Verfasst von: Juri Kembügler

Betreuende Lehrkraft: Dr. Sandra Wortmann

Ethereum Chat-App

Eine besondere Lernleistung zum Thema „Decentralised Networks and Decentralised Finance“ im Fach Informatik am Städtischen Gymnasium Sundern

8. Juni 2020 bis 22. April 2021

Inhalt

Abkürzungsverzeichnis

API:	Application Programming Interface
CeFi:	Centralised Finance
DAO:	Decentralised Autonomous Organisation
dApps:	Decentralised Applications
DeFi:	Decentralised Finance
EVM:	Ethereum Virtual Machine
GWEI:	Giga-Wei
IDSA:	International Data Spaces Association
IPFS:	Inter Planetary File System
NFT:	Non Fungible Token
SEPA:	Single Euro Payments Area
USDC:	US-Dollar Coin
USDT:	Tether US-Dollar
WBTC:	Wrapped Bitcoin
XAUT:	Tether Gold

1 Einleitung

Die Welt hat sich in den letzten Jahrzenten zu einer digitalen Dienstleistungsgesellschaft entwickelt, in der nahezu alles online erledigt wird. Auch Zahlungen werden zum Großteil im Internet ausgeführt. Dies geschieht nicht nur im Online-Handel, sondern auch bei Bezahlungen im Einzelhandel mit der Kredit- oder Girokarte. Vielerorts ist es üblich, mit dem Smartphone oder der Smartwatch zu bezahlen.

Um eine Zahlung auszuführen, werden Dienstleister einer dritten Partei benötigt. Diese Dienstleister können Banken oder Online-Bezahldienste wie zum Beispiel *PayPal* und *Sofortüberweisung* sein. Im Falle einer Bezahlung per Smartphone oder Smartwatch stehen große Konzerne wie *Google* oder *Apple* als Vermittler dazwischen. Die Verantwortung über die Transaktion von einer Partei zu einer anderen wird also an eine dritte zentrale Partei abgetreten, die zur Ausführung dieser Transaktion alle dafür erforderlichen bzw. verfügbaren Informationen erhält. Diese Informationen enthalten nicht nur Daten über die Konten, zwischen denen das Geld transferiert werden soll, sondern auch der Personen, denen die Konten zugeordnet werden können, und in einigen Fällen darüber, zu welchem Zweck die Transaktion stattfinden soll. So weiß die dritte Partei bei einem Bezahlvorgang, wer der Inhaber der Karte oder des Endgerätes ist, in welchem Supermarkt er wieviel Geld bezahlt, und in einigen Fällen auch, was er gekauft hat.

Verbraucher müssen nicht nur darauf vertrauen, dass Dienstleister die Transaktionen korrekt ausführen, sondern es muss ebenfalls in Betracht gezogen werden, dass Zahlungsdaten weitergegeben werden, etwa an Werbetreibende oder zur Bonitätsprüfung. Eine völlig anonyme Art Geld zu transferieren, wie es mit analogem Geld, also Bargeld, der Fall ist, lässt sich mit unserem heutigen Finanzsystem bei immateriellen Transaktionen nicht gewährleisten.

Einen Lösungsansatz bieten dezentrale Kryptowährungen. Dabei erlauben dezentrale Infrastrukturen Transaktionen ohne dritte Partei und bieten Anonymität.

Grundlage der vorliegenden Arbeit ist die Entwicklung einer Anwendung, die es ermöglicht, mit anderen Personen in Kontakt zu treten, sich zu vernetzen und zu kommunizieren. Die verknüpften Personen können dann über die Anwendung untereinander anonyme Zahlungen basierend auf einer Kryptowährung ausführen.

2 Ausgangslage

2.1 Kryptowährungen

2.1.1 Entstehung von Kryptowährungen

Wie in [1] ausgeführt wird, lässt sich eine Kryptowährung als eine durch Kryptographie gesicherte digitale Währung bezeichnen. Dabei soll diese eine unabhängige, dezentrale und sichere Alternative zu unserem gegenwärtigen Zahlungssystem darstellen.

Die erste Kryptowährung eCash wurde 1983 von dem amerikanischen Kryptographen David Chaum entwickelt. Wie er in [2] darstellt, sollte es mit eCash möglich sein, Transaktionen zwischen zwei Parteien digital ausführen zu können. Dafür sollte die neuartige kryptografische Technik „Blinde Signaturen“ zum Einsatz kommen. Die Währung konnte sich nicht durchsetzen; das dahinter stehende Unternehmen *DigiCash* musste 1998 Insolvenz anmelden.

Die erste Kryptowährung, die die Anerkennung der breiten Masse erreichte, war 2009 Bitcoin. Sie wurde während der Finanzkrise 2008 von dem Pseudonym Satoshi Nakamoto entwickelt und stellt zusammen mit der Blockchain-Technologie eine grundlegende Neuerung in der Entwicklung von Kryptowährungen dar. Basierend auf Bitcoin haben sich seit 2008 viele neue Kryptowährungen entwickelt. Nach [3] sind mittlerweile mehr als 4.500 Kryptowährungen auf CoinMarketCap gelistet.

Unter den vielen digitalen Währungen ist auch die Kryptowährung Ether. Sie ist die Währung auf der dezentralisierten Plattform Ethereum.

2.1.2 Funktionsweise von Transaktionen in einer Blockchain

Wie in [1] beschrieben, ist eine Blockchain als ein dezentrales, kryptographisch gesichertes „Verzeichnis“ aller im Netzwerk getätigten Transaktionen zu verstehen. Dieses Verzeichnis nennt man „Ledger“. Das Besondere ist, dass die in der Blockchain gespeicherten Transaktionen öffentlich von jedem einsehbar sind, diese aber keiner Person zuzuordnen sind. Somit ist zum Beispiel das Bitcoin-Netzwerk anonym.

Damit es einem Nutzer möglich ist, eine Bitcoin-Transaktion auszuführen, benötigt er ein kryptografisch generiertes Schlüsselpaar. Dieses besteht aus dem „Public Key” und dem „Private Key”. Beide Schlüssel bestehen aus einer Reihenfolge von Zahlen und

Buchstaben. Der Public Key ist öffentlich und lässt sich von jedem einsehen, während der Private Key privat ist und nur dem Eigentümer bekannt ist. Aus dem Private Key lassen sich mehrere dazugehörige Public Keys erzeugen, so dass ein Konto mehrere verschiedene Adressen besitzen kann.

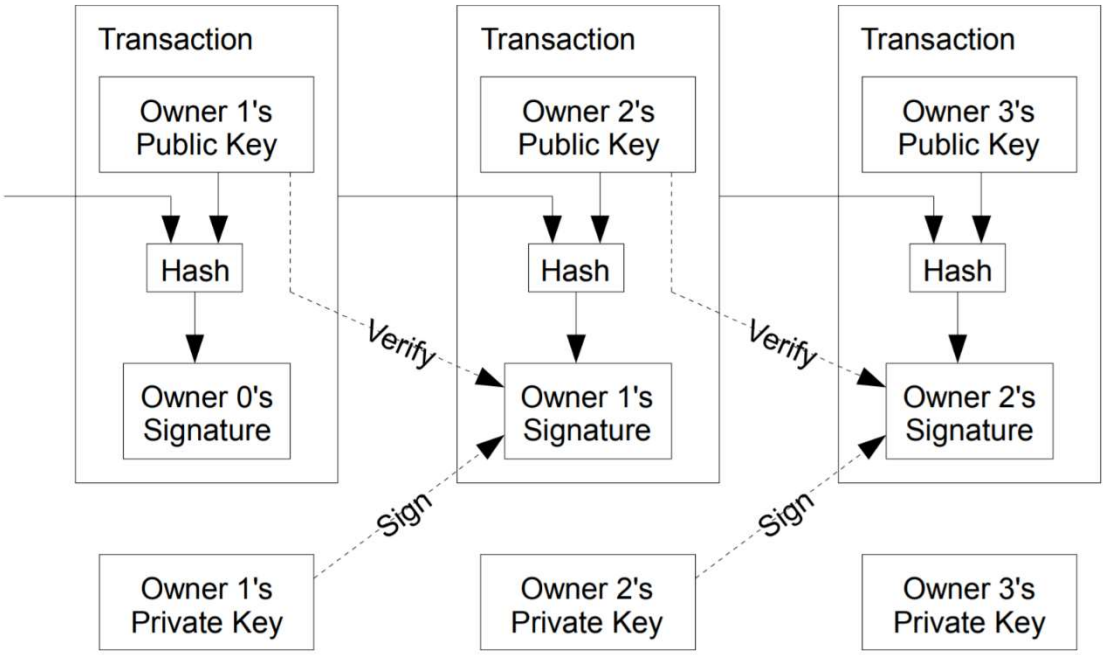

Abbildung 1: Transaktionen im Bitcoin Netzwerk [4]

Der Public Key bildet die Adresse eines Bitcoin-Kontos, die sogenannte „Wallet", an die man Bitcoins senden kann. Mit dem Private Key und den Transaktionsdaten lässt sich dann eine kryptographische Signatur erstellen. Aus den Transaktionsdaten, der Signatur und dem Public Key lässt sich dann verifizieren, ob es sich tatsächlich um eine zum Private Key passende Signatur handelt. So kann eine Transaktion signiert und verifiziert werden, ohne dass der Private Key öffentlich bekannt sein muss, denn es wird nur die Signatur an das Netzwerk geschickt. Jede Transaktion erhält einen Hashwert (vgl. *Abbildung 1*). Ein Hashwert bildet dabei einen einzigartigen Wert, der mit der Hashfunktion „SHA-256" gebildet wird. Eine kryptografische Hashfunktion liefert bei gleicher Eingabe immer denselben einzigartigen Wert. Aus dem Wert allein lässt sich jedoch nicht die Eingabe wiederherstellen [5]. Ein kryptografischer Hashwert lässt sich also als Fingerabdruck von Daten verstehen. Somit ist jede Signatur einzigartig, obwohl die gleiche Transaktion ausgeführt wird. Die Signatur lässt sich mit physischen Unterschriften vergleichen. Allerdings sind die Signaturen im Bitcoin-Netzwerk einzigartiger und sicherer. [4]

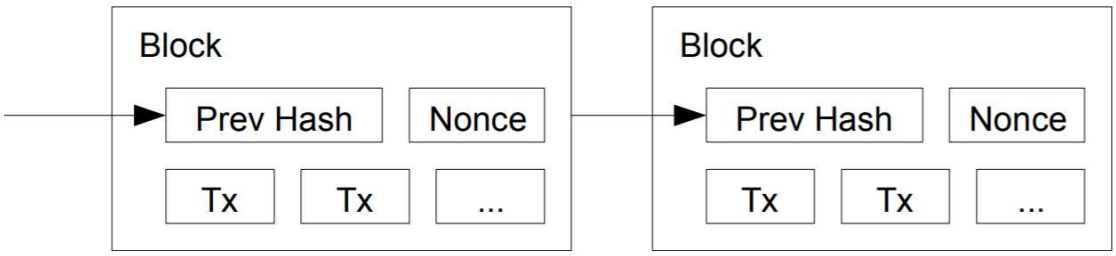

Abbildung 2: Die Bitcoin-Blockchain [4]

Ist eine Transaktion mit dem Private Key signiert und mit einem dazugehörigen Public Key verifiziert, werden die Transaktionen in einem Block gespeichert. Dieser Block wird an die Blockchain angehängt. Dabei stellt die Blockchain eine einfach verkettete Liste dar, in der jeder Block auf den nachfolgenden Block zeigt, in dem der Nachfolger den Hash des Vorgängers enthält (vgl. *Abbildung 2*).

Damit ein Block im Netzwerk verifiziert werden kann, muss eine bestimmte Nummer gefunden werden, so dass der Hash des Blockes mit einer bestimmten Anzahl an Nullen beginnt. Diese Nummer wird „Nonce“ genannt und ebenfalls im Block gespeichert. Um die Nonce zu finden, wird jede mögliche Zahlenkombination ausprobiert, bis der Hashwert mit einer bestimmten Anzahl Nullen beginnt. Derjenige, der die Nonce findet, erhält den “Blockreward” und die Transaktionsgebühren, die der Nutzer angeben kann, damit seine Transaktion bevorzugt ausgeführt wird. Damit die Menge der Bitcoins begrenzt bei 21 Millionen liegt, wird der Blockreward alle 210.000 Blöcke halbiert. Dieser Vorgang wird als „Halving“ bezeichnet. Satoshi Nakamoto bezeichnet das Finden einer Nonce als Proof-of-Work [4], da durch diesen Prozess mit physischer Rechenleistung ein Block geprüft wird. Dadurch, dass hier neue Bitcoins erschaffen werden, wird dieser Vorgang „Mining“ genannt. Dieser soll die Sicherheit im Netzwerk garantieren. Die Anzahl der Nullen wird dann durch die sogenannte „Difficulty“ angegeben, die sich aus der Rechenleistung im Netzwerk ergibt. Denn desto höher die Difficulty, desto mehr Nullen müssen am Anfang des Hashwertes stehen, und mit jeder weiteren Null wächst die benötigte Arbeit exponentiell, um die Nonce zu finden. Um einen Anreiz zu schaffen, die Nonce eines Blockes zu finden, erhält derjenige, der die Nonce findet, eine Belohnung in Form von Bitcoin. Die Difficulty wird im Bitcoin-Netzwerk so angepasst, dass es bei einer entsprechenden Rechenleistung

ca. 10 min dauert, bis die Nonce gefunden wird und so der Block und die darin enthaltenen Transaktionen verifiziert werden. [4]

Die einzelnen Transaktionen werden in einem so genannten „Merkle Tree“ gespeichert. Ein Merkle Tree lässt sich wie ein Binärbaum verstehen, nur dass beim Merkle Tree aus den Blättern des Baumes die Wurzel erzeugt wird (vgl. *Abbildung 3*). Dazu werden in den Blättern die Hashwerte der Transaktionen gespeichert. Aus zwei Blättern wird der darüber liegende Knoten gebildet. Er enthält einen Hashwert, der aus den beiden Blättern gebildet wird. Am Ende bleibt nur noch die Wurzel des Baumes übrig. Dieser Hashwert wird auch „Root-Hash“ genannt und im Block gespeichert.

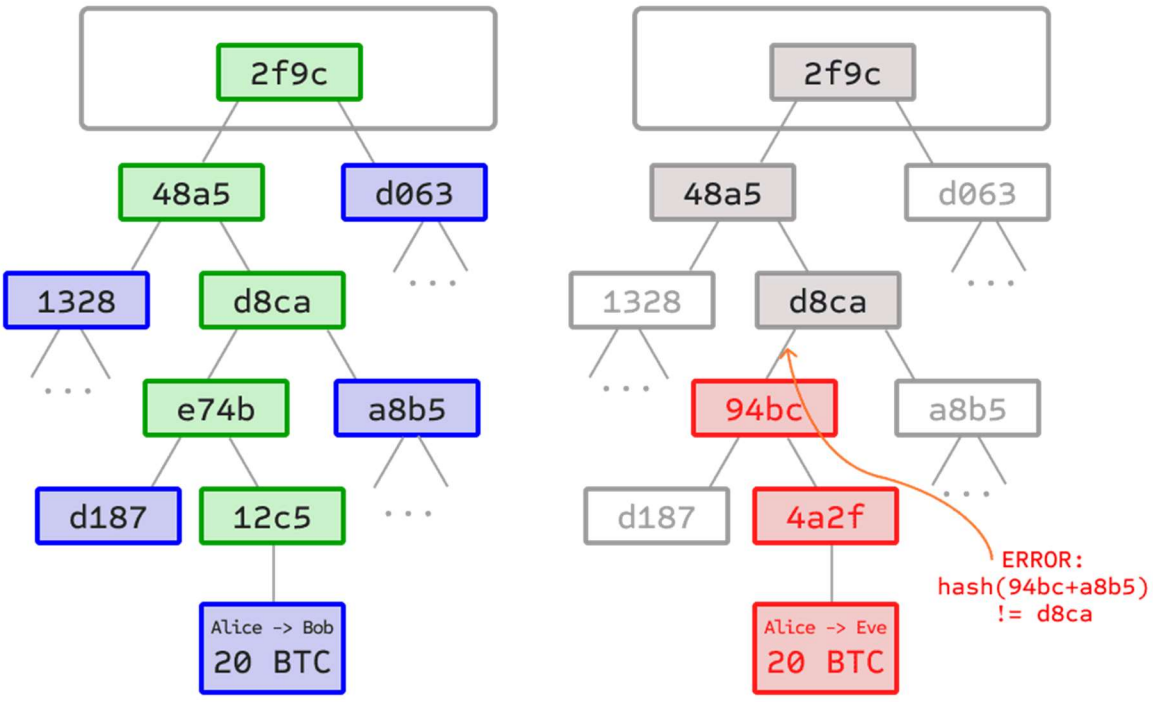

Abbildung 3: Merkle Tree in Bitcoin [6]

Dadurch, dass ein Hashwert einzigartig ist und sich bei nur kleinen Änderung deutlich vom Vorgänger unterscheidet, spiegelt sich die Veränderung einer Transaktion direkt im Root-Hash wider. Durch diese Veränderung wird der Hashwert eines ganzen Blockes geändert und damit der Hashwert aller folgenden Blöcke. Somit ist die Nonce der folgenden Blöcke nicht mehr gültig, und es müsste für alle folgenden Blöcke eine neue Nonce gefunden werden. Es ist so unmöglich, nachträglich Transaktionen in der Blockchain zu verändern. Laut Buterin [6] ergibt diese Art der Datenstruktur der Blockchain eine große Skalierbarkeit, da immer nur der Hashwert gespeichert wird und

somit viele Transaktionen sicher und mit geringem Aufwand gespeichert werden können. [4]

Damit jedoch ein Konsens zu einer Währung entsteht, gibt es die sogenannten „Nodes“. Sie speichern die gesamte Blockchain und suchen nach neuen Transaktionen, um diese in einen neuen Block zu speichern. Die Node prüft, ob die Transaktionen valide sind, und versucht, für den Block die passende Nonce zu finden. Ist die Nonce gefunden, wird der Block an die Blockchain angehängt und an alle anderen Nodes im Netzwerk kommuniziert. Diese prüfen dann, ob die Nonce stimmt, also ob passend zur Difficulty genug Nullen am Anfang des Hashwertes stehen. Die Nodes suchen gleichzeitig nach vielen anderen Blockchains im Netzwerk. Akzeptiert wird nur die Blockchain mit den meisten Blöcken. Ein hinzugefügter Block wird erst dann akzeptiert, wenn dieser auch nach längerer Zeit noch in der Blockchain vorkommt, also von besonders vielen Nodes akzeptiert wurde. Dies macht die Blockchain besonders sicher und dezentral. Denn um eine Transaktion zu fälschen, müsste man mindestens 51 % der Rechenleistung im Netzwerk haben, um über einen längeren Zeitraum die weiteren Blöcke schneller mit einer Nonce zu validieren, als es der Rest des Netzwerks tut – ein so genannter 51-%-Angriff [7]. Das Betreiben einer Node ist für jeden Nutzer ohne großen Hardwareaufwand (solange man nicht minen möchte) möglich und erlaubt es ihm, zur Sicherheit im Netzwerk beizutragen. [4]

2.1.3 Ethereum und Smart Contracts

Ether ist nach [3] die derzeit zweitgrößte Kryptowährung hinter Bitcoin und weist eine Marktkapitalisierung von rund 200 Milliarden US-Dollar auf. Vitalik Buterin beschreibt Ethereum als „Betriebssystem für die Blockchain“, auf der man immer wieder neue Anwendungen für verschiedenen Anwendungsfälle entwickeln kann [8]. Diese Anwendungen werden dann dezentral auf der Blockchain ausgeführt und nicht wie herkömmliche Anwendungen zentral auf einem Server.

Im White Paper zu Ethereum [6] verdeutlicht Buterin, wie man die Blockchain-Technologie nutzen kann, um nicht nur Transaktionen zu speichern und so einen Konsens einer Währung zu erreichen, sondern auch um Programme auf der Blockchain ausführen zu können. Wie bei Bitcoin kommt bei Ethereum ebenfalls die Blockchain-Technologie zum Einsatz. Die Funktionsweise ist bei beiden Kryptowährungen gleich. Als Hash-Algorithmus wird bei Ethereum Ethash verwendet, und die Bestätigungszeit

eines Blockes liegt im Schnitt bei nur 13,4s und ist damit deutlich schneller als im Bitcoin-Netzwerk [9] [10].

Um Programme auf der Ethereum-Blockchain ausführen zu können, gibt es die sogenannten „Smart Contracts“. Sie stellen eine Art Vertrag im Netzwerk dar und sind durch die Blockchain von jedem einsehbar. Sie werden durch programmierbaren Code definiert und genauso ausgeführt, wie sie programmiert sind. Ausgeführt wird ein solcher Vertrag zum Beispiel, wenn eine bestimmte Menge Ether an den Contract überwiesen wurde. Eine Anwendung kann in Ethereum also mit solchen Contracts geschrieben werden. Im Ethereum-Netzwerk gibt es zwei Arten von Accounts: Nutzer-Accounts, die mit einem Private Key wie bei Bitcoin kontrolliert werden, und Contracts, die durch den vordefinierten Code kontrolliert werden. Mit diesen Contracts lassen sich autonome dezentrale Organisationen bilden, die sogenannten Decentralised Autonomous Organisations (DAO).

Damit solche Contracts gespeichert und der Code im Netzwerk ausgeführt werden kann, besteht ein Block im Netzwerk nicht einfach nur aus Transaktionen, sondern aus Funktionen, die eine Transaktion, aber auch andere Vorgänge definieren können, die dann von der Ethereum Virtual Machine (EVM) ausgeführt werden. Die EVM befindet sich auf der Blockchain und damit auf jeder Node des Ethereum-Netzwerkes. Das bedeutet, dass der Code eines Smart Contracts nahezu gleichzeitig auf jeder Node und damit im Ethereum-Netzwerk ausgeführt wird, sobald ein Block validiert wurde [11]. [6]

Auf der Ethereum-Blockchain lassen sich demnach dezentrale und sichere Anwendungen entwickeln, die nicht nur Geld dezentral transferieren, sondern auch Daten dezentral und sicher speichern. Diese Anwendungen werden aufgrund ihrer Dezentralität auch decentralised applications (dApps) genannt. Um auf Ethereum Smart Contracts auszuführen und Daten speichern zu können, muss eine Gebühr entrichtet werden, denn eine Node führt solche Operationen wie Transaktionen in der EVM aus. Diese Gebühren werden als Giga-Wei (GWEI) angegeben. Dabei ist Wei die kleinste Einheit im Netzwerk und entspricht 10^{-18} Ether [12]. Ether bzw. GWEI ist eine Art Treibstoff im Netzwerk und wird daher auch als Gas bezeichnet. [6]

Im Gegensatz zum Bitcoin-Netzwerk ist das Ethereum-Netzwerk nicht statisch und wird von vielen verschiedenen Entwicklerteams weltweit aus dem Ethereum-

Ökosystem weiterentwickelt. Das größte Upgrade seit dem Launch 2015 stellt Ethereum 2.0 dar. Mit diesem Upgrade soll das Ethereum-Netzwerk skalierbarer, sicherer und effizienter gemacht werden. Dies soll durch den Wechsel von einem Proof-of-Work-Konsensus zu einem Proof-of-Stake-Konsensus erreicht werden. Dabei sollen durch das Halten von Ether neue Blocks validiert werden. Damit fällt der rechenaufwändige Mining-Prozess weg, und das Netzwerk wird ökonomischer und dezentraler, da nicht große Rechenzentren die Blocks validieren, sondern diejenigen mit einer besonders hohen Menge Ether. Mit geregelten Staking-Pools, Zusammenschlüsse vieler kleiner Ether-Besitzer, kann jeder, auch mit nur wenig Ether, am Staking teilnehmen. Vor allem sollen durch das Upgrade niedrigere Gebühren im Netzwerk anfallen, was nicht nur Transaktionen billiger macht, sondern auch dApps mit Smart Contracts ansprechender und durch mehr Dezentralität sicherer macht. [13] [14]

2.1.4 Potenzial und aktueller Anwendungszweck von Kryptowährungen

Mittlerweile weisen Kryptowährungen eine globale Marktkapitalisierung von 1,97 Billionen US-Dollar auf [3]. Darin ist nicht nur Bitcoin enthalten, sondern mittlerweile über 4.500 andere Kryptowährungen, die sich auf andere Anwendungsfälle spezialisiert haben (vgl. Kap. 2.1.1). Doch welches Potenzial haben Kryptowährungen in unserer vernetzten Welt und wie werden sie bereits eingesetzt?

Kryptowährungen bieten mit Hilfe von Kryptografie, dezentralen Netzwerken und der Blockchain-Technologie ein unabhängiges, globales, anonymes und freies Zahlungsmittel. Dadurch, dass diese Technologie jedem zur Verfügung steht, der einen Internetzugang besitzt, und jeder daran mitarbeiten kann, um sie zu verbessern, stellen Kryptowährungen ein deutlich demokratischeres Geldsystem dar, als das gegenwärtige. Das Besondere an Kryptowährungen ist nicht nur die Technologie, sondern auch die Globalität. So sind Kryptowährungen vor allem in Ländern wie Nigeria, Vietnam und Südafrika beliebt [15]. Hierbei handelt es sich um Entwicklungs- oder Schwellenländer. Hier gibt es eher schwache und instabile Währungen. Bürgerinnen und Bürger versuchen dort, in stabile Währungen wie den US-Dollar oder Euro zu wechseln. Diese sind jedoch von Behörden vor Ort oft stark reguliert. Daher bieten Kryptowährungen Menschen einen sicheren und anonymen Weg, Werte aufzubewahren. Da Wallets mit einer sogenannten „Seed Phrase“, also einer zufälligen

Reihenfolge von 12 bis 24 Wörtern, gesichert sind, müssen sich Besitzer von Kryptowährungen nur diese Seed Phrase merken. Mit dieser Phrase können Menschen überall ihre Wallets und Coins wiederherstellen und über ihr Geld verfügen. Da die Phrase nicht verschriftlicht ist, ist der Geldbesitz vor illegalem oder autoritärem Zugriff geschützt.

Kryptowährungen und vor allem die ihr zugrunde liegende Blockchain-Technologie sind auch für die breite Anwendung in der Wirtschaft attraktiv. Blockchain macht nicht nur Transaktionen im Finanzsektor, sondern auch den Austausch von Geschäfts- und Industriedaten sicher, kalkulierbar und schnell. Die von Industrieunternehmen getragene *International Data Spaces Association (IDSA)* hat eine Referenzarchitektur entwickelt, die einen sicheren Austausch und die Verknüpfung von Daten innerhalb von Geschäftsökosystemen ermöglichen soll [16]. Reinhold Achatz, Vorstandsvorsitzender von *IDSA*, betrachtet Blockchain als adäquate, ausgereifte und anerkannte Technologie für den Betrieb von datengetriebenen Geschäftsökosystemen [17].

Decentralised Finance (DeFi) ist im Kryptowährungs-Sektor beliebt. DeFi soll traditionelle Finanzdienstleistungen wie Kredite oder Börsen in den Kryptowährungs-Sektor bringen [18]. Die DeFi-Bewegung wird von der Idee geleitet, dass das Finanzsystem nicht von einer zentralen dritten Partei kontrolliert wird, sondern dezentral ist. Die meisten DeFi Projekte werden auf der Ethereum-Plattform entwickelt und nutzen die Smart Contract-Technologie. So lassen sich mit Smart Contracts auf der Ethereum-Plattform eigene Protokolle für Tokens programmieren, um zum Beispiel *Stablecoins* zu schaffen, die sich an US-Dollar oder Euro orientieren. Zudem gibt es dezentralisierte Börsen wie UniSwap, auf denen man einzelne Tokens untereinander handeln kann [19]. Auf dezentralen Peer-to-Peer-Lending-Plattformen wie Compound können Tokens gegen Zinsen verliehen oder geliehen werden [20].

Weitere Projekte im DeFi-Bereich gibt es in der Kultur: Zum Beispiel werden mit Non Fungible Tokens (NFT) digitale Kunstwerke wie Bilder oder Musikstücke auf der Ethereum-Blockchain gespeichert und einem bestimmten Account zugeordnet. So sind Fälschungen und unerlaubte Vervielfältigung von Kunstwerken nicht mehr möglich.

Wie viele Menschen Kryptowährungen nutzen, lässt sich wegen der Anonymität von Kryptowährungen schwer herausfinden, auch wenn jede Transaktion öffentlich verfügbar ist. Mit so genannten „OnChain-Analysen“ lässt sich näherungsweise ermitteln, wie viele Nutzer es gibt. Derzeit sind es zwischen 1,2 und 1,3 Millionen aktive Adressen im Bitcoin-Netzwerk [21]. Diese Adressen können jedoch nicht nur von Menschen, sondern auch von Computern bedient werden, und Menschen können auch mehrere Adressen gleichzeitig benutzen. Ebenfalls nicht enthalten sind Menschen, die Kryptowährungen zwar haben, diese jedoch nicht aktiv als Zahlungsmittel nutzen und selten Transaktionen ausführen. Laut GlassnodeStudio gibt es ca. 30 Millionen Adressen im Bitcoin-Netzwerk, die Bitcoins besitzen [22]. Auch hier ist es möglich, dass ein Mensch Zugriff auf mehrere Adressen hat oder es keinen Zugriff mehr auf die Coins gibt, da der Private Key oder die Seed Phrase verloren gegangen ist.

2.2 Mängel derzeitiger Transaktionsplattformen

Im traditionellen Finanzsystem gibt es zwei Formen des Geldbesitzes: Bargeld und Online-Geld. Bargeld ist physisch vorhanden und durch Wasserzeichen und andere Schutzmechanismen vor Fälschung geschützt. Kleine Geldmengen lassen sich gut in einer Geldbörse oder einem Sparschwein aufbewahren. Bei großen Geldmengen wird die Aufbewahrung wegen der Sicherheit und der physischen Menge schwierig. Wer dieses Geld online zur Verfügung haben möchte, muss Kunde bei einer Bank sein und darauf vertrauen, dass diese das Geld sicher aufbewahrt, Transaktionen korrekt ausführt und vor allem Nutzerdaten und Nutzerverhalten sicher speichert.

Um Zahlungen im Geschäft oder Online-Handel zu tätigen, kommen Kreditkarten zum Einsatz. Hier ist nicht nur die Bank für die korrekte Ausführung von Transaktionen verantwortlich, sondern auch Unternehmen wie *Visa* oder *MasterCard*. Soll Geld von Person zu Person gesendet werden, kann das Verfahren der Single Euro Payments Area (SEPA) genutzt werden. Hier werden Zahlungen innerhalb im Euroraum direkt zwischen den Banken ausgeführt. Einfach und flexibel lässt sich Geld mit Online-Bezahldiensten wie *PayPal* oder *Sofortüberweisung* verschicken. Inzwischen lässt sich mit dem Smartphone oder der Smartwatch und den Diensten von *Google Pay* oder *Apple Pay* bezahlen. Über Plattformen wie *VimPay* ist es ebenfalls möglich, mit anderen Menschen über einen Chat in Kontakt zu treten und dieser Person Geld zu senden.

Diese Transaktionsformen haben ein Problem gemeinsam: Eine dritte zentrale Partei ist für die Transaktionsausführung zwingend erforderlich (vgl. Kap. 1), und Verbraucherinnen und Verbraucher müssen darauf vertrauen, dass Zahlungen korrekt ausgeführt werden. Die Bank oder andere Zahlungsdienstleister sind dafür verantwortlich, dass nicht zu viel oder zu wenig abgebucht und das korrekte Zielkonto gefunden wird. Zudem ist für den Verbraucher unklar, welche Firmen und Banken an den Transaktionen beteiligt sind und welche Daten an wen weiter gegeben werden.

Kryptowährungen sind im Gegensatz zum traditionellen Finanzsystem neutral. Möchte man ein Konto bei einer Bank eröffnen, wird eine Vielzahl persönlicher Daten eingefordert und gründlich überprüft, bevor es möglich ist, das Bankkonto zu nutzen. Darüber hinaus sind Transaktionen teuer. So zahlt ein Geschäftskunde bei *PayPal* für eine Transaktion von 2.000 Euro etwa 50 Euro Gebühren [23]. Für Privatkunden ist eine Transaktion in den meisten Fällen kostenlos, geht diese jedoch ins Ausland, können die Gebühren bis zu 3,99 Euro betragen [24]. Auch können Transaktionen lange dauern. Bei der Bezahlung mit der Kreditkarte dauert eine Transaktion bis zu 30 Tage. In dieser Zeit haben Transaktionssender und -empfänger keine Sicherheit darüber, ob die Transaktion vom Kreditinstitut genehmigt oder zurückgezogen wird. Auch bei *PayPal* dauert es einige Tage, bis eingegangenes Geld verfügbar ist. In diesem Zeitraum ist das Geld Kapital für die dritte Partei.

Ein weiteres Problem herkömmlicher Transaktionsplattformen ist deren zentrale Struktur. Banken und andere Zahlungsdienstleister sind durch ihre Zentralität anfällig für Hackerangriffe. So haben meist nur wenige Computer die Kontrolle über das gesamte Bankensystem. Zum Beispiel wurden beim „Carbanak-Vorfall“ [25] ungefähr 1,2 Milliarden US-Dollar erbeutet. Hier wurde das Geld von mehreren Tausend Kunden entwendet, allerdings nur eine Institution gehackt. Bei Kryptowährungen wie Bitcoin oder Ethereum verfügt jeder Geldeigentümer über die Private Keys und hat diesen sicher verwahrt. Sollte es hier zu einem Hack kommen, ist nur die einzelne Person betroffen und nicht das gesamte Netzwerk. Selbst bei einem 51%-Angriff (vgl. Kap. 2.1.2) sind die Coins in einer Wallet sicher, da man nachträglich keine Transaktionen in der Blockchain ändern kann [7].

2.3 Stand der Forschung

Gemäß Satoshi Nakamoto setzt Bitcoin auf ein neues Privatsphäre-Modell [4]. Er beschreibt, dass das traditionelle Finanzsystem Privatsphäre erreicht, indem nur eine limitierte Anzahl an Menschen Zugang zu Transaktionsinformationen hat, während der Öffentlichkeit die Identitäten verborgen bleiben, d. h. sie weiß nicht, von wem das Geld zu wem transferiert wird. Das von Nakamoto skizzierte neue Modell setzt hingegen darauf, die Transaktionen der Öffentlichkeit zur Verfügung zu stellen. Diese werden aber von einzelnen Identitäten getrennt. Somit weiß man, dass Geld überwiesen wurde, jedoch nicht von wem zu wem. Erreicht wird dies dadurch, dass in der Blockchain nur Adressen vermerkt sind. Diese Adressen sind ein aus dem Private Key erzeugter Public Key (vgl. Kap. 2.1.2). Um mehr Privatsphäre zu erreichen, lassen sich auch mehrere Public Keys für denselben Private Key erzeugen, sodass bei jeder Transaktion ein anderer Public Key in der Blockchain vermerkt werden kann. Andreas M. Antonopoulos schätzt die Anonymität derzeit geringer ein, als sie sein sollte, sie könne aber durch neue kryptografische Methoden anonymer werden [26]. Tatsächlich gibt es mittlerweile Möglichkeiten, Coins anonymer zu benutzen. Zum Beispiel garantiert die Wasabi-Wallet mit vermischten Transaktionen eine höhere Anonymität als normale Transaktionen im Bitcoin-Netzwerk [27].

Anotonopoulos betrachtet die Anonymität einer Währung als sehr wichtig für die Selbstbestimmung und freie Meinungsäußerung einer Gesellschaft. Kryptowährungen eignen sich, Geld über nationale Grenzen ohne Einmischung und Überwachung seitens Dritter zu transferieren. So stünde Geld beispielsweise für politische Kampagnen zur Verfügung, die sonst finanziellen Restriktionen unterlägen; mit den Kampagnen könnten Meinungen ausgedrückt werden [26]. Zudem haben weltweit nur 69 % der Bevölkerung ein Konto bei einer Finanzinstitution [28]. Das bedeutet, dass 31 % der Weltbevölkerung kein Konto und möglicherweise keinen Zugang zu herkömmlichen finanziellen Dienstleistungen haben und ihr etwaiges Barvermögen nur eingeschränkt einsetzen können.

Durch die Anonymität wird davon ausgegangen, dass Kryptowährungen für Illegale Zwecke wie Drogen und Waffenhandel genutzt werden. Bei der Visual Objects Digital Currency Survey [29] gaben 38 % der Befragten an, dass sie mit Kryptowährungen Essen kaufen, und 34 % beschaffen Kleidung. 29 % der Befragten nutzen

Kryptowährungen als Geldanlage oder Wertspeicher, und 21 % teilten mit, von Kryptowährungen Gold zu kaufen. Vergleichsweise wenige Menschen gaben an, Waffen (15 %) oder Drogen (11 %) zu kaufen. Auch wenn 15 und 11 % für fragwürdige Einkäufe hoch erscheinen mögen, zeigt die Befragung, dass akzeptierte Aktivitäten bei Kryptowährungen wesentlich stärker vertreten sind.

Kryptowährungen verursachen einen hohen Energieverbrauch. Allein Bitcoin verbraucht pro Jahr ungefähr 144,9 TWh [30]. Das ist etwa so viel Strom, wie Polen (141 TWh) oder Ägypten (150 TWh) verbrauchen [31]. Somit verbraucht eine digitale Währung als einzelne Anwendung so viel Strom wie ein ganzes Land. Jedoch steht ein Land nicht im Verhältnis zu einem globalen Finanzsystem. Daher ist der Stromverbrauch mit dem traditionellen Finanzsystem zu vergleichen. So verbraucht das globale Bankensystem etwa 650 TWh/Jahr [32]. Das ist etwa 4,5 Mal so viel Strom wie Bitcoin. Ulrich Gallersdörfer sieht das Problem nicht in den Kryptowährungen selbst, sondern dort, woher diese ihre Energie beziehen [33]. Wie bei Rechenzentren von großen Internetkonzernen und Streaming-Plattformen sollte man sich auch bei Kryptowährungen damit auseinandersetzten, wie der Energiebedarf generell nachhaltiger gestaltet werden könne.

Bitcoin und andere Kryptowährungen können eine stabile Währung für Menschen sein, deren Land instabil und von Korruption geprägt ist. Antonopoulos begründet jedoch, dass Bitcoin auch der privilegierten Weltbevölkerung dient: Kryptowährungen böten günstigere und schnellere Möglichkeiten, Transaktionen auszuführen als das traditionelle Finanzsystem, und selbst der Euro ist von einer kleinen Inflation betroffen. Darüber hinaus nennt Antonopoulos die Neutralität von Kryptowährungen als großen Vorteil. Jeder könne durch die Dezentralität am Netzwerk partizipieren und dessen Dienste nutzen. So könne sich jemand einen Kredit in Form von Kryptowährungen nehmen, ohne dass diese Person auf ihre Herkunft, Religion, Hautfarbe und Qualifikation geprüft würde. [34]

Eine besonders große Rolle für die Zukunft können laut Fabian Schär DeFi-Produkte haben. Er erläutert, wie DeFi-Projekte traditionelle Finanzprodukte ersetzen und neue Produkte zur Verfügung stellen. Die Technologie der Smart Contracts könne in den nächsten Jahren einen großen Bestandteil im Kryptowährungs-Sektor darstellen. [35]

2.4 Dezentrale Transaktionsmechanismen als Lösungsansatz

Traditionelle Finanzkonzepte sind durch ihre zentrale Struktur sehr ineffizient und teuer, so dass Transaktionen bis zu einen ganzen Monat dauern können, und gerade bei großen Summen und internationalen Transaktionen fallen hohe Gebühren an (vgl. Kap. 2.2). Zudem sind sie nur wenig anonym, basieren auf Vertrauen und erst nach einer ausführlichen Identitätsprüfung ist es dem Nutzer möglich, online Geld zu verwenden. Zudem werden Währungen im herkömmlichen Sinne von einer zentralen Instanz gesteuert. So haben Zentralbanken Einfluss auf den Wert einer Währung.

Damit eine Anwendung Geld online versenden kann, müssen Nutzer mit Banken, Kreditinstituten und Zahlungsdienstleistern Verträge schließen oder eine Geschäftsbeziehung eingehen. Dabei müssen sowohl der Nutzer als auch die Anwendung darauf vertrauen, dass Transaktionen korrekt durchgeführt werden. Zudem können die Transaktionen sehr lange dauern, und bei großen Geldmengen können hohe Gebühren für eine Transaktion anfallen. Ein weiteres Problem bei der Entwicklung einer globalen Transaktionsanwendung sind die unterschiedlichen Währungssysteme. So muss bei grenzüberschreitenden Transaktionen – außer innerhalb der Eurozone – immer ein Wechsel zwischen den beteiligten Währungen stattfinden. Dies bedeutet Aufwand selbst bei kleinen Beträgen, da hier an vielen verschiedenen Börsen Währungspaare gehandelt werden müssen, damit Geld an jede Nation gesendet werden kann.

Kryptowährungen schaffen mit ihrer dezentralen Struktur eine globale Währung, die überall online verfügbar ist. Sie bieten Privatpersonen durch ihre dezentrale Struktur deutlich mehr Sicherheit und Kontrolle über ihr Geld. Zudem sind sie demokratischer, da jeder Zugang zum Netzwerk hat und daran partizipieren kann. Wer Kryptowährungen besitzen möchte, muss keine aufwändigen Identitätsprüfungen durchlaufen und hat direkten Zugang zu vielen verschiedenen Finanzprodukten.

Die Plattform Ethereum bietet mit Smart Contracts die Möglichkeit, neue Anwendungen auf der Ethereum-Blockchain zu programmieren (vgl. Kap. 2.1.3). So ist es möglich, dezentrale Anwendungen zu programmieren, ohne eine eigene Blockchain mit eigener Währung zu entwickeln. Um eine globale Transaktionsanwendung zu entwickeln, eignet sich Ethereum besonders gut. Neben Ethereum gibt es Alternativen wie Cardano oder Algorand, die geringere Transaktionskosten aufweisen als

Ethrereum. Diese sind jedoch noch unausgereift, so dass es noch nicht möglich ist, Smart Contracts zu programmieren, oder die Entwickler-Community ist noch klein, so dass es schwierig ist, gute Dokumentationen oder Threads in Foren zu finden.

3 Die Entwicklung einer Anwendung für dezentrale Transaktionen

3.1 Ziele und Herangehensweise

3.1.1 Aufgaben und Anwendungszweck der Anwendung

Kryptowährungen und die Blockchain-Technologie sind mittlerweile kein Pionierthema mehr, sondern sie erreichen breite Bevölkerungsgruppen. Es ist in den letzten Jahren durch ein großes Angebot an verschiedenen Krypto-Börsen und Wallet-Anbietern einfacher geworden, Kryptowährungen zu besitzen und zu handeln. Jedoch gibt es neben Finanzprodukten und einigen wenigen sozialen Netzwerken wie *Cent* und *CryptoKitties* nur wenige Anwendungsfälle für Verbraucher. Daher werden Kryptowährungen vor allem als spekulatives Asset gehandelt.

Die zu entwickelnde Anwendung soll den analogen und anonymen Bargeldtransfer in Kombination mit einem Kommunikationsdialog online ermöglichen.

Bisher war es möglich, mit Messenger-Diensten einen Dialog zu führen. Wollte nun die eine Partei der anderen Partei online Geld schicken, musste auf eine dritte Partei wie eine Bank oder *PayPal* zurückgegriffen werden. Wollte man völlig anonym bleiben, hatten Pioniere die Möglichkeit, Kryptowährungen zu benutzen, indem sie die Adressen austauschten und dann mit ihrer Wallet eine Überweisung tätigten. Jenseits davon blieb nur der physische Austausch von Bargeld.

Mit der Anwendung soll es nun möglich sein, sich wie bei einer herkömmlichen Messenger-Anwendung zu vernetzen, also mit anderen Personen global zu kommunizieren. Gleichzeitig soll die Anwendung dem Benutzer die Möglichkeit geben, die Kryptowährung Ether zu verwenden. So sollen Personen nicht nur kommunizieren, sondern untereinander auch einfach, schnell und günstig Geld versenden können. Dabei soll dies für den Nutzer völlig anonym funktionieren.

3.1.2 Umsetzung

Die Anwendung soll für jeden nutzbar und über das Internet erreichbar sein. Dementsprechend soll eine Web-Anwendung entwickelt werden, die über den Browser erreichbar ist. Der Nutzer soll über einen benutzerfreundlichen Login mit E-Mail und Passwort auf die Chat-Seite gelangen. Dort soll er dann mit anderen Nutzern in einem Eins-zu-eins-Chat kommunizieren. Über „Settings“ soll der Nutzer von der Chat-Seite

zu den Einstellungen gelangen, wo er neben Einstellungen über Nutzerdaten auch den Button „Connect Wallet“ finden wird, über den er seine Wallet mit der Anwendung verbinden kann.

Um die Messenger-Anwendung so benutzerfreundlich wie möglich zu gestalten und sie auch Nutzern ohne Vorkenntnisse im Bereich Kryptowährungen zugänglich zu machen, ist eine Wallet keine Voraussetzung für die Nutzung der Anwendung. So werden Daten wie Nachrichten und Nutzerdaten nicht auf der Ethereum-Blockchain gespeichert. Stattdessen soll die Messenger-Anwendung zunächst konventionell Daten zentral auf einem Server speichern. Die Anwendung soll nur dann mit dem Ethereum-Netzwerk kommunizieren, um Zahlungen dezentral über das Netzwerk sicher, günstig, schnell und anonym abwickeln zu können.

3.2 Aufbau und Funktionsweise der Anwendung

Die Anwendung soll als Web-Anwendung über einen Browser erreichbar sein (vgl. Kap. 3.1.2). Um dies zu realisieren, werden die JavaScript-Bibliothek React.js und die Programmiersprache JavaScript verwendet. React.js ermöglicht es, mit JavaScript Benutzeroberflächen und Webserver für Webanwendungen zu programmieren. Dabei kann eine Webanwendung mit React.js-JSX-Elementen dynamisch und interaktiv gestaltet werden. React.js soll dann mit der Datenbank, Ethereum und dem Nutzer kommunizieren.

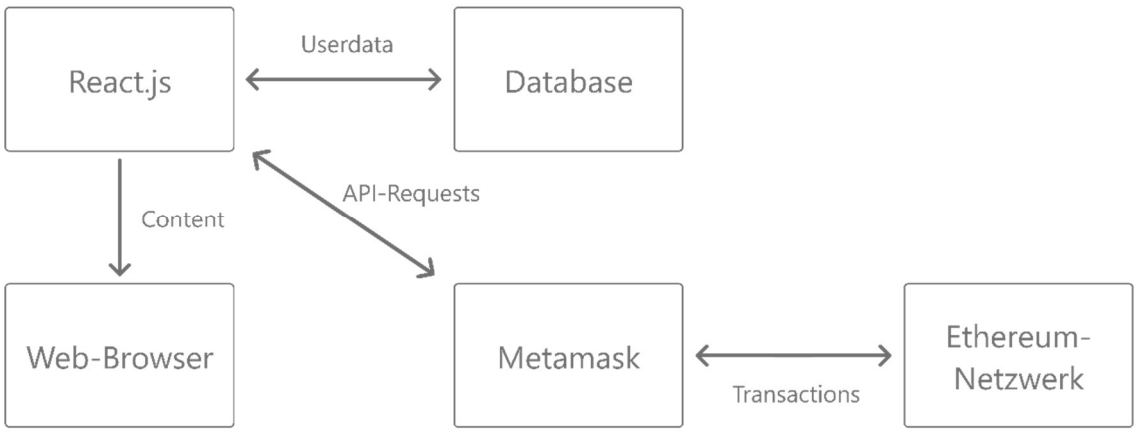

Abbildung 4: Server-Kommunikation in der Anwendung vereinfacht dargestellt

Wie in *Abbildung 4* dargestellt, kommuniziert der React.js-Server mit der Datenbank, die im vorliegenden Fall durch *Google Firebase* abgebildet wird. Damit der Nutzer sein Ether sicher verwahren und trotzdem in einer Webanwendung verwenden kann, gibt es für den Browser die Extension Metamask. Metamask bildet eine im Browser integrierte Ethereum-Wallet, mit der Anwendungen über die web3.js-API kommunizieren können. Das bedeutet, die Anwendung kann mit Metamask und der Zustimmung des Nutzers Transaktionen im Ethereum-Netzwerk durchführen. dApps wie Compound, CryptoKitties und Cent nutzen ebenfalls Metamask, wodruch der Nutzer nur eine Wallet im Browser besitzen muss, um in vielen verschiedenen Anwendungen Ether zu verwenden.

3.2.1 Kommunikation mit dem Nutzer – React.js

Die Webanwendung besteht aus drei Webseiten. Dem Login bzw. der Registrierung, der Homepage und den Settings. Auf der Login-Seite kann sich der Nutzer mit seinen Login-Daten einloggen und wird auf die Homepage weitergeleitet. Auf der Registrierungsseite kann sich der Nutzer mit Username, E-Mail und Passwort registrieren und wird dann ebenfalls auf die Homepage weitergeleitet. Auf der Homepage kann der Nutzer über die Liste der anderen Nutzer auf die einzelnen Chats mit den jeweiligen Personen zugreifen. Klickt der Nutzer auf einen anderen Nutzer, erscheint der Chatverlauf mit dem jeweiligen Nutzer; zudem gibt es die Möglichkeit, über ein Textfeld und den Button „Send“ eine Nachricht an den anderen Nutzer zu schicken.

Haben beide Nutzer die Metamask Wallet mit der Anwendung verknüpft, so erscheint ein Textfeld und ein Button, mit denen der Nutzer Ether an den anderen Nutzer senden kann. Klickt der Nutzer auf die Settings in der Navigationsbar, gelangt er zu den Einstellungen. Dort findet der Nutzer die „Wallet Settings“ mit dem Button „Connect Wallet“ und den Reiter „Personal Information“, in denen Informationen, wie Username, E-Mail und Passwort geändert werden können.

Die einzelnen Seiten werden in React.js als Components programmiert. Die Components können dann andere Components enthalten, die einzelne Elemente auf der Webseite darstellen. In diesen Components wird der darzustellende Inhalt mit der JSX Syntax programmiert.

```
10
11
12  const Message = (props) => {
13
14      let {index, chat} = props;
15
16      const auth = useSelector( selector: state => state.auth);
17
18      return (
19          <div
20              style={{width: '100%'...}}
26              key={index}>
27              <div className={chat.user_uid_Sender === auth.uid ? 'messageStyle sender' : 'messageStyle receiver'}...>
44              <p className="message-Time"...>
47          </div>
48      );
49  }
50
51  export default Message
52
```

Abbildung 5: React Component Message

JSX wird in der Return-Funktion der Component übergeben und bildet HTML-Code mit JavaScript-Elementen ab. Die JavaScript-Elemente lassen es zu, dass die eigentlich statische HTML-Syntax hier programmiert werden kann und sich dynamisch anpasst. So enthält das ClassName-Attribut in dem div in Zeile 27 unterschiedliche Werte in Abhängigkeit davon, ob die Bedingung erfüllt ist oder nicht.

Um Nutzereingaben in React.js direkt zu verarbeiten, gibt es mit JSX die Möglichkeit, die Eingabe in ein bestimmtes Feld direkt in einem State der Component abzuspeichern. States von Components lassen sich mit Attributen aus der objektorientierten Programmierung gleichsetzen.

3.2.2 Kommunikation mit Ethereum – Metamask

Bei Metamask handelt es sich um eine Wallet, die sich direkt im Browser befindet (vgl. Kap. 3.2). Sie lässt sich als Browser Extension in den meisten Browsern installieren. In dieser Wallet kann der Nutzer sich eine neue Wallet mit neuem Public Key und Private Key erstellen, an die er Ether senden kann. Es gibt zudem die Möglichkeit, eine bestehende Wallet mit dem Private Key zu importieren.

Mit der web3.js API lässt sich mit der Metamask Wallet kommunizieren, sofern der Nutzer die Extension im Browser installiert hat. Damit der Nutzer auch auf die Wallet zugreifen kann, muss er die Zustimmung geben, dass die Anwendung auf die Wallet zugreifen darf. Stimmt der Nutzer zu, wird in der Wallet die jeweilige Domain der

Anwendung gespeichert. Der Nutzer kann diese jederzeit entfernen, sodass die Anwendung nicht mehr mit der Wallet kommunizieren darf.

Mit verschiedenen API Requests kann die Anwendung Daten aus der Wallet abfragen und Aufträge an die Wallet schicken. Zum Beispiel kann die Anwendung die Adressen, die aktuelle Bilanz und ggf. die verschieden Tokens abfragen.

```
62  // ETH Transaction
63  const sendETH = () => {
64
65      if (!isNaN(amount) && !amount == '') {
66          ethereum
67              .request({
68                  method: 'eth_sendTransaction',
69                  params: [
70                      {
71                          from: accounts[0], // Adresse des Senders
72                          to: chatUser.ETH_Adress, // Adresse des Empfängers
73                          value: web3.utils.toHex(web3.utils.toWei(amount)) // Menge an Ether, die überwiesen werden soll
74                      },
75                  ],
76              })
77              .then((txHash) => {
78                  web3.eth.getTransactionReceipt(txHash, callback: (e :Error) => e)
79                      .then(result => {
80                          submitTransaction(txHash, web3.utils.fromWei(result.gasUsed.toString()));
81                      })
82              })
83              .catch((error) => console.error);
84      } else {
85          console.log('amount is not a number')
86      }
87  }
```

Abbildung 6: Funktion sendETH aus HomePage

Mit diesen Requests kann die Anwendung von der Wallet Daten abfragen, aber auch Transaktionen beauftragen. Diese Requests wird, wie in *Abbildung 6* dargestellt, mit „*eth_sendTransaction*“ aufgerufen. Als Parameter muss angegeben werden, von welcher Adresse welcher Betrag zu welcher Adresse überwiesen wird. Als Senderadresse wird die Adresse angegeben, die gerade mit der Wallet verbunden ist, und als Empfängeradresse wird die Adresse verwendet, die der andere Nutzer zuletzt mit der Anwendung verknüpft hatte. Alle Werte müssen als hexadezimale Zahl übergeben werden. Die Adressen werden als Hexadezimal-String gespeichert und können Problemlos übergeben werden. Der Nutzer gibt den Betrag, der überwiesen werden soll, in Ether und als Dezimalzahl an. Somit muss die Zahl zuerst in Wei, also die kleinste Einheit im Netzwerk, und dann in die entsprechende Hexadezimalzahl umgewandelt werden. Beispiel: 1 ETH = 10^{18} Wei; 10^{18} Wei = 0xde0b6b3a7640000 Wei

(Hexadezimalzahlen werden im Ethereum-Netzwerk beginnend mit 0x gekennzeichnet).

3.2.3 Datenspeicherung und Verarbeitung – Firebase

Chat und Nutzerdaten müssen zentral gespeichert werden, damit keine Ethereum Wallet zum Nutzen der Anwendung vorausgesetzt werden muss (vgl. Kap. 3.1.2). Gespeichert werden die Daten auf *Google Firebase*, da es dort die Möglichkeit gibt, Nutzer zu verwalten und Daten in Echtzeit abzurufen und zu bearbeiten. *Firebase* bietet an, Daten im JSON-Format zu speichern. Nach [36] basiert das JSON-Format auf der Programmiersprache JavaScript und lässt sich demnach gut in dieser Sprache verwenden. *Firebase* wurde wegen seiner Benutzerfreundlichkeit gewählt, eine Alternative ist *MongoDB*.

Damit React.js mit *Firebase* kommunizieren kann, um Daten abzurufen und zu verarbeiten, wird die JavaScript-Bibliothek Redux.js verwendet. Mit Redux.js werden die Daten aus der Datenbank abgerufen, um sie dann gebündelt und organisiert den React Components zu Verfügung zu stellen. So muss nicht in jede React Component einzeln eine Datenabfrage programmiert werden. Dies erleichtert die Verwendung von Daten in einer Anwendung. [37]

Damit sich ein Nutzer registrieren und einloggen kann, wird sein Nutzername, die E-Mail und das Passwort verschlüsselt gespeichert (vgl. *Abbildung 7*). Zusätzlich wird gespeichert, ob der Nutzer gerade online ist, damit dies anderen Nutzern angezeigt werden kann, wann der Nutzer einen Account erstellt hat und eine einzigartige ID, damit in anderen Objekten der Datenbank auf dieses Objekt verwiesen werden kann. Hat der Nutzer eine Metamask Wallet mit der Anwendung verbunden, wird die Adresse ebenfalls gespeichert, damit andere Nutzer Transaktionen an diese Adresse tätigen können.

Abbildung 7: Nutzerdaten im JSON-Format

Damit Transaktionen und Nachrichten im Chat angezeigt werden können, werden diese ebenfalls gespeichert (vgl. *Abbildung 8*). Auch hier werden ein Zeitstempel, der anzeigt, wann die Nachricht oder Transaktion gesendet wurde, und eine einzigartige ID gespeichert. Um nachvollziehen zu können, wer die Nachricht gesendet hat und wer sie erhalten soll, wird die ID des jeweiligen Nutzers entweder als Sender oder Empfänger im JSON-Objekt gespeichert. Um festzustellen, ob es sich um eine Nachricht oder eine Transaktion handelt, wird dies ebenfalls als String gespeichert. Handelt es sich um eine Nachricht, wird unter *message* die Nachricht gespeichert. Handelt es sich um eine Transaktion, wird unter *transaction* ein weiteres Objekt mit den Transaktionsdaten gespeichert. Dieses Objekt enthält den Hashwert der Transaktion, die Adresse des Senders und des Empfängers, die Menge an Ether die überwiesen wurde in Ether sowie die Transaktionsgebühr in Ether.

```
[
  {
    "createdAt": 1617974210368,
    "isViewed": false,
    "message": "Hi",
    "type": "text",
    "user_uid_Receiver": "aoa6g1HQajfXi704EGxt3vnjix93",
    "user_uid_Sender": "k5ZZbq3ZeERT1L91l9fJk4bIFOA3",
    "uid": "ed21af664939158df27776873f129999bec3e41b6b0d554a10e4b3411bc7b689"
  },
  {
    "createdAt": 1617974185888,
    "isViewed": true,
    "transaction": {
      "txHash": "0x235d31cbedac86418c7fa852a5842ade17c026ca6bb377b36dabef13210ff053",
      "from": "0x18ed89f4b1e7b2230c9d13e3121f5ba9fcab544b",
      "to": "0xe66ad4a5dc0afd16d8a8f1c3339c48ccce7946f3",
      "value": 14.9995,
      "gasUsed": 0.000000000000021
    },
    "type": "transaction",
    "user_uid_Receiver": "kFKiZIWgRceJvwUDSgOzv3bzBsI3",
    "user_uid_Sender": "CAfZzBmCbKUgNnnygvTrm8rRvQL2",
    "uid": "27a73e24a3e5b8806697d54ade102182b6921bfdcef4629ef4741c816fa9f3fb"
  }
]
```


Abbildung 8: Nachrichten und Transaktionen im JSON-Format

Der Einfachheit halber werden hier Daten wie die Nachrichten und Transaktionen nicht verschlüsselt. Dies könnte jedoch mit der Ende-zu-Ende-Verschlüsselung ergänzt werden. Nach [38] ist es mit der Ende-zu-Ende-Verschlüsselung möglich, dass nur der Sender und Empfänger die Nachrichten lesen können. Auf dem Server sind die Daten jedoch verschlüsselt und nicht für die Anwendung lesbar.

3.3 Konventionelle und innovative Elemente der Anwendung

Die Anwendung besteht aus verschiedenen Elementen, um eine Messenger-Anwendung zu gestalten, die dezentrale Transaktionen unterstützt. Dabei greift sie sowohl auf konventionelle als auch auf innovative Elemente zurück.

Die Anwendung besitzt zwei grundlegende Funktionen: die Messenger-Funktion und die Funktion, dezentrale Transaktionen im Ethereum-Netzwerk an andere Nutzer auszuführen. Das Konzept einer Messenger-Anwendung ist konventionell. Der erste

Messenger-Dienst *ICQ* erschien im Jahr 1996. Seitdem haben sich viele weitere Messenger-Dienste wie *WhatsApp*, *Telegram*, *Signal* oder *Threema* entwickelt. In der vorliegenden Anwendung werden die Daten wie bei konventionellen Messenger-Diensten zentral auf einem Server gespeichert. Dementsprechend sind Nutzerdaten und Chatnachrichten von den Vor- und Nachteilen einer zentralen Serverstruktur betroffen. Es muss einer dritten Partei für die Aufbewahrung der Daten vertraut werden, und es besteht eine höhere Anfälligkeit gegenüber Hackerangriffen. Dafür können Daten deutlich schneller zugestellt werden.

Die Anwendung hingegen verbindet dieses konventionelle Konzept eines zentralen Messenger-Dienstes mit dem innovativen Element des dezentralen Transaktionsmechanismus mit Ethereum. Auch etablierte Messenger-Dienste wie *Signal* integrieren Kryptowährungen [39]. Dort kommt jedoch nur die Kryptowährung MobileCoin zum Einsatz. Das Innovative an der Verwendung von Ethereum und Metamask ist, dass hier nicht nur eine Kryptowährung zum Einsatz kommen kann, sondern auch Tokens verwendet werden können. Zudem ist Metamask bereits in viele andere dApps integriert und wird daher von jenen bereits verwendet, die dApps und Ethereum nutzen.

3.4 Ausblick

Die vorliegende Anwendung bietet mit der Transaktionsfunktion mehr als eine herkömmliche Messenger-Anwendung. Hinsichtlich der dezentralen Transaktionen ist sie zwar noch einseitig, kann aber durch zusätzliche Funktonen erweitert werden.

Kryptowährungen befinden sich im stetigen Wandel und werden durch viele neue Projekte ergänzt. Resultate dieser Projekte können in die Anwendung integriert werden, zum Beispiel um diese sicherer und benutzerfreundlicher zu machen.

3.4.1 Dezentrale Datenspeicherung mit IPFS

Die dezentrale Datenspeicherung im Ethereum-Netzwerk birgt für die Anwendung zwei Herausforderungen: Zum einen muss ein Nutzer eine Ethereum Wallet besitzen, damit ihm die Daten zugeordnet werden können, und zum anderen werden Datenspeicherungen auf der EVM wie Transaktionen behandelt [6]. Somit müsste beim Speichern eines Nutzers oder einer Nachricht jedes Mal eine Transaktionsgebühr

entrichtet werden. Einen Lösungsvorschlag bietet hier das Inter Planentary File System (IPFS).

IPFS wurde 2015 von Juan Benet entwickelt und stellt ein dezentrales Peer-to-Peer File System da. Hier werden Dateien dezentral auf einem Netzwerk von Nodes gespeichert. Die Dateien werden in einem IPFS Object gespeichert. Dieses Objekt besitzt einen Hashwert, mit dem sich das Objekt im Netzwerk identifizieren lässt. Das IPFS Object enthält die Daten der Datei und Links, in denen Hashwerte anderer IPFS Objects gespeichert werden können, um auf andere Dateien im Netzwerk zu verlinken. Soll eine Datei im Netzwerk abgerufen werden, ist der Hashwert anzugeben. Die Nodes suchen dann im Netzwerk nach diesem Hashwert und geben die Datei zurück. Dadurch, dass eine Datei mit dem Hash im Netzwerk gespeichert wird, ist es nicht möglich, die Datei zu verändern, ohne den Hash zu verändern.

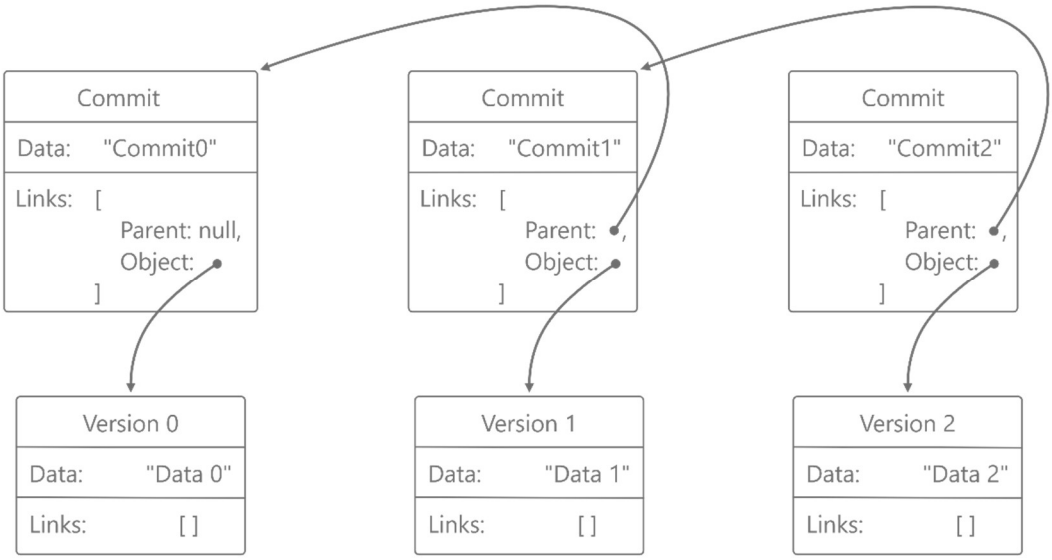

Abbildung 9: Visualisierung von Version Control in IPFS nach [40]

Möchte man eine Datei verändern, nutzt man einen Versionsverlauf (vgl. *Abbildung 9*). Dieser besteht aus mehreren IPFS Objects, die Commits genannt werden und auf das IPFS Object mit der aktuellen Datei und auf den Vorgänger-Commit verlinken. [40]

IPFS bietet die Möglichkeit, Dateien fälschungssicher speichern und Änderungen nachvollziehen zu können. IPFS hat jedoch Nachteile. Es besteht die Gefahr, dass bei der Speicherung einer Datei auf nur wenigen Nodes, diese die Datei verlieren oder

offline gehen. Somit lässt sich die Datei im Netzwerk nicht mehr wiederfinden, und der Nutzer kann die Datei nicht mehr abrufen. Dies versucht nach [41] das Projekt Filecoin mit einem Proof-of-Replication verfahren zu lösen, indem Nodes dafür belohnt werden, dass sie Speicherplatz zur Verfügung stellen, und eine bestimmte Anzahl an Nodes vorausgesetzt wird, um eine Datei zu speichern. So müssten nicht einzelne Nutzer die Speicherung bezahlen, sondern die Anwendung. Da die Anwendung keine Gebühren von den Nutzern verlangt, müsste sie den Speicherplatz mit Werbung oder Datenhandel finanzieren. Nach einer geeigneten Lösung wird gesucht.

3.4.2 Integrierung von ERC-20 Tokens

Die Ethereum Blockchain ermöglicht es, mit Smart Contracts und dem ERC-20-Standard Tokens auf der Blockchain zu speichern. Diese Tokens können verschiedene Zwecke erfüllen. Sie können zum Beispiel ein Punktesystem auf einer Online Plattform darstellen oder ganz eigene Währungen darstellen. Unter anderem lassen sich mit ERC-20 Tokens auch bestehende konventionelle Währungen wie US-Dollar oder Euro abbilden. Sie können zudem Gold oder andere Kryptowährungen wie Bitcoin auf der Ethereum Blockchain darstellen. [42]

Diese Tokens lassen sich genauso wie Ether in einer Ethereum Wallet aufbewahren, auch in der Metamask Wallet. Die Anwendung könnte verschiedene Tokens unterstützen. Dadurch hätten Nutzer die Möglichkeit, nicht nur Ether zu versenden, sondern auch Tokens, die andere Währungen abbilden. So wäre es ihnen mit Tether-USD (USDT) oder USD Coin (USDC) möglich, über die Anwendung US-Dollar an andere Personen zu versenden. Mit dem Wrapped-BTC (WBTC) oder Tether-Gold (XAUT) Token könnten Anwender auch Bitcoin oder Gold über die Ethereum-Blockchain versenden.

Da der Ether großen Preisschwankungen ausgesetzt ist, könnten diese Tokens es Nutzern ermöglichen, stabile Währungen zu versenden und kein hohes Verlustrisiko einzugehen, wenn sie Coins an andere Personen versenden möchten.

3.4.3 Gruppenchats mit Smart Contracts

Bisher ist es in der vorliegenden Anwendung möglich, mit einzelnen Nutzern zu kommunizieren. Gruppenchats, wie sie andere Messenger-Dienste anbieten, können noch implementiert werden. Dort könnten Nutzer einfach Geld an Mitglieder einer

Gruppe senden. Zudem wäre es mit Smart Contracts möglich, dass eine Gruppe gemeinsam auf ein bestimmtes Ziel spart.

Gruppenmitglieder würden Ether oder andere Tokens an einen Smart Contract senden, nachdem sie zuvor festgelegt haben, wie viel Geld gespart und an welche Adresse das gesammelte Geld bei Erreichen des Sparziels gesendet werden soll. Der Smart Contract würde jedes Mal, wenn Geld zu ihm transferiert wird, überprüfen, wie viel Geld sich im Contract befindet. Sobald dieser Betrag der Zielgröße entspricht oder sie überschreitet, werden die Coins an die angegebene Adresse ausgezahlt. Soll der Contract aufgelöst werden, bevor das Ziel erreicht wird, so kann dies als Message an eine Transaktion angefügt und so dem Contract mitgeteilt werden. Erhält der Contract diese Nachricht, sendet er die erhaltenen Coins wieder an die entsprechenden Adressen zurück.

Damit wäre es Nutzern möglich, nicht nur in einer Gruppe zu chatten, sondern auch gemeinsam Geld zu sammeln. Dieses Geld kann für verschiedene Zwecke verwendet werden, wobei der Sparzweck Dritten gegenüber anonym bleibt. Dadurch, dass dies über einen Smart Contract abgewickelt wird, muss nicht einer dritten Partei vertraut werden, dass das Geld korrekt ausgezahlt wird. Der Smart Contract würde von der EVM stets so wie programmiert ausgeführt werden.

4 Dokumentation

4.1 Darstellung des Arbeitsprozesses

4.1.1 Erste Planung und Skizzen

Am Beginn der Entwicklung standen Überlegungen, wie die Anwendung aussehen soll und was sie leisten muss. Es wurde eine grobe Skizze der Anwendung angefertigt, die den Aufbau und die Funktionsweise der Anwendung verdeutlicht.

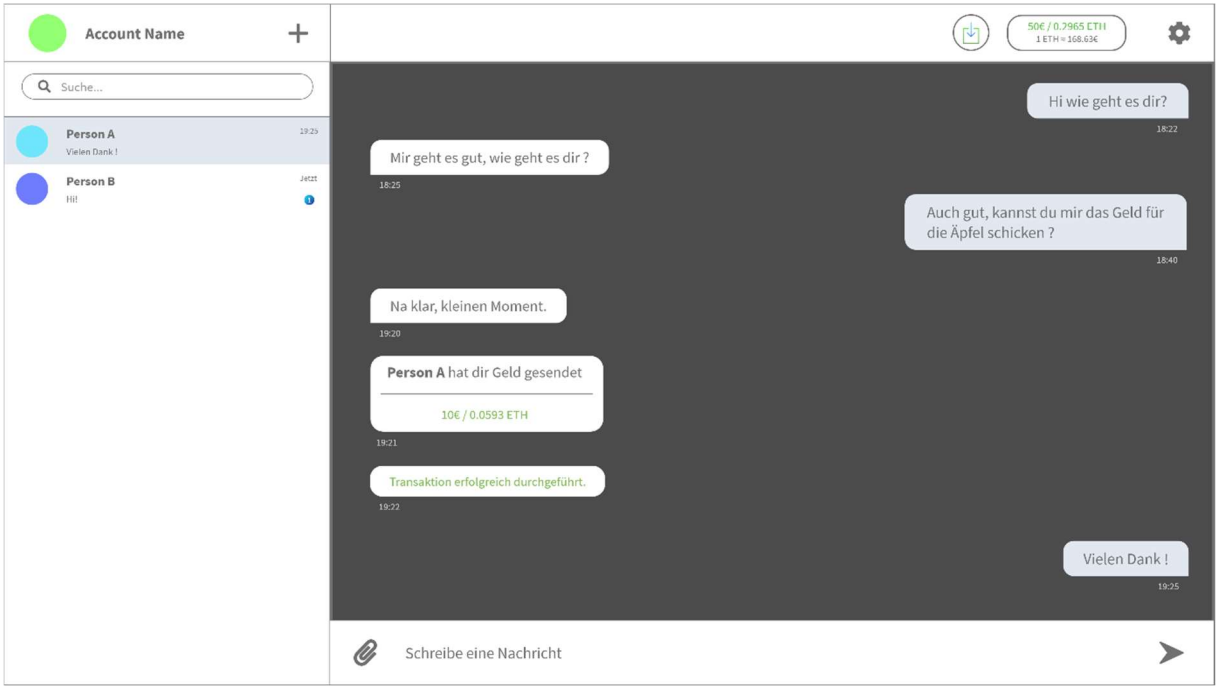

Abbildung 10: Erste Skizze der Anwendung

Abbildung 10 zeigt die Skizze der Messenger-Anwendung, in der es möglich sein soll, mit anderen Nutzern zu kommunizieren und, wie im Chatverlauf ersichtlich, Ether zu versenden. In dieser Skizze wird noch davon ausgegangen, dass es möglich ist, dem Nutzer eine eigene Wallet in der Anwendung bereitzustellen. So müsste der Nutzer sein Ether immer an die Wallet in der Anwendung senden, wenn er dort über sein Geld verfügen möchte. Dies ist jedoch für den Nutzer umständlich, da bei jeder Transaktion Gebühren anfallen und er sein Ether nicht in anderen Anwendungen nutzen kann. Darüber hinaus müsste die Anwendung die Private Keys verwalten und sicher aufbewahren. Dies könnte ein Sicherheitsrisiko für die Anwendung sein, da jemand mit den erbeuteten Keys über die Coins der Nutzer verfügen könnte (vgl. Kap. 2.1.1).

Da die Browser Wallet Metamask als besonders sicher und etabliert gilt und von vielen anderen dApps verwendet wird, wurde entschieden, diese im Browser als Extension zu installieren (vgl. Kap. 3.2.2). Dabei werden Private Keys im eigenen Browser gespeichert statt bei einer dritten Partei. Nutzer können hier nicht nur neue Coins an die Wallet senden, sondern auch bestehende Wallets mit dem Private Key importieren.

4.1.2 Entwicklung einer ersten Messenger-Anwendung

Anhand der zuvor angefertigten Planung und Skizze wurde dann eine Webanwendung programmiert, die den Zweck eines Messengers erfüllt (Weitere Abbildungen siehe Anhang).

Damit eine Webanwendung dynamisch auf Nutzereingaben reagieren kann, wird ein Webserver benötigt. Da Vorerfahrungen mit JavaScript und Node.js vorhanden sind und React.js eine weit verbreitete JavaScript-Bibliothek ist, wurde als Webserver Node.js in Verbindung mit React.js ausgewählt.

Es wurde mit der Programmierung einer einfachen Chat-Anwendung begonnen, um den Umgang mit einer Frontend-Bibliothek wie React.js zu erlernen und die Datenspeicherung in der Anwendung zu erproben. Um den Chat zu realisieren, wurden zwei Server programmiert: ein reiner Node.js-Server und ein React.js-Server. Der Node.js-Server kommunizierte nur mit der Datenbank, um Daten organisiert abzurufen und gebündelt an den React.js-Server zu leiten. Die Daten wurden mit Socket.IO zwischen den Servern ausgetauscht. Dieser hat dann die Daten verarbeitet und diese als Webseite an den Browser und damit an den Nutzer weitergegeben.

In dieser Version der Anwendung war es nur möglich, in einem einzelnen Chat-Raum mit allen Nutzern zu kommunizieren, da nur der Inhalt und der Absender einer Nachricht gespeichert wurden. So konnte die Anwendung Nachrichten zwar einem Nutzer zuordnen, aber nicht für wen diese bestimmt war.

4.1.3 Erweiterte Planungen

Die erste Version der Anwendung war eine gute Möglichkeit, sich an das Arbeiten mit einer Bibliothek wie React.js zu gewöhnen und den Aufbau einer Messenger-Anwendung kennenzulernen. Die bisher programmierte Anwendung erfüllte jedoch nicht das Ziel eines Eins-zu-eins-Chats mit anderen Nutzern, und Zahlungen über das

Ethereum-Netzwerk waren ebenfalls noch nicht möglich. Zudem war der Aufbau der Anwendung durch die zwei Server komplex und ineffizient.

Daher wurde ein neues Datenbanksystem modelliert, das nicht nur speichert, von wem eine Nachricht stammt, sondern auch, an wen diese gesendet werden soll. Zudem sollte dieses Modell auch die Ethereum-Adressen der Nutzer speichern, damit diese für Transaktionen genutzt werden konnten. Dieses Modell entspricht dem aktuell verwendeten (vgl. Kap. 3.2.3).

Um die Server-Struktur einfacher und effizienter zu gestalten, sollten nicht mehr zwei Server zum Einsatz kommen, welche getrennt voneinander das Backend und Frontend der Anwendung abbilden, sondern nur ein React.js Server, der sowohl Nutzereingaben als auch Daten aus der Datenbank verwalten kann. Damit der React.js-Server mit der Datenbank reibungslos und strukturiert kommunizieren kann, sollte Redux.js zum Einsatz kommen. Mit der Bibliothek sollte es möglich sein, innerhalb von React.js Daten aus der Datenbank abzufragen und organsiert an die einzelnen React.js Components weiterzugeben (vgl. Kap. 3.2.3).

4.1.4 Entwicklung einer privaten Messenger-Anwendung

Nach dem neuen Plan wurde eine zweite Version der Anwendung entwickelt. Da sich die Server-Struktur der neuen Anwendung erheblich von der ersten Version unterscheiden sollte, wurde ein neuer React.js-Server programmiert. Damit wurde auch ein Repository auf GitHub angelegt, in dem der Source Code gesichert und dokumentiert wird (vgl. Kap. A.1.1).

Für die neue Anwendung wurde als Datenbank *Google Firebase* eingesetzt, da hier besonders einfach Nutzer und Daten verwaltet werden können (vgl. Kap. 3.2.3). Da es nun mit dem neuen Datenbankensystem möglich war, Nachrichten dem Sender und Empfänger zuzuordnen, konnte in dieser Version der Anwendung eine Liste der Nutzer programmiert werden. Dies geschah in Anlehnung an *Abbildung 10* und ermöglicht dem Nutzer, einen Chat mit einem anderen Nutzer zu öffnen oder zu beginnen. Es war nun möglich, in einem Chat nicht nur die Nachrichten der beiden Nutzer anzuzeigen, sondern im Chat konnten diese auch nach Empfänger und Sender geordnet angezeigt werden.

Da der Nutzer auch Einstellungen seiner Login-Daten und später auch eine Wallet mit der Anwendung verbinden können sollte, wurde die Settings-Seite programmiert. Hier ist es dem Nutzer möglich, seine Nutzerdaten wie Nutzername, E-Mail und Passwort zu ändern.

4.1.5 Erweiterung der Anwendung durch die Metamask Wallet

Die Anwendung erlaubt es den Nutzern, untereinander zu kommunizieren. Mit der Metamask Wallet und der Web3.js API sollte nun ermöglicht werden, dass Nutzer sich über dezentrale Transaktionen gegenseitig Geld senden können.

Damit Nutzer die Metamask Wallet in der Anwendung verwenden können, müssen sie diese zuerst mit der Anwendung verbinden und der Anwendung Zugriff auf die Wallet gewähren. Dies soll der Nutzer in den Einstellungen festlegen können. Dazu wurde ein Button auf der Settings-Seite programmiert, mit dem die Anwendung eine Anfrage an die Metamask Extension sendet – sofern der Nutzer diese Extension im Browser installiert hat –, in der der Zugriff auf die Wallet erfragt wird. Stimmt der Nutzer dem zu, kann die Anwendung auf die Adressen der Wallet zugreifen und Anfragen für Transaktionen an die Wallet senden. Die aktuelle Adresse wird in der Datenbank gespeichert.

Da ein Nutzer seine Wallet mit der Anwendung verbinden kann und die Ethereum-Adresse in der Datenbank gespeichert wird, hat die Anwendung ausreichend Informationen, um eine Transaktion an die Metamask Wallet weiterzuleiten. Damit dies möglich ist, wurde ein weiteres Textfeld hinzugefügt, in das der Nutzer die zu versendende Menge Ether hineinschreiben kann.

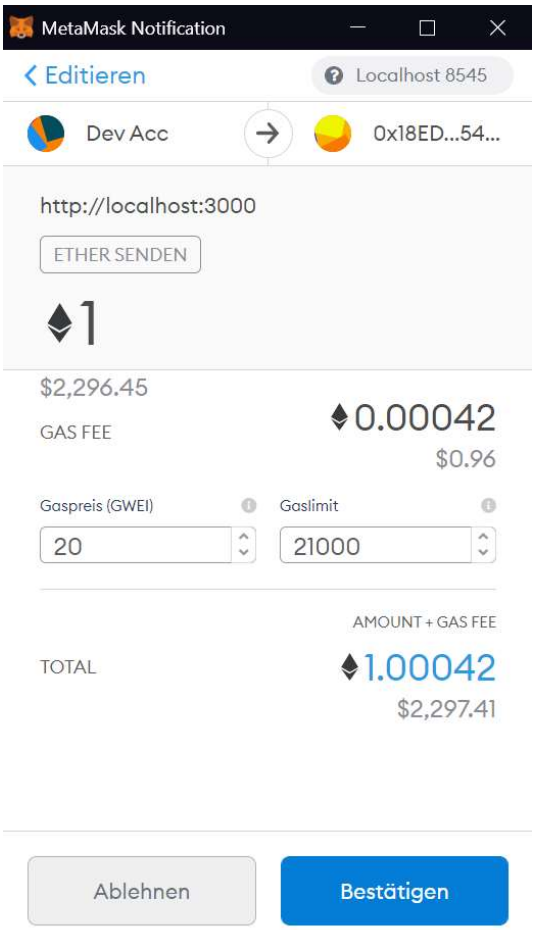

Abbildung 11: Metamask Transaktions Bestätigung

Klickt der Nutzer auf den Button „ETH“, öffnet sich ein Fenster der Metamask Extension, in der alle Details zur Transaktion dargestellt sind (vgl. *Abbildung 11*). Die Transaktionsgebühren kann der Nutzer selbst anpassen. Mit dem Button Bestätigen wird die Transaktion bestätigt und Metamask signiert diese mit dem Private Key. Die Transaktion kann zu diesem Zeitpunkt noch abgebrochen werden, indem der Nutzer auf den Button Ablehnen klickt. Der Server speichert die Transaktion als Nachricht in der Datenbank und zeigt sie im Chatverlauf an (vgl. Kap. 3.2.3).

5 Kritische Reflexion

5.1 Rahmenbedingungen der Arbeit

Zu Beginn der Arbeit waren Kryptowährungen kaum verbreitet und wurden von der Gesellschaft wenig wahrgenommen oder mit Zurückhaltung betrachtet. Diese war zum Beispiel darin begründet, dass Kryptowährungen mit illegalen Darknet-Aktivitäten in Zusammenhang gebracht wurden. Zudem führte der Crash der Krypto-Blase im Jahr 2017 dazu, dass Kryptowährungen als Investments mit sehr hohem Risiko angesehen wurden.

Seit Ende 2020 ist das Interesse an Kryptowährungen stark angestiegen [43]. Nachdem sie beim „Corona-Crash“ Anfang 2020 auf einen neuen Tiefstand gesunken waren, erreichen Kryptowährungen seit Ende 2020 immer wieder neue Allzeithochs [3]. Das Vertrauen der Bevölkerung in Kryptowährungen und die dahinter stehende Blockchain-Technologie ist erstarkt. Auch nach DeFi ist die Nachfrage stark gestiegen; zum Beispiel hat sich das Angebot an USDC auf Compound von Februar bis April 2021 fast verdreifacht [44], und die Liquidität auf Uniswap hat sich von Januar bis April 2021 von 2,16 Milliarden auf 8,27 Milliarden US-Dollar nahezu vervierfacht [45]. Diese Entwicklung bestätigt die Relevanz dezentraler Transaktionsmechanismen und deren Plattformen.

Auch bestätigt das hohe Handelsvolumen, das sich auf Uniswap zwischen ein und zwei Milliarden US-Dollar pro Tag [45] beläuft, dass dezentrale Anwendungen stabil und skalierbar sind. Einer globalen Transaktionsanwendung mit dezentralen Mechanismen muss daher eine hohe Bedeutung beigemessen werden, und eine hohe Nachfrage ist zu erwarten. Dies hat die Absicht, die vorliegende Anwendung zu entwickeln, bestätigt und bekräftigt.

Zudem haben sich die Potenziale und Funktionalitäten von dApps in Kombination mit der Metamask Wallet als sehr gut herausgestellt. So kann jeder Nutzer individuell bestimmen, wo und wie er seine Metamask Wallet einsetzen möchte. Die Anwendung hat keine Kontrolle über die Private Keys und kann somit auch nicht über die Coins der Nutzer verfügen. Der Nutzer kann also anonym entscheiden, wie und wo er sein Geld einsetzt. So kann er die Wallet, die er in der Anwendung verwendet, auch in anderen dApps wie Compound oder Uniswap nutzen. So kann der Nutzer seine Coins nicht nur

in der Anwendung versenden, sondern auch als Liquidität auf Uniswap bereitstellen oder auf Compound gegen Zinsen verleihen.

5.2 Herausforderungen während des Entwicklungsprozesses

Während die Anwendung entwickelt wurde, haben sich Situationen und Erkenntnisse ergeben, die den Entwicklungsprozess gebremst und beeinflusst haben.

So war die Bibliothek React.js. eine Herausforderung: Sie erleichterte zwar die Entwicklung einer dynamischen Webseite, dennoch gibt es deutliche Unterschiede im Gegensatz zu der Entwicklung mit reinem JavaScript und Node.js. React.js bietet die neue Syntax JSX, mit der sich die dynamischen Elemente einer Webseite entwickeln lassen. Diese Syntax besteht aus HTML- und JavaScript-Elementen (vgl. Kap. 3.2.1). Mit JSX ist es in React.js möglich, dass HTML-Elemente Funktionen und States einer React.js Component aufrufen zu können. Da der Webserver normalerweise nur ein HTML-Dokument statisch auf eine Webseite projiziert, musste die direkte Interaktion zwischen HTML und dem JavaScript Server erst erlernt werden.

Eine weitere Herausforderung war die Kommunikation über eine API mit der Datenbank oder der Metamask Wallet. Bei dieser Art von Kommunikation handelt es sich um asynchrone Vorgänge im Programm, in JavaScript als „Async-Functions“ bezeichnet. Diese asynchronen Vorgänge zeichnen sich dadurch aus, dass Folgeschritte zeitversetzt ausgeführt werden. Beispiel: Für eine Datenbankabfrage wird festgelegt, dass das Programm so lange wartet, bis Daten vorhanden sind, bevor diese verarbeitet werden. Wird im Programm nicht festgelegt, dass auf die Daten gewartet wird, führt das Programm die folgenden Befehle parallel aus, ohne dass Daten vorhanden sind [46]. Diese Art von JavaScript-Funktionen war zunächst zu erlernen.

Die Verwendung von Redux.js war ebenfalls ungewohnt. Redux.js arbeitet mit sogenannten „Reducern“, in denen die abgefragten Daten aus der Datenbank zusammengeführt und organisiert den Components zur Verfügung gestellt werden. Dies vereinfacht zwar später den Zugriff und die Übersichtlichkeit über die Daten aus der Datenbank, ist jedoch etwas aufwändiger und ungewohnt zu programmieren.

Die größte Herausforderung während des Entwicklungsprozesses war die Integrierung von Ethereum. Zu Beginn der Entwicklung war unklar, ob es möglich ist, Nachrichten und Nutzerdaten zentral oder dezentral auf der Blockchain zu speichern. Nach

ausführlicher Prüfung der Technologie und der Ziele der Anwendung musste die dezentrale Datenspeicherung ausgeschlossen werden. Zudem liegen Dokumentationen für die Entwicklung einer dApp nur in geringem Maße vor, und nur wenige Entwickler hatten sich zuvor mit dieser Art von Programmieren auseinandergesetzt. So war es schwierig, herauszufinden, wie eine dezentrale Anwendung entwickelt werden könnte, und geeignete Lösungen für Probleme und Fehler zu finden.

6 Wertende Zusammenfassung des Ergebnisses

Kryptowährungen sind seit 2009 mit Bitcoin in Verwendung und erlangen immer größere Beliebtheit. Sie haben die Art der Bezahlung über das Internet mit dezentralen Netzwerken und kryptografischen Verfahren stark verändert und bieten den Menschen neue Möglichkeiten, Geld online zu nutzen und darüber zu verfügen. Darüber hinaus sind Kryptowährungen verglichen mit konventionellen Geldsystemen globaler, freier, demokratischer und ökonomischer.

Neuere Kryptowährungen nutzen die für Bitcoin entwickelte Blockchain-Technologie, um Probleme zu lösen oder neue Funktionen zu erschaffen. So lassen sich für die Kryptowährung Ethereum mit Blockchain dezentrale Anwendungen programmieren, die gleiche oder neue Funktionalitäten bieten wie bisherige Finanzdienstleistungen. Diese Anwendungen werden als DeFi bezeichnet und stellen einen Gegenentwurf zum traditionellen, zentralen Finanzsystem, Centralised Finance (CeFi) genannt, dar. Bei den neuartigen dezentralen Finanzanwendungen ist eine dritte Partei, die Dienstleistungen bereitstellt und Finanzflüsse kontrolliert, überflüssig. Das spart verglichen mit dem traditionellen Finanzsystem Aufwand und Energie und macht Finanztransaktionen anonymer.

Durch ihren privateren, günstigeren und freieren Charakter bieten sich Kryptowährungen an, um eine Anwendung zu entwickeln, die es Menschen nicht nur ermöglicht, sich global zu verknüpfen, sondern auch global Geld zu versenden. Ethereum als größte Kryptowährungsplattform und Ether als zweitgrößte Kryptowährung sind sehr gut für die Entwicklung einer solchen Anwendung geeignet. In Kombination mit der Browser Wallet Metamask lässt sich Ethereum benutzerfreundlich in die Anwendung Integrieren, so dass Nutzer an andere Nutzer Coins senden können, dabei aber stets die Kontrolle über ihre Coins behalten.

Die Anwendung soll benutzerfreundlich sein und daher kein Vorwissen über Kryptowährungen und Ethereum voraussetzen. Lediglich für die Nutzung von Ether in der Anwendung muss der Nutzer eine Wallet bedienen können. Damit dies möglich ist, werden Nutzer- und Chatdaten im JSON-Format zentral auf der Datenbank *Google Firebase* gespeichert. Die Anwendung läuft auf einem React.js-Server, damit sie dynamisch auf Nutzereingaben reagieren und diese verarbeiten kann.

Die Anwendung besteht aus einer Chat-Seite, auf der Nutzer nach Registrierung und Login in einem Eins-zu-eins-Chat miteinander chatten können. In den Einstellungen auf der Settings-Seite können Nutzer Ihre Login-Daten ändern, aber auch eine Metamask Wallet mit der Anwendung verknüpfen. Hat der Chat-Partner ebenfalls eine Wallet mit seiner Anwendung verknüpft, können sich beide Ether über die Ethereum-Blockchain senden. Die Transaktionen werden dezentral und durch die Blockchain-Technologie sicher ausgeführt.

Mit neuesten Entwicklungen aus dem Bereich der Kryptowährungen lässt sich die Anwendung noch verbessern oder erweitern. Mit IPFS könnte es möglich sein, die Teile der Anwendung, die bislang zentral betrieben werden, auch dezentral zu speichern und auszuführen. Im Filecoin-Projekt wird versucht, mit dem Proof-of-Replication-Verfahren Probleme, die auf IPFS auftreten, zu lösen. Auch davon könnte das Bestreben, eine in jeder Hinsicht dezentrale Anwendung zu entwickeln, profitieren.

Die Ethereum-Plattform bietet die Möglichkeit, mit Smart Contracts direkt auf der Blockchain zu programmieren. Diese Technologie bietet mit dem ERC-20 Standard die Möglichkeit, eigene Tokens auf der Ethereum-Blockchain zu programmieren. Diese können zum Beispiel herkömmliche Währungen wie den US-Dollar abbilden. Die Tokens können in die Anwendung integriert werden und so dem Nutzer ermöglichen, auch andere Währungen an andere Nutzer zu senden.

Die Smart Contract-Technologie kann ebenfalls in der Anwendung verwendet werden, um es in einem Gruppenchat zu ermöglichen, dass mehrere Gruppenmitglieder auf ein Ziel sparen können. Auch dies könnte dezentral ohne eine dritte Partei, die das gesparte Geld verwaltet, realisiert werden.

Mit der Anwendung lässt sich sicher und einfach, unabhängig und effizient Geld an andere Menschen versenden, die die Anwendung ebenfalls nutzen. Dabei sind sowohl die Anwendung als auch die verwendete Währung neutral gegenüber Religion, Nationalität, kultureller Herkunft oder sozialem Status. Anders als bei traditionellen Finanzsystemen finden Transaktionen dezentral statt – sie lassen sich von Dritten, zum Beispiel Behörden, kaum kontrollieren. Neue Technologien bergen noch viel Potenzial, um die Anwendung zu verbessern und weiterzuentwickeln.

Abbildungsverzeichnis

Literaturverzeichnis

[1] *CoinMarketCap,* „What Is a Cryptocurrency?", 2021
https://coinmarketcap.com/alexandria/glossary/cryptocurrency
(zuletzt abgerufen am 23.03.2021)

[2] *Chaum, David,* „Blind signatures for untraceable payments", Boston Springer-Verlag 1983, S. 199-203
https://www.chaum.com/publications/Chaum-blind-signatures.PDF (zuletzt abgerufen am 23.03.2021)

[3] *CoinMarketCap,* „Today's Cryptocurrency Prices by Market Cap", 2021
https://coinmarketcap.com/ (zuletzt abgerufen am 29.03.2021)

[4] *Nakamoto, Satoshi,* „Bitcoin: A Peer-to-Peer Electronic Cash System", 2008
https://bitcoin.org/bitcoin.pdf (zuletzt abgerufen am 29.03.2021)

[5] *CoinMarketCap,* „What Is a Cryptographic Hash Function?", 2021
https://coinmarketcap.com/alexandria/glossary/cryptographic-hash-function
(zuletzt abgerufen am 31.03.2021)

[6] *Buterin, Vitalik,* „A Next-Generation Smart Contract and Decentralized Application Platform", 2013
https://ethereum.org/de/whitepaper/ (zuletzt abgerufen am 29.03.2021)

[7] *Beigel, Ofir,* „51% Attack Explained – a Beginner's Guide", 99Bitcoins 2020
https://99bitcoins.com/51-percent-attack/
(zuletzt abgerufen am am 30.03.2021)

[8] *Buterin, Vitalik,* „Understanding the Ethereum Blockchain Protocol", London 2015, min. 2:00–4:00
https://archive.devcon.org/devcon-1/details/
(zuletzt abgerufen am am 30.03.2021)

[9] *BitInfoCharts,* „Ethereum (ETH) statistiken und informationen", 2021
https://bitinfocharts.com/de/ethereum/ (zuletzt abgerufen am am 30.03.2021)

[10] *Ward, Chris,* „Ethash", Ethereum Wiki 2020
https://eth.wiki/en/concepts/ethash/ethash
(zuletzt abgerufen am am 30.03.2021)

[11] *Beregszászi, Alex,* „Ethereum Virtual Machine (EVM)", Ethereum.org 2021
https://ethereum.org/en/developers/docs/evm/
(zuletzt abgerufen am am 30.03.2021)

[12] *Wood, Gavin,* „Ethereum: A secure decentralised generalised transaction ledger", Petersburg Version 41c1837 2021
https://github.com/ethereum/yellowpaper
(zuletzt abgerufen am am 03.04.2021)

[13] *Ethereum.org,* „A digital future on a global scale", 2021
https://ethereum.org/en/eth2/vision/ (zuletzt abgerufen am am 02.04.2021)

[14] *Dog, Decentralized,* „A Dive Into Ethereum 2.0", CoinMarketCap 2020 https://coinmarketcap.com/alexandria/article/a-dive-into-ethereum-2-0 (zuletzt abgerufen am am 02.04.2021)

[15] *Statista Global Consumer Survey,* „Wie verbreitet ist Crypto-Währung", 2020 https://de.statista.com/infografik/22561/anteil-der-krypto-nutzer-in-ausgewaehlten-laendern/ (zuletzt abgerufen am am 03.04.2021)

[16] *IDSA,* „International Data Spaces (IDS)", International Data Spaces Association Dortmund 2019 https://internationaldataspaces.org/wp-content/uploads/IDSA-Brochure-IDS-Standard-for-Data-Sovereignty-Indispensible-Element-for-Data-Ecosystems.pdf (zuletzt abgerufen am 15.04.2021)

[17] *Achatz, Reinhold,* „Blockchain Technology in IDS", International Data Spaces Association Dortmund 2019 https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Position-Paper-Blockchain-Technology-in-IDS.pdf (zuletzt abgerufen am 15.04.2021)

[18] *CoinMarketCap,* „What Is Decentralized Finance (DeFi)?", 2021 https://coinmarketcap.com/alexandria/glossary/defi (zuletzt abgerufen am 03.04.2021)

[19] *Uniswap,* „Swap", Uniswap 2021 https://app.uniswap.org/#/swap (zuletzt abgerufen am 17.04.2021)

[20] *Compound,* „Dashboard", Compound Labs 2021 https://app.compound.finance/ (zuletzt abgerufen am 17.04.2021)

[21] *GlassnodeStudio,* „Bitcoin: Number of Active Addresses", Glassnode 2021 https://studio.glassnode.com/metrics?a=BTC&category=Addresses&m=addresses.ActiveCount (zuletzt abgerufen am 04.04.2021)

[22] *GlassnodeStudio,* „Bitcoin: Number of Addresses with a Non-Zero Balance", Glassnode 2021 https://studio.glassnode.com/metrics?a=BTC&category=Addresses&m=addresses.NonZeroCount (zuletzt abgerufen am 04.04.2021)

[23] *PayPal Inc.,* „Vorzugskonditionen für gewerbliche Verkäufer", 2021 https://www.paypal.com/de/webapps/mpp/merchant-fees (zuletzt abgerufen am 06.04.2021)

[24] *PayPal Inc.,* „PayPal-Gebühren für Privatkunden", 2021 https://www.paypal.com/de/webapps/mpp/paypal-fees (zuletzt abgerufen am 06.04.2021)

[25] *Devereux, Charlie; Wild, Franz; Robinson, Edward:* „The Biggest Digital Heist in History Isn't Over Yet", Bloomberg L.P. 2018 https://www.bloomberg.com/news/features/2018-06-25/the-biggest-digital-heist-in-history-isn-t-over-yet (zuletzt abgerufen am 05.04.2021)

[26] *Antonopoulos, Andreas M.*, „Andreas M. Antonopoulos über Bitcoin, Datenschutz, Menschenrechte und die Zukunft von Bitcoin“, Anita Posch 2018 https://bitcoinundco.com/de/andreas-antonopoulos-bitcoin-zukunft-podcast/ (zuletzt abgerufen am 06.04.2021)

[27] *Kuhlmann, Max,* „Bitcoin anonym aufbewahren: Wasabi-Wallet verspricht mehr Privacy“, BTC ECHO 2018 https://www.btc-echo.de/bitcoin-anonym-aufbewahren-wasabi-wallet-verspricht-mehr-privacy/ (zuletzt abgerufen am 06.04.2021)

[28] *Demirgüç-Kunt, Asli et al.*, „The Global Findex Database”, World Bank Group 2017 https://openknowledge.worldbank.org/bitstream/handle/10986/29510/211259ov.pdf (zuletzt abgerufen am 18.04.2021)

[29] *Clark, Emily,* „Cryptocurrency Statistics: What Are The Myths & Realities?”, Visual Objects 2020 https://visualobjects.com/web-development/understanding-cryptocurrency-myths-realities (zuletzt abgerufen am 06.04.2021)

[30] *Cambridge Centre for Alternative Finance,* „Cambridge Bitcoin Electricity Consumption Index”, 2021 https://cbeci.org/ (zuletzt abgerufen am 06.04.2021)

[31] *LaenderDaten,* „Stromverbrauch“, Lexas 2015 https://www.laenderdaten.de/energiewirtschaft/elektrische_energie/stromverbrauch.aspx (zuletzt abgerufen am 06.04.2021)

[32] *Bain, Tyler,* „Introducing CBEI: A new way to measure Bitcoin Network electrical consumption”, Bitcoin Magazine 2020 https://bitcoinmagazine.com/business/introducing-cbei-a-new-way-to-measure-bitcoin-network-electrical-consumption (zuletzt abgerufen am 06.04.2021)

[33] *Gallersdörfer, Ulrich,* „Experten-Interview: „Das Bitcoin-Netzwerk emittiert so viel CO2 wie Las Vegas oder ganze Staaten““, Kryptoszene 2020 https://kryptoszene.de/news/experten-interview-das-bitcoin-netzwerk-emittiert-so-viel-co2-wie-las-vegas-oder-ganze-staaten/ (zuletzt abgerufen am 06.04.2021)

[34] *Antonopoulos, Andreas M.*, „Bitcoin: Revolution des Geldsystems oder digitales Gold?“, Finanzfluss 2021, min. 7:00–19:00 https://www.youtube.com/watch?v=QiEz1Aw8BEk (zuletzt abgerufen am 06.04.2021)

[35] *Schär, Fabian,* „Krypto-Experte Schär: «DeFi ist höchst interessant – aber für Spekulanten extrem riskant»“, Watson 2021 https://www.watson.ch/wirtschaft/wissen/813917630-krypto-experte-schaer-defi-ist-hoechst-interessant-aber-extrem-riskant (zuletzt abgerufen am 06.04.2021)

[36] *Json.org,* „Introducing JSON“
https://www.json.org/json-en.html (zuletzt abgerufen am 09.04.2021)

[37] *Abramov, Dan,* „Redux Essentials, Part 1: Redux Overview and Concepts”, Redux.js.org 2021
https://redux.js.org/tutorials/essentials/part-1-overview-concepts
(zuletzt abgerufen am 09.04.2021)

[38] *Kaspersky Team,* „Was eine Ende-zu-Ende-Verschlüsselung ist und warum Sie eine benötigen“, AO Kaspersky Lab. 2020
https://www.kaspersky.de/blog/what-is-end-to-end-encryption/25190/
(zuletzt abgerufen am 09.04.2021)

[39] *Beuth, Patrick,* „Signal testet Bezahlfunktion mit Kryptowährung“, Der Spiegel Hamburg 2021
https://www.spiegel.de/netzwelt/apps/messenger-app-signal-testet-bezahlfunktion-mit-kryptowaehrung-a-f03e2d66-be58-44fd-b0fd-acfe697078b2
(zuletzt abgerufen am 10.04.2021)

[40] *Benet, Juan,* „IPFS - Content Addressed, Versioned, P2P File System”, 2014
https://ipfs.io/ipfs/QmV9tSDx9UiPeWExXEeH6aoDvmihvx6jD5eLb4jbTaKGps
(zuletzt abgerufen am 09.04.2021)

[41] *Benet, Juan; Daylrymple, David; Greco, Nicola:* „Proof of Replication”, 2017
https://research.filecoin.io/papers (zuletzt abgerufen am 10.04.2021)

[42] *Slyghtlyfloating,* „ERC-20 Token Standard“, Ethereum.org 2021
https://ethereum.org/en/developers/docs/standards/tokens/erc-20/
(zuletzt abgerufen am 10.04.2021)

[43] *Google Trends,* „Suchtrend von Kryptowährung, Bitcoin und DeFi“, Google Ireland Limited Irland 2021
https://trends.google.de/trends/explore?date=today%205-y&q=%2Fm%2F0vpj4_b,%2Fm%2F05p0rrx,DeFi
(zuletzt abgerufen am 16.04.2021)

[44] *Compound.finance,* „USD Coin“, Compound Labs 2021
https://compound.finance/markets/USDC (zuletzt abgerufen am 16.04.2021)

[45] *Uniswap,* „Uniswap Analytics“, Uniswap 2021
https://info.uniswap.org/home (zuletzt abgerufen am 16.04.2021)

[46] *MDN Web Docs,* „Async function”, Mozilla 2021
https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Statements/async_function (zuletzt abgerufen am 17.04.2021)

Anhang

A.1 Source Code

A.1.1 Repository

Der Source Code der Anwendung befindet sich auf GitHub. Dort sind die einzelnen Produktionsschritte der finalen Version der Anwendung durch das Version Control System „Git“ abgespeichert.

Zu finden ist das Repository unter https://github.com/Jufg/Chat-App-ETH.

A.1.2 Hilfsmittel

Node.js Dependencies:

- React.js:
  https://reactjs.org/
- Firebase:
  https://www.npmjs.com/package/firebase
- Fontawesome:
  https://fontawesome.com/how-to-use/on-the-web/using-with/react
- Redux:
  https://redux.js.org/
- React-Redux:
  https://www.npmjs.com/package/react-redux
- Redux-thunk:
  https://www.npmjs.com/package/redux-thunk
- Web3
  https://github.com/ChainSafe/web3.js

Programme:

- JetBrains WebStorm:
  https://www.jetbrains.com/webstorm/
- Adobe XD:
  https://www.adobe.com/de/products/xd.html
- Ganache:
  https://www.trufflesuite.com/ganach

A.2 Grafiken zur Anwendung

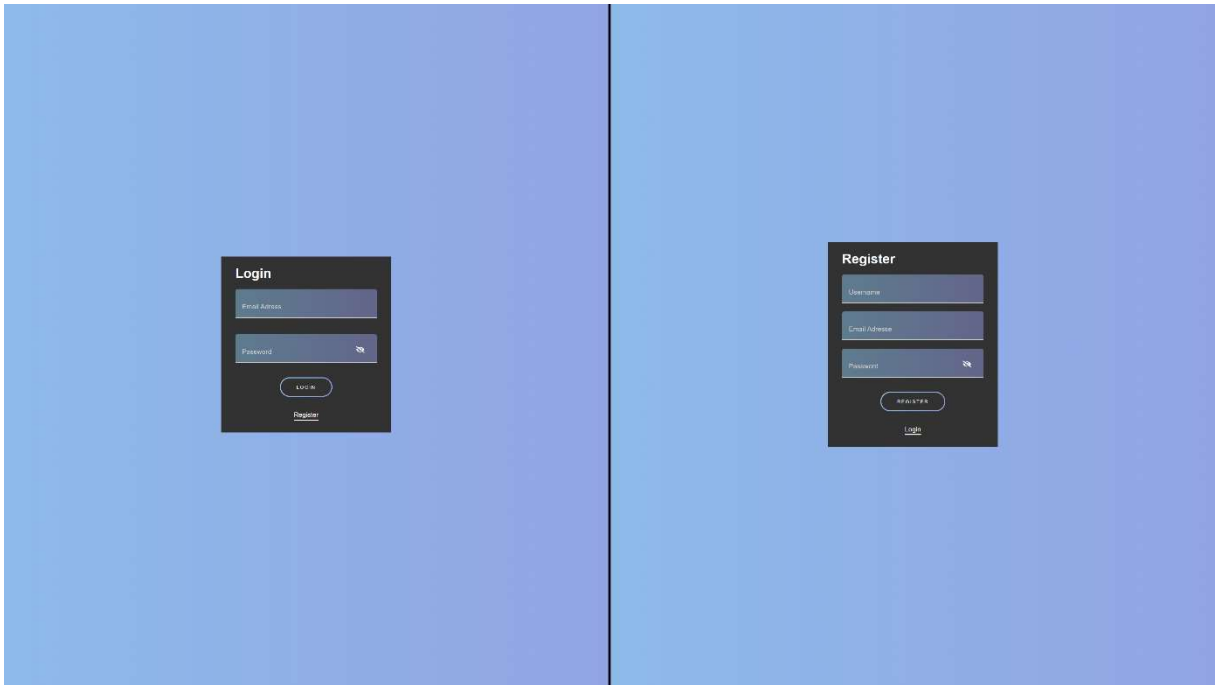

Abbildung A.12: Login/Register-Seite

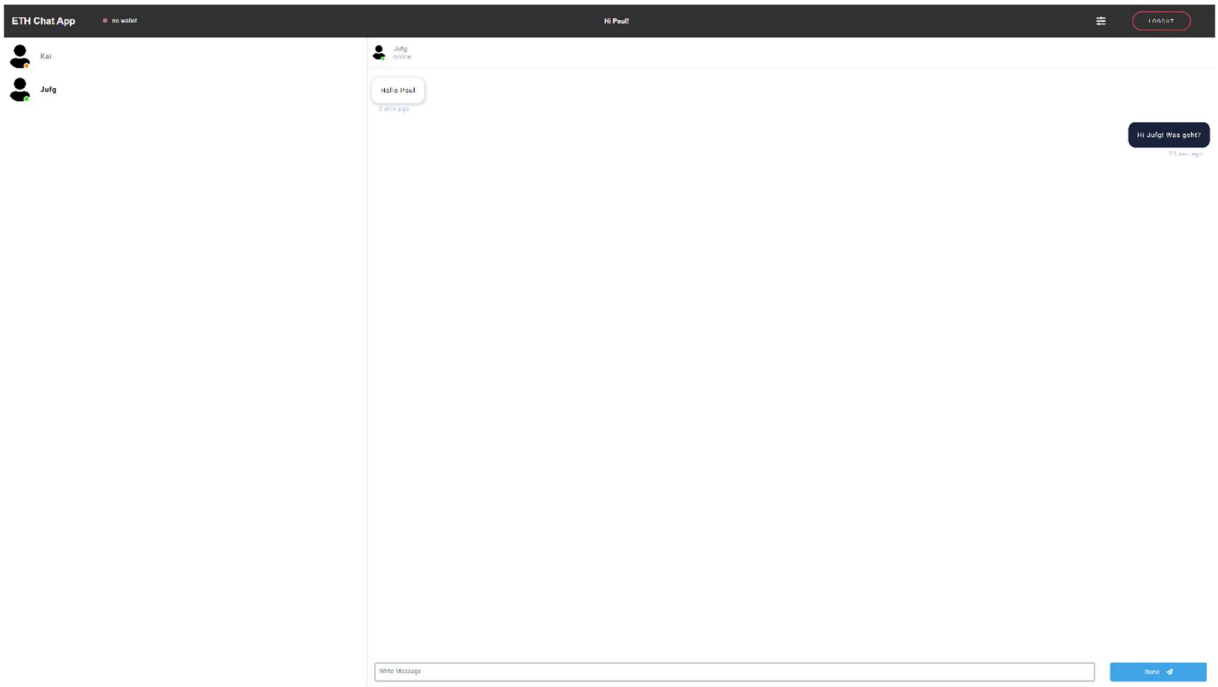

Abbildung A.13: Chat in der Anwendung ohne Wallet

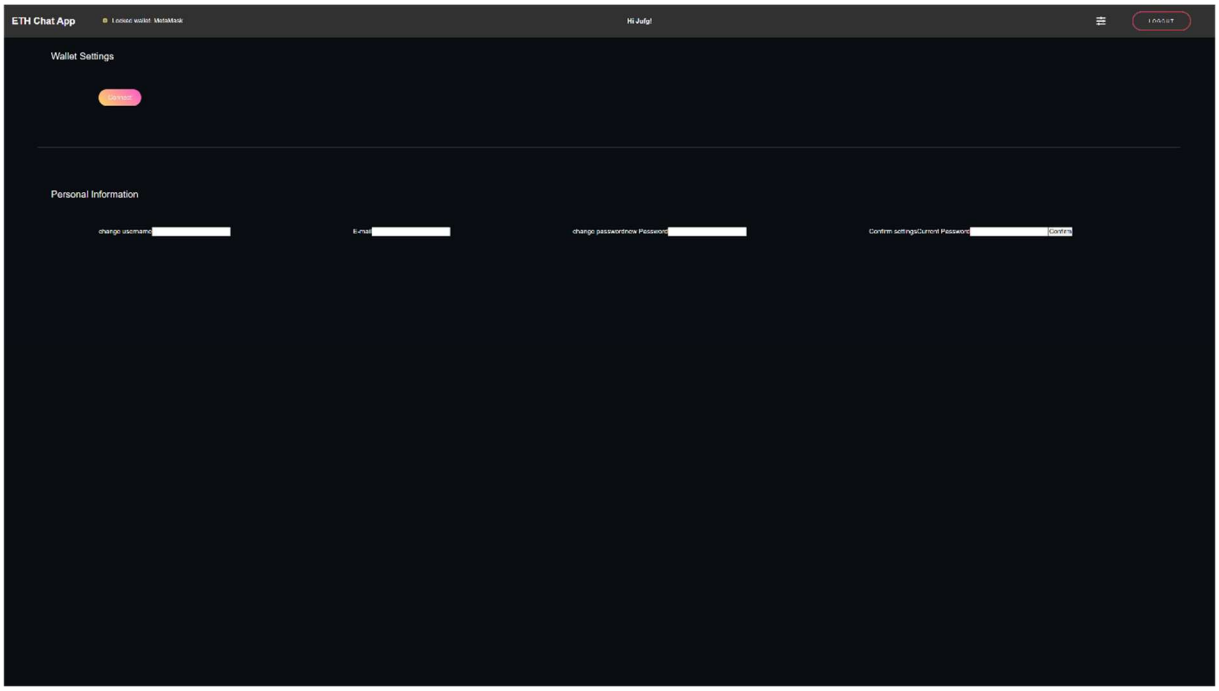

Abbildung A.14: Settings-Seite

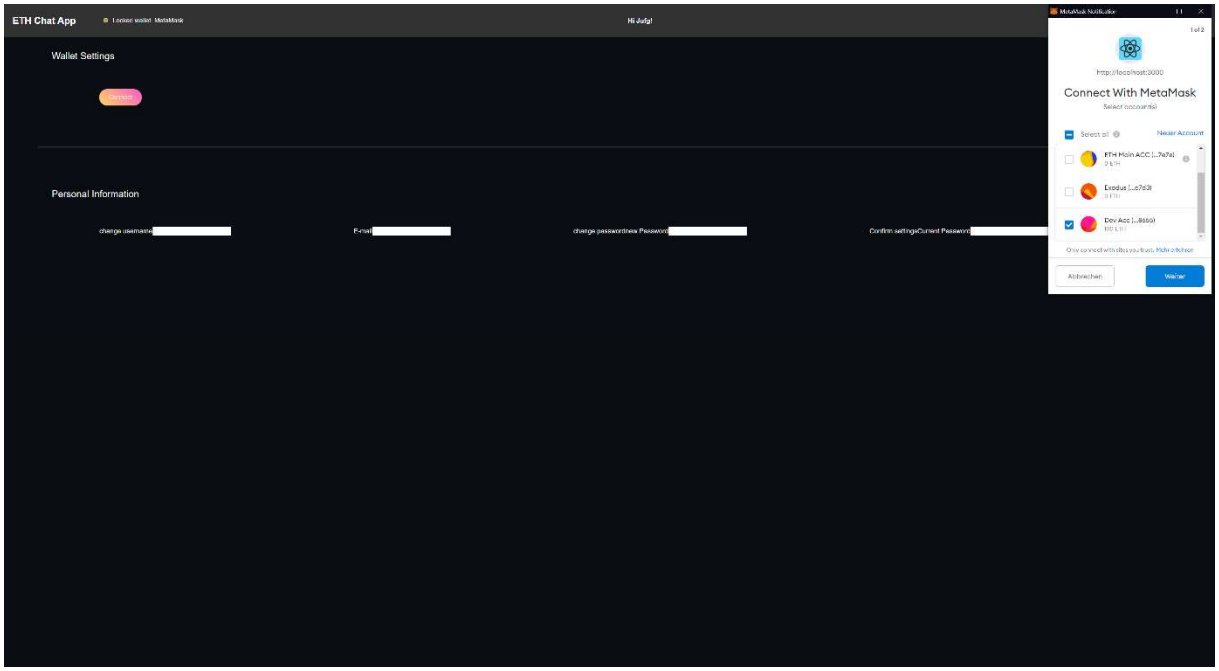

Abbildung A.15: Wallet verknüpfen, Schritt 1

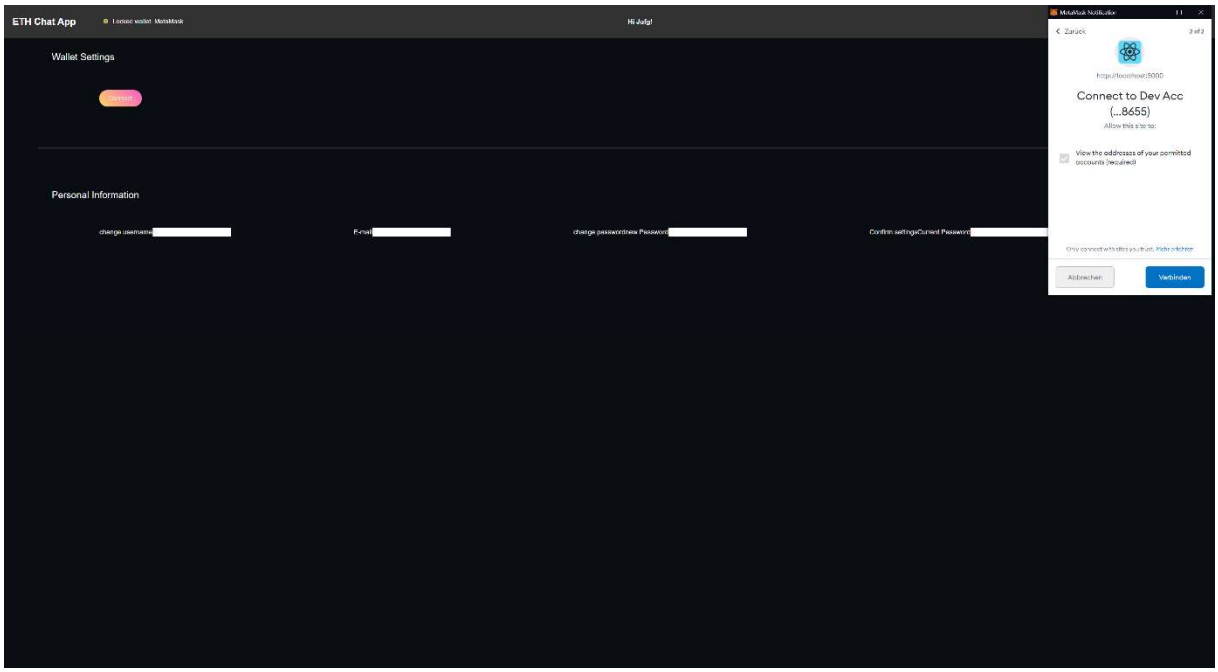

Abbildung A.16: Wallet verknüpfen, Schritt 2

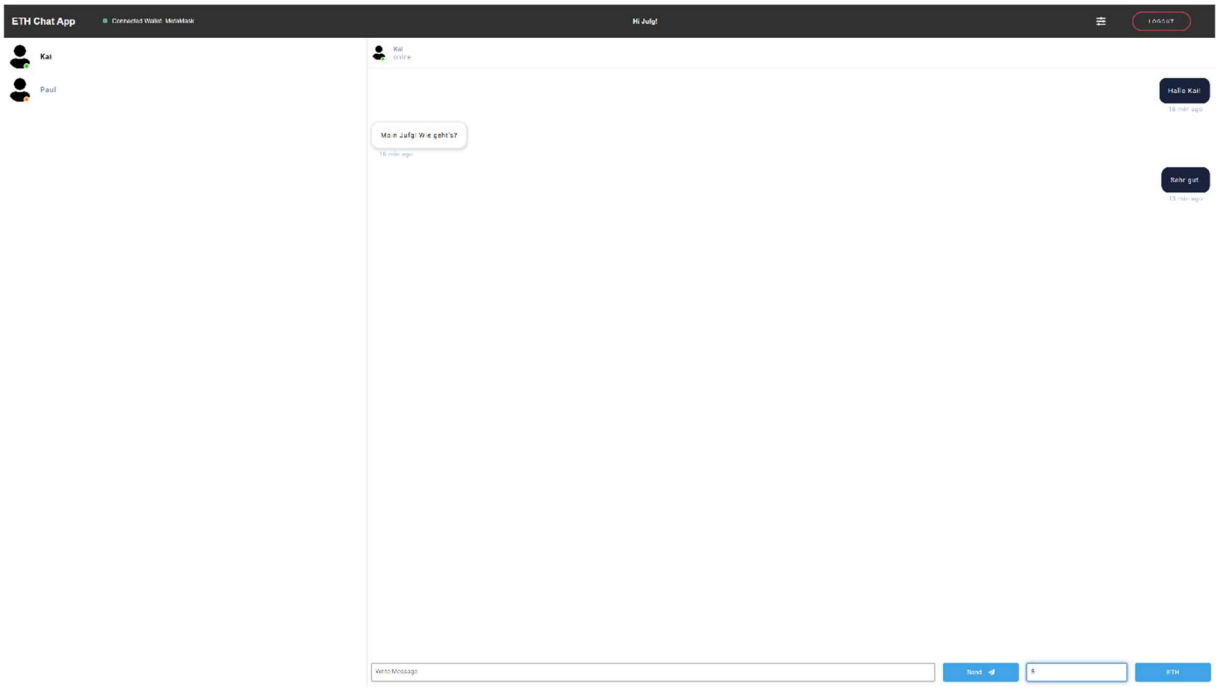

Abbildung A.17: Chat mit verknüpfter Wallet

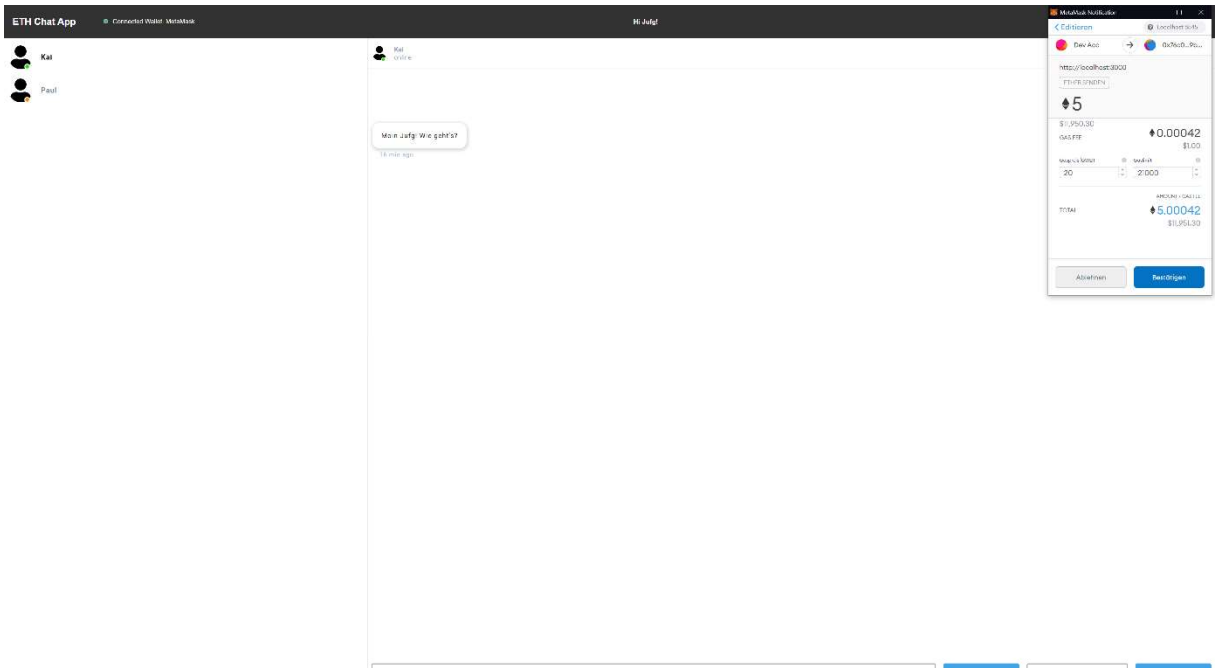

Abbildung A.18: Transaktion an anderen Nutzer

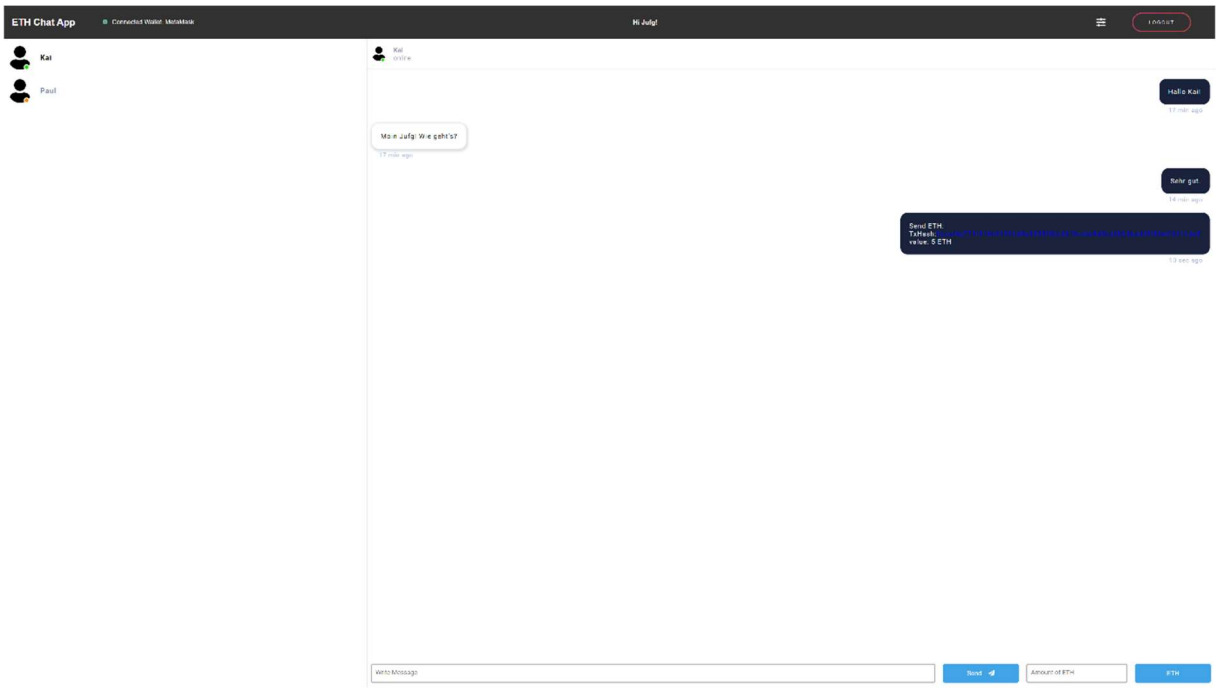

Abbildung A.19: Chat nachdem die Transaktion ausgeführt wurde

Über den folgenden Link lassen sich alle in dieser Arbeit verwendeten Grafiken in voller Auflösung aufrufen und herunterladen:

- https://ipfs.io/ipfs/QmZ4DCyanzBi9fiGVLGghKjAXjUk6Z3m46dKbEEEZAqTV2

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Abschlussarbeit selbstständig und ohne Inanspruchnahme fremder Hilfe angefertigt habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder inhaltlich entnommenen Stellen kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Sundern, den 22.04.2021

Juri Kembügler