.setSubject("TestMail");
31     message.setText("This is a test mail ...");
32
33     transport.sendMessage(message, addresses);
34     transport.close();
35 }
36 }

```

Tabelle 3: Nutshell Textnachricht mit Java Mail

Ohne den Code näher zu erläutern, kann man auf den ersten Blick feststellen, dass die Anzahl der notwendigen Zeilen weit größer ist als bei SimpleEmail. Auch der Komplexitätsgrad nimmt zu, was den Code schwieriger lesbar macht. Die beiden Beispiele aus Tabelle 1 und Tabelle 2 erfüllen genau dieselbe Aufgabe.

5.2 MultiPartEmail

Möchte man eine E-Mail senden und dieser Nachricht eine Datei anhängen, bleibt der Code zum Senden der Nachricht im wesentlichen der Selbe. Mit dem Unterschied, dass die Klasse MultiPartEmail⁸ verwendet wird.

```
1 import org.apache.commons.mail.*;
2 ...
3 // Create the attachment
4 EmailAttachment attachment = new EmailAttachment();
5     attachment.setPath("alicepic.jpg");
6     attachment.setDisposition(EmailAttachment.ATTACHMENT);
7     attachment.setDescription("Picture of Alice");
8     attachment.setName("Alice");
9 // Create the email message
10 MultiPartEmail email = new MultiPartEmail();
11     email.setHostName("smtp.googlemail.com");
12     email.addTo("bob@gmx.at", "Bob");
13     email.setFrom("alice@gmail.com", "Alice");
14     email.setSubject("The picture");
15     email.setMsg("Here is the picture you wanted");
16 // add the attachment
17     email.attach(attachment);
18 // send the email
19     email.send();
```

Tabelle 4: Nutshell Nachricht und Anhang mit MultiPartEmail

8 <http://commons.apache.org/proper/commons-email/userguide.html> Zugriff am 18.06.2013

Die für den Anhang relevanten Zeilen werden mit Hilfe der Klasse `EmailAttachment` ergänzt. Die Verwendung dieser Klasse findet man inklusive Kommentar in den Zeilen 3 bis 8. Nachdem das Attachment fertiggestellt ist, muss noch der Code zum Versenden der Nachricht geschrieben werden. Wie bereits erwähnt bleibt der Code annähernd gleich. Einziger Unterschied ist die Verwendung einer anderen Klasse.

Sollte man die Datei welche hinzugefügt werden soll nicht lokal gespeichert haben, ist es auch möglich Zeile 5 des Codes mit folgender Zeile zu ersetzen:

```
attachment.setURL(new URL("http://www.alicespage.at/images/alicepic.jpg"));
```

Der Rest bleibt unverändert.

Möchte man in Java ohne die Commons E-Mail Klassenbibliothek eine Nachricht mit der selben Funktionalität wie das Beispiel in Tabelle 3 versenden benötigt man rund 60 Zeilen Code.

5.3 Resume

Die Erkenntnisse, die man aus diesem Kapitel gewinnen sind, dass die Java Klassenbibliothek Commons E-Mail mehrere Vorteile bietet. Dank der Verwendung der Klassenbibliothek ist der leicht lesbare Code deutlich kürzer als der vergleichbare Java Code, der ohne Commons E-Mail auskommen muss. Sollte man den vorhandenen Code anpassen müssen, geht dies dank der „usability“ des Codes schneller als sonst.

Es gibt noch weitere Klassen innerhalb von Commons E-Mail mit denen es möglich ist, Nachrichten die mit HTML formatiert werden, zu senden. In dieser Arbeit werden diese Klassen nicht behandelt, da das Grundmuster ähnlich wie in den gezeigten Klassen ist.

Die Nützlichkeit der Commons Projekte ist unbestritten. Die hier gezeigte E-Mail Klassenbibliothek ist nur eines von vielen Projekten, die das Programmieren wesentlich erleichtern.

Zusammenfassung

Diese Seminararbeit befasst sich mit dem Thema E-Mail und bietet auch technisch weniger bewanderten Lesern einen einfachen Einstieg in die relevanten Technologien und Methoden beim Versand von E-Mail Nachrichten. Ausgehend von der ursprünglichen Entwicklung der E-Mail in den 1970ern nahm die Erfolgsgeschichte der E-Mail ihren Lauf. Bedingt durch die enorme Popularität der E-Mail kommt es zunehmend häufiger zu Problemen. Spam-Mails oder Phishing-Mails werden heutzutage immer öfter eingesetzt.

Die wichtigsten Protokolle beim E-Mail Versand sind SMTP, POP und IMAP. Deren Funktionen werden im einzelnen erklärt. Mit Hilfe eines einfachen Modells wird der Ablauf eines Sendevorganges dargestellt und erläutert. Dieses wird in ein komplexeres Modell überführt, welches die beteiligten Hard- und Softwarekomponenten sowie das Zusammenspiel der Protokolle zeigt.

Ein weiterer Teil dieser Seminararbeit beschäftigt sich mit der Apache Software Foundation. Hier wird ebenfalls die Geschichte und Entstehung behandelt. Der Fokus der Arbeit liegt auf dem Apache Commons E-Mail Projekt.

Dieses Projekt stellt eine einfache Programmierschnittstelle zum Versand von E-Mails zur Verfügung. Dadurch kann menschenfreundlicherer Code verwendet werden, der gleichzeitig kürzer und kompakter ist.

Ausblick

In dieser Arbeit wurden die Kommunikationsregeln der für den E-Mail Versand wichtigen Protokolle SMTP, POP3 und IMAP nicht behandelt. Ebenso wurde das Thema bezüglich Spam nur angeschnitten und Spamblocker wurden nicht behandelt.

In der heutigen Zeit spielt Security eine immer größere Rolle in allen Bereichen mit Kontakt zum Internet. Dazu zählt natürlich auch die Sicherheit beim E-Mail Versand. Dieses Thema wurde in dieser Arbeit ebenfalls nicht behandelt und bietet Möglichkeiten für eine Vielzahl an weiteren Arbeiten.

Eng mit dem Security Problem verknüpft, spielt Kryptographie eine große Rolle beim Versand von gesicherten E-Mail Nachrichten.

In Bezug auf die Apache Software Foundation wurde nur ein Bruchteil der Arbeit die dort geleistet wird vorgestellt. Es gibt eine große Anzahl potentieller Themen, wenn man an die rund 140 Apache Projekte denkt.

Literaturverzeichnis

Günther, U. & Wyss, E. L. (1996). E-Mail-Briefe – eine neue Textsorte zwischen Mündlichkeit und Schriftlichkeit.

Haase, M., Huber, M., Krumeich, A. & Rehm, G. (1997). Internetkommunikation und Sprachwandel.

Quellenverzeichnis

<http://www.viruslist.com/de/spam/analysis?pubid=200883818> Zugriff am 15.06.2013

<http://www.amacando.de/geschichte-der-email.php> Zugriff am 15.06.2013

<http://de.kioskea.net/contents/1103-struktur-einer-e-mail-header-und-textkorper> Zugriff am 15.06.2013

[http://de.wikipedia.org/wiki/Header_\(E-Mail\)](http://de.wikipedia.org/wiki/Header_(E-Mail)) Zugriff am 15.06.2013

<http://de.kioskea.net/contents/38-die-mail-protokolle-smtp-pop3-imap4> Zugriff am 12.06.2013

http://de.wikipedia.org/wiki/E-mail#Zustellung_einer_E-Mail_.28Prinzip.29 Zugriff am 12.06.2013

<http://de.kioskea.net/contents/1096-funktionsweise-von-e-mails-mta-mda-mua> Zugriff am 12.06.2013

http://de.wikipedia.org/wiki/Apache_Software_Foundation Zugriff am 12.06.2013

<http://httpd.apache.org/docs/1.3/misc/FAQ#name> Zugriff am 15.06.2013

<http://www.apache.org/foundation/how-it-works.html> Zugriff am 15.06.2013

http://en.wikipedia.org/wiki/Apache_Commons Zugriff am 15.06.2013

<http://commons.apache.org/> Zugriff am 18.06.2013

<http://commons.apache.org/proper/commons-email/> Zugriff am 18.06.2013

<http://commons.apache.org/proper/commons-email/userguide.html> Zugriff am 18.06.2013