Technische Hochschule Ingolstadt

Fakultät Elektrotechnik und Informatik

Konzeptionierung eines Briefmarkensystems für E-Mails zur Bepreisung der Aufmerksamkeit des Empfängers

Bachelorarbeit zur Erlangung des Grades

eines Bachelor of Science Informatik

Erstprüfer: Prof. Dr.-Ing. Hans-Joachim Hof

Zweitprüfer: Prof. Dr.-Ing. Ernst-Heinrich Göldner

Eingereicht von: Timo Meilinger

Ausgabedatum: 30. Oktober 2018

Abgabedatum: 11. Januar 2019

Inhaltsverzeichnis

Abbildungsverzeichnis III

Quellcodeverzeichnis IV

Abkürzungsverzeichnis V

1 Einleitung 1

2 Grundlagen 3

2.1 Blockchain 3

2.1.1 Ethereum Netzwerk 4

2.1.2 Token im Ethereum Netzwerk 4

2.2 E-Mail 5

2.2.1 Aufbau von E-Mail Nachrichten 7

2.2.2 Versand von E-Mail Nachrichten 8

3 Verwandte Arbeiten 10

3.1 Hashcash 10

3.2 SAVE - Sender Address Verification Extension 11

4 Analyse der Umgebungsbedingungen und aktuellen Situation 13

4.1 Aktuelle Herausforderung im E-Mail System 13

4.2 Verschiedene Kommunikationsarten 14

4.3 Verschiedene Nutzer des E-Mail Systems 16

5 Erweiterung des E-Mail Systems **18**

5.1 Grundkonzept für ein Briefmarkensystem für E-Mails 18

5.2 Anforderungen an das Briefmarkensystem 19

5.3 Aufbau des Briefmarkensystems 20

5.4 Konzeptionierung der Briefmarken 23

5.5 Konzeptionierung der Änderungen des E-Mail Aufbaues 24

5.6 Konzept zur sicheren Validierung der Zuordnung Account-ID zur E-Mail Adresse 26

6 Prototypumsetzung des Konzeptes **30**

6.1 Extension für Thunderbird 31

6.1.1 Sortierung der E-Mails 31

6.1.2 Versand der E-Mails 33

6.1.3 Auflösung E-Mail Adresse zu Account-ID der Blockchain 34

6.2 Verwaltungsprogramm für Briefmarken 36

6.2.1 Versand von Briefmarken 36

6.2.2 Validierung der Briefmarken 37

6.2.3 Ermittlung der Anzahl verfügbarer Briefmarken 39

6.3 ERC20 Token im Ethereum Netzwerk 39

7 Evaluierung des Nutzens eines Briefmarkensystems für E-Mails **41**

8 Zusammenfassung und Ausblick **44**

Literaturverzeichnis **VI**

Abbildungsverzeichnis

1 Aufbau der Blockchain 3
2 MUA Beispiel Weboberfläche Gmail von Google 6
3 MUA Beispiel Thunderbird von Mozilla 6
4 Transportweg einer E-Mail 9
5 Grundsätzlicher Aufbau 20
6 Ablaufdiagramm des Versands von Briefmarken 21
7 Ablaufdiagramm der Sortierung von Briefmarken 22
8 Registrierungsprozess beim Root of Trust 27
9 Verifikation einer Zuordnung beim Root of Trust 29
10 E-Mail Sortierung 'StampSort' starten 31
11 Verification-Object von Thunderbird erhalten 32
12 Versendete Briefmarke zur E-Mail hinzufügen 33
13 Account-ID zum E-Mail Header anfügen 34
14 Einstellungen für die Thunderbird Extension 35
15 Account-ID zu einer E-Mail Adresse finden 36
16 Fenster Stamp Functions 36
17 Fenster Send Stamp 37
18 Pop-up Fenster Send Stamp 37
19 Fenster Verify Stamps 38
20 Pop-up Fenster Verify Stamps 38
21 Anzahl an verfügbaren Briefmarken bestimmen 39

Quellcodeverzeichnis

1 E-Mail mit Header und Body 7
2 JSON-Object in der Headereigenschaft 'Stamp' 34
3 Code des ERC20 Tokens TestStamp 40

Abkürzungsverzeichnis

ASCII	American Standard Code for Information Interchange
DNS	Domain Name System
ESMTP	Extended Simple Mail Transfer Protocol
ETH	Ethereum
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IMAP	Internet Message Access Protocol
MTA	Mail Transfer Agent
MUA	Mail User Agent
POP3	Post Office Protocol 3
SMTP	Simple Mail Transfer Protocol

1 Einleitung

E-Mails stellen heute eine der wichtigsten Kommunikationsarten in der privaten und geschäftlichen Kommunikation dar.[15] Hierbei kann in zwei Arten von Nachrichten unterschieden werden. Einerseits in gewollte Kommunikation, beziehungsweise Informationsaustausch und andererseits in ungewollte Kommunikation, wie beispielsweise Werbung oder sogenannte Phishing E-Mails. Allgemein werden diese ungewollten Nachrichten als Spam bezeichnet. Das BSI - Bundesamt für Sicherheit in der Informationstechnik definiert Spam E-Mails wie folgt: 'Bei Spam-E-Mail handelt es sich um unverlangt zugesandte Werbe-E-Mails, häufig mit pornographischen Inhalten. In der Regel werden diese E-Mails mit gefälschten Absender-Angaben über Rechner von Dritten verteilt. Insbesondere wechseln sowohl die Absende-, als auch die Betreff-Angaben häufig, damit sie nicht zu leicht als Spam erkannt werden können.'[4] Weiter definiert der Duden Spam als 'unerwünschte massenhaft per E-Mail oder auf ähnlichem Wege versandte Nachrichten'.[7] In dieser Arbeit soll der Begriff Spam und Spam-E-Mail nach diesen beiden Definitionen verwendet werden.

Wie wichtig und verletzlich die E-Mail Kommunikation ist, zeigt auch eine Studie des GDV - Gesamtverband der Deutschen Versicherungswirtschaft e. V.. In dieser ist zu lesen, dass im Mittelstand die meisten erfolgreichen Cyberangriffe über E-Mails ausgeführt wurden.[8] Hierdurch wird klar, dass der Informationsaustausch über E-Mails nicht nur Chancen und neue Möglichkeiten bietet, sondern auch Gefahren birgt. Diese können durch Schulung der Empfänger reduziert werden. Zu erkennen, welche Nachrichten vertrauenswürdig sind ist hierbei von elementarer Bedeutung. Somit ist ein zusätzliches Kriterium zum Erkennen von 'gefährlichen E-Mails' sinnvoll.

In dieser Arbeit soll die Idee von Prof. Dr. Hans-Joachim Hof und Stanley Chow zu einem Konzept für ein Briefmarkensystem für E-Mails ausgearbeitet werden. Diese ist ein neuartiger Ansatz, wobei der Sender einer E-Mail, dem Empfänger der Nachricht für seine Aufmerksamkeit 'bezahlt'. Die 'Bezahlung' erfolgt durch wiederverwendbare Briefmarken. Somit kann der Empfänger der Briefmarke diese nutzen, um eine neue Nachricht mit dieser zu versenden. Durch diesen Umstand wird ein ausgeglichenes Verhältnis von empfangenen und gesendeten Nachrichten angestrebt. Hierfür ist ein Konzept auszuarbeiten, diese Briefmarken durch eine Blockchain zu verwalten. Weiter muss die Integration der Briefmarken in einer Nachricht ausgearbeitet und eine Möglichkeit gefunden werden, diese in einer E-Mail festzuhalten. Hierbei sollen die Briefmarken wiederverwendbar sein. Durch diese Eigenschaft kann der Empfänger erhaltene Briefmarken nutzen, um eigene E-Mails mit Briefmarken zu versenden. Weiter soll ein Prototyp für dieses Konzept implementiert werden, um die Funktionsweise aufzuzeigen und die Umsetzbarkeit zu beweisen. Weiter soll ausgearbeitet werden, ob mit diesem Konzept Spam reduziert und dadurch die Anzahl an unerwünschter Nachrichten dezimiert werden kann.

Zunächst werden Grundlagen, die für diese Arbeit relevant sind, diskutiert. Es werden verwandte Arbeiten vorgestellt und Unterschiede zu diesen aufgezeigt. Die anschließende Analyse wird Aufschluss geben, welche Nutzer und Verhalten bei der Ausarbeitung des Konzeptes zu berücksichtigen sind. Im Design wird der Aufbau und die einzelnen Komponenten (Briefmarken, E-Mail) des Konzeptes vorgestellt. Weiter wird ein Prototyp Programm präsentiert, das die Funktionsweise aufzeigt und die Umsetzbarkeit des Konzeptes beweisen wird. In der Evaluierung werden die Vor- und Nachteile des Konzeptes diskutiert und aufgezeigt, dass durch dieses eine aktive Spamreduktion möglich ist. Abschließend wird in der Zusammenfassung die Arbeit nochmals komprimiert vorgestellt und ein Ausblick beschrieben, wie dieses Konzept in Zukunft erweitert werden kann und welches Potenzial dieses Konzept für weitere Funktionalitäten bietet.

2 Grundlagen

Nachfolgend sollen grundlegende Informationen zu den verwendeten Komponenten erläutert werden, welche als Wissensgrundlage für die weitere Arbeit dienen.

2.1 Blockchain

In der Blockchain werden einzelne Transaktionen zu Blöcken zusammengefasst. Über die einzelnen Blöcke wird ein Hash-Wert gebildet. Dieser Wert wird in dem darauffolgenden Block als vorangegangener Block vermerkt. Hierdurch ist der Block nicht mehr veränderbar, da sich durch jede Änderung der Hash-Wert des Blocks ändert. Aufgrund dessen sind keine nachträglichen Manipulationen möglich und die Integrität über die einzelnen Blöcke sowie Transaktionen sichergestellt.[5]

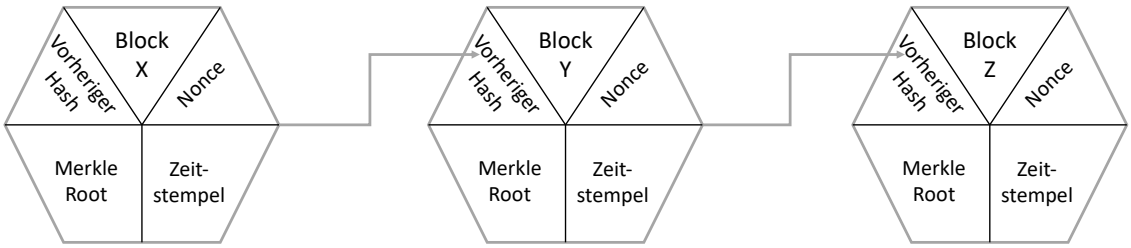

Abbildung 1: Aufbau der Blockchain[1]

Die Transaktionen werden unter dem Merkle Root des Blocks gespeichert. Dieser setzt sich aus den einzelnen Transaktionen und deren Hash-Werten zusammen. Um eine Transaktion zu veranlassen, muss diese durch den privaten Schlüssel des Senders signiert werden. Im Netzwerk der Blockchain kann nun von jedem, mithilfe des öffentlichen Schlüssel

[1] Abbildung 4 in Anlehnung an https://decentralbox.com/was-ist-eine-blockchain/

des Senders, die Echtheit der Transaktion verifiziert werden.[5] Jeder mit dem privaten Schlüssel eines Wallets kann Transaktionen veranlassen. Deshalb muss dieser private Schlüssel geheim gehalten werden und darf Dritten nicht zur Verfügung stehen. Transaktionen erfolgen teils anonym, da sie einem Pseudonym (Account-ID) zugeordnet werden können, jedoch keiner Person. Die Account-ID ist aus dem privaten Schlüssel ableitbar.

2.1.1 Ethereum Netzwerk

Das Ethereum Netzwerk stellt eine Blockchain dar. Dies ist jedoch eine spezielle Blockchain, da nicht nur Transaktionen ausgeführt werden können. Es ist möglich 'Verträge' zu hinterlegen, die bestimmte Funktionen enthalten und somit eine große Vielfalt an speziellen Transaktionen und Verbindlichkeiten anbieten.[5]

Aufgrund dieser Funktionalität kann ein eigener 'Vertrag' hinterlegt werden, der die normalen Transaktionseigenschaften einer Kryptowährung enthält. So kann die Blockchain des Ethereum Netzwerks genutzt werden, um eine eigene Kryptowährung zu definieren. Für das Ausführen einer Funktion von einem 'Vertrag' muss sogenanntes 'Gas' bezahlt werden. Dieses wird in der Währung Ethereum abgerechnet. Das 'Gas' stellt die Gebühr für die Ausführung einer Funktion dar und wird vom Auftraggeber bezahlt. Die Kryptowährung Ethereum wird auch in der Blockchain des Ethereum Netzwerkes verwaltet.

2.1.2 Token im Ethereum Netzwerk

Token stellen eine besondere 'Vertragsform' im Ethereum Netzwerk dar. Diese können als Kryptowährung gesehen werden. Die am meist verbreitetet Form eines Tokens ist der ERC20 Token. Damit ein 'Vertrag' ERC20 kompatible ist, muss er folgende Funktionen enthalten:[5]

- totalSupply() : Gibt die Gesamtanzahl an existierenden Token zurück.
- balanceOf() : Gibt die Anzahl an Tokens zurück, die ein bestimmter Account besitzt.
- transfer(), transferFrom() : Transferiert einen Token.
- approve(), allowance() : Signiert (erlaubt) die Transferierung eines Tokens.

Diese Standardisierung ermöglicht es Token in verschiedenen Wallets zu transferieren. Weiter können Token in vielen Bereichen verwendet werden, aufgrund der Personalisierbarkeit der einzelnen Funktionen und der freien Erstellung von neuen Token. So können diese als Währung verwendet werden oder auch eine Ressource darstellen, die in begrenzter Anzahl zur Verfügung steht und somit digital sowie sicher abgebildet wird. So ergeben sich viele Einsatzmöglichkeiten und der Kreativität sind hierbei nur wenige Grenzen gesetzt. Auch dieser Umstand trägt zu der Beliebtheit von Token im Ethereum Netzwerk bei.

2.2 E-Mail

Das E-Mail System besteht aus zwei Komponenten, einerseits dem Mail Transfer Agents (MTAs) und andererseits den Mail User Agents (MUAs). Diese werden teils auch Message Transfer und Message User Agents genannt. Wie der Name vermuten lässt ist der MTA für den Transport der Nachrichten verantwortlich und vermittelt daher E-Mails über verschiedene Teilnetzwerke an deren Bestimmungsort. Als Protokoll zwischen den MTAs wird das SMTP (Simple Mail Transfer Protocol) bzw. des ESMTP (Extended Simple Mail Transport Protocol) verwendet. Das ESMTP ist die neuere Version des SMTP Protokolls. Weiter in dieser Arbeit soll nicht zwischen SMTP und ESMTP unterschieden werden. Beide werden als SMTP Protokoll bezeichnet. Der MUA wird vom Benutzer benötigt, um die Interaktion mit E-Mails für diesen einfacher und übersichtlicher zu gestalten. Der MUA kann auch als E-Mail-Editor gesehen werden, da dieser alle nötigen Hilfsmittel zur Erstellung, Bearbeitung und Verwaltung elektronischer Post enthält.[11]

Als Beispiele für verbreitete MTAs sind Gmail, web.de, GMX und T-Online zu nennen.[2] Diese Anbieter sind jedoch nicht nur MTAs, sondern bieten auch eine Weboberfläche zur Verwaltung der E-Mails, wodurch Sie einen MUA bereitstellen. Denn ein MUA kann sowohl ein E-Mail Client sein, der lokal auf einen Rechner installiert wird oder eine Weboberfläche, über die E-Mails verwaltet und versendet werden können. Somit sind diese Weboberflächen als Beispiele für MUA zu nennen und weiter auch E-Mail Programme wie: Apple Mail, Mozilla Thunderbird oder Microsoft Outlook.[3] Jedoch ist auch jedes andere

[2] google.com/gmail, web.de, gmx.net, t-online.de/email (Aufgerufen am 29.12.2018)

[3] support.apple.com/mail, thunderbird.net, products.office.com/de-de/outlook/ (Aufgerufen am 29.12.2018)

Programm zur Verwaltung von E-Mails, das auf einem Smartphone oder einem Computer installiert ist, ein MUA. Hierdurch wird ersichtlich, dass es sehr viele verschiedene MUAs mit unterschiedlichen Funktionen und grafischen Layouts gibt.

Um zwei Beispiele für solche MUA zu zeigen ist in Abbildung 2 die Weboberfläche von Google Gmail zu sehen. Ein weiteres Beispiel ist in Abbildung 3 dargestellt. Hierbei handelt es sich um das Programm Thunderbird von Mozilla.

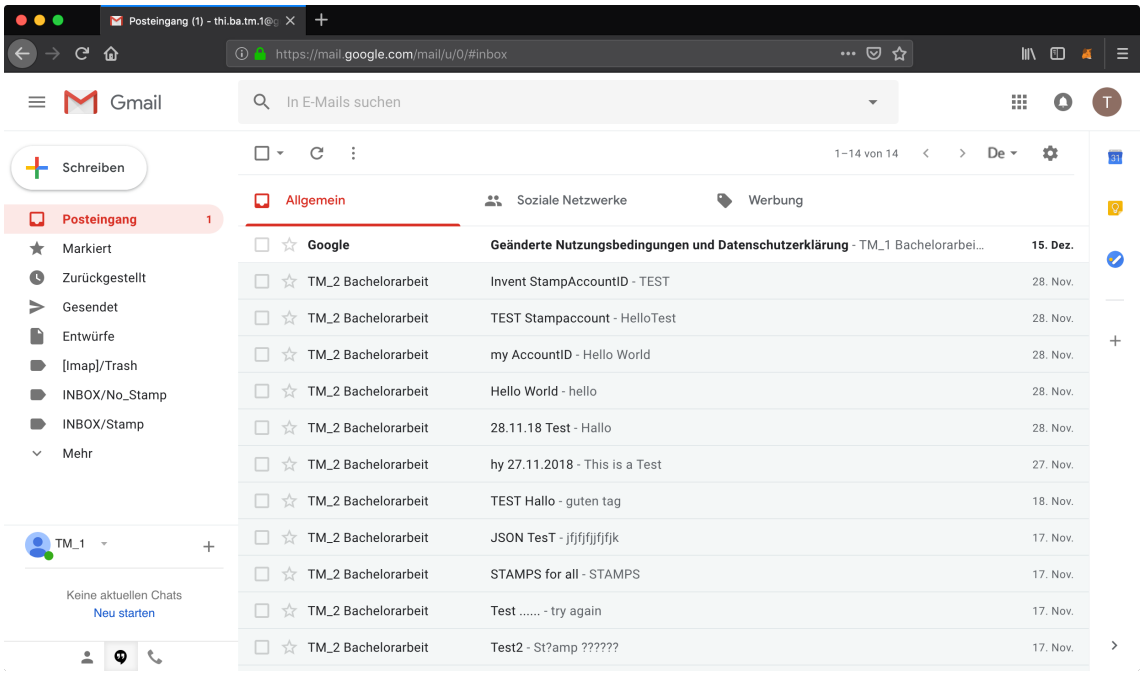

Abbildung 2: MUA Beispiel Weboberfläche Gmail von Google

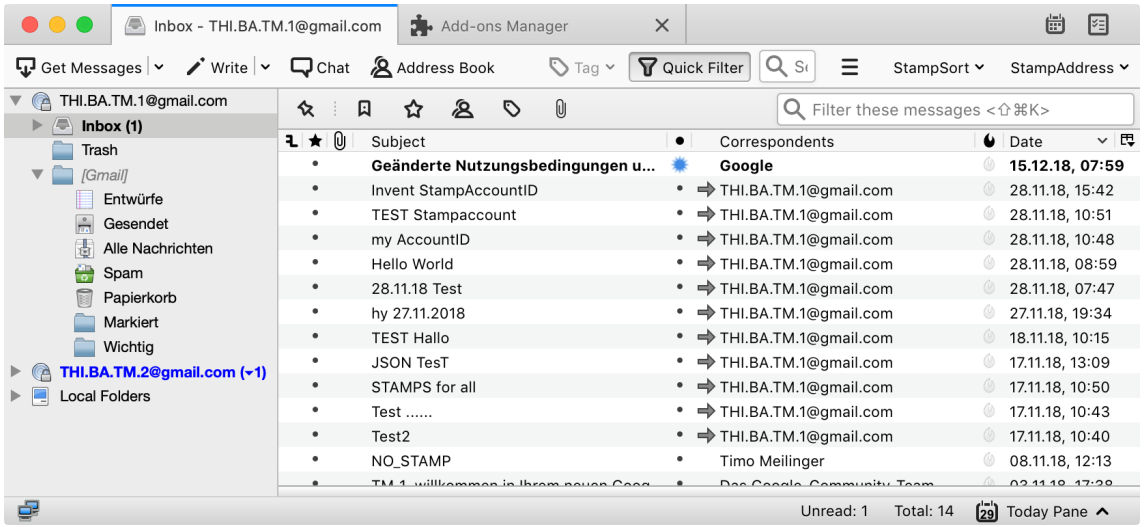

Abbildung 3: MUA Beispiel Thunderbird von Mozilla

2.2.1 Aufbau von E-Mail Nachrichten

Eine E-Mail ist in zwei Bereiche unterteilt. In den sogenannten Header und den Body einer Nachricht.

Der Header stellt wichtige Informationen bezüglich einer E-Mail bereit. Diese können sein: Empfänger, Betreff wie auch Details zur Codierung der Anhänge sowie viele weitere Einzelheiten. Dem Benutzer wird nur ein Teil dieser Informationen angezeigt, da diese beispielsweise zur Interpretierung der Anhänge relevant für den MUA sind, jedoch keinerlei Relevanz für den menschlichen Nutzer darstellen. Somit sollen dem Verwender nur für ihn sachdienliche Informationen bereitgestellt werden und keine Einzelheiten zu technischen Details. Viele MUAs codieren auch Informationen zu der zum Versand verwendeten Software mit in den Header.

```
1  Delivered-To: thi.ba.tm.2@gmail.com
2  Received: by 2002:a02:2585:0:0:0:0:0 with SMTP id g127csp10176131jag;
3          Sat, 29 Dec 2018 02:49:15 -0800 (PST)
4  Return-Path: <thi.ba.tm.1@gmail.com>
5  Received: from mail-sor-f41.google.com (mail-sor-f41.google.com. [209.85.220.41])
6          by mx.google.com with SMTPS id 51sor21346332wra.51.2018.12.29.02.49.14
7          for <THI.BA.TM.2@gmail.com>
8          (Google Transport Security);
9          Sat, 29 Dec 2018 02:49:15 -0800 (PST)
10 Received-SPF: pass (google.com: domain of thi.ba.tm.1@gmail.com designates 209.85.220.41
11         as permitted sender) client-ip=209.85.220.41;
12 Authentication-Results: mx.google.com;
13         dkim=pass header.i=@gmail.com header.s=20161025 header.b=IRT7CIdv;
14         spf=pass (google.com: domain of thi.ba.tm.1@gmail.com designates 209.85.220.41 as permitted sender)
15         smtp.mailfrom=thi.ba.tm.1@gmail.com;
16 Return-Path: <thi.ba.tm.1@gmail.com>
17 Received: from Timos-MBP.fritz.box ([194.135.129.204])
18         by smtp.gmail.com with ESMTPSA id g16sm31655039wru.41.2018.12.29.02.49.13
19         for <THI.BA.TM.2@gmail.com>
20         (version=TLS1_2 cipher=ECDHE-RSA-AES128-GCM-SHA256 bits=128/128);
21         Sat, 29 Dec 2018 02:49:13 -0800 (PST)
22 From: TM_1 Bachelorarbeit <thi.ba.tm.1@gmail.com>
23 X-Google-Original-From: TM_1 Bachelorarbeit <THI.BA.TM.1@gmail.com>
24 To: THI.BA.TM.2@gmail.com
25 Subject: Hallo Welt !
26 Message-ID: <eeeabfac-c29e-98e6-f866-d1a6fc33d99b@gmail.com>
27 Date: Sat, 29 Dec 2018 11:49:12 +0100
28 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:60.0) Gecko/20100101 Thunderbird/60.3.2
29 MIME-Version: 1.0
30 Content-Type: text/plain; charset=utf-8; format=flowed
31 Content-Transfer-Encoding: 7bit
32 Content-Language: en-US
33
34 Hallo Welt !
```

Quellcode 1: E-Mail mit Header und Body

In dem Quellcode 1 ist eine E-Mail im Plaintextformat zu sehen. Damit die Übersichtlichkeit gegeben ist, wurden mehrere Header entfernt. Dennoch ist zu sehen, dass viele Informationen in den Headern der E-Mail zu finden sind. Der Betreff der E-Mail ist in dem Header 'Subject' zu finden, in diesem Fall 'Hallo Welt !'. Weiter ist die Nachricht und somit der Body der E-Mail die letzte Zeile, mit dem Inhalt 'Hallo Welt !'. Zu sehen ist, dass mehrere Informationen von MTA wie 'Received', mit Details zu der Herkunft der Mail, hinzugefügt wurden. Auch Informationen vom MUA sind in den Header codiert worden, wie die Eigenschaft 'User-Agent', welche angibt, dass die E-Mail mit Mozilla Thunderbird verschickt wurde.

Im Body befindet sich die eigentliche Nachricht. Hierbei kann der Text als sogenannter 'Plain Text' oder im HTML-Format codiert sein. 'Plain Text' bedeutet, es werden nur die einzelnen Zeichen ohne jeglicher Formatierung in der E-Mail versendet. Texte im HTML-Format sind hingegen formatiert und können auch Bilder enthalten. Zum Vergleich: Websites werden auch in HTML formatiert und die Nachrichten können hierdurch die gleiche Designvielfalt aufweisen. Weiter sind im Body Anhänge zu finden, die mit der E-Mail versendet werden sollen. Auch solche müssen in eine ASCII-Zeichenfolge codiert sein, wenn sie mit einer E-Mail versendet werden sollen.[17]

2.2.2 Versand von E-Mail Nachrichten

Die digitalen Nachrichten werden vom Nutzer im MUA verfasst. Dieser fügt weitere Informationen an die E-Mail hinzu und codiert die Nachricht und Anhänge in die richtigen Formate. Hierfür werden wichtige Details im Header der E-Mail hinterlegt. Weiter übersendet der MUA die versendete E-Mail an den MTA über das SMTP Protokoll. In diesem Kontext muss sich der MUA authentifizieren, damit sichergestellt ist, dass nur autorisierte Individuen E-Mails an den MTA versenden können. Durch diese Eigenschaft soll Spam vermieden werden, da ansonsten der MTA als 'Open-Relay' bezeichnet wird, denn dieses kann als Verbreitungswerkzeug von Spam Nachrichten verwendet werden, wenn jedes Individuum Nachrichten versenden kann. Der MTA ermittelt mithilfe der E-Mail Adresse und des DNS Protokolls die IP-Adresse des empfangenden MTAs. Hat er diese ermittelt, übersendet er die Nachricht mithilfe des SMTP Protokolls an diesen.

Der MUA des adressierten Individuums kann die E-Mail mithilfe des IMAP oder POP3 Protokolls bei dem MTA abrufen und dem Benutzer formatiert und decodiert anzeigen.[17] Der Versandprozess einer E-Mail soll in Abbildung 4 schematisch dargestellt werden.

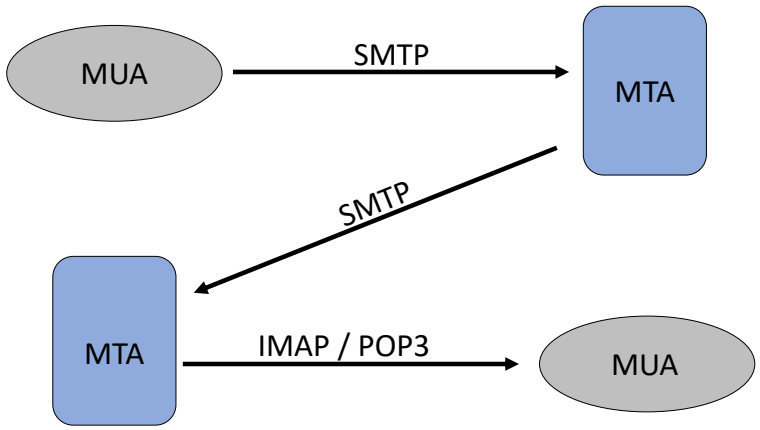

Abbildung 4: Transportweg einer E-Mail[4]

[4] Abbildung 4 in Anlehnung an [17] S. 257

3 Verwandte Arbeiten

Weiter sollen Arbeiten mit ähnlichen Konzepten für die Bepreisung der Aufmerksamkeit oder die Verifikation des Empfängers vorgestellt werden. Es soll auch differenziert werden, worin der Unterschied zu dieser Arbeit liegt.

3.1 Hashcash

Die Idee von Hashcash ist, dass der Sender Rechenaufwand und somit Zeit investiert. Der Empfänger soll nachprüfen können, ob der Sender diesen investiert hat. Durch dieses Kriterium kann Spam erkannt werden.[2] Da Spam Sender die Nachrichten an eine große Masse von Empfängern senden, ist es diesen nicht möglich, für jeden Empfänger Rechenaufwand und dadurch Zeit zu investieren.

Konkret ist das Verfahren wie folgt umgesetzt. Der Sender fügt eine neue Header Eigenschaft hinzu. Diese enthält die E-Mail Adresse des Empfängers, das Datum und eine Zufallszahl. Über diese Header Eigenschaft wird ein Hash-Wert gebildet. Beginnt dieser nicht mit einer bestimmten Anzahl an Nullen, wird die Zufallszahl so oft durch eine andere ersetzt, bis diese Eigenschaft gegeben ist.[1, 2]

Somit basiert das System darauf, Hash-Kollisionen zu finden. Dies ist nur durch Brute-Force möglich, daher werden viele Versuche benötigt bis eine geeignete Zufallszahl gefunden wurde. Die Länge des Hash-Wertes und die Anzahl an vorangegangenen Nullen müssen in diesem Konzept so gewählt werden, dass diese Aufgabe nicht zu langsam, jedoch auch nicht zu schnell lösbar ist.

Dieses Konzept enthält eine Schwachstelle, denn das Finden der Hash-Kollisionen auf einem System (simultan) benötigt sehr viel Rechenaufwand und Zeit. Diese Aufgabe ist

jedoch sehr gut parallelisierbar. Das bedeutet, viele unabhängige Individuen mit unterschiedlicher Rechenzeit können diese Aufgabe aufteilen. Wenn viele E-Mails versendet werden wollen, und es gleich viele Individuen gibt, reicht es, wenn jeder von diesen eine Kollision für eine Adresse findet. Hier bieten sich sogenannte Botnetze an. Dies sind Zusammenschlüsse von vielen Computern, denen über das Internet Befehle erteilt werden können. Meist sind diese ungewollt durch Schadsoftware an ein solches Netz angeschlossen. Somit kann das Auffinden von Hash-Kollisionen an diese einzelnen Rechner aufgeteilt werden und auch Spam-Sender können beweisen das für den Versand Rechenzeit investiert wurde, auch wenn dies nicht ihre eigene war.[9]

Der Unterschied zu dieser Arbeit ist, dass bei Hashcash der Sender mit Rechnenzeit bezahlt und nicht mit einer eigenen Währung, die zum Empfänger transferiert wird. Weiter zielt das Konzept dieser Arbeit auf ein ausgeglichenes Sendeverhalten ab. Das bedeutet, ein Individuum sendet maximal soviel, wie dieser auch an Nachrichten erhält. Dies ist bei Hashcash nicht der Fall, da bei diesem die Anzahl an gesendeten Nachrichten nicht limitiert ist und der Sender für jede Nachricht Rechenzeit investiert. Solange dieser Rechenzeit aufwendet, gibt es keine Einschränkungen bezüglich der Anzahl an gesendeten Nachrichten. Bei dem Konzept dieser Arbeit hingegen ist die Anzahl an maximal gesendeten Nachrichten durch die Anzahl an erhaltener Nachrichten limitiert. Dadurch wird ein ausgeglichenes Verhältnis zwischen gesendeten und empfangenen Nachrichten angestrebt.

3.2 SAVE - Sender Address Verification Extension

Die Idee von SAVE (Sender Address Verification Extension) ist, Spam durch die Verifikation des Senders zu verhindern. Hierfür wird ein Challenge/Response Verfahren verwendet. Das bedeutet, der Empfänger triggert den Sender an seine Identität bzw. Existenz zu beweisen. Dieser Beweis kann 'manuell' oder 'automatisch' lösbar sein. Manuell lösbar bedeutet, ein Mensch muss die Aufgabe lösen, wie beispielsweise ein reCAPTCHA (Google).[5] Diese Tests stellen einen Turingtest da, indem bewiesen wird, dass das Individuum kein Computer ist, sondern ein Mensch. Automatisch lösbar hingegen sind Aufgaben,

[5] Weitere Details unter: `https://www.google.com/recaptcha` (aufgerufen am 06.12.2018)

die durch Rechenaufwand von einem Computer gelöst werden können, wie finden von Hash-Kollisionen. Durch das Lösen der Aufgabe muss Rechenzeit oder Zeit des Senders investiert werden, dies ist bei großen Mengen an Spam E-Mails nicht möglich. Durch dieses Prinzip funktioniert die Reduzierung von Spam aufgrund der Verifikation des Senders.[3]

Im Grundsatz beruht dieses Konzept darauf, dass der Sender Zeit investieren muss. Entweder durch eine Aufgabe, die ein Mensch lösen muss oder durch ein rechenintensives Rätsel, das ein Computer mit vielen Versuchen und langer Rechenzeit bearbeiten kann. Mit dem Unterschied zu Hashcash, dass der Sender dies nicht bei dem Versenden einer Nachricht investieren muss. Der Empfänger muss diese Verifizierung auslösen und gibt somit vor, wann der Sender Zeit bzw. Rechenzeit investieren muss. Weiter muss in diesem Konzept vom Sender nicht bei jeder Nachricht Zeit oder Rechenzeit investiert werden, da nur eine einmalige Verifizierung des Senders erfolgt.

Der Unterschied zu dieser Arbeit ist, dass bei SAVE der Sender Zeit investieren muss. Bei dieser Ausarbeitung ist es eine eigene Währung, die dem Sender nur begrenzt zur Verfügung steht. Zeit als Währung und eine eigene Kryptowährung können dahingehend differenziert werden, dass die Zeit von anderen Individuen wie zum Beispiel Bots investiert werden kann. Die Kryptowährung steht dem Sender nur in dem Umfang zur Verfügung, wie viele E-Mails dieser erhält, also andere Individuen diesen damit bezahlen.

Weiter schließen sich das Konzept dieser Arbeit und SAVE nicht gegenseitig aus. Eine Kombination aus beiden kann ein interessantes Konzept darstellen, da es durch SAVE möglich ist zu verifizieren, dass es sich um einem menschlichen Sender handelt.

4 Analyse der Umgebungsbedingungen und aktuellen Situation

Die Ausgangssituation und Umgebungsbedingungen, sowie beteiligte Individuen sollen nachfolgend analysiert werden. Dies ist nötig, um die notwendigen Komponenten und Bedürfnisse der einzelnen Individuen zu erfassen.

4.1 Aktuelle Herausforderung im E-Mail System

Der Versand von E-Mails ist sehr kostengünstig bzw. kostenfrei. Alles was für den Versand einer E-Mail benötigt wird, ist eine Mail-Adresse an einen Mail-Server. Diese sind bei diversen Freemail Anbietern kostenfrei zu erhalten. Es kann auch ein eigener Mail-Server dank OpenSource Lösungen auf dem eigenen Computer kostenfrei installiert werden. Mithilfe dessen kann an beliebig viele Empfänger eine E-Mail mit beliebigen Inhalt versendet werden.[12] Diese Situation erklärt durchaus das hohe Aufkommen von E-Mail-Nachrichten. So ist die Anzahl der versendeten E-Mails (ohne Spam) in Deutschland im Jahr 2018 bei 917 Milliarden Nachrichten. Dies sind mehr als doppelt so viele Mails wie noch im Jahr 2012 (432 Milliarden).[16]

Derzeit gibt es viele E-Mail Nachrichten, die vom Empfänger unerwünscht sind, so beispielsweise Werbung, Phishing oder allgemein Spam-Nachrichten. Im September 2018 war der Anteil an Spam E-Mails im gesamten E-Mail Verkehr bei 53,49%.[10] Versuche, anhand von gut gestalteten Phishing-Nachrichten an interessante Informationen, wie Login-

Daten oder Bankverbindungen, zu gelangen, basieren auf einer hohen Empfängerzahl. Da mehrere Tausende diese Nachricht erhalten, hofft der Sender, dass nur wenige einzelne auf diese Nachricht antworten und Opfer seines Betruges werden. Bei anderen Formen des Spams, wie Werbung für verschiedenste Produkte, basiert auf dem Prinzip auf hohen Empfängerzahl. Somit ist festzuhalten, dass der Versand einer E-Mail an möglichst viele Empfänger, die Grundlage für Spam E-Mails darstellt. Um solche Nachrichten zu verhindern oder den Empfänger zu schützen, wird heutzutage oft ein Spam-Filter verwendet. Dieser versucht, diese Nachrichten anhand von verschiedenen Kriterien zu erkennen und dem Empfang in einem gesonderten Bereich anzuzeigen. Diese Filter sind nur mäßig effektiv, da der Erfolg stark von den Filterkriterien abhängig ist. Solche können zum einen Absender-Adresse oder bestimmte Wortgruppen im Betreff, aber auch sonstige Inhalte im Text der E-Mail sein. Diese Filter kommen jedoch aufgrund der Komplexität dieser Suche und Menge von Nachrichten an ihre Grenzen. Daher werden andere Konzepte benötigt, um die Empfänger solcher Nachrichten zu schützen.

4.2 Verschiedene Kommunikationsarten

Die Verwendung des E-Mail Systems ist von verschiedenen Nutzergruppen und Verhaltensweisen geprägt. Daher sollen verschiedene Arten der E-Mail Nutzung betrachtet werden. Weiter soll auch analysiert werden wie ein Briefmarkensystem mit wiederverwendbaren Briefmarken angewendet werden könnte.

Individuelle E-Mails

Die Kommunikation von einem Individuum zu einem anderen stellt eine wichtige Form der E-Mail Kommunikation dar. Bei dieser Art des Austausches sind zwei Akteure beteiligt, der Sender und der Empfänger. Dies ist die einfachste Form einer E-Mail, da lediglich zwei Individuen beteiligt sind und es nur einen Empfänger der Nachricht gibt. Hierbei ist das Anwenden des Briefmarkensystems sehr intuitiv, da eine Briefmarke vom Sender zum Empfänger übermittelt wird.

E-Mails mit wenigen Empfängern

Die Kommunikation von einem Individuum zu mehreren ist nicht zu vernachlässigen. Hierbei handelt es sich um die verbreitete CC-, BCC-Funktion und kleine Verteilerlisten bzw. mehrere Empfänger einer E-Mail. Die Grenze zwischen mehreren und vielen Empfängern ist schwer zu ziehen. In diesem Kontext ist davon auszugehen, dass es sich um mehrere Empfänger handelt, wenn die Anzahl unter zehn ist, aber vor allem, wenn von den meisten oder allen eine Antwort erwartet wird.

Diese Abgrenzung ist wichtig, da E-Mails mit wenigen Empfängern als eine eins zu eins Kommunikation abgebildet werden kann, indem der Sender jedem Empfänger eine Briefmarke übermittelt.

E-Mails mit vielen Empfängern

Bei der Übermittlung von E-Mails von einem zu vielen, sendet ein Individuum an viele Empfänger die gleiche oder eine ähnliche Nachricht. Hierbei ist es nicht selten, dass mehrere hunderte diese Informationen erhalten. Beispiele für diese sind Newsletter, Werbemails und auch Phishing oder allgemein Spam Nachrichten.

Das Anwenden eines Briefmarkensystems auf solche E-Mails ist nur schwer oder gar nicht möglich, dar die Kommunikation nicht mehr oder nur unter großem Aufwand auf eine eins zu eins Kommunikation abgebildet werden kann. Auch wird für den Versand solcher Nachrichten eine große Anzahl an Briefmarken benötigt oder die Nachrichten müssen ohne Briefmarken versendet werden.

In diesem Zusammenhang muss zwischen unerwünschten und erwünschter Nachrichten unterschieden werden. Falls unerwünschte Nachrichten nicht mehr mit Briefmarken versendet werden können, ist dies ein Erfolg für das Briefmarkensystem. Jedoch gibt es auch erwünschte Nachrichten, die an viele Empfänger adressiert sind. Beispiele hierfür sind vom Nutzer erwünschte Newsletter oder Mailinglisten. Für diese vom Empfänger gewünschte Kommunikation ist eine Lösung zu finden.

4.3 Verschiedene Nutzer des E-Mail Systems

Bei dem Versand von E-Mail Nachrichten sind verschiedene Individuen beteiligt. Deren Verhalten und potenziellen Nutzen aus der Verwendung eines Briefmarkensystems für E-Mails soll analysiert werden.

Empfänger von E-Mail Nachrichten

Die Intention des Empfängers ist es, Nachrichten zu erhalten. Dieser will Informationen verarbeiten, die ihm andere zur Verfügung stellen oder mit anderen Individuen kommunizieren. Hierbei kann es auch beabsichtigt sein, Informationen zu verbreiten, da Fragen des Senders in einer Antwort-Mail beantwortet werden.

Der Empfänger von E-Mail Nachrichten profitiert durch die Vorsortierung und erhält die Informationen die ihn interessierten in einem gesonderten Postfach. Daher kann dieser schnell und einfach die wichtigsten E-Mails erkennen. Weiter ist es für diesen leichter, die potenzielle Echtheit von wichtigen Nachrichten zu erkennen, da ein weiteres Kriterium eingeführt wird, dass von Phishing- und Spam-Nachrichten nicht erfüllt wird. Dadurch profitiert der Empfänger von E-Mails durch die erleichterte Erkennung von betrügerischen Nachrichten.

Sender von E-Mail Nachrichten

Die Intention des Senders ist es Nachrichten zu versenden. Dieser will Informationen verbreiten und anderen zur Verfügung stellen oder mit einem anderen Individuum kommunizieren. Hierbei kann auch beabsichtigt sein, Informationen zu erhalten, da Fragen an den Empfänger gestellt werden können.

Die Sender zahlen in diesem Kontext für die Aufmerksamkeit des Empfängers. Was negativ klingt, aber nicht nachteilig zu beurteilen ist, da der Sender auch Nachrichten und Briefmarken erhält, die er selbst als Sender wieder verschicken kann. Weiter kann er davon ausgehen, dass die Nachricht schneller beantwortet wird, da der Empfangene die Nachrichten mit Briefmarken priorisieren wird.

Vermittler von E-Mail Nachrichten

Unter den Vermittlerdiensten sind Mail Transfer Agents (MTA) zu verstehen. Diese sind für die Übermittlung der E-Mails verantwortlich. Meist sind diese als Freemail Anbieter bekannt. Diese Dienste wickeln den gesamten Sende und Empfangsprozess der E-Mail Nachrichten ab.

Das Briefmarkensystem kann auf verschiedenen Ebenen implementiert werden. Einerseits auf Nutzerebene, andererseits aber auch auf Serverebene. Bei der Implementierung auf Serverebene können auch MTAs profitieren, da diese die Briefmarken erhalten und dafür die E-Mails in die verschiedenen Inboxes aufteilen. Weiter werden keine Briefmarken benötigt, wenn Sender und Empfänger den selben Freemail Anbieter nutzen, da die Nachricht den Server nie verlassen muss und der MTA die E-Mail direkt ohne Briefmarke in die richtige Inbox legen kann. Die dadurch überschüssigen Briefmarken kann der MTA verkaufen, wodurch sich ein neues Geschäftsmodell ergibt. Denn Firmen können diese Briefmarken kaufen und damit für die Aufmerksamkeit bestimmter Nutzer zahlen, wenn diese E-Mails von dieser Firma erhalten. Da diese Nachrichten, aufgrund der Briefmarken, den Firmen Geld kosten, werden diese jedoch nur sparsam und mit bedacht verwendet. Weiter kann es auch für privat Personen interessant sein Briefmarken bei den MTAs zu erwerben, wenn diese eine größere Anzahl an E-Mails versenden wollen. Als Beispiel ist hier die Einladung zu einer Geburtstagsfeier oder Hochzeitsfeier anzubringen.

5 Erweiterung des E-Mail Systems

Das E-Mail System muss in der Funktionalität erweitert werden, um das Konzept eines Briefmarkensystems für E-Mails verwirklichen zu können. Diese konzeptionellen Erweiterungen sollen nachfolgend vorgestellt werden.

5.1 Grundkonzept für ein Briefmarkensystem für E-Mails

Ein neuartiger Ansatz ist, dass der Sender einer E-Mail, den Empfänger der Nachricht für seine Aufmerksamkeit 'bezahlt'. Diese Bezahlung kann als eine Art wiederverwendbare Briefmarke gesehen werden, die der Sender dem Empfänger übermittelt.

In einer normalen Nutzung von E-Mails werden ungefähr genau so viele Nachrichten empfangen wie versendet. Hierdurch gleicht sich das Vergeben und Erhalten von Briefmarken aus. Im Falle von Phishing oder Spam-Nachrichten ist es den Sendern nicht möglich Briefmarken zu senden, da in diesen Fällen tausende Nachrichten gesendet, jedoch nie welche empfangen werden.

Somit können diese Briefmarken als Währung betrachtet werden, da mit jeder empfangenen E-Mail das 'Guthaben' um eins inkrementiert wird. Beim Senden wird nicht nur die E-Mail an den Empfänger übermittelt, sondern auch die Bezahlung für seine Aufmerksamkeit. Somit sinkt das eigene Guthaben von Briefmarken und das des Empfängers steigt. Bei verbreiteter Kommunikation mit darauf folgenden Antworten wird der Sender zum Empfänger und erhält die zuvor übermittelte Briefmarke zurück. Hierdurch entsteht ein Gleichgewicht bei ausgeglichenem Senden und Empfangen von E-Mail Nachrichten.

5.2 Anforderungen an das Briefmarkensystem

Folgende Anforderungen ergeben sich an ein Briefmarkensystem für E-Mails. Das System muss einfach nutzbar sein und auch ohne der Unterstützung des MTA verwendet werden können. Jedoch sollte es auch erweiterbar und sowohl auf Ebene des MTA als auch des MUA implementierbar sein. Diese Eigenschaft ist wichtig, damit das System für eine große Menge von Menschen nutzbar wird, denn hierdurch kann das System ohne Unterstützung des Mail Anbieters genutzt werden. Zusätzlich ergeben sich für Freemail Anbieter neue Einnahmequellen zur Finanzierung ihrer Dienste.

Weiter muss die Integrität der Briefmarken sichergestellt werden. Dies sollte mithilfe einer Blockchain geschehen, da hierdurch die Unveränderbarkeit der vorangegangenen Transaktionen sichergestellt ist. Diese Anforderung muss eingehalten werden, um die Übermittlung der Briefmarken nachvollziehbar zu gestalten und sicher zu stellen, dass die Menge an verfügbaren Briefmarken nicht manipuliert werden kann.

Das E-Mail System soll lediglich erweitert werden und alle Funktionen des bisherigen E-Mail Aufbaues müssen erhalten bleiben. Dies beinhaltet unter anderem, das eine E-Mail mehrere Empfänger haben kann, sowie die Nutzung der CC und BCC Funktion. Um eine sinnvolle Erweiterung des E-Mail Systems vorzunehmen, ist es essenziell, die bisherigen Funktionalitäten nicht einzuschränken. Die Nutzer sind es gewohnt alle bisher vorhanden Features zur Verfügung zu haben, bzw. haben ihre Arbeitsprozesse darauf abgestimmt.

Diese Anforderungen müssen in dem Konzept berücksichtigt und umgesetzt werden.

5.3 Aufbau des Briefmarkensystems

Bei dem Versand einer E-Mail wird zuerst dem Empfänger eine Briefmarke übermittelt. In der Nachricht wird dem Empfänger die Transaktions-ID des Vorgangs mitgeteilt. Anhand dieser kann der Empfangende die Übermittlung der Briefmarke verifizieren und so feststellen, ob der Sender für die Aufmerksamkeit bezahlt hat. Damit es diesem möglich ist eine Briefmarke zu übermitteln, muss der Sender die Account-ID des Empfängers bekannt sein. Auf diese Problemstellung wird im Abschnitt 5.4 näher eingegangen.

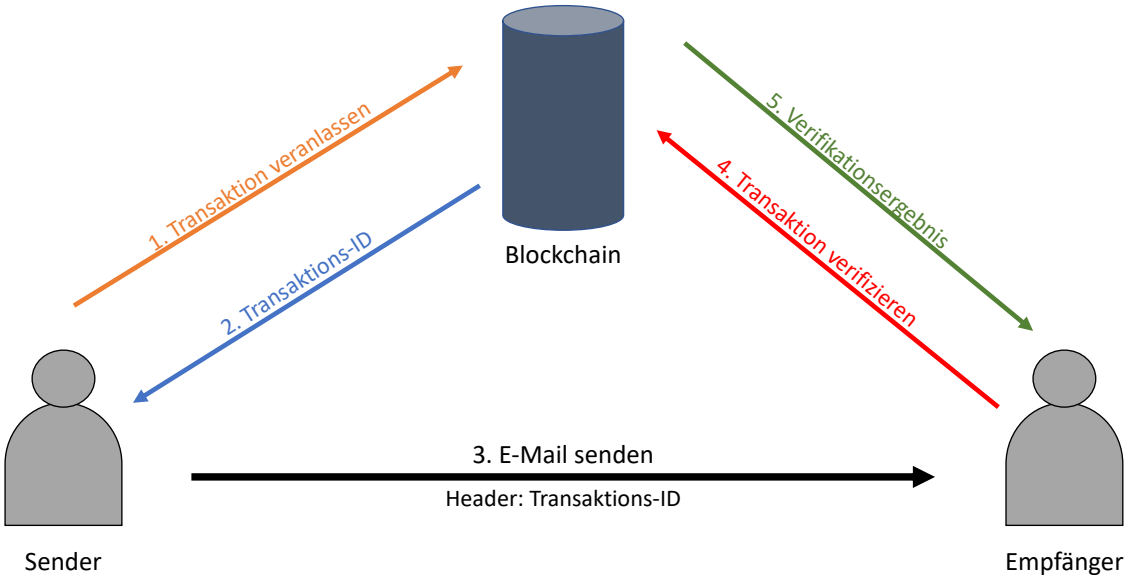

Abbildung 5: Grundsätzlicher Aufbau

Wie in der Abbildung 5, durch den orangenen Pfeil dargestellt, muss der Sender erst eine Transaktion der Briefmarke veranlassen. Für diese erhält er eine Transaktions-ID von der Blockchain, blau dargestellt. Bei dem Senden der E-Mail fügt der Sender diese ID im Header der E-Mail mit an. Hierbei ist zu beachten, dass die E-Mail erst versendet werden sollte, wenn die Transaktion in der Blockchain erfolgreich war. Anhand der eindeutigen Transaktions-ID kann der Empfänger die Transaktionsdetails anfordern, rot dargestellt. Durch diese kann verifiziert werden, ob eine valide Transaktion vom Sender zum Empfänger ausgeführt wurde, grün dargestellt.

Das Versenden von Nachrichten erfolgt nach dem in Abbildung 6 dargestellten Ablauf. Hierfür muss zuerst die Account-ID des Empfängers ermittelt werden, ob der Sender eine verfügbare Briefmarke besitzt. Ist dies der Fall, wird eine Briefmarke versendet. Die

zugehörige Transaktions-ID wird in der E-Mail hinzugefügt, da anhand von dieser der Empfänger die Transaktion verifizieren kann.

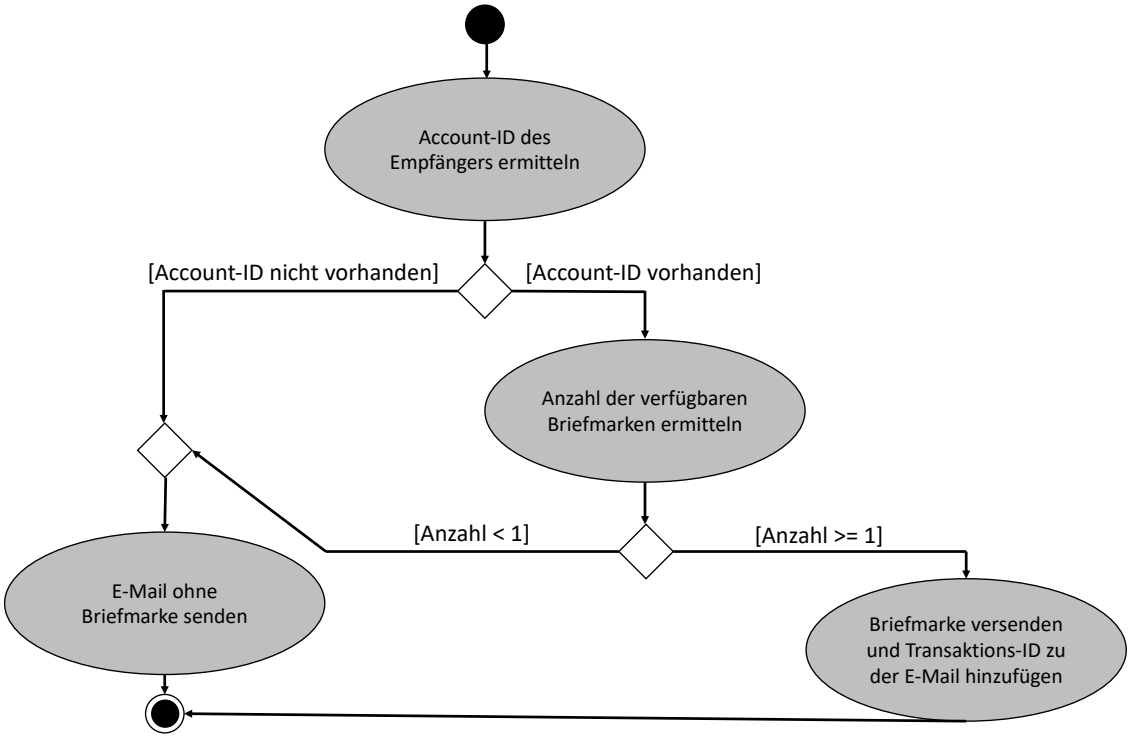

Abbildung 6: Ablaufdiagramm des Versands von Briefmarken

Das Sortieren der E-Mails erfolgt nachdem in Abbildung 7 dargestellten Ablauf. Hier muss zuerst festgestellt werden, ob die Unterordner 'Stamp' und 'No_Stamp' existieren. Falls nicht, werden diese erstellt. Demnach wird für jede Nachricht überprüft, ob eine Briefmarke vorhanden ist. Ist dies der Fall, muss diese verifiziert werden, ob diese auch gültig ist. Wenn dies zutrifft, wird die Nachricht in den Ordner 'Stamp' verschoben. Falls keine oder eine ungültige Briefmarke vorhanden ist, wird die Nachricht in den Unterordner 'No_Stamp' verschoben. Wenn alle Nachrichten überprüft und in einen der beiden Unterordner verschoben wurden, ist die Sortierung abgeschlossen.

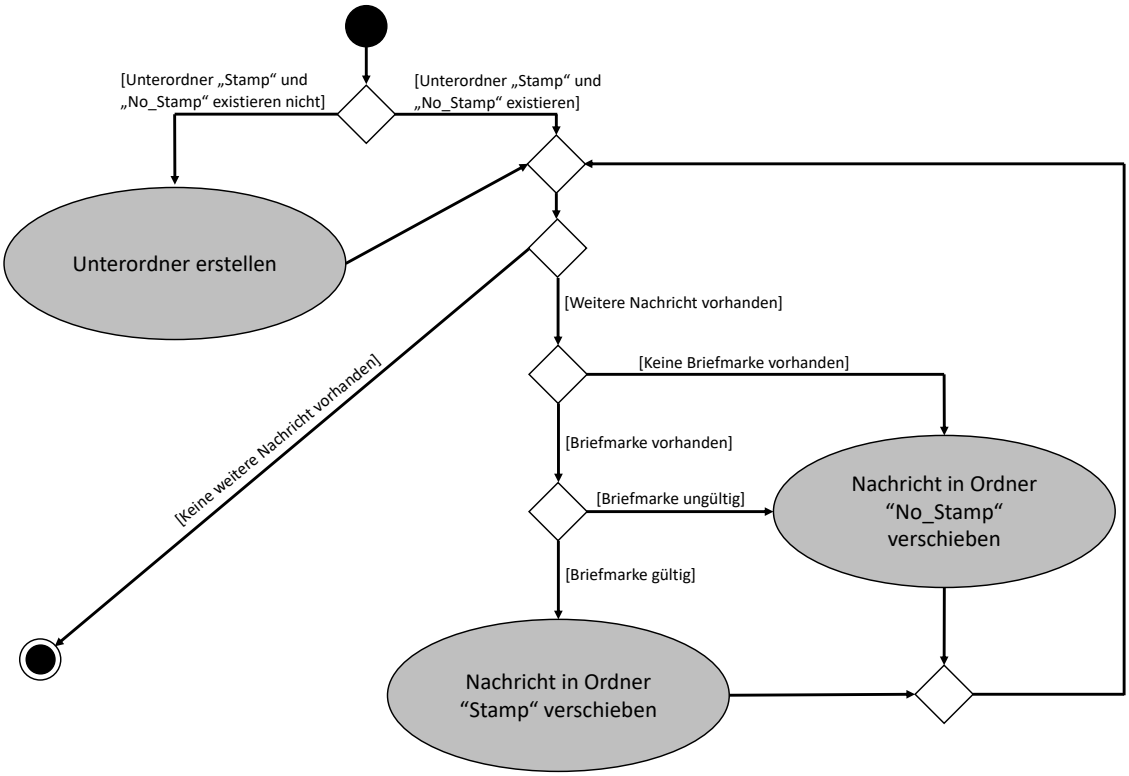

Abbildung 7: Ablaufdiagramm der Sortierung von Briefmarken

5.4 Konzeptionierung der Briefmarken

Der Empfänger wird von dem Sender für seine Aufmerksamkeit bezahlt. Die Briefmarken stellen in diesem Zusammenhang das Zahlungsmittel dar. Aufgrund dessen können die Briefmarken als Währung für die Aufmerksamkeit der Person gesehen werden.

Um die Integrität der Währung 'Briefmarken' sicherzustellen, ist es sinnvoll die Transaktionen in einer Blockchain festzuhalten. Hierdurch wird jeder Block von Transaktionen durch den darauf folgenden Block verifizierbar und es ist keine nachträgliche Manipulation möglich. Aufgrund dieser Nachvollziehbarkeit kann das Guthaben jedes einzelnen Nutzers berechnet werden. Daraus folgt, durch die Verwendung der Blockchain ist keine Manipulation am Guthaben der einzelnen Nutzer möglich.

Jede einzelne Transaktion muss innerhalb der Blockchain mithilfe des privaten Schlüssels signiert werden. Daher kann nur der Inhaber der privaten Schlüssels Transaktionen von einem Konto veranlassen. Das verhindert Missbrauch, jedoch ist darauf zu achten, das der private Schlüssel nicht weiter gegeben wird und nur dem autorisierten Individuum bekannt ist. Diese Signierung mit dem privaten Schlüssel kann von jedem Individuum in der Blockchain nachvollzogen werden, da der öffentliche Schlüssel gleichzeitig die Account-ID ist.[6]

Zu beachten ist, dass die Transaktions-ID dem Empfänger in der E-Mail mitgeteilt wird. Jedoch handelt es sich hierbei nicht um die Briefmarke, sondern durch diese Transaktions-ID kann auf die Übermittlung der Briefmarke in der Blockchain verwiesen werden. Hierdurch wird der Erhalt der Briefmarke für denn Empfänger verifizierbar. Somit wird die eigentliche Briefmarke nicht in der E-Mail mitgeschickt, den in dieser ist nur der Verweis auf die Details der Übermittlung. Die Briefmarke wird in der Blockchain zum Empfänger transferiert.

Damit ein Sender dem Empfänger Briefmarken übermitteln kann, muss diesem die Account-ID bekannt sein. Deshalb muss eine Möglichkeit geschaffen werden, die eigene Account-ID bei dem Senden einer E-Mail dem Empfänger mitzuteilen, auch ohne eine Briefmarke zu senden. Weiter muss diese Account-ID bei dem Empfänger extrahiert und abgespeichert werden, damit dieser dem Sender später Briefmarken übermitteln kann.

5.5 Konzeptionierung der Änderungen des E-Mail Aufbaues

In der E-Mail muss dem Empfänger kenntlich gemacht werden, dass der Sender für die Aufmerksamkeit des Empfängers gezahlt hat. Hierfür bietet sich der Header einer E-Mail an, denn dieser ist dem Mail User Agent (MUA) und Mail Transfer Agent (MTA) bekannt. Weiter erhält dieser viele Informationen, die für MUA und MTA wichtig sind, jedoch dem menschlichen Benutzer nicht interessieren. Für diesen soll das System funktionieren, aber gleichzeitig nicht mit Details überfordern. Daher werden die Header Informationen von MUA und MTA verarbeitet und nur wenige relevante Details aus diesen dem Verwender dargestellt.

Eine für den Empfänger wichtige Information ist, ob der Sender für dessen Aufmerksamkeit bezahlt hat. Anhand der Transaktions-ID kann der Empfänger die Bezahlung des Senders verifizieren. Für diese Verifizierung bietet sich die Transaktions-ID an, da anhand dieser alle relevanten Informationen abgerufen werden können.

Die empfangenen E-Mails werden aufgeteilt in zwei Kategorien. Diese sind Nachrichten mit und ohne einer validen Bezahlung. Dementsprechend werden die Nachrichten auf zwei Ordner aufgeteilt, um diese für den Nutzer getrennt darzustellen. Somit werden alle E-Mails mit Bezahlung gesondert abgelegt und können vom Nutzer priorisiert werden. Weiter erfolgt hierdurch eine Vorsortierung für den Nutzer. In diesem Kontext ist es möglich, eine Whitelisting Funktionalität zu integrieren für E-Mails ohne Briefmarken, die jedoch vom Empfänger gewünscht sind. Beispiele für solche Nachrichten sind vom Nutzer erwünschte Newsletter oder Mailinglisten. Der Vorteil dieses Systems liegt darin, dass der Nutzer selbst Adressen bestimmen kann, die auch ohne der Verwendung von Briefmarken, so behandelt werden, als wären Sie mit einer Briefmarke verschickt worden. Weiter ist eine Whitelisting Funktionalität wichtig, um nicht nur einzelne Adressen zulassen zu können, sondern auch Domains. Dies ist vor allem für Unternehmen essenziell, da hierdurch alle Adressen der eigenen Firma in dieser Whitelist aufgenommen sind, indem die Domain auf diese Liste gesetzt wird. Aber auch in der privaten Nutzung des Konzeptes ergeben sich viele Vorteile wenn vertraute Domains auf die Whitelist gesetzt werden können und

nicht nur einzelne Adressen. Hierdurch wird die Verwaltung von Whitelists einfacher und übersichtlicher.

Die Sortierung der E-Mails in Nachrichten mit und ohne Briefmarken und das Transferieren dieser, kann an verschiedenen Stellen des Versand- und Empfangsprozesses geschehen. Einerseits kann diese Logik auf Ebene des Mail Transfer Agents geschehen, dies bietet Vorteile für die Freemail-Anbieter, da diese eine zusätzliche Einnahmequelle erhalten. Die Integration beim MTA bietet einen weiteren Vorteil, da die Briefmarken zum MTA und nicht zu dem einzelnen Nutzer transferiert werden. Hierdurch ist in der Blockchain nicht mehr nachvollziehbar, welche Adressen miteinander kommunizieren, sondern nur noch, zwischen welchen MTAs ein Nachrichtenaustausch stattfindet. Durch diese Eigenschaft steigt die Privatsphäre der Verwender, da es für Dritte nicht mehr möglich ist eine Verbindung zwischen einzelnen Adressen herzustellen. Aus diesen beiden Gründen ist eine Umsetzung bei den MTAs nicht nur möglich, sondern auch sehr sinnvoll.

Andererseits kann dieses Verfahren auch bei dem Mail User Agent umgesetzt werden. Dies birgt den Vorteil, dass die Logik in dem Mailprogramm implementiert wird und somit einfach und ohne Unterstützung des Mail-Anbieters verwendet werden kann. Diese Variante ist für den Start eines solchen Systems sinnvoll, da jeder E-Mail Nutzer unabhängig von der Unterstützung seines Mail-Anbieters entscheiden kann dieses System zu nutzen. Die Integration bei den MTAs würde eine breite Unterstützung dieses Systems bei den Mail Transfer Agents voraussetzen. Beide Umsetzungsformen schließen sich nicht aus, denn das System funktioniert auch, wenn einerseits ein MTA diese Funktion anbietet und andererseits ein Verwender dieses System bei dem MUA nutzt. Der große Vorteil des Systems ist, dass es für jeden nutzbar ist, unabhängig von der Unterstützung seines MTA.

5.6 Konzept zur sicheren Validierung der Zuordnung Account-ID zur E-Mail Adresse

Für die Übermittlung von Briefmarken ist sowohl die E-Mail Adresse als auch die Account-ID der Blockchain nötig. Zweite wird dem Empfänger für zukünftige Nachrichten in dem Header einer E-Mail mitgeteilt. Da Header einer E-Mail manipuliert werden können, beispielsweise durch einen Man-in-the-Middle Angriff, muss eine Möglichkeit geschaffen werden, die Zuordnung von Account-ID zu E-Mail Adresse sicher verifizieren zu können. Dies ist einerseits durch Zertifikate möglich, hierfür muss die Zuordnung von Mail-Adresse zu Account-ID bestätigt werden. Andererseits ist es möglich diese Zuordnung in der Blockchain zu verwalten. Hierfür soll folgend ein Konzept erläutert werden. Die Umsetzung und weitere Ausarbeitung soll kein Teil dieser Arbeit sein.

Der Root of Trust stellt den Vertrauensanker für weiteres Vertrauen dar. Dass bedeutet man geht davon aus, dass dieser eine Zuordnung verifiziert hat und sich dieser sicher ist, wenn er diese bestätigt. Im nachfolgenden Konzept der Verifikation wäre der E-Mail Provider ein guter Root of Trust, da E-Mails zu diesem nicht den Server des Providers verlassen und somit nicht über das Internet geschickt werden. Sondern es sich auf dem Server des Providers um eine Kopierfunktion handelt und somit eine Manipulation von Dritten an dieser E-Mail unwahrscheinlicher ist. Jedoch sollte dieses Konzept auch mit einem Root of Trust sicher sein der nicht der zugehörige E-Mail Provider ist und E-Mails zu diesem manipuliert werden können. Weiter ist eine zeitliche Begrenzung von wenigen Stunden für diese Verifikation sinnvoll und sollte auch eingehalten werden, um Manipulationen zu erschweren. Außerdem muss bei der Umsetzung berücksichtigt werden, dass die E-Mails des Root of Trust Transport verschlüsselt an den Empfangenden MTA versendet werden. Ein sinnvoller Zeitpunkt der Verifizierung der Zuordnung von Account-ID zur E-Mail Adresse ist, bei der Einrichtung des Systems oder vor der erstmaligen Verwendung. Daher wurde in folgendem Konzept auf eine Transferirrung von Briefmarken verzichtet und anstelle von dieser eine Transferirrung von Ethereum verwendet.

Um einem Individuum die Zuordnung von E-Mail Adresse zu Account-ID zu beweisen, muss der Nachweis des Besitzes des E-Mail Accounts aber auch der des PrivateKeys

erbracht werden. Der Beweis des Besitzes eines E-Mail Accounts kann erbracht werden, indem der Erhalt einer E-Mail von diesem Account nachgewiesen wird. Der Besitz des PrivateKeys kann durch eine valide Transaktion in der Blockchain erbracht werden.

Aus diesen Definitionen folgt, dass der Root of Trust durch Erhalt einer E-Mail und dem Eingang einer Transaktion in der Blockchain die Zuordnung eines Individuums bestätigen kann. Somit könnte eine Verifizierung durch den Root of Trust wie folgt durchgeführt werden.

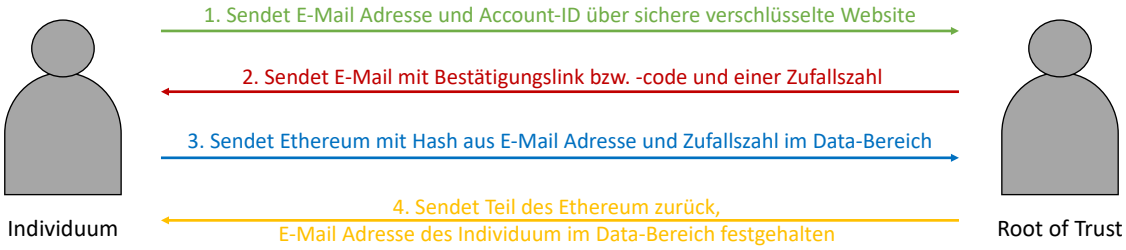

Abbildung 8: Registrierungsprozess beim Root of Trust

1. Der zu Verifizierende sendet seine E-Mail und Account-ID dem Root of Trust über eine sichere zum Transport verschlüsselte Website.
2. Der Root of Trust sendet dem zu Verifizierenden eine E-Mail mit einem Geheimnis oder Bestätigungslink. Den Erhalt muss der zu Verifizierende beweisen. Weiter bestätigt er durch den Link oder das Geheimnis, dass die angegebene Account-ID zu seiner E-Mail Adresse gehört. Zusätzlich übersendet der Root of Trust eine Zufallszahl.
3. Der zu Verifizierende sendet dem Root of Trust Ethereum und hält in dem Data Bereich einen Hash aus der E-Mail Adresse und der erhaltenen Zufallszahl fest. Eventuell ist eine neue Funktion in dem Briefmarken-Token nötig.
4. Der Root of Trust prüft alle Informationen, wie die Bestätigung des Links oder die Kenntnis über das Geheimnis und die erhaltene Transaktion mit dem Hash aus E-Mail Adresse und Zufallszahl im Data Bereich. Wenn all diese Informationen korrekt sind, sendet der Root of Trust dem zu Verifizierenden einen Teil des Ethereums zurück und hält in dem Data Bereich die gehashte E-Mail Adresse fest.

Durch den Punkt 1 erhält der Root of Trust die E-Mail Adresse und die Account-ID, dessen Zuordnung ein Individuum bei diesem bestätigen lassen will. Daraufhin bestätigt der Root of Trust durch den Punkt 2, ob es dem Individuum möglich ist, E-Mails zu empfangen und einen darin enthaltenen Link oder Geheimnis zu bestätigen. Weiter wird hier dem Individuum die Chance gegeben, die Zuordnung abzulehnen, falls dieser der Inhaber der E-Mail Adresse ist, jedoch nicht die angegebene Account-ID besitzt. Bei dem Punkt 3 bestätigt das Individuum, das es den PrivateKey zu der angegebenen Account-ID besitzt und die im Data Bereich eingegebene E-Mail Adresse verifizieren möchte. Eine Transaktion kann nur mit dem korrekten PrivateKey ausgelöst werden und somit zeigt der Inhaber des PrivateKeys, dass er diese E-Mail Adresse mit seiner Account-ID bestätigen will. Weiter bestätigt das Individuum die Kenntnis über die Zufallszahl, die vom Root of Trust versendet wurde. Durch die erhaltene Zufallszahl wird ein Man-in-the-Middle Angriff abgewehrt, bei dem die Account-ID ersetzt werden könnte. Wenn alle diese Kriterien erfüllt wurden und somit der Root of Trust überzeugt wurde, dass das Individuum Inhaber der E-Mail und des PrivateKeys ist, bestätigt er diese Zuordnung durch die Rücküberweisung der Briefmarken mit der gehashten E-Mail Adresse des Individuums in dem Data Bereich. Hierdurch ist die Zuordnung in der nicht manipulierbaren Blockchain enthalten und kann dort eingesehen werden. Weiter ist denkbar, dass der Root of Trust nur einen Teil des erhaltenen Ethereums zurück zum Individuum schickt, damit dessen Aufwand entlohnt wird.

Damit ein anderer eine erhaltene Zuordnung von diesem Root of Trust bestätigen lassen kann, muss dieser dem Root of Trust die E-Mail und Account-ID nennen. Weiter erhält er von dem Root of Trust die verifizierte Zuordnung zurück und als Beweis die Transaktions-ID der Rücküberweisung. Das Individuum kann somit prüfen, ob die Nachricht auf dem Weg vom Root of Trust manipuliert wurde. Durch die Transaktions-ID kann überprüft werden, von welchem Account zu welchem Account der Transfer stattfand und im Data-Bereich kann die E-Mail Adresse verifiziert werden. Da die Blockchain nicht manipuliert werden kann, ist eine Manipulation durch Dritte ausgeschlossen.

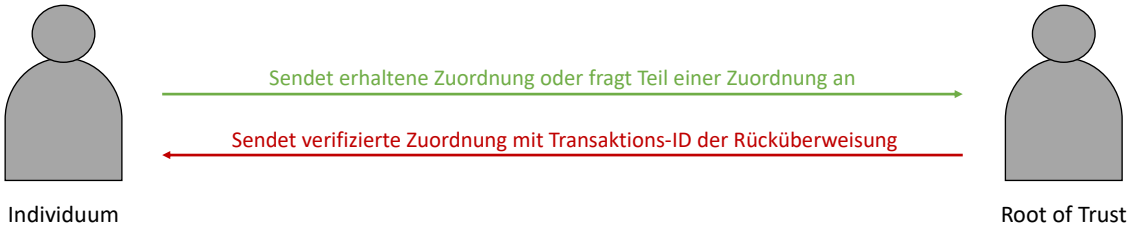

Abbildung 9: Verifikation einer Zuordnung beim Root of Trust

Als Nachteil dieses Konzeptes ist zu nennen, dass die E-Mail Adressen in der Blockchain frei zugänglich sind. Hierdurch kann ein Spam-Sender neue E-Mail Adressen durch diese finden. Da diese als Hash in der Blockchain gespeichert sind, ist jedoch ein nicht zu vernachlässigender Aufwand nötig, um von den Hash-Werten auf die E-Mail Adressen zu schließen. Denn Hash-Werte können nicht zurück zu dem ursprünglichen Wert gerechnet werden, jedoch ist es möglich über sogenannte Rainbowtables auf die Ursprungswerte zu schließen. Dies ist möglich, da die E-Mail Adressen relativ kurz sind. Ein weiterer Grund ist, dass viele Nutzer eine E-Mail Adresse von einem großen Freemail Anbieter besitzen und die Nutzer oft ihre Namen als Adresse verwenden. Hierdurch ist es theoretisch möglich durch die Kombination von häufigen Namen und den Adressen von großen Freemail Anbietern eine Datenbank von möglichen E-Mail Adressen und deren Hash-Werten zu erstellen. Jedoch stellt dies einen nicht zu vernachlässigenden Aufwand dar, denn es müssten viele Kombinationen von möglichen Namen und Freemail Anbietern kombiniert und in dieser Datenbank abgespeichert werden. Weiter zielt diese Arbeit auf die Verhinderung von Spam durch das Nutzen von Briefmarken ab, daher können die Spam Nachrichten aufgrund fehlender Briefmarken erkannt und separiert dargestellt werden.

6 Prototypumsetzung des Konzeptes

Die Implementierung des Konzeptes wurde in zwei Programme aufgeteilt. Einerseits in eine Extension für den E-Mail Client Thunderbird, andererseits in ein eigenständiges Programm zur Verwaltung der Briefmarken. Eine anfangs angestrebte komplette Implementierung in der Extension für Thunderbird war aus mehreren Gründen nicht möglich.

Thunderbird beendet Skripte mit einer zulangen Laufzeit automatisch. Die Dauer der Kommunikation mit dem Ethereum Netzwerk kann nicht vorausgesagt werden und die Ausführung des Skriptes wird durch Thunderbird beendet, bevor eine Antwort der Blockchain erhalten wurde. Weiter werden zur Kommunikation mit der Blockchain Bibliotheken aus NodeJS benötigt. Diese müssten mithilfe von Browserify zu einem Skript gebunden werden, damit sie in Thunderbird ausführbar sind. Hierbei ergibt sich jedoch die Komplikation, dass diese Skripte sehr groß werden und somit nicht vollkommen ausgeführt sind, bevor die Ausführung von Thunderbird beendet wird. Durch die vorzeitige Beendigung sind diese Bibliotheken nicht lauffähig und somit nicht verwendbar.

Ein weiterer Grund, der gegen die komplette Implementierung in Thunderbird spricht, ist die sogenannte 'same-origin policy'. Diese erlaubt in Browsern keine HTTP-Request von Skripten zu einer anderen Website (Domain) als gerade dargestellt.[6] Da Thunderbird auf dem Browser Firefox basiert, gilt diese Richtlinie auch in Thunderbird. Weiter ist es auch nicht möglich diese Richtlinie zu erfüllen, da verschiedene Schnittstellen benötigt werden und der Autor keinen Einfluss auf diese hat. Hierdurch führt die 'same-origin policy' zu einer Blockierung der Aufrufe von Blockchainschnittstellen.

[6] Für weitere Details zu dieser Richtlinie siehe: [14] und auch [13]

Aus diesen Gründen wurde die Implementierung des Konzeptes in zwei Programme unterteilt. Diese sollen nachfolgend vorgestellt werden.

6.1 Extension für Thunderbird

In der Extension für Thunderbird sind Funktionalitäten die direkt mit der E-Mail Interaktion verbunden sind gelöst. Es erfolgen die Sortierung, der Versand und die Auflösung von E-Mail Adressen zu Account-IDs der Blockchain.

6.1.1 Sortierung der E-Mails

Die E-Mails sollen in zwei Kategorien unterteilt werden, 'Stamp' und 'No_Stamp' und somit auch in diese zwei Unterordner verschoben werden. In dem Ordner 'Stamp' befinden sich E-Mails mit einer validen Bezahlung. Im Gegensatz zu dem Ordner 'No_Stamp' in diesem befinden sich alle Nachrichten ohne einer validen Bezahlung. Durch diese Unterteilung kann der Nutzer seine Aufmerksamkeit gezielt auf die E-Mails richten, deren Empfänger für seine Aufmerksamkeit bezahlt hat.

Konkret funktioniert dies über den Button 'StampSort' wie in Abbildung 10 dargestellt, durch den sich ein Kontextmenü öffnet.

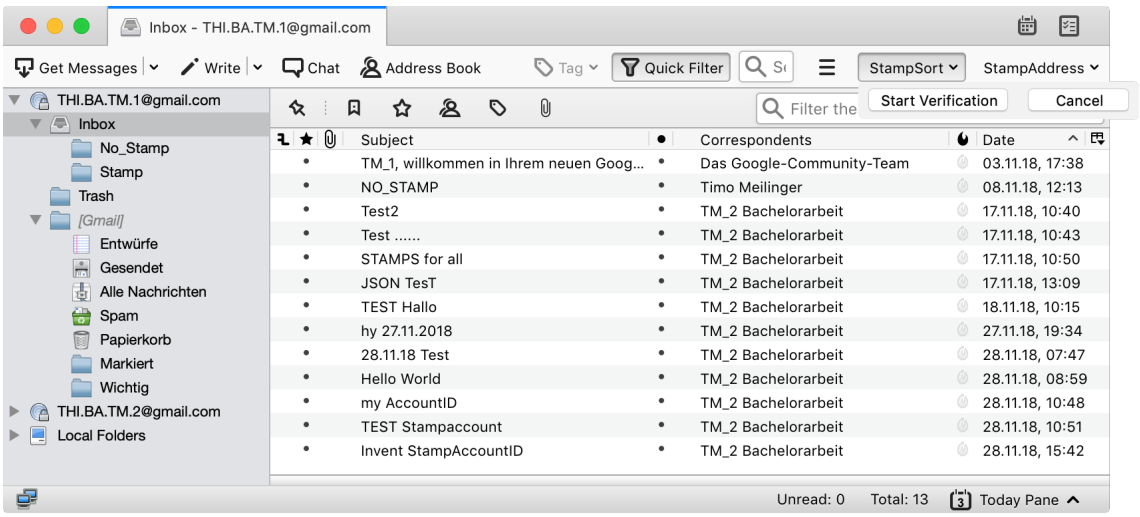

Abbildung 10: E-Mail Sortierung 'StampSort' starten

Wird der Button ’Start Verification’ betätigt, überprüft das Programm, ob die zwei Unterordner ’Stamp’ und ’No_Stamp’ existieren. Falls nicht, werden diese erstellt. Es werden alle Nachrichten durchsucht und, falls vorhanden, deren Transaktions-ID extrahiert. Weiter werden E-Mails ohne der Headereigenschaft ’Stamp’ sofort in den Unterordner ’No_Stamp’ verschoben. Demnach ändert sich das Kontextmenü ab und zeigt ein JSON-Object an. Dieses muss der Nutzer in das Verwaltungsprogramm der Briefmarken zur Validierung kopieren. Das Ergebnis der Validierung des zusätzlichen Programms muss in dieses Fenster wieder eingefügt werden.

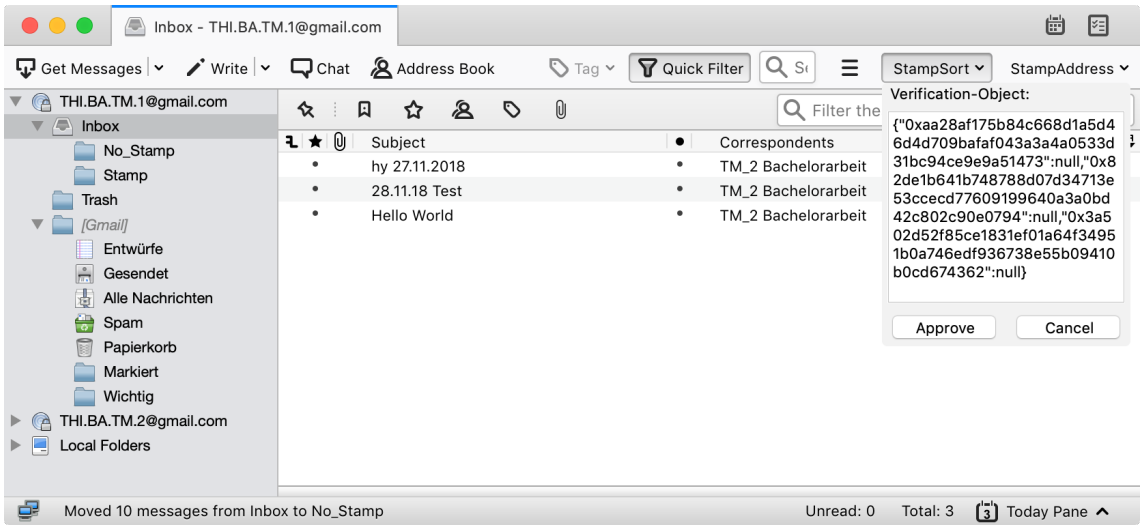

Abbildung 11: Verification-Object von Thunderbird erhalten

Nach der Betätigung des Buttons ’Approve’ überprüft das Programm nochmals alle E-Mails in dem Ordner. Nun kann die Extension anhand der validierten Transaktions-IDs entscheiden, welche der E-Mails eine valide Transaktion enthält und welche nicht. Anhand dieser Entscheidung werden die E-Mails in die Ordner ’Stamp’ und ’No_Stamp’ sortiert. Hierbei ist zu beachten, dass die Briefmarke nur für eine E-Mail gültig ist. Wenn mehrere Nachrichten auf die selbe Transaktions-ID verweisen, wird die erste E-Mail in den Ordner ’Stamp’ verschoben und darauffolgende Nachrichten in den Ordner ’No_Stamp’.

6.1.2 Versand der E-Mails

Bei dem Versand von E-Mails kann vom Benutzer eine Briefmarke an den Empfänger transferiert werden. Konkret wird dazu der Header einer E-Mail um die Eigenschaft 'Stamp' erweitert. In dieser wird ein JSON-Objekt gespeichert in dem zu der Account-ID des Empfängers die Transaktions-ID der Briefmarke zugeordnet gespeichert wird. Jeder Empfänger kann zu seiner Account-ID die zugehörige Transaktions-ID ermitteln, um die Briefmarkentransaktion zu verifizieren.

In der Umsetzung ist dies mit dem 'StampSend' Button möglich. Wenn dieser betätigt wird, öffnet sich ein Kontextmenü, hier wird die 'Recipient-Address' und 'TX-Hash' abgefragt. Unter der 'Recipient-Address' ist die Account-ID des Empfängers zu verstehen. Der 'TX-Hash' ist die Transaktions-ID der Briefmarke zu dem Empfänger. Hier muss das Ergebnis des zusätzlichen Programms eingetragen werden, das in einem Pop-up Fenster angezeigt wird. Zu sehen in Abbildung 18.

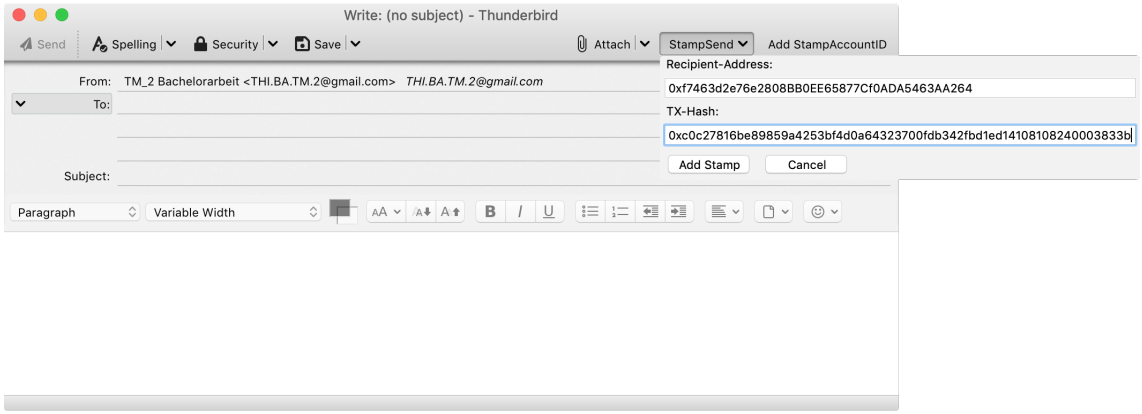

Abbildung 12: Versendete Briefmarke zur E-Mail hinzufügen

Nach der Bestätigung wird überprüft, ob das genannte JSON-Objekt in dem Header schon vorhanden ist und wenn ja, wird dieses erweitert bzw. aktualisiert. Ist dies noch nicht vorhanden, wird es inkl. der Transaktion erstellt und der E-Mail hinzugefügt. Der genaue Aufbau und ein Beispiel des JSON-Objekts wird in Kapitel 6.1.3 aufgezeigt.

6.1.3 Auflösung E-Mail Adresse zu Account-ID der Blockchain

Hierzu kann vom Nutzer bei dem Versand einer E-Mail die Account-ID des Senders in die Header Eigenschaft 'Stamp' hinzugefügt werden. Hierfür wird in dem JSON-Objekt die Zuordnung 'sender:{ID:AccountIDdesSenders}' hinzugefügt. Dies geschieht mit dem Button 'Add StampAccountID'.

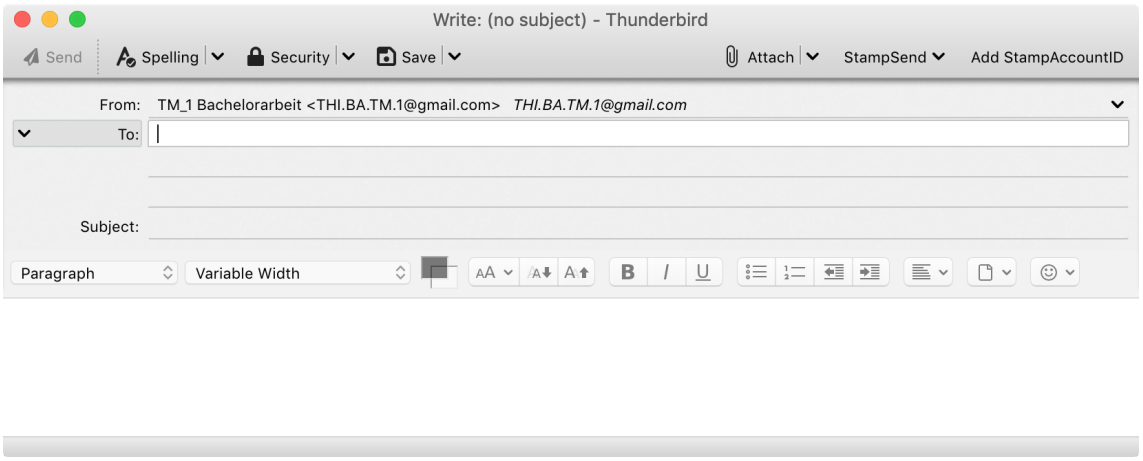

Abbildung 13: Account-ID zum E-Mail Header anfügen

Nach der Betätigung dieses Buttons wird, wie in der 'SendStamp' Methode, überprüft ob der Header 'Stamp' schon existiert. Demnach wird dieser erweitert oder gegebenenfalls neu erstellt.

```
{"sender":
  {"ID":"0xf7463d2e76e2808BB0EE65877Cf0ADA5463AA264"},
"recipients":
  {"0xA465515A64b563a20984020E8bD9A29331AFD668":
    {"transaction":"0xca6f31f92b476c9aeb0f70680f992d4b1ba26495b8b27fcd4a4ab05243be857b"}
  }
}
```

Quellcode 2: JSON-Object in der Headereigenschaft 'Stamp'

In dem Quellcode 2 soll anhand eines Beispiels das JSON-Objekt im Header 'Stamp' veranschaulicht werden. Das Format des JSON-Objekt bringt den Vorteil, das es um beliebige Eigenschaften erweitert werden kann, sollte das Briefmarkensystem für E-Mails weiter entwickelt werden.

Der zweite Teil der Auflösung von E-Mail Adresse zu Account-ID beinhaltet die Auswertung von empfangenen Nachrichten und der Speicherung dieser Adressen. Hierfür wird bei der Sortierfunktion im Zuge der Überprüfung, ob ein 'Stamp' Header existiert, auch geprüft, ob in diesem die Eigenschaft 'sender:{ID:AccountIDdesSenders}' enthalten ist. Sollte dies der Fall sein, wird zu der E-Mail Adresse die zugehörige Account-ID abgespeichert. Diese gespeicherten Zuordnungen können in den Einstellungen verändert werden.

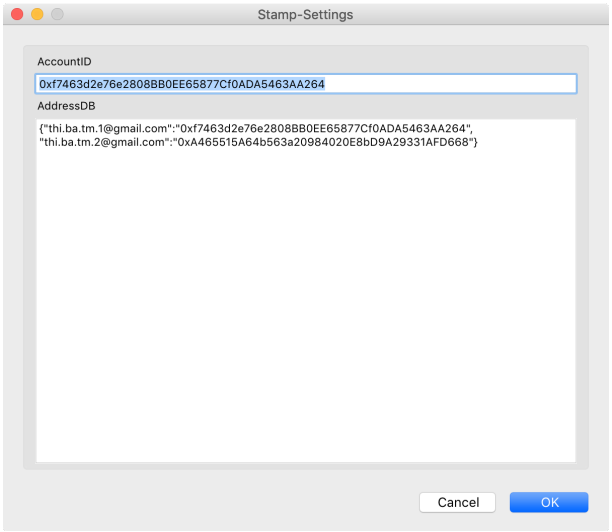

Abbildung 14: Einstellungen für die Thunderbird Extension

Weiter kann über die Eingabe einer E-Mail Adresse diese Zuordnung durchsucht werden. Hierfür muss nur diese in das Kontextmenü von 'StampAddress' eingeben werden. Das Programm durchsucht die gespeicherten Zuordnungen und gibt gegebenenfalls die zugehörige Account-ID zurück. Die Groß- und Kleinschreibung wird bei der E-Mail Adresse nicht berücksichtigt.

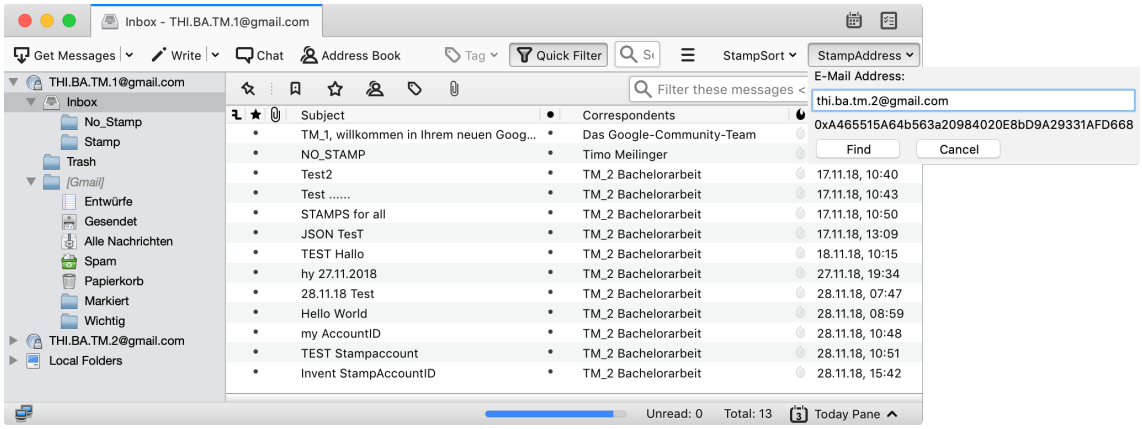

Abbildung 15: Account-ID zu einer E-Mail Adresse finden

6.2 Verwaltungsprogramm für Briefmarken

Dieses Programm übernimmt alle Aufgaben die mit der Verwaltung von Briefmarken zu tun haben. Die Aufgaben hierbei sind der Versand von Briefmarken, die Validierung von Briefmarken bzw. von Transaktions-IDs und die Ermittlung der Anzahl von verfügbaren Briefmarken.

Hierzu sind die Account-ID und der PrivateKey des Senders erforderlich und müssen in dem Fenster 'Stamp Functions' hinterlegt werden. Der private Schlüssel wird nur für den Versand von Briefmarken benötigt.

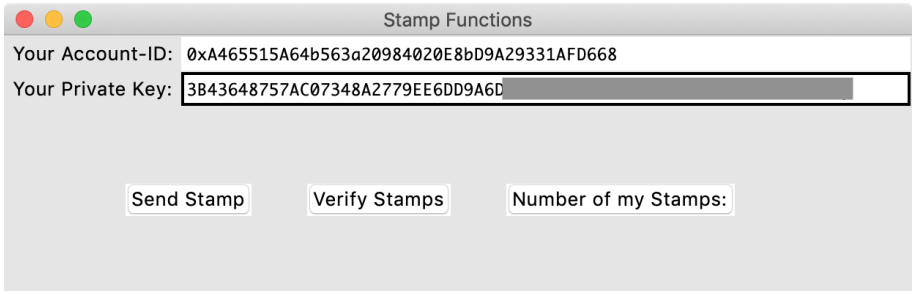

Abbildung 16: Fenster Stamp Functions

6.2.1 Versand von Briefmarken

Für den Versand von Briefmarken müssen die Account-ID und der PrivateKey des Senders hinterlegt werden. Durch klicken des Buttons 'Send Stamp' öffnet sich ein neues Fenster.

In diesem muss nur die Account-ID des Empfängers eingetragen werden und das Senden durch den Button 'Send' bestätigt werden.

Abbildung 17: Fenster Send Stamp

Wird dieser Button betätigt überprüft das Programm die Anzahl der verfügbaren Briefmarken des Senders. Hat dieser eine oder mehrere verfügbar, erstellt das Programm eine neue Transaktion für den hinterlegten ERC20 Token. Diese neue Transaktion wird mithilfe des PrivateKeys signiert und zu dem Ethereum Netzwerk übermittelt. Hierdurch erhält das Programm eine Transaktions-ID. War die Übermittlung erfolgreich, öffnet sich ein Pop-up Fenster in dem die Account-ID des Empfängers und die Transaktions-ID angezeigt werden. Diese müssen wie oben beschrieben nur noch in die Extension von Thunderbird kopiert werden.

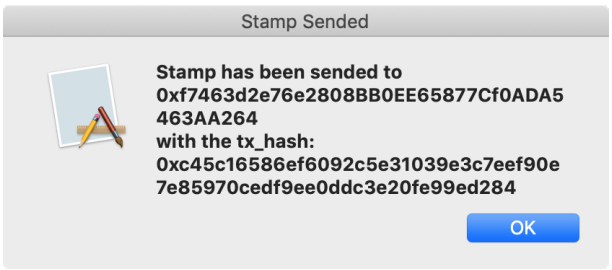

Abbildung 18: Pop-up Fenster Send Stamp

6.2.2 Validierung der Briefmarken

Für die Validierung von Briefmarken bzw. Transaktions-IDs muss die Account-ID des Empfängers hinterlegt sein. Durch klicken des Buttons 'Verify Stamps' öffnet sich ein neues Fenster. In dieses muss das 'Verify Object' von der Thunderbird Extension kopiert werden. Dies ist möglich mit dem Button 'Paste Clipboard', da hierdurch die Zwischenablage in das Textfeld kopiert wird. Durch den Button 'Verify' wird die Überprüfung der einzelnen Transaktionen gestartet.

Abbildung 19: Fenster Verify Stamps

Das Programm analysiert dann das eingefügte JSON-Objekt und fragt an der Blockchain zu jeder Transaktions-ID die Details an. Falls der Blockchain die Transaktions-ID nicht bekannt ist, wird diese für ungültig erklärt. Ansonsten wird in den Details überprüft, ob die Transaktion eine Übermittlung des korrekten ERC20 Token darstellt. Weiter wird überprüft, ob die Account-ID der Transaktion mit der des Empfängers übereinstimmt und so der Token auch wirklich an den Empfänger der E-Mail transferiert wurde. Stimmen diese beiden Details, wird die Transaktion für valide erklärt. Dies wird für alle Transaktionen in dem JSON-Objekt wiederholt. Nach Abschluss der Überprüfung aller Transaktionen wird ein JSON-Objekt mit allen Ergebnissen in die Zwischenablage kopiert und der Nutzer durch ein Pop-up Fenster aufgefordert, dieses Ergebnis in die Thunderbird Extension zurück zu kopieren. Hier werden die E-Mails anhand der Ergebnisse in die entsprechenden Ordner sortiert.

Abbildung 20: Pop-up Fenster Verify Stamps

6.2.3 Ermittlung der Anzahl verfügbarer Briefmarken

Für die Ermittlung der Anzahl verfügbarer Briefmarken muss die Account-ID eingetragen werden. Durch die Betätigung des Buttons 'Number of my Stamps' erscheint rechts neben dem Button die Anzahl der verfügbaren Briefmarken für die eingetragene Account-ID.

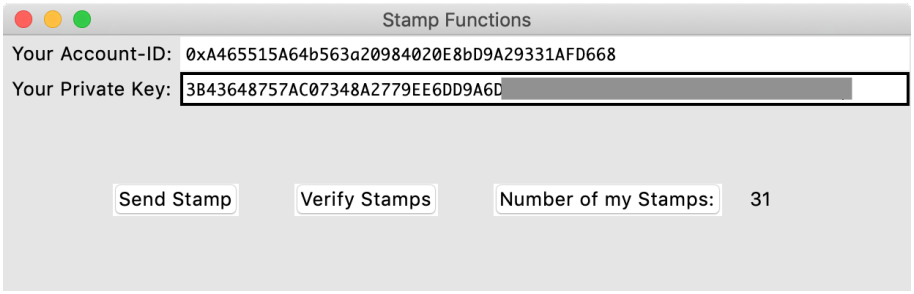

Abbildung 21: Anzahl an verfügbaren Briefmarken bestimmen

Das Programm führt die Funktion 'balanceOf(Account-ID)' des hinterlegten ERC20 Tokens aus. Dieser Aufruf liefert die Anzahl an verfügbaren Briefmarken, die neben dem Button eingeblendet wird.

6.3 ERC20 Token im Ethereum Netzwerk

Als Vorlage für den ERC20 Token Contract wurde eine OpenSource Vorlage aus Github verwendet und abgeändert.[7] Dies beinhaltet Änderungen zu Namen und Anzahl an Coins, die erstellt werden sollen. In das Ethereum Netzwerk deployed wurde dieser Vertrag mithilfe von der Remix-Solidity-IDE.[8] Hierdurch kann der Vertrag über eine Website ohne lokale Installationen compiliert und deployed werden.

Der ERC20 Token wurde 'TestStamp' genannt und erhielt die Abkürzung 'TSTAMP'. Dieser Name wurde gewählt, da der Vertrag in das Testnetz Rinkeby und nicht in das Main Ethereum Netzwerk deployed wurde. Hierdurch ergeben sich keinerlei Änderungen in der Implementierung des Tokens oder des Verwaltungsprogramms für Briefmarken. Weiter wurden 1.000.000.000.000 (1 Billionen) Einheiten dieses Tokens erstellt. Diese

[7] https://github.com/ConsenSys/Tokens/tree/master/contracts/eip20 (Aufgerufen am 30.11.2018)
[8] https://remix.ethereum.org/ (Aufgerufen am 30.11.2018)

Anzahl wurde beabsichtigt so groß gewählt, damit gezeigt werden kann, dass dieses System auch mit einer großen Anzahl an Verwendern genutzt werden kann. Erzeugt wurden diese Einheiten ohne Dezimalstellen, da eine Briefmarke nicht weiter unterteilt werden soll.

```
pragma solidity ^0.4.21;

import "./EIP20Interface.sol";

contract TestStamp is EIP20Interface {
  uint256 constant private MAX_UINT256 = 2**256 - 1;
  mapping (address => uint256) public balances;
  mapping (address => mapping (address => uint256)) public allowed;

  string public name;                   //Name of the Token
  uint8 public decimals;                //How many decimals to show.
  string public symbol;                 //Identifier of the Token

  function TestStamp() public {
    balances[msg.sender] = 1000000000000;   // Give the creator all initial tokens
    totalSupply = 1000000000000;            // Update total supply
    name = "TestStamp";                   // Set the name for display purposes
    decimals = 0;                         // Amount of decimals for display purposes
    symbol = "TSTAMP";                    // Set the symbol for display purposes
  }

  function transfer(address _to, uint256 _value) public returns (bool success) {
    require(balances[msg.sender] >= _value);
    balances[msg.sender] -= _value;
    balances[_to] += _value;
    emit Transfer(msg.sender, _to, _value);
    return true;
  }

  function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
    uint256 allowance = allowed[_from][msg.sender];
    require(balances[_from] >= _value && allowance >= _value);
    balances[_to] += _value;
    balances[_from] -= _value;
    if (allowance < MAX_UINT256) {
      allowed[_from][msg.sender] -= _value;
    }
    emit Transfer(_from, _to, _value);
    return true;
  }

  function balanceOf(address _owner) public view returns (uint256 balance) {
    return balances[_owner];
  }

  function approve(address _spender, uint256 _value) public returns (bool success) {
    allowed[msg.sender][_spender] = _value;
    emit Approval(msg.sender, _spender, _value);
    return true;
  }

  function allowance(address _owner, address _spender) public view returns (uint256 remaining) {
    return allowed[_owner][_spender];
  }
}
```

Quellcode 3: Code des ERC20 Tokens TestStamp

7 Evaluierung des Nutzens eines Briefmarkensystems für E-Mails

Die Implementierung und der daraus folgende Prototyp beweisen, dass eine Umsetzung des Konzeptes von wiederverwendbaren Briefmarken für das E-Mail System möglich ist. Auch wird die Aufmerksamkeit des Empfängers durch das Senden der Briefmarken bepreist, ohne die durchschnittliche und normale Verwendung von E-Mails einzuschränken. Da bei dieser Nutzung ein Gleichgewicht zwischen senden und empfangen herrscht, gibt es für diese Verwender keinerlei Einschnitte.

Alle Anforderungen an das Briefmarkensystem aus Abschnitt 5.2 wurden mit dem ausgearbeiteten Konzept eingehalten. Das System ist sowohl auf Ebene des MTA als auch MUA implementierbar. Weiter erfolge die Verwaltung der Briefmarken in einer Blockchain. Diese stellt, wie gefordert, die Integrität der Währung sicher. Zusätzlich sind alle Funktionen des E-Mail Systems erhalten geblieben. Es können auch E-Mails mit mehreren Empfängern versendet werden und die CC und BCC Funktion sind weiter nutzbar. Unter dem Vorbehalt, der Sender besitzt genügend Briefmarken. Daraus ergibt sich, dass alle Anforderungen im ausgearbeiteten Konzept eingehalten wurden.

Weiter gibt es zu beurteilen ob es durch dieses Konzept möglich ist, Spam zu verhindern bzw. zu reduzieren. Hierfür muss betrachtet werden, dass Spam-Sender mehrere Tausende E-Mails innerhalb weniger Stunden senden. Innerhalb dieses Konzeptes ist es nicht möglich, eine große Anzahl an Nachrichten zu versenden, beziehungsweise diese E-Mails werden beim Empfänger gesondert dargestellt. Den Spam-Sender kann es in diesem Konzept nicht möglich sein, große Mengen von Briefmarken zu besitzen, weil diese hierfür auch große Mengen von Nachrichten empfangen oder diese Briefmarken käuflich erwerben

müssten. Jedoch erhalten Spam-Sender keine oder nur sehr wenige Nachrichten und dadurch auch nur eine sehr geringe Anzahl an Briefmarken. Hieraus kann gefolgert werden, durch empfangene Nachrichten erhalten diese viel weniger Briefmarken als benötigt. Ein käuflicher Erwerb von Briefmarken ist für die Spam-Sender auch keine Lösung, da diese große Summen investieren müssten, um tausende E-Mails täglich versenden zu können. Hierfür sind die Erträge durch Spam Nachrichten zu gering. Somit ist es den Spam-Sendern keinesfalls möglich, genügend Briefmarken zu besitzen, was die Schlussfolgerung zu lässt, dass die unerwünschten Nachrichten gesondert dargestellt werden und bei den Empfängern weniger Beachtung finden. Aufgrund dessen werden Spam Nachrichten weniger lukrativ und die Anzahl an Spam-Sendern wird kleiner, wodurch auch die Anzahl an Spam E-Mails geringer wird.

Daraus folgt, dass nicht nur die Aufmerksamkeit der Empfänger durch dieses System bepreist wird, sondern auch eine Reduzierung von Spam zur Folge hat. Dies ist als großer Vorteil zu sehen.

Der Mehraufwand, der für den Sender und dem Empfänger durch das System entsteht, ist überschaubar. Dieser Mehraufwand kann in der weiteren Entwicklung dieses Systems noch verkleinert werden. Dies ist beispielsweise durch eine Implementierung des Systems auf Ebene der MTAs möglich. Der Mehraufwand durch dieses System kann somit nur als ein geringer Nachteil bezeichnet werden.

Weiter muss analysiert werden, wie groß die Verzögerung beim Senden einer E-Mail, durch die Transaktionsausführung in der Blockchain, ist. Hierbei muss unterschieden werden, ob bei dem Versand die Ausführung der Transaktion in der Blockchain abgewartet werden soll oder ob die E-Mail versendet wird, auch wenn die Transaktion noch nicht abgeschlossen ist. Die Zeit bis zum Abschluss einer Transaktion ist abhängig von dem sogenannten bezahlten 'Gas'. Hierbei wird zwischen drei Preisen unterschieden. Dem günstigsten Preis, wobei die Ausführung weniger als 30 Minuten in Anspruch nimmt. Weiter gibt es die mittlere oder Standard Preisklasse, für eine Transaktion mit dieser Höhe von bezahlten 'Gas' wird weniger als fünf Minuten benötigt. Für die dritte Preisklasse die schnellste und teuerste wird für eine Ausführung einer Transaktion, weniger als zwei Minuten in Anspruch

genommen.[9] Es ist somit festzuhalten, dass die Verzögerung beim Senden einer E-Mail mit der Höhe des bezahlten 'Gas' vom Sender bestimmt wird. Die Verzögerung kann auf unter fünf Minuten festgeschrieben werden. Dies ist möglich, wenn der Standard 'Gas' Preis bezahlt wird. Somit ist die Verzögerung für die E-Mails beschränkbar und einstellbar, durch den Sender und weiter handelt es sich nur um wenige Minuten. Diese Verzögerung bei dem Versand einer E-Mail ist vertretbar und stellt keine große Einschränkung dar. Ebenso ist denkbar, dass die E-Mail versendet wird, auch wenn die Transaktion noch nicht abgeschlossen ist. In diesem Fall benötigt die Nachricht auch Zeit, bis sie übermittelt wurde. Hierdurch wird die Wartezeit auf den Empfänger übertragen. Dieser wartet bei der Validierung der Briefmarken, bis die Transaktion abgeschlossen ist.

Zusammenfassend kann festgehalten werden, dass die zusätzliche Zeit, die für den Versand von Briefmarken benötigt wird einen vertretbaren Einsatz darstellt, da nur wenige Minuten für eine E-Mail zusätzlich benötigt werden. Weiter ist es für den Sender möglich, durch die Zahlung eines höheren 'Gas' Preises, diese Zeit zu verringern. Der Sender bestimmt, wie schnell die E-Mail versendet wird.

Der Sender muss, wie beschrieben, für den Versand einer Briefmarke sogenanntes 'Gas' bezahlen. Hierbei ist einerseits der Gaspreis und andererseits die Anzahl an verwendeten Einheiten zu beachten. Bei einem durchschnittlichen Gaspreis und Gasverbrauch werden ca. 0,1 Cent für eine Transaktion fällig.[10] Die hierdurch entstehenden zusätzlichen Kosten für jede Transaktion sind vertretbar.

Abschließend ist festzustellen, dass die Vorteile dieses Systems, die Nachteile bei weitem überschatten. Der Mehraufwand der durch das System entsteht, ist überschaubar und unterliegt dem Nutzen des Briefmarkensystems für E-Mails zur Bepreisung der Aufmerksamkeit des Empfängers. Somit ist das Konzept eine sinnvolle Erweiterung des E-Mail Systems. Dieses sollte weiter umgesetzt und verbreitet werden, da viele Nutzer davon profitieren können.

[9] Der aktuelle 'Gas' Preis für die jeweilige Preiskategorie ist unter https://ethgasstation.info/ einsehbar. (Aufgerufen am 28.12.2018)

[10] Gerechnet mit Gasverbrauch von 3000 Einheiten bei einem Gaspreis von 3.1 Gwei und einem Umrechnungskurs von 1 ETH entspricht 132.33€. (Stand 04.01.2019) Berechnung: $3000 * 3,1 Gwei = 9300 Gwei = 0,0000093 ETH \approx 0,0012 EUR$

8 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Konzept für ein Briefmarkensystem für E-Mails ausgearbeitet. Dieses bepreist erfolgreich die Aufmerksamkeit des Empfängers. Die Kombination aus der Blockchain und der Erweiterung des E-Mail Headers stellen eine sinnvolle und funktionsfähige Implementierungsmöglichkeit für dieses Konzept dar. Somit kann festgehalten werden, dass eine Erweiterung der E-Mail, um ein Briefmarkensystem sinnvoll und wünschenswert ist. In dieser Arbeit sollte das zugehörige Konzept mit einer Prototyp-Implementierung ausgearbeitet werden. Dies ist auch erfolgreich gelungen. In Zukunft sollte dieses Konzept in ein produktives System ausgearbeitet werden. Hierfür gibt es noch viele weitere sinnvolle Funktionen, die in dieses Konzept eingebaut werden können.

So könnten durch eine Whitelistingfunktion auch Nachrichten ohne Briefmarken priorisiert werden, wenn der Empfänger Interesse an diesen hat und diese in dem Ordner 'Stamp' auffinden möchte. Ein Beispiel für den sinnvollen Einsatz sind vom Empfänger gewollte Newsletter E-Mails. Da es den Newsletter-Sendern in diesem Konzept nicht möglich ist, genügend Briefmarken zu besitzen, können so gewollte Newsletter erfolgreich in dieses Konzept eingearbeitet werden.

Weiter ergibt sich die Möglichkeit dieses Konzept auf der Ebene des MTA zu implementieren. Hierdurch geht der Mehraufwand für die Nutzer gegen Null und für die Freemail Anbieter ergibt sich ein neues Geschäftsmodell. Denn wenn Nachrichten von den Freemail Anbieter zu diesem versendet werden, wird keine Briefmarke benötigt. Die überflüssigen Briefmarken kann dieser verkaufen. Somit ergibt sich eine neue Einnahmequelle für die Freemail Anbieter und die Verwendung wird für die einzelnen Nutzer leichter.

Ein weiterer Punkt, der weiterentwickelt werden kann, ist die Umsetzung des in Abschnitt 5.6 beschriebenen Konzeptes zur Validierung der Zuordnung von E-Mail Adresse zu

Account-ID. Durch dieses Verfahren ist es möglich, den Empfänger der Briefmarken zu verifizieren, damit dieser mit dem Empfänger der E-Mail übereinstimmt. Dies ist notwendig, damit kein Angreifer die Account-IDs in den E-Mail Headern manipuliert und dadurch Briefmarken erhält, die diesem nicht zustehen.

Da die Account-ID der Hash Wert des öffentlichen Schlüssels eines Individuums ist, wäre auch eine Verschlüsselung der E-Mail mit dem öffentlichen Schlüssel der Blockchain eines Accounts denkbar. Hierdurch wird es möglich, dass E-Mails verschlüsselt werden können ohne das ein Austausch eines weiteren Schlüssels benötigt wird. Auch ist denkbar, die E-Mail bei dem Versenden mit dem privaten Schlüssel der Blockchain zu signieren. Hierdurch kann der Empfänger mit dem öffentlichen Schlüssel des Senders die Signatur prüfen und so mögliche Manipulationen an der E-Mail erkennen. Dies zeigt auf, dass aufgrund der vorhanden Schlüssel der Blockchain sich weitere interessante Möglichkeiten ergeben, die den Versand von E-Mails sicherer gestalten.

Aus diesen Punkten ergibt sich, dass in dieser Arbeit vorgestellte Konzept noch Potenzial für Erweiterungen bietet. Das Grundkonzept wurde ausgearbeitet und durch die Prototyp Implementierung umgesetzt. Das Konzept eines Briefmarkensystems für E-Mails zur Bepreisung der Aufmerksamkeit des Empfängers ist hierdurch abgeschlossen und sollte in Zukunft weiter ausgebaut sowie in einem produktiven System umgesetzt werden.

Literaturverzeichnis

[1] Adam Back. *Hashcash - Amortizable Publicly Auditable Cost-Functions*. Website. Aug. 2002. URL: http://www.hashcash.org/papers/amortizable.pdf (aufgerufen am 05. 12. 2018).

[2] Adam Back. *hashcash.org*. Website. URL: http://www.hashcash.org/ (aufgerufen am 05. 12. 2018).

[3] Roland Bless, Michael Conrad und Hans-Joachim Hof. *Spam Protection by using Sender Address Verification Extension (SAVE)*. Website. Aug. 2005. URL: http://telematics.tm.kit.edu/publications/Files/112/SAVE.pdf (aufgerufen am 06. 12. 2018).

[4] BSI - Bundesamt für Sicherheit in der Informationstechnik. *Spam E-Mails*. Website. URL: https://bsi.bund.de/DE/Service/FAQ/SpamEMails/faq_node.html (aufgerufen am 29. 12. 2018).

[5] Vikram Dhillon, David Metcalf und Max Hooper. *Blockchain Enabled Applications*. Apress, 2017. ISBN: 978-1-4842-3081-7. DOI: 10.1007/978-1-4842-3081-7.

[6] Daniel Drescher. *Blockchain Basics*. Apress, 2017. ISBN: 978-1-4842-2604-9. DOI: 10.1007/978-1-4842-2604-9.

[7] Dudenredaktion (o. J.) *'Spam' auf Duden online*. Website. URL: https://www.duden.de/node/695539/revisions/1771479/view (aufgerufen am 07. 12. 2018).

[8] GDV - Gesamtverband der Deutschen Versicherungswirtschaft e. V.
Cyberrisiken im Mittelstand - Ergebnisse einer Forsa-Befragung im Frühjahr 2018.
Website. URL: https://www.gdv.de/resource/blob/32708/d3d1509dbb080d899fbfb7162ae4f9f6/cyberrisiken-im-mittelstand-pdf-data.pdf (aufgerufen am 29. 11. 2018).

[9] Ralf Hund, Matthias Hamann und Thorsten Holz.
„Towards Next-Generation Botnets“.
In: *2008 European Conference on Computer Network Defense*. IEEE, Dez. 2008.
DOI: 10.1109/ec2nd.2008.11.

[10] Kaspersky Lab. *Spam and phishing in Q3 2018*. Website. 2018.
URL: https://securelist.com/spam-and-phishing-in-q3-2018/88686/
(aufgerufen am 15. 12. 2018).

[11] Franz-Joachim Kauffels. *Grundlagen der Netzwerktechnik*. 6. Auflage.
Stuttgart: UTB, 2007. ISBN: 978-3-8252-8377-3.

[12] Ralf T. Kreutzer. *E-Mail-Marketing kompakt*.
Springer Fachmedien Wiesbaden, 2018. ISBN: 978-3-658-20990-2.
DOI: 10.1007/978-3-658-20990-2.

[13] Mozilla Developer Network. *Cross-Origin Resource Sharing (CORS)*. Website.
URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
(aufgerufen am 29. 11. 2018).

[14] Mozilla Developer Network. *Same-origin policy - Web security*. Website.
URL: https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy (aufgerufen am 02. 12. 2018).

[15] SevenOneMedia. *Media Activity Guide 2018*. Website. 2018.
URL: https://www.sevenonemedia.de/documents/924471/1111769/Media+Activity+Guide+2018/0d7f33af-210a-682c-6c21-9aad7ae863f8
(aufgerufen am 07. 12. 2018).

[16] WEB.DE. *E-Mail-Volumen in Deutschland auf Rekordwert – 771 Milliarden E-Mails verschickt*. Website. URL: `https://newsroom.web.de/2018/02/06/e-mail-volumen-in-deutschland-2017-auf-rekordwert-771-milliarden-e-mails-verschickt/` (aufgerufen am 30. 12. 2018).

[17] Harald Zisler. *Computer-Netzwerke. Grundlagen, Funktionsweise, Anwendung*. 4., aktualisierte und erweiterte Auflage. Rheinwerk Computing. Bonn: Rheinwerk, 2016. ISBN: 978-3-8362-4322-3.

Ehrenwörtliche Erklärung

Ich erkläre hiermit, dass ich die Arbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Ingolstadt, den 11. Januar 2019

Timo Meilinger