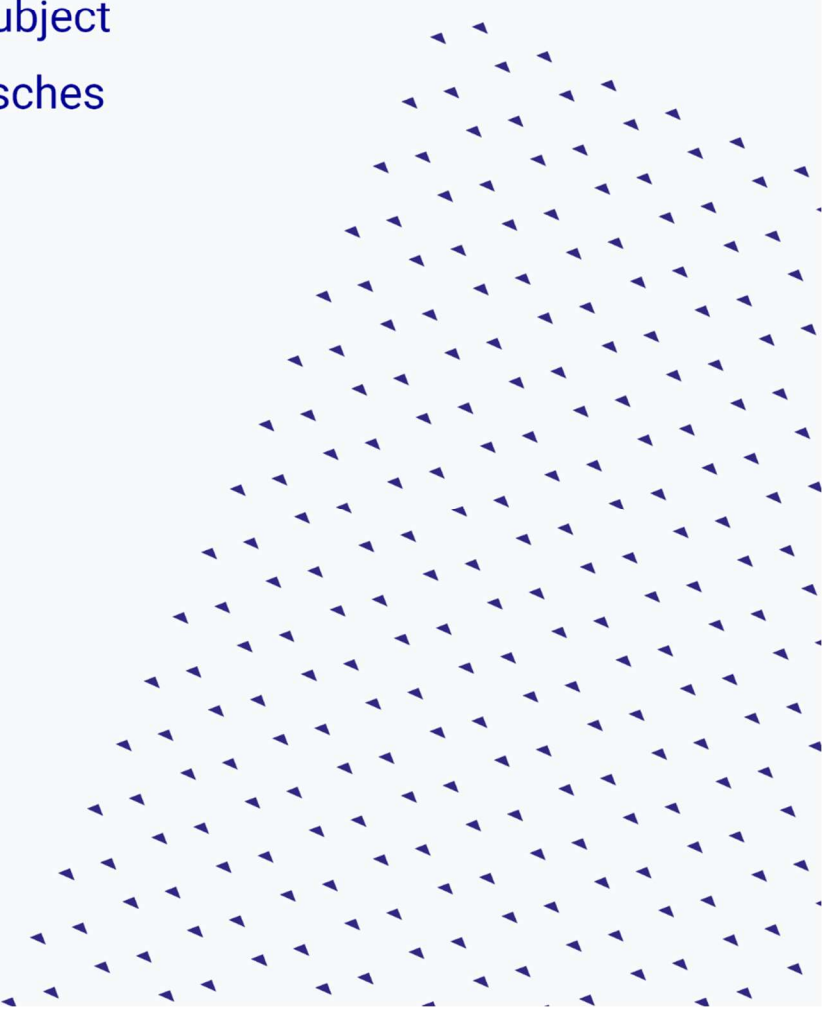




# Ethereum Chat-App

A special learning achievement on the topic "Decentralised Networks and Decentralised Finance" in the subject computer science at the Städtisches Gymnasium Sundern

8. Juni 2020 bis 22. April 2021



## Table of Contents

List of abbreviations .....	IV
1 Introduction .....	1
2 Initial position .....	2
2.1 Cryptocurrencies.....	2
2.1.1 Development of cryptocurrencies .....	2
2.1.2 How transactions work in a blockchain.....	2
2.1.3 Ethereum and Smart Contracts .....	6
2.1.4 Potential and current purpose of cryptocurrencies .....	8
2.2 Weaknesses of current transaction platforms.....	9
2.3 Current state of research .....	11
2.4 Decentralised transaction mechanisms as a solution approach .....	13
3 The development of an application for decentralised transactions .....	15
3.1 Objectives and methodology .....	15
3.1.1 Functions and purpose of the application.....	15
3.1.2 Realisation.....	15
3.2 Structure and functioning of the application.....	16
3.2.1 Communication with the user – React.js .....	17
3.2.2 Communication with Ethereum – Metamask .....	18
3.2.3 Data storage and processing – Firebase .....	20
3.3 Conventional and innovative elements of the application .....	22
3.4 Outlook .....	23
3.4.1 Decentralised data storage with IPFS.....	23
3.4.2 ERC-20 token integration .....	25
3.4.3 Group chats with smart contracts.....	25
4 Documentation.....	27
4.1 Representation of the work process .....	27
4.1.1 First planning and sketches.....	27
4.1.2 Development of a first messenger application .....	28
4.1.3 Extended planning .....	28
4.1.4 Development of a private messenger application .....	29
4.1.5 Extending the application with the Metamask Wallet .....	30
5 Critical reflection .....	32
5.1 General conditions of the paper .....	32

5.2	Challenges during the development process .....	33
6	Evaluative summary of the result .....	34
	List of figures .....	V
	Bibliography.....	VI
	Appendix.....	X

## List of abbreviations

API:	Application Programming Interface
CeFi:	Centralised Finance
DAO:	Decentralised Autonomous Organisation
dApps:	Decentralised Applications
DeFi:	Decentralised Finance
EVM:	Ethereum Virtual Machine
GWEL:	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 Introduction

In recent decades, the world has developed into a digital service society in which almost everything is done online. Also Payments are mostly done on the Internet. This even happens when paying in retail stores with a credit or girocard. In many places, it is common to pay with a smartphone or smartwatch.

In order to execute a payment, third party service providers are required. These service providers can be banks or online payment services such as *PayPal* and *Sofortüberweisung*. In the case of payment by smartphone or smartwatch, large companies like *Google* or *Apple* stand in between as intermediaries. The responsibility for the transaction from one party to another is therefore transferred to a third party, which receives all the information necessary or available to execute the transaction. This information contains not only data on the accounts between which the money is to be transferred, but also of the persons to which the accounts can be assigned, and in some cases, the purpose for which the transaction is to take place. In this way, the third-party knows who the holder of the card or device is, in which supermarket he spends how much money and in some cases, what he has bought.

Consumers not only need to trust that service providers will execute transactions correctly, it must also be taken into account that payment data will be shared, for example with advertisers or for credit rating. A completely anonymous way of transferring money, as it is the case with analogue money, so cash, cannot be guaranteed with our current financial system for immaterial transactions.

Decentralised cryptocurrencies offer a solution. Decentralised infrastructures allow transactions without a third party and offer anonymity.

The basis of the present paper is the development of an application that makes it possible to get in touch with other people, to network and communicate. The linked people can then use the application to make anonymous payments to each other based on a cryptocurrency.

## **2 Initial position**

### **2.1 Cryptocurrencies**

#### **2.1.1 Development of cryptocurrencies**

As explained in [1], a cryptocurrency can be described as a digital currency secured by cryptography. It is intended to be an independent, decentralised and secure alternative to our current payment system.

The first cryptocurrency eCash was developed in 1983 by the American cryptographer David Chaum. As he describes in [2], eCash was supposed to make it possible to execute transactions between two parties digitally. The new cryptographic technique "blind signatures" was to be used for this. The currency did not succeed; the company behind it, DigiCash, had to file for bankruptcy in 1998.

The first cryptocurrency which achieved mainstream recognition was Bitcoin in 2009. It was developed during the financial crisis of 2008 by the pseudonym Satoshi Nakamoto and, together with blockchain technology, it represents a fundamental innovation in the development of cryptocurrencies. Based on Bitcoin, many new cryptocurrencies have developed since 2008. According to [3], more than 4,500 cryptocurrencies are now listed on CoinMarketCap.

Among the many digital currencies is the cryptocurrency Ether. It is the currency on the decentralised platform Ethereum.

#### **2.1.2 How transactions work in a blockchain**

As described in [1], a blockchain is to be understood as a decentralised, cryptographically secured "directory" of all transactions made in the network. This directory is called "ledger". The unique aspect is that the transactions stored in the blockchain can be viewed publicly by anyone, but they cannot be assigned to a person. This means that the Bitcoin network, for example, is anonymous.

In order for a user to be able to execute a Bitcoin transaction, he needs a cryptographically generated key pair. This consists of the "Public Key" and the "Private Key". Both keys consist of a sequence of numbers and letters. The public key is public and can be seen by anyone, while the private key is private and only known by the

owner. From the private key several associated public keys can be generated, so that one account can have several different addresses.

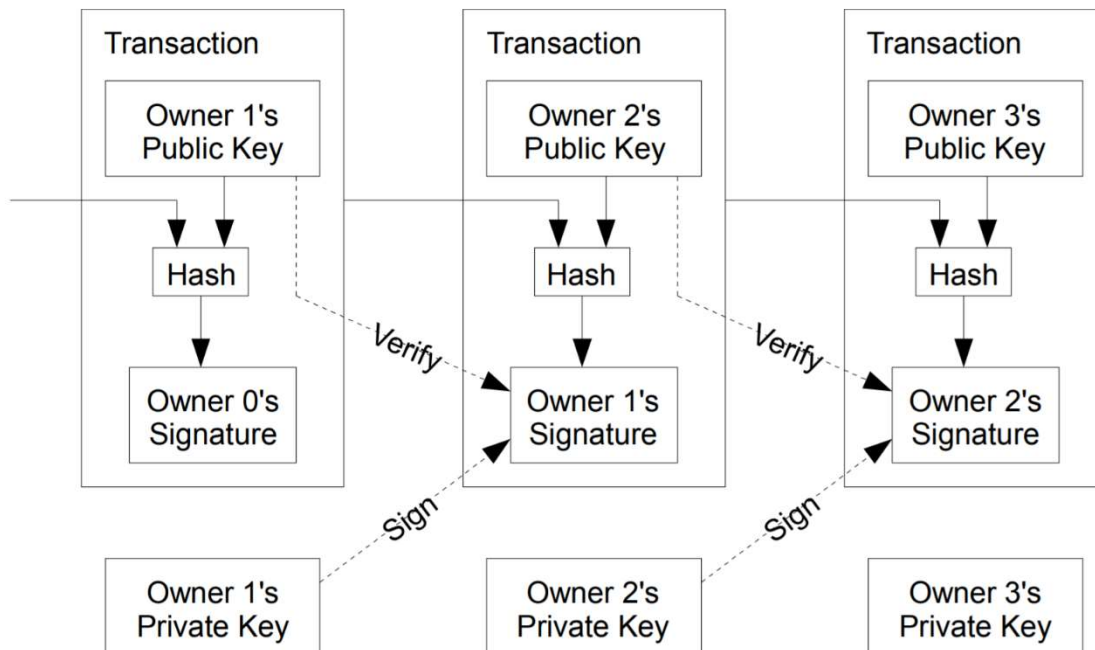


Figure 1: Transactions in the Bitcoin network [4]

The public key is the address of a Bitcoin account, the so called "wallet ", to which bitcoins can be sent. The private key and the transaction data can then be used to create a cryptographic signature. The transaction data, the signature and the public key can then be used to verify whether the signature belongs to the private key. In this way, a transaction can be signed and verified without the private key having to be publicly known, because only the signature is sent to the network. Each transaction receives a hash value (cf. Figure 1). A hash value is a unique value that is formed using the hash function "SHA-256". A cryptographic hash function always returns the same unique value for the same input. The input cannot be restored from the value [5]. A cryptographic hash value can therefore be understood as a fingerprint of data. Each signature is therefore unique, even though the same transaction is executed. The signature can be compared with physical signatures. But the signatures on the Bitcoin network are more unique and secure. [4]

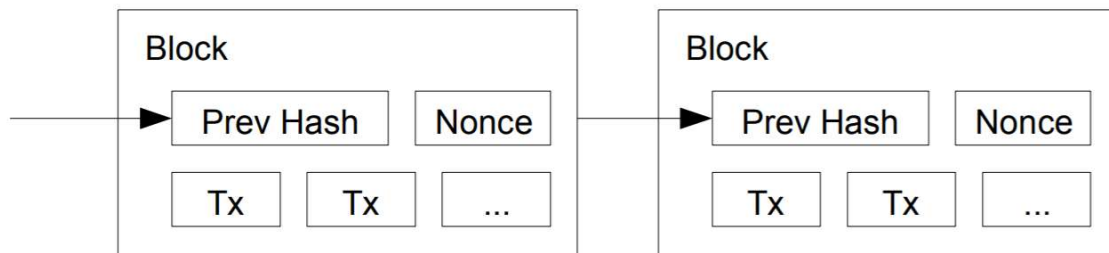


Figure 2: The Bitcoin-Blockchain [4]

Once a transaction has been signed with the private key and verified with an associated public key, the transactions are stored in a block. This block is attached to the blockchain. Thereby, the blockchain represents a simple chained list in which each block points to the following block in which the successor contains the hash of the predecessor (cf. Figure 2).

In order for a block to be verified on the network, a certain number must be found so that the hash of the block starts with a certain number of zeros. This number is called "nonce" and is also stored in the block. To find the nonce, every possible combination of numbers is tried until the hash value starts with a certain number of zeros. The one, who finds the nonce receives the "block reward" and the transaction fees that the user can provide so that his transaction is preferentially executed. In order to keep the amount of bitcoins limited at 21 million, the block rewards are halved every 210,000 blocks. This process is called "halving". Satoshi Nakamoto describes finding a nonce as proof-of-work [4], as this process uses physical computing power to check a block. Because new Bitcoins are created here, this process is called "mining". This is to guarantee security in the network. The number of zeros is then indicated by the so called "Difficulty", which results from the computing power in the network. Because the higher the Difficulty, more zeros must be placed at the beginning of the hash value, and with each additional zero, the work required to find the nonce grows exponentially. To create an interest in finding the nonce of a block, the person who finds the nonce receives a reward in the form of Bitcoin. The Difficulty is adjusted in the Bitcoin network in such a way that, with the appropriate computing power, it takes about 10 minutes to find the nonce and verify the block and the transactions it contains. [4]

The individual transactions are stored in a so called "Merkle Tree". A Merkle Tree can be understood like a binary tree, except that in the Merkle Tree the root is generated



from the leaves of the tree (cf. Figure 3). For this purpose, the hash values of the transactions are stored in the leaves. From two leaves the node above it is created. It contains a hash value that is formed from the two leaves. In the end, only the root of the tree remains. This hash value is also called "root hash" and is stored in the block.

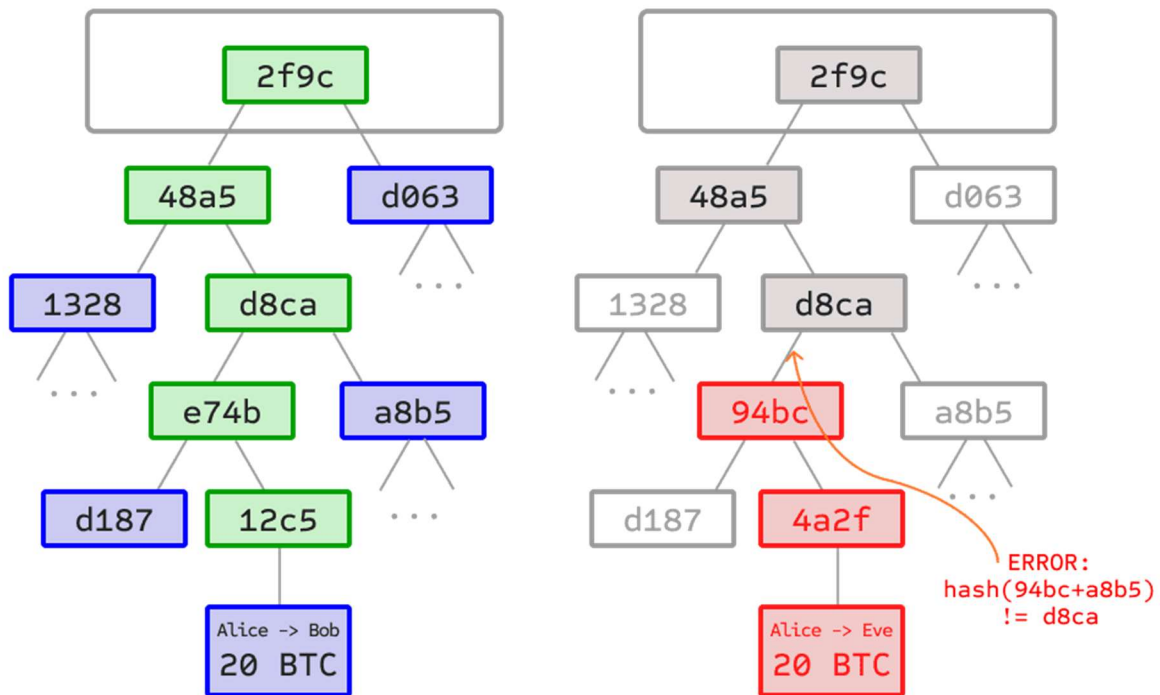


Figure 3: Merkle Tree in Bitcoin [6]

Because a hash value is unique and differs significantly from its predecessor with only a small change in the input, the change in a transaction is reflected directly in the root hash. This change changes the hash value of an entire block and therefore the hash value of all following blocks. Therefore, the nonce of the following blocks is no longer valid, and a new nonce would have to be found for all following blocks. It is then impossible to later change transactions in the blockchain. According to Buterin [6], this type of data structure of the blockchain results in great scalability, as only the hash value is stored and therefore many transactions can be stored securely and with little effort. [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.

In order for a consensus to emerge on a currency, there are the so called "nodes". They store the entire blockchain and search for new transactions to store in a new block. The node checks whether the transactions are valid and tries to find the matching nonce for the block. Once the nonce is found, the block is appended to the blockchain and communicated to all other nodes in the network. These then check whether the nonce is correct, i.e. whether there are enough zeros at the beginning of the hash value to match the difficulty. The nodes at the same time search for many other blockchains in the network. Only the blockchain with the most blocks are accepted. An added block is only accepted if it is still present in the blockchain after a long time, i.e. if it has been accepted by a particularly large number of nodes. This makes the blockchain particularly secure and decentralised. Because in order to forge a transaction, it would be necessary to have at least 51 % of the computing power in the network to validate the other blocks with a nonce faster than the rest of the network it does over a longer period of time - a so-called 51 % attack [7]. Running a node is possible for any user without much hardware effort (as long as the user doesn't want to mine) and allows him to contribute to the security of the network. [4]

### **2.1.3 Ethereum and Smart Contracts**

According to [3], Ether is currently the second largest cryptocurrency behind Bitcoin and has a market capitalisation of around 200 billion US dollars. Vitalik Buterin describes Ethereum as an "operating system for the blockchain" on which new applications for various use cases can always be developed [8]. These applications are then executed decentrally on the blockchain and not centrally on a server like conventional applications.

In the white paper of Ethereum [6], Buterin explains how blockchain technology can be used not only to store transactions and thereby achieve a consensus of a currency, but also to execute programmes on the blockchain. Like Bitcoin, Ethereum also uses blockchain technology. The way both cryptocurrencies work is the same. Ethereum uses Ethash as its hash algorithm, and the confirmation time of a block is on average only 13.4s, which is significantly faster than in the Bitcoin network [9] [10].

In order to be able to execute programmes on the Ethereum Blockchain, there are so-called "smart contracts". They represent a kind of contract in the network and are visible to everyone through the blockchain. They are defined by programmable code

and executed exactly as they are programmed. Such a contract is executed, for example, when a certain amount of Ether has been transferred to the contract. An application can therefore be written in Ethereum with such contracts. There are two types of accounts in the Ethereum network: User accounts, which are controlled with a private key like in Bitcoin, and contracts, which are controlled by the predefined code. These contracts can be used to form autonomous decentralised organisations (DAOs).

In order for such contracts to be stored and the code to be executed on the network, a block on the network does not simply consist of transactions, but of functions that can define a transaction, but also other operations, which are then executed by the Ethereum Virtual Machine (EVM). The EVM is located on the blockchain and thus on every node of the Ethereum network. This means that the code of a smart contract is executed almost at the same time on each node and therefore on the Ethereum network as soon as a block has been validated [11]. [6]

Accordingly, decentralised and secure applications can be developed on the Ethereum blockchain that not only transfer money decentrally, but also store data decentrally and securely. These applications are also called decentralised applications (dApps) because of their decentralised nature. In order to execute smart contracts and store data on Ethereum, a fee must be paid because a node performs such operations like transactions in the EVM. These fees are expressed as Giga-Wei (GWEI). Wei is the smallest unit in the network and is equivalent to  $10^{-18}$  ether [12]. Ether or GWEI is a kind of fuel in the network and is therefore also called gas. [6]

In contrast to the Bitcoin network, the Ethereum network is not static and is being further developed by many different developer teams worldwide from the Ethereum ecosystem. The biggest upgrade since its launch in 2015 is Ethereum 2.0. This upgrade is intended to make the Ethereum network more scalable, secure and efficient. This is to be achieved by moving from a proof-of-work consensus to a proof-of-stake consensus. In the process, new blocks are to be validated by holding Ether. This eliminates the computationally intensive mining process and makes the network more economical and decentralised, as it is not large data centres that validate the blocks, but those with a particularly high amount of Ether. With regulated staking pools, associations of many small Ether holders, anyone can participate in staking, even with

only a little amount of Ether. Most importantly, the upgrade should result in lower fees on the network, which not only makes transactions cheaper, but also makes dApps with smart contracts more appealing and more secure through more decentralisation. [13] [14]

#### **2.1.4 Potential and current purpose of cryptocurrencies**

Cryptocurrencies now have a global market capitalisation of 1.97 trillion US dollars [3]. This includes not only Bitcoin, but also more than 4,500 other cryptocurrencies that have specialised in other use cases (cf. chapter 2.1.1). But which potential do cryptocurrencies have in our networked world and how are they already being used?

Cryptocurrencies use cryptography, decentralised networks and blockchain technology to offer an independent, global, anonymous and free means of payment. Because this technology is available to anyone with internet access and anyone can work to improve it, cryptocurrencies are a much more democratic monetary system than the current one. What is special about cryptocurrencies is not only the technology, but also the global dimension. Cryptocurrencies are particularly popular in countries like Nigeria, Vietnam and South Africa [15]. These are developing or emerging countries. There are rather weak and unstable currencies. Citizens there try to change from their currencies into stable currencies like the US dollar or the euro. However, these are often highly regulated by local authorities. Therefore, cryptocurrencies offer people a safe and anonymous way to store value. Since wallets are secured with a so-called "seed phrase", i.e. a random sequence of 12 to 24 words, owners of cryptocurrencies only have to remember this seed phrase. With this phrase, people can everywhere recover their wallets and coins and dispose of their money. Since the phrase is not written down, the possession of money is protected from illegal or authoritarian access.

Cryptocurrencies and especially the blockchain technology on which they are based are also attractive for broad application in the economy. Blockchain makes not only transactions in the financial sector, but also the exchange of business and industrial data secure, calculable and fast. The *International Data Spaces Association* (IDSA), which is supported by industrial companies, has developed a reference architecture to enable the secure exchange and linking of data within business ecosystems [16].

Reinhold Achatz, chairman of IDSA, considers blockchain to be an adequate, mature and recognised technology for the operation of data-driven business ecosystems [17].

Decentralised Finance (DeFi) is popular in the cryptocurrency sector. DeFi aims to bring traditional financial services such as loans or exchanges into the cryptocurrency sector [18]. The DeFi movement is guided by the idea that the financial system is not controlled by a central third party, but is decentralised. Most DeFi projects are being developed on the Ethereum platform and use smart contract technology. For example, smart contracts on the Ethereum platform can be used to programme their own protocols for tokens in order to create stablecoins that are based on US dollars or euros. There are also decentralised exchanges such as UniSwap, on which individual tokens can be traded among themselves [19]. On decentralised peer-to-peer lending platforms like Compound, tokens can be lent or borrowed for interest [20].

There are other projects in the DeFi field in culture: for example, Non Fungible Tokens (NFT) are used to store digital artworks such as pictures or music on the Ethereum blockchain and assign them to a specific account. This makes counterfeiting and unauthorised copying of artworks no longer possible.

How many people use cryptocurrencies is difficult to find out because of the anonymity of cryptocurrencies, even if every transaction is publicly available. With so-called "OnChain analyses", it is possible to approximate how many users there are. There are currently between 1.2 and 1.3 million active addresses on the Bitcoin network [21]. However, these addresses can be operated not only by humans but also by computers, and humans can also use several addresses simultaneously. Also not included are people who have cryptocurrencies but do not actively use them as a means of payment and rarely make transactions. According to GlassnodeStudio, there are approximately 30 million addresses on the Bitcoin network that own Bitcoins [22]. Again, it is possible that one person has access to multiple addresses or there is no longer access to the coins because the private key or seed phrase has been lost.

## **2.2 Weaknesses of current transaction platforms**

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

In the traditional financial system, there are two forms of money ownership: cash and online money. Cash is physically present and protected from counterfeiting by watermarks and other protective mechanisms. Small amounts of money can be stored well in a purse or piggy bank. For large amounts of money, storage becomes difficult because of security and the physical quantity. Anyone who wants to have this money available online must be a customer of a bank and trust that the bank will keep the money safe, carry out transactions correctly and, above all, store user data and user behaviour securely.

To make payments in shops or online commerce, credit cards are used. Here, it is not only the bank that is responsible for the correct execution of transactions but also companies such as *Visa* or *MasterCard*. If money is to be sent from person to person, the Single Euro Payments Area (SEPA) procedure can be used. Here, payments are made directly between banks within the euro area. Money can be sent easily and flexibly with online payment services such as *PayPal* or *Sofortüberweisung*. Meanwhile, it is possible to pay with your smartphone or smartwatch and the services of *Google Pay* or *Apple Pay*. Via platforms such as *VimPay*, it is also possible to get in touch with other people via chat and send money to that person.

These forms of transactions have one problem in common: a third central party is absolutely necessary for the execution of the transaction (cf. chapter 1), and consumers have to trust that payments will be executed correctly. The bank or other payment service providers are responsible for ensuring that not too much or too less is debited and that the correct destination account is found. In addition, it is unclear to the consumer which companies and banks are involved in the transactions and which data is given to them.

Cryptocurrencies are neutral, unlike the traditional financial system. If one wants to open an account at a bank, a large amount of personal data is requested and thoroughly checked before it is possible to use the bank account. Moreover, transactions are expensive. A business customer at PayPal, for example, pays 50 euros for a transaction of 2,000 euros [23]. For private customers, a transaction is free in most cases, but if it goes abroad, the fees can be up to 3,99 euros [24]. Transactions can also take a long time. When paying by credit card, a transaction takes up to 30 days. During this time, the transaction sender and recipient have no assurance as to

whether the transaction will be approved or revoked by the credit institution. With PayPal, too, it takes a few days until money received is available. During this period, the money is capital for the third party.

Another problem with traditional transaction platforms is their centralised structure. Banks and other payment service providers are vulnerable to hacker attacks due to their centrality. Thus, usually only a few computers have control over the entire banking system. For example, in the "Carbanak incident" [25] about 1.2 billion US dollars were looted. Here, the money of several thousand customers was stolen, but only one institution was hacked. In the case of cryptocurrencies such as Bitcoin or Ethereum, each owner of the money has the private keys and has kept them safe. Should a hack occur here, only the individual person is affected and not the entire network. Even in the case of a 51 % attack (cf. chapter 2.1.2), the coins in a wallet are safe, as it is not possible to change any transactions in the blockchain afterwards [7].

### **2.3 Current state of research**

According to Satoshi Nakamoto, Bitcoin relies on a new privacy model [4]. He describes that the traditional financial system achieves privacy by allowing only a limited number of people to have access to transaction information, while hiding the identities from the public, i.e. they do not know from whom the money is transferred to whom. The new model outlined by Nakamoto, on the other hand, relies on making the transactions available to the public. However, these are separated from individual identities. In this way, it is known that money was transferred, but not from who to whom. This is achieved by only storing addresses in the blockchain. These addresses are a public key generated from the private key (cf. chapter 2.1.2). To achieve more privacy, several public keys can be generated for the same private key, so that a different public key can be written to the blockchain for each transaction. Andreas M. Antonopoulos currently estimates anonymity to be lower than it should be, but it could become more anonymous through new cryptographic methods [26]. In fact, there are now ways to use coins more anonymously. For example, the Wasabi wallet with mixed transactions guarantees higher anonymity than normal transactions in the Bitcoin network [27].

Antonopoulos considers the anonymity of a currency to be very important for the self-determination and free expression of a society. Cryptocurrencies are suitable for

transferring money across national borders without interference and monitoring from third parties. For example, money would be available for political campaigns that would otherwise be subject to financial restrictions; the campaigns could be used to express opinions [26]. Moreover, only 69 % of the population worldwide have an account at a financial institution [28]. This means that 31 % of the world's population do not have an account and may not have access to traditional financial services, limiting the use of any cash they may have.

Because of the anonymity, cryptocurrencies are assumed to be used for illegal purposes such as drugs and arms dealing. In the Visual Objects Digital Currency Survey [29], 38 % of respondents said they use cryptocurrencies to buy food and 34 % to buy clothes. 29 % of respondents use cryptocurrencies as an investment or store of value, and 21 % reported using cryptocurrencies to buy gold. Comparatively less people reported buying weapons (15 %) or drugs (11 %). Although 15 and 11 % may seem high for questionable purchases, the survey shows that accepted activities are much more represented in cryptocurrencies.

Cryptocurrencies cause high energy consumption. Bitcoin alone consumes about 144.9 TWh per year [30]. That is about as much electricity as Poland (141 TWh) or Egypt (150 TWh) consume [31]. Therefore, a digital currency as a single application consumes as much electricity as an entire country. However, a country is not in relation to a global financial system. For this reason, the electricity consumption can be compared with the traditional financial system. For example, the global banking system consumes about 650 TWh/year [32]. That is about 4.5 times as much power as bitcoin. Ulrich Gellersdörfer sees the problem not in the cryptocurrencies themselves, but where they get their energy from [33]. As with the data centres of large internet corporations and streaming platforms, cryptocurrencies should also be looked at to see how the energy requirements can be made more sustainable in general.

Bitcoin and other cryptocurrencies can be a stable currency for people whose country is unstable and marked by corruption. However, Antonopoulos reasons that Bitcoin also serves the world's privileged population: cryptocurrencies offer cheaper and faster ways to carry out transactions than the traditional financial system, and even the euro is affected by a little inflation. Antonopoulos also cites the neutrality of cryptocurrencies as a major advantage. Thanks to the decentralised nature of the



network, anyone can participate and use its services. This way, someone could take out a loan in the form of cryptocurrencies without that person being checked for their origin, religion, skin colour and qualifications. [34]

According to Fabian Schär, DeFi products can play a particularly important role for the future. He explains how DeFi projects replace traditional financial products and make new products available. The technology of smart contracts could be a big component in the cryptocurrency sector in the next few years. [35]

## **2.4 Decentralised transaction mechanisms as a solution approach**

Traditional financial concepts are very inefficient and expensive because of their centralised structure, so transactions can take up to a whole month, and especially large sums and international transactions incur high fees (cf. chapter 2.2). In addition, they are only slightly anonymous, are based on trust and only after a detailed identity check is it possible for the user to use money online. Moreover, currencies in the traditional sense are controlled by a central authority. So central banks have an influence on the value of a currency.

In order for an application to send money online, users must enter into contracts or a business agreement with banks, credit institutions and payment service providers. In doing so, both the user and the application must trust that transactions are carried out correctly. In addition, transactions can take a very long time, and large amounts of money can incur high fees for a transaction. Another problem in developing a global transaction application are the different currency systems. This means that for cross-border transactions – except within the Eurozone – there must always be an exchange between the currencies involved. This means effort even for small amounts, as currency pairs have to be traded here on many different exchanges so that money can be sent to each nation.

With their decentralised structure, cryptocurrencies create a global currency that is available everywhere online. They offer private individuals significantly more security and control over their money through their decentralised structure. They are also more democratic, as everyone has access to the network and can participate in it. Those who want to own cryptocurrencies do not have to go through elaborate identity checks and have direct access to many different financial products.

With smart contracts, the Ethereum platform offers the possibility of programming new applications on the Ethereum blockchain (cf. chapter 2.1.3). This makes it possible to programme decentralised applications without having to develop a blockchain with its own currency. To develop a global transaction application, Ethereum is particularly well suited. Besides Ethereum, there are alternatives such as Cardano or Algorand, which have lower transaction fees than Ethereum. However, these are still undeveloped, so it is not yet possible to programme smart contracts, or the developer community is still small, so it is difficult to find good documentation or threads in forums.

### **3 The development of an application for decentralised transactions**

#### **3.1 Objectives and methodology**

##### **3.1.1 Functions and purpose of the application**

Cryptocurrencies and blockchain technology are no longer a pioneering topic, but are reaching broad sections of the population. It has become easier to own and trade cryptocurrencies in recent years with a wide range of different crypto exchanges and wallet providers. However, apart from financial products and a few social networks like *Cent* and *CryptoKitties*, there are only a few use cases for consumers. Therefore, cryptocurrencies are mainly traded as speculative assets.

The application that is to be developed is to enable the analogue and anonymous cash transfer in combination with a communication dialogue online.

Until now, it was possible to have a dialogue with messenger services. If one party wanted to send money to the other party online, a third party such as a bank or *PayPal* had to be used. If people wanted to remain completely anonymous, pioneers had the option of using cryptocurrencies by exchanging addresses and then making a transfer with their wallet. Beyond that, only the physical exchange of cash remained.

With the application, it should now be possible to network like a conventional messenger application, i.e. to communicate globally with other people. At the same time, the application is supposed to give the user the possibility to use the cryptocurrency Ether. People should not only be able to communicate, but also to send money to each other easily, quickly and cheaply. This should work completely anonymously for the user.

##### **3.1.2 Realisation**

The application is to be usable by everyone and accessible via the internet. Accordingly, a web application is to be developed that is accessible via the browser. The user is to access the chat page via a user-friendly login with e-mail and password. There, they can then communicate with other users in a one-to-one chat. Via "Settings", the user is supposed to get from the chat page to the settings, where, in addition to settings about user data, he will also find the "Connect Wallet" button, which he can use to connect his wallet to the application.

In order to make the Messenger application as user-friendly as possible and to make it accessible to users with no previous knowledge of cryptocurrencies, a wallet is not a requirement for using the application. So, data such as messages and user data are not stored on the Ethereum blockchain. Instead, the Messenger application should first store data centrally on a server in the conventional way. The application should only then communicate with the Ethereum network in order to be able to process payments decentrally via the network securely, cheaply, quickly and anonymously.

### 3.2 Structure and functioning of the application

The application should be accessible as a web application via a browser (cf. chapter 3.1.2). To realise this, the JavaScript library React.js and the programming language JavaScript are used. React.js makes it possible to programme user interfaces and web servers for web applications with JavaScript. In this way, a web application can be designed dynamically and interactively with React.js-JSX elements. React.js should then communicate with the database, Ethereum and the user.

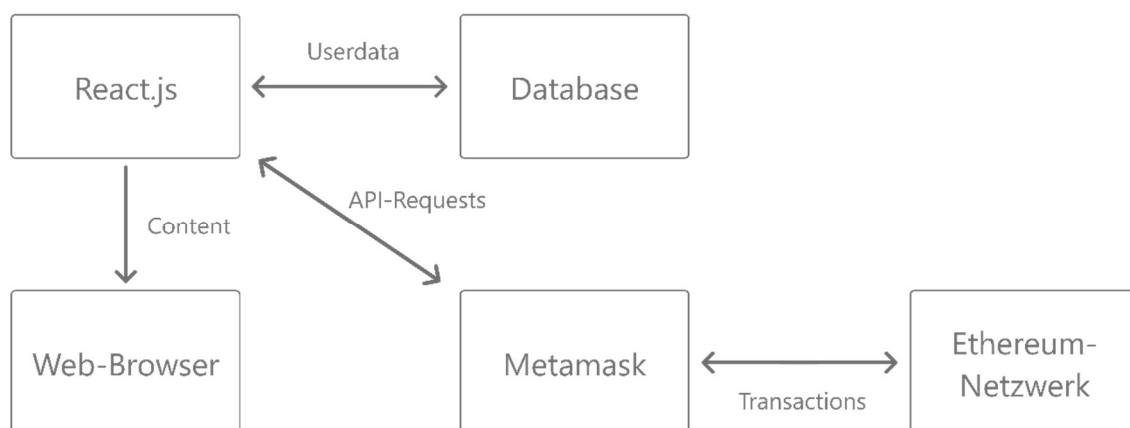


Figure 4: Server communication in the application shown in simplified form

As shown in *Figure 4*, the React.js server communicates with the database, which in this case is represented by Google Firebase. To enable users to store their Ether securely and still use it in a web application, the Metamask extension is available for the browser. Metamask forms an Ethereum wallet integrated in the browser, with which applications can communicate via the web3.js API. This means the application

can perform transactions on the Ethereum network with Metamask and the user's permission. dApps like Compound, CryptoKitties and Cent also use Metamask, which means the user only needs to have one wallet in the browser to use Ether in many different applications.

### **3.2.1 Communication with the user – React.js**

The web application consists of three web pages. The login or registration, the homepage and the settings. On the login page, the user can log in with his login data and will be redirected to the homepage. On the registration page, the user can register with his or her username, e-mail and password and is then also redirected to the homepage. On the homepage, the user can access the individual chats with the particular persons via the list of other users. If the user clicks on another user, the chat history with the respective user appears; in addition, there is the option to send a message to the other user via a text field and the "Send" button.

If both users have linked the Metamask Wallet to the application, a text field and a button appear with which the user can send Ether to the other user. If the user clicks on the settings in the navigation bar, he or she is taken to the settings. There, the user will find the "Wallet Settings" with the button "Connect Wallet" and the tab "Personal Information", where information such as username, email and password can be changed.

The individual pages are programmed in React.js as components. The components can then contain other components that display individual elements on the website. The content to be displayed is programmed in these components using the JSX syntax.

```
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%'}}
21       key={index}>
22       <div className={chat.user_uid_Sender === auth.uid ? 'messageStyle sender' : 'messageStyle receiver'}...>
23       <p className="message-Time"...>
24     </div>
25   );
26 }
27
28 export default Message
29
```

Figure 5: React Component Message

JSX is passed in the return function of the component and represents HTML code with JavaScript elements. The JavaScript elements allow the actually static HTML syntax to be programmed here and to adapt dynamically. For example, the ClassName attribute in the div in line 27 contains different values depending on whether the condition is true or not.

In order to process user input directly in React.js, JSX offers the option of storing the input in a specific field directly in a state of the component. States of components can be equated with attributes from object-oriented programming.

### 3.2.2 Communication with Ethereum – Metamask

Metamask is a wallet that is located directly in the browser (cf. chapter 3.2). It can be installed as a browser extension in most browsers. In this wallet, the user can create a new wallet with a new public key and private key to which he can send Ether. It is also possible to import an existing wallet with the private key.

The web3.js API can be used to communicate with the Metamask Wallet if the user has installed the extension in the browser. In order for the user to also access the wallet, he or she must give the permission for the application to access the wallet. If the user agrees, the respective domain of the application is stored in the wallet. The user can remove this at any time so that the application is no longer allowed to communicate with the wallet.

With various API requests, the application can retrieve data from the wallet and send orders to the wallet. For example, the application can request the addresses, the current balance and, if applicable, the various tokens.

```
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_Address, // 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 }
```

Figure 6: Function sendETH from HomePage

With these requests, the application can request data from the wallet, but also order transactions. This request is called up with "eth\_sendTransaction", as shown in Figure 6. As a parameter, it must be specified from which address which amount is to be transferred to which address. The sender address is the address that is currently linked to the wallet and the recipient address is the address that the other user last linked to the application. All values must be passed as hexadecimal numbers. The addresses are stored as hexadecimal strings and can be transferred without any problems. The user specifies the amount to be transferred in Ether and as a decimal number. Therefore, the number must first be converted into Wei, the smallest unit in the network, and then into the corresponding hexadecimal number. Example: 1 ETH =  $10^{18}$  Wei;  $10^{18}$  Wei = 0xde0b6b3a7640000 Wei (Hexadecimal numbers are marked in the Ethereum network starting with 0x).

### 3.2.3 Data storage and processing – Firebase

Chat and user data must be stored centrally so that no Ethereum wallet is required to use the application (cf. chapter 3.1.2). The data is stored on Google Firebase, as it offers the possibility to manage users and to retrieve and edit data in real time. Firebase offers to store data in JSON format. According to [36], the JSON format is based on the JavaScript programming language and can therefore be used well in this language. Firebase was chosen because of its user-friendliness; an alternative is *MongoDB*.

In order for React.js to communicate with Firebase to retrieve and process data, the JavaScript library Redux.js is used. Redux.js is used to retrieve data from the database and then make it available to the React Components in a bundled and organised way. This means that a data query does not have to be programmed into each React Component individually. This makes it easier to use data in an application. [37]

In order for a user to register and log in, his or her username, email and password in encrypted form are stored (cf. *Figure 7*). In addition, it is stored whether the user is currently online so that this can be displayed to other users, when the user created an account and a unique ID so that this object can be referred to in other objects in the database. If the user has connected a Metamask Wallet to the application, the address is also stored so that other users can make transactions to this address.



```
1  [
2  {
3      "username": "Kai",
4      "email": "kai@gmx.de",
5      "ETH_Address": "0x18ed89f4b1e7b2230c9d13e3121f5ba9fcab544b",
6      "password": "E10ADC3949BA59ABBE56E057F20F883E",
7      "isOnline": true,
8      "createdAt": 1617974144224,
9      "uid": "aoa6g1HQajfXi704E6xt3vnjix93"
10 },
11 {"username": "Jufg"...},
19 {"username": "Paul"...}
27 ]
```

Figure 7: User data in JSON-Format

In order for transactions and messages to be displayed in the chat, they are also stored (cf. Figure 8). Again, a timestamp indicating when the message or transaction was sent and a unique ID are stored. In order to be able to track who sent the message and who should receive it, the ID of the respective user is stored in the JSON object either as sender or recipient. To determine whether it is a message or a transaction, this is also stored as a string. If it is a message, the message is saved under *message*. If it is a transaction, another object with the transaction data is saved under *transaction*. This object contains the hash value of the transaction, the address of the sender and the recipient, the amount of Ether transferred in Ether and the transaction fee in Ether.

```
1  [
2    {
3      "createdAt": 1617974210368,
4      "isViewed": false,
5      "message": "Hi",
6      "type": "text",
7      "user_uid_Receiver": "aoa6g1HQajfXi704EGxt3vnjix93",
8      "user_uid_Sender": "k5ZZbq3ZeERT1L91l9fJk4bIF0A3",
9      "uid": "ed21af664939158df27776873f129999bec3e41b6b0d554a10e4b3411bc7b689"
10   },
11   {
12     "createdAt": 1617974185888,
13     "isViewed": true,
14     "transaction": {
15       "txHash": "0x235d31cbedac86418c7fa852a5842ade17c026ca6bb377b36dabef13210ff053",
16       "from": "0x18ed89f4b1e7b2230c9d13e3121f5ba9fcab544b",
17       "to": "0xe66ad4a5dc0afd16d8a8f1c3339c48ccce7946f3",
18       "value": 14.9995,
19       "gasUsed": 0.000000000000021
20     },
21     "type": "transaction",
22     "user_uid_Receiver": "kFKiZIWgRceJvwUDSg0zv3bzBsI3",
23     "user_uid_Sender": "CAfZzBmCbKUgNnnygvTrm8rRvQL2",
24     "uid": "27a73e24a3e5b8806697d54ade102182b6921bfdcef4629ef4741c816fa9f3fb"
25   }
26 ]
```

Figure 8: Messages and transactions in JSON-Format

For simplicity, data such as the messages and transactions are not encrypted here. However, this could be supplemented with end-to-end encryption. According to [38], with end-to-end encryption it is possible that only the sender and receiver can read the messages. On the server, however, the data is encrypted and not readable by the application.

### 3.3 Conventional and innovative elements of the application

The application consists of various elements to design a messenger application that supports decentralised transactions. It draws on both conventional and innovative elements.

The application has two basic functions: the messenger function and the function to execute decentralised transactions on the Ethereum network to other users. The

concept of a messenger application is conventional. The first messenger service, *ICQ*, appeared in 1996. Since then, many other messenger services have developed, such as *WhatsApp*, *Telegram*, *Signal* or *Threema*. In the present application, data is stored centrally on a server, as in conventional messenger services. Accordingly, user data and chat messages are affected by the advantages and disadvantages of a central server structure. A third party must be trusted to store the data and there is a higher vulnerability to hacker attacks. On the other hand, data can be delivered much faster.

The application, on the other hand, combines this conventional concept of a central messenger service with the innovative element of the decentralised transaction mechanism with Ethereum. Established messenger services such as *Signal* also integrate cryptocurrencies [39]. However, only the cryptocurrency *MobileCoin* is used there. The innovative thing about using Ethereum and *Metamask* is that not only can a cryptocurrency be used here, it is also possible to use tokens. In addition, *Metamask* is already integrated into many other dApps and is therefore already used by those who use dApps and Ethereum.

### **3.4 Outlook**

With its transaction function, the present application offers more than a conventional messenger application. With a view to decentralised transactions, it is still one-sided, but can be extended by additional functions

Cryptocurrencies are in a constant state of change and are supplemented by many new projects. Results of these projects can be integrated into the application, for example to make it more secure and user-friendly.

#### **3.4.1 Decentralised data storage with IPFS**

The decentralised data storage in the Ethereum network poses two challenges for the application: First, a user must have an Ethereum wallet in order for the data to be associated with them, and second, data storage on the EVM is treated like transactions [6]. Therefore, a transaction fee would have to be paid each time a user or a message is stored. The Inter Planetary File System (IPFS) offers a possible solution.

IPFS was developed in 2015 by Juan Benet and is a decentralised peer-to-peer file system. Here, files are stored decentrally on a network of nodes. The files are stored in an IPFS object. This object has a hash value that can be used to identify the object in the network. The IPFS object contains the data of the file and links in which hash values of other IPFS objects can be stored in order to link to other files in the network. If a file is to be retrieved in the network, the hash value must be specified. The nodes then search the network for this hash value and return the file. Because a file is stored on the network with the hash, it is not possible to change the file without changing the hash.

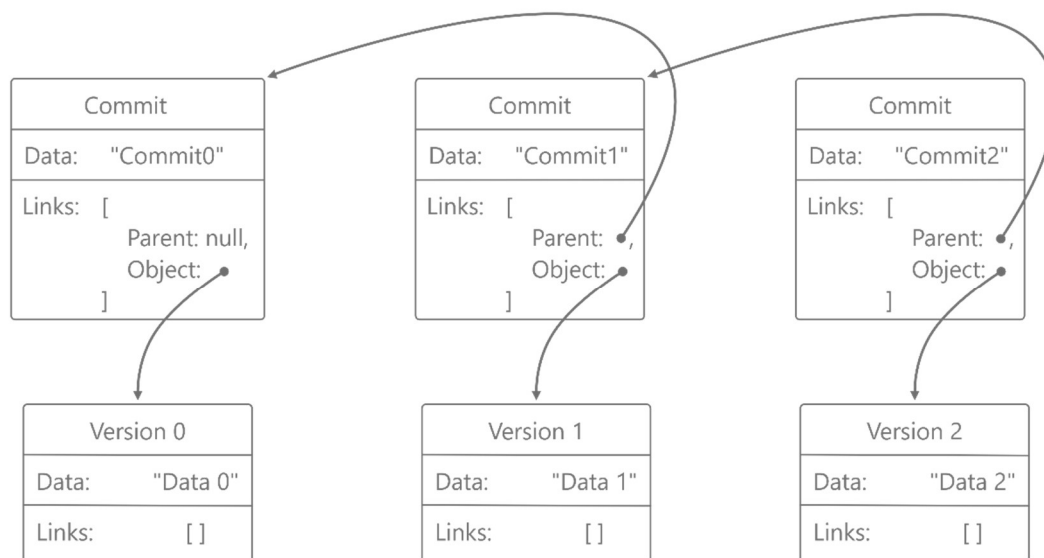


Figure 9: Visualisation of version control in IPFS according to [40]

If you want to change a file, you use a version history (cf. Figure 9). This consists of several IPFS objects called commits, which link to the IPFS object with the current file and to the previous commit. [40]

IPFS offers the possibility to store files in a tamper-proof way and to be able to track changes. However, IPFS has disadvantages. There is a risk that if a file is stored on only a few nodes, they will lose the file or go offline. This means that the file can no longer be found in the network and the user can no longer retrieve the file. According to [41], the Filecoin project is trying to solve this with a proof-of-replication procedure by rewarding nodes for providing storage space and requiring a certain number of

nodes to store a file. This way, individual users would not have to pay for storage, but the application would. Since the application does not charge users, it would have to finance the storage space with advertising or data trading. A suitable solution is being searched for.

### **3.4.2 ERC-20 token integration**

The Ethereum Blockchain makes it possible to store tokens on the blockchain using smart contracts and the ERC-20 standard. These tokens can serve different purposes. For example, they can represent a points system on an online platform or represent entirely separate currencies. Among other things, ERC-20 tokens can also be used to represent existing conventional currencies such as US dollars or euros. They can also represent gold or other cryptocurrencies such as Bitcoin on the Ethereum blockchain. [42]

These tokens can be stored in an Ethereum wallet in the same way as Ether, including in the Metamask wallet. The application could support different tokens. This would allow users to send not only Ether, but also tokens that represent other currencies. So, with Tether-USD (USDT) or USD Coin (USDC), they would be able to send US dollars to other people via the application. With the Wrapped-BTC (WBTC) or Tether-Gold (XAUT) token, users could also send bitcoin or gold via the Ethereum blockchain.

As Ether is exposed to large price fluctuations, these tokens could allow users to send stable currencies and not take a high risk of loss when they want to send coins to other people.

### **3.4.3 Group chats with smart contracts**

So far, it is possible to communicate with individual users in the present application. Group chats, as offered by other messenger services, can still be implemented. There, users could simply send money to members of a group. In addition, with smart contracts, it would be possible for a group to save together towards a specific goal

Group members would send Ether or other tokens to a smart contract after previously determining how much money to save and to which address the collected money should be sent when the savings target is reached. The smart contract would check how much money is in the contract every time money is transferred to it. As soon as

this amount equals or exceeds the target amount, the coins are paid out to the specified address. If the contract is to be dissolved before the target is reached, this can be attached to a transaction as a message and thereby communicated to the contract. If the contract receives this message, it sends the received coins back to the corresponding addresses.

This would enable users not only to chat in a group, but also to collect money together. This money can be used for various purposes, whereby the savings purpose remains anonymous to third parties. By doing this via a smart contract, there is no need to trust a third party that the money will be paid out correctly. The smart contract would always be executed by EVM as programmed.

## 4 Documentation

### 4.1 Representation of the work process

#### 4.1.1 First planning and sketches

At the beginning of the development, there were considerations about what the application should look like and what it has to do. A rough sketch of the application was made, which clarifies the structure and functionality of the application.

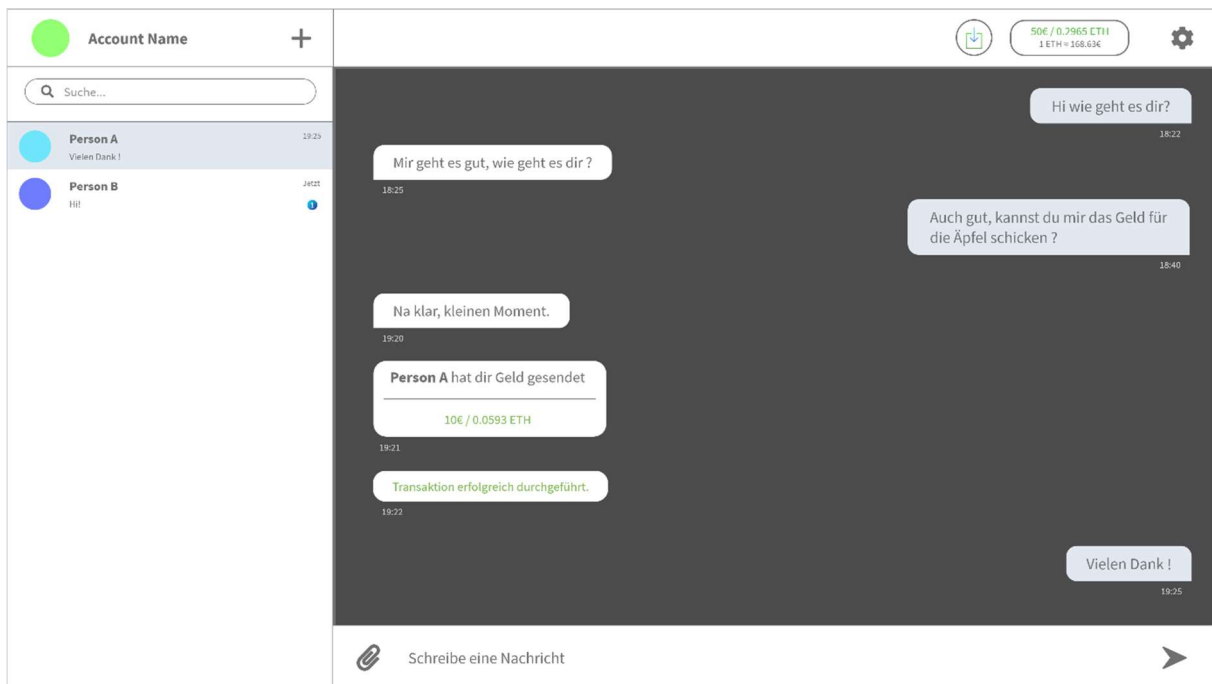


Figure 10: First sketch of the application

Figure 10 shows the sketch of the messenger application in which it should be possible to communicate with other users and, as can be seen in the chat history, to send Ether. In this sketch, it is still assumed that it is possible to provide the user with their own wallet in the application. In this way, the user would always have to send his Ether to the wallet in the application if he wants to dispose of his money there. However, this is inconvenient for the user because fees are charged for each transaction and they cannot use their Ether in other applications. In addition, the application would have to manage the private keys and keep them safe. This could be a security risk for the application, as someone could dispose of the user's coins with the captured keys (cf. chapter 2.1.1).

Since the browser wallet Metamask is considered particularly secure and established and is used by many other dApps, it was decided to install it in the browser as an extension (cf. chapter 3.2.2). Private keys are stored in the user's own browser instead of with a third party. Users can not only send new coins to the wallet, but also import existing wallets with the private key.

#### **4.1.2 Development of a first messenger application**

Based on the planning and sketch made beforehand, a web application was then programmed that fulfils the purpose of a messenger (See appendix for further illustrations).

A web server is needed so that a web application can react dynamically to user input. Since previous experience with JavaScript and Node.js is available and React.js is a widely used JavaScript library, Node.js in conjunction with React.js was chosen as the web server.

Programming of a simple chat application was started in order to learn how to use a frontend library like React.js and to test the data storage in the application. To realise the chat, two servers were programmed: a pure Node.js server and a React.js server. The Node.js server communicated only with the database in order to retrieve data in an organised way and to pass it bundled to the React.js server. The data was exchanged between the servers using Socket.IO. This then processed the data and passed it on as a web page to the browser and therefore to the user

In this version of the application, it was only possible to communicate with all users in a single chat room, as only the content and sender of a message were stored. So the application could assign messages to a user, but not to who it was intended for.

#### **4.1.3 Extended planning**

The first version of the application was a good way to get used to working with a library like React.js and to learn how to build a messenger application. However, the application programmed so far did not fulfil the goal of a one-to-one chat with other users, and payments via the Ethereum network were also not yet possible. In addition, the construction of the application was complex and inefficient due to the two servers.



Therefore, a new database system was modelled that not only stores who a message comes from, but also to whom it should be sent. In addition, this model should also store the Ethereum addresses of the users so that these could be used for transactions. This model corresponds to the one currently used (cf. chapter 3.2.3).

In order to make the server structure more simple and efficient, two servers should no longer be used, which separately represent the backend and frontend of the application, but only one React.js server, which can manage both user input and data from the database. Redux.js should be used so that the React.js server can communicate easily and in a structured way with the database. With the library, it should be possible to query data from the database within React.js and pass it on to the individual React.js components in an organised manner (cf. chapter 3.2.3).

#### **4.1.4 Development of a private messenger application**

According to the new plan, a second version of the application was developed. Since the server structure of the new application was to differ significantly from the first version, a new React.js server was programmed. With this, a repository was also created on GitHub, where the source code is saved and documented (cf. chapter A.1.1).

*Google Firebase* was used as the database for the new application, as it is particularly easy to manage users and data here (cf. chapter 3.2.3). Since it was now possible with the new database system to assign messages to the sender and recipient, a list of users could be programmed in this version of the application. This was done in the same way as in *Figure 10* and allows the user to open or start a chat with another user. It was now possible not only to display the messages of the two users in a chat, but in the chat, they could also be displayed according to recipient and sender.

Since the user should also be able to connect settings of his login data and later also a wallet to the application, the Settings page was programmed. Here, the user can change his or her user data such as username, email and password.

#### **4.1.5 Extending the application with the Metamask Wallet**

The application allows users to communicate with each other. With the Metamask Wallet and the Web3.js API, it should now be possible for users to send money to each other via decentralised transactions.

In order for users to be able to use the Metamask Wallet in the application, they must first connect it to the application and allow the application access to the wallet. The user should be able to define this in the settings. For this purpose, a button was programmed on the settings page with which the application sends a request to the Metamask extension - provided the user has installed this extension in the browser - asking for access to the wallet. If the user agrees, the application can access the addresses of the wallet and send requests for transactions to the wallet. The current address is stored in the database.

Since a user can connect their wallet to the application and the Ethereum address is stored in the database, the application has enough information to forward a transaction to the Metamask wallet. To make this possible, another text field was added in which the user can write the amount of Ether to be sent.

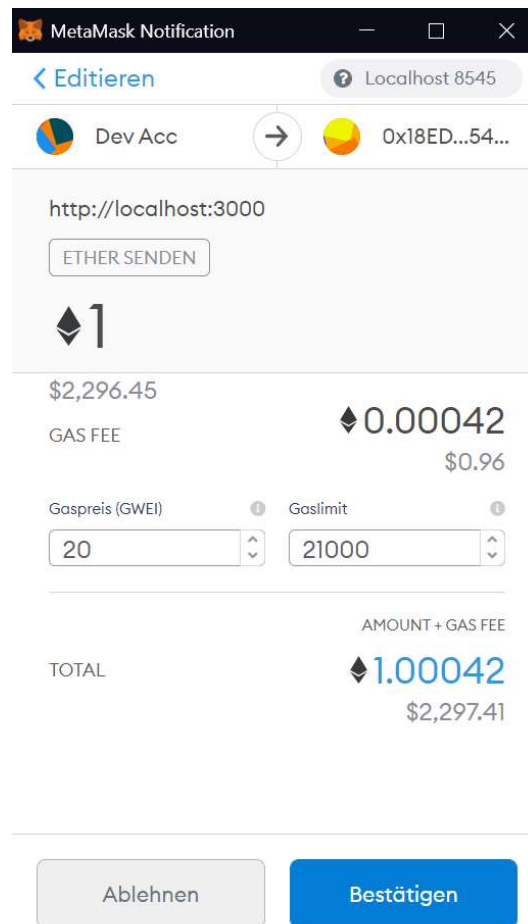


Figure 11: Metamask Transaction Confirmation

If the user clicks on the button "ETH", a window of the Metamask Extension opens in which all details of the transaction are displayed (cf. *Figure 11*). The user can adjust the transaction fees himself. The transaction is confirmed with the Confirm button and Metamask signs it with the private key. The transaction can still be cancelled at this point by clicking the Decline button. The server saves the transaction as a message in the database and displays it in the chat history (cf. chapter 3.2.3).

## 5 Critical reflection

### 5.1 General conditions of the paper

At the beginning of the work, cryptocurrencies were rarely used and were little noticed by society or viewed with restraint. This was caused, for example, by the fact that cryptocurrencies were associated with illegal darknet activities. In addition, the crash of the crypto bubble in 2017 led to cryptocurrencies being seen as investments with very high risk.

Since the end of 2020, interest in cryptocurrencies has increased sharply [43]. After falling to a new low in the "covid crash" at the beginning of 2020, cryptocurrencies have repeatedly reached new all-time highs since the end of 2020 [3]. Public confidence in cryptocurrencies and the blockchain technology behind them has strengthened. Demand has also surged for DeFi; for example, USDC supply on Compound nearly tripled from February to April 2021 [44], and liquidity on Uniswap nearly quadrupled from \$2.16 billion to \$8.27 billion from January to April 2021 [45]. This development confirms the relevance of decentralized transaction mechanisms and their platforms.

Also, the high trading volume, which amounts to between one and two billion US dollars per day on Uniswap [45], confirms that decentralized applications are stable and scalable. Therefore, a global transaction application with decentralized mechanisms must be given high importance and high demand is expected. This has confirmed and reinforced the intention to develop the present application.

In addition, the potential and functionalities of dApps in combination with the Metamask Wallet have proven to be very good. Every user can decide individually where and how he wants to use his Metamask Wallet. The application has no control over the private keys and can therefore not dispose of the user's coins. The user can anonymously decide how and where to use his money. So, he can use the wallet he uses in the application also in other dApps like Compound or Uniswap. In this way, the user can not only send his coins in the application, but also provide them as liquidity on Uniswap or lend them on Compound for interest.

## 5.2 Challenges during the development process

As the application has been developed, situations and insights have arisen that have slowed down and influenced the development process.

For example, the library React.js. was a challenge: although it facilitated the development of a dynamic website, there are significant differences in contrast to the development with pure JavaScript and Node.js. React.js offers the new syntax JSX, which can be used to develop the dynamic elements of a web page. This syntax consists of HTML and JavaScript elements (cf. chapter 3.2.1). With JSX it is possible in React.js to call HTML elements functions and states of a React.js component. Since the web server normally only projects an HTML document statically onto a web page, the direct interaction between HTML and the JavaScript server had to be learned first.

Another challenge was communicating via an API with the database or the Metamask wallet. This type of communication involves asynchronous operations in the program, referred to as "async functions" in JavaScript. These asynchronous operations are characterized by the fact that subsequent steps are executed with a time delay. Example: For a database query, it is specified that the program waits until data is available before processing it. If it is not specified in the program to wait for data, the program executes the following commands in parallel without data being present [46]. This type of JavaScript functions had to be learned first.

The use of Redux.js was also unfamiliar. Redux.js works with so-called "Reducers", in which the queried data from the database is merged and made available to the components in an organized manner. This simplifies later access and clarity over the data from the database but is a bit more complex and unfamiliar to program.

The biggest challenge during the development process was the integration of Ethereum. At the beginning of the development, it was unclear whether it was possible to store messages and user data centrally or decentrally on the blockchain. After extensive review of the technology and the goals of the application, decentralized data storage had to be ruled out. In addition, documentation for the development of a dApp is not widely available, and few developers had previously explored this type of programming. So, it was difficult to figure out how a decentralized application could be developed and to find suitable solutions to problems and errors.

## 6 Evaluative summary of the result

Cryptocurrencies have been in use since 2009 with Bitcoin and are gaining popularity. They have greatly changed the way payments are made over the Internet with decentralized networks and cryptographic processes, offering people new ways to use and dispose of money online. Moreover, compared to conventional monetary systems, cryptocurrencies are more global, free, democratic and economical.

Newer cryptocurrencies use the blockchain technology developed for Bitcoin to solve problems or create new functions. For example, for the cryptocurrency Ethereum, blockchain can be used to program decentralized applications that offer the same or new functionalities as previous financial services. These applications are called DeFi and represent a counter design to the traditional centralized financial system, called Centralised Finance (CeFi). In the innovative decentralized finance applications, there is no need for a third party to provide services and control financial flows. Compared to the traditional financial system, this saves effort and energy and makes financial transactions more anonymous.

Due to their more private, cheaper, and freer nature, cryptocurrencies offer themselves to develop an application that allows people not only to connect globally, but also to send money globally. Ethereum, as the largest cryptocurrency platform, and Ether, as the second largest cryptocurrency, are very well suited for developing such an application. In combination with the browser wallet Metamask, Ethereum can be integrated into the application in a user-friendly way, so that users can send coins to other users, but always keep control over their coins.

The application is supposed to be user-friendly and therefore does not require any prior knowledge about cryptocurrencies and Ethereum. Only for the use of Ether in the application, the user must be able to operate a wallet. To make this possible, user and chat data is stored centrally in JSON format on the *Google Firebase* database. The application runs on a React.js server so that it can react dynamically to user input and process it.

The application consists of a chat page where users can chat with each other in a one-to-one chat after registering and logging in. In the settings on the settings page, users can change their login data, but also link a Metamask wallet to the application. If the

chat partner has also linked a wallet to their application, both can send Ether to each other via the Ethereum blockchain. The transactions are executed decentrally and securely through blockchain technology.

With the latest developments from the field of cryptocurrencies, the application can still be improved or expanded. With IPFS, it could be possible to store and execute the parts of the application that have been operated centrally so far in a decentralized manner. The Filecoin project is trying to use proof-of-replication to solve problems that arise on IPFS. The effort to develop a decentralized application in every aspect could also benefit from this.

The Ethereum platform offers the possibility to program directly on the blockchain with smart contracts. With the ERC-20 standard, this technology offers the possibility to program own tokens on the Ethereum blockchain. These can, for example, represent conventional currencies such as the US dollar. The tokens can be integrated into the application and thereby enable the user to send other currencies to other users as well.

Smart contract technology can also be used in the application to allow multiple group members to save towards a goal in a group chat. This could also be implemented in a decentralized manner without a third party managing the money saved.

The application can be used to send money securely and easily, independently and efficiently to other people who also use the application. Both the application and the currency used are neutral with regard to religion, nationality, cultural origin or social status. Unlike traditional financial systems, transactions take place in a decentralized manner – they can hardly be controlled by third parties, for example authorities. New technologies still hold a lot of potential to improve and further develop the application.

## List of figures

Figure 1: Transactions in the Bitcoin network [4].....	3
Figure 2: The Bitcoin-Blockchain [4].....	4
Figure 3: Merkle Tree in Bitcoin [6].....	5
Figure 4: Server communication in the application shown in simplified form .....	16
Figure 5: React Component Message.....	18
Figure 6: Function sendETH from HomePage.....	19
Figure 7: User data in JSON-Format.....	21
Figure 8: Messages and transactions in JSON-Format.....	22
Figure 9: Visualisation of version control in IPFS according to [40].....	24
Figure 10: First sketch of the application .....	27
Figure 11: Metamask Transaction Confirmation .....	31
Figure A.12: Login/Register-Page.....	XI
Figure A.13: Chat in the app without wallet .....	XI
Figure A.14: Settings page .....	XII
Figure A.15: Linking the wallet, Step 1 .....	XII
Figure A.16: Linking the wallet, Step 2 .....	XIII
Figure A.17: Chat with linked wallet .....	XIII
Figure A.18: Transaction to other user .....	XIV
Figure A.19: Chat after the transaction has been executed.....	XIV



## Bibliography

- [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 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 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](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“, Lexus 2015  
[https://www.laenderdaten.de/energiewirtschaft/elektrische\\_energie/stromverbrauch.aspx](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] Gellersdörfer, 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](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](https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Statements/async_function) (zuletzt abgerufen am 17.04.2021)

## Appendix

### A.1 Source Code

#### A.1.1 Repository

The source code of the application is located on GitHub. There, the individual production steps of the final version of the application are stored by the version control system "Git".

The repository can be found at <https://github.com/Jufg/Chat-App-ETH>.

#### A.1.2 Auxiliaries

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 Graphics of the application

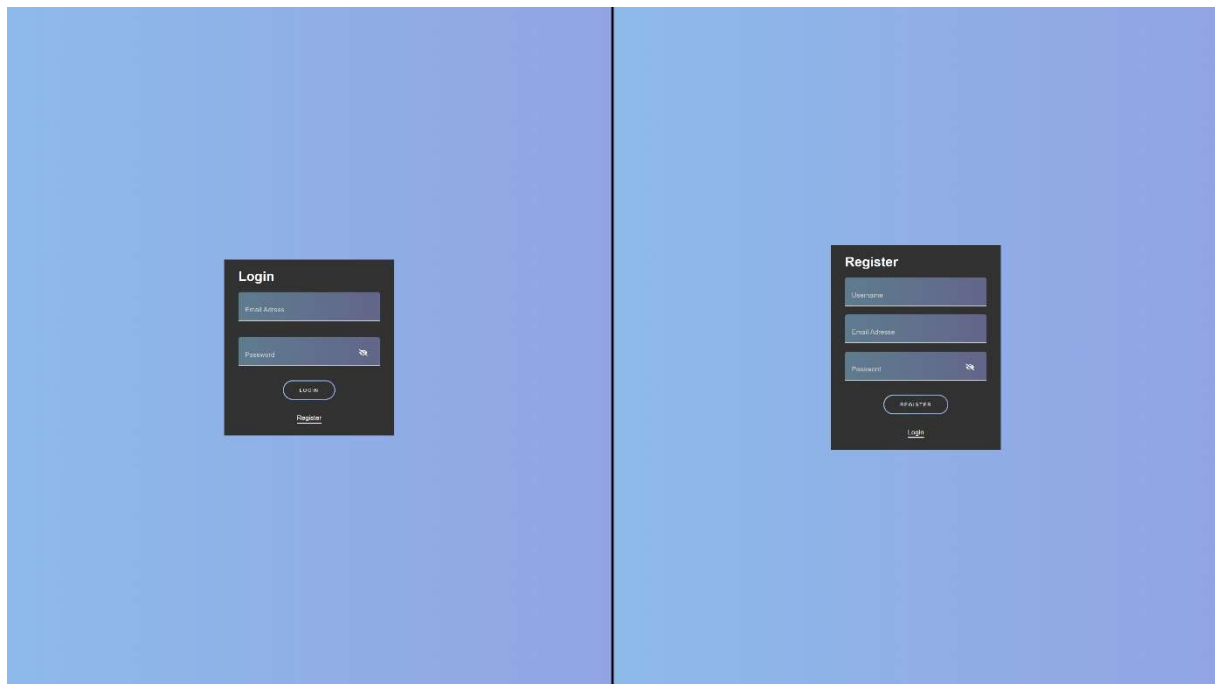


Figure A.12: Login/Register-Page

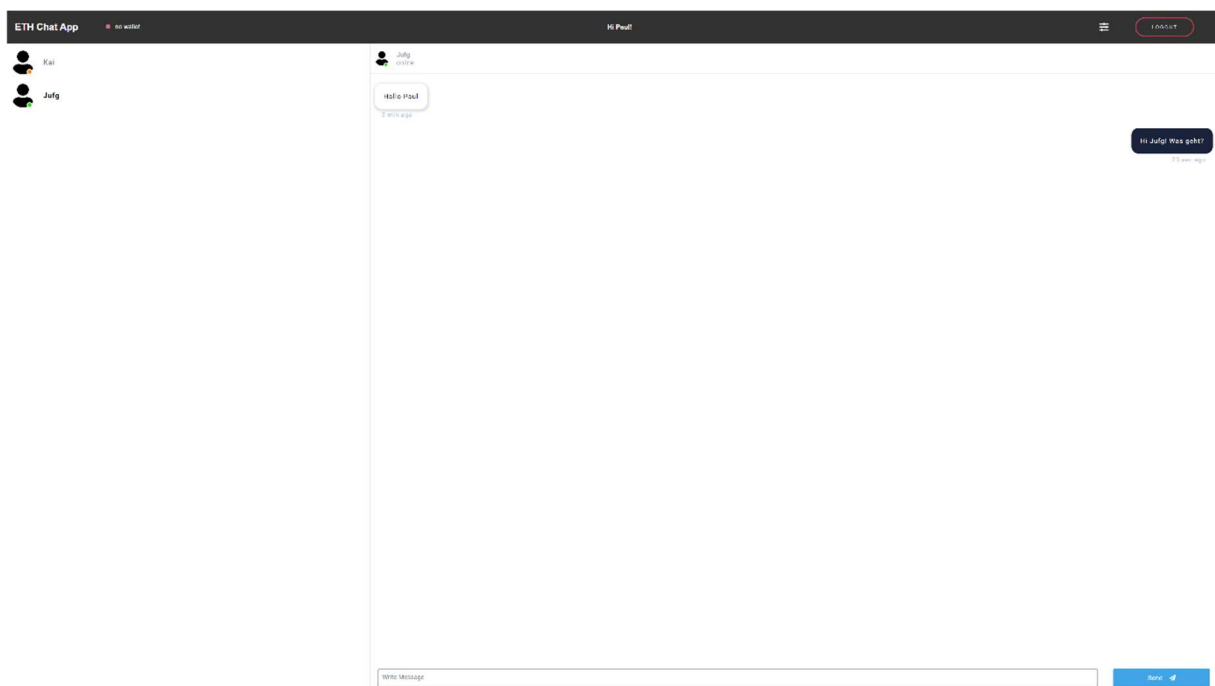


Figure A.13: Chat in the app without wallet

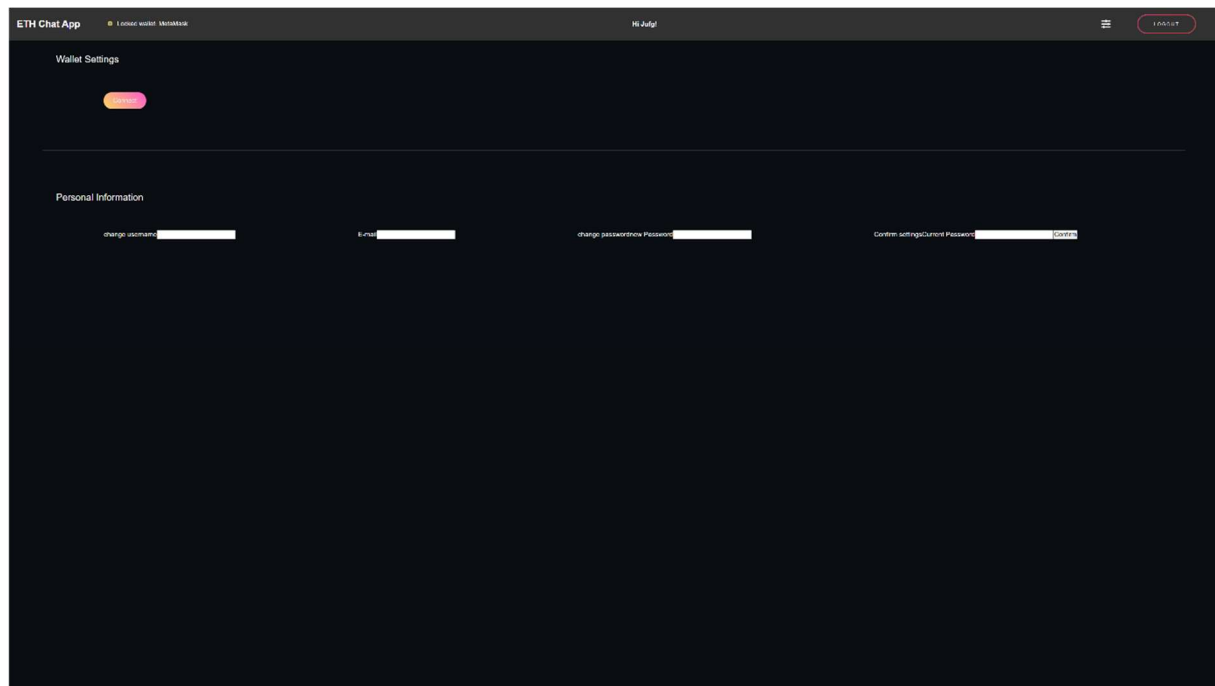


Figure A.14: Settings page

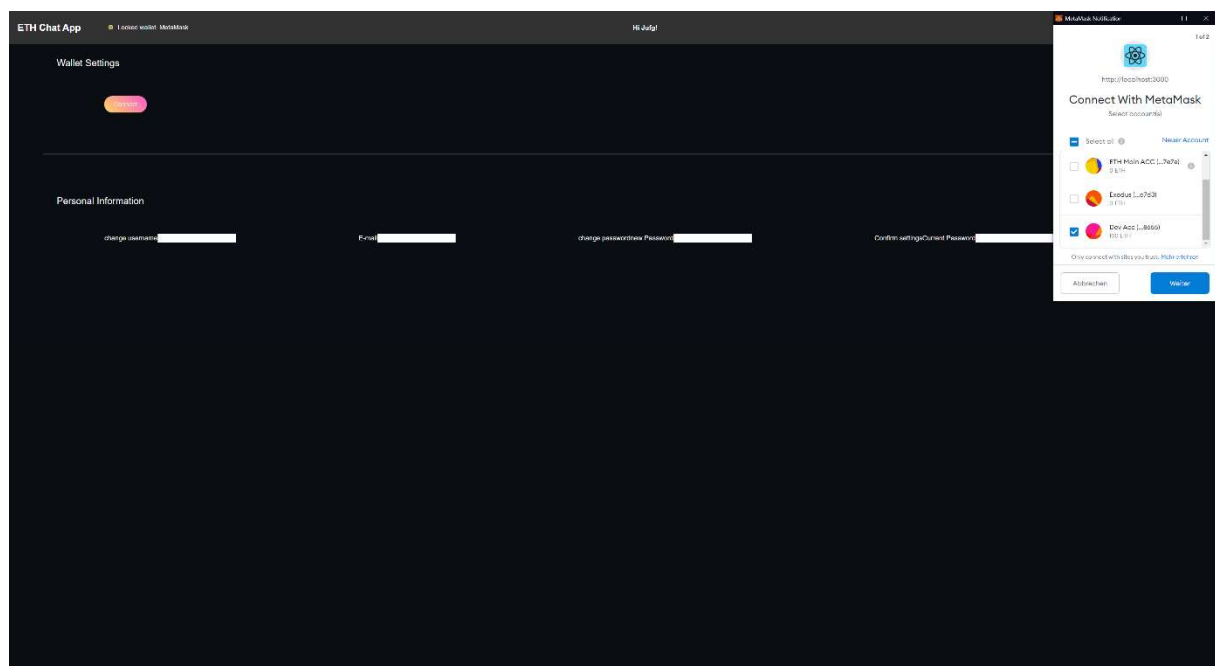


Figure A.15: Linking the wallet, Step 1

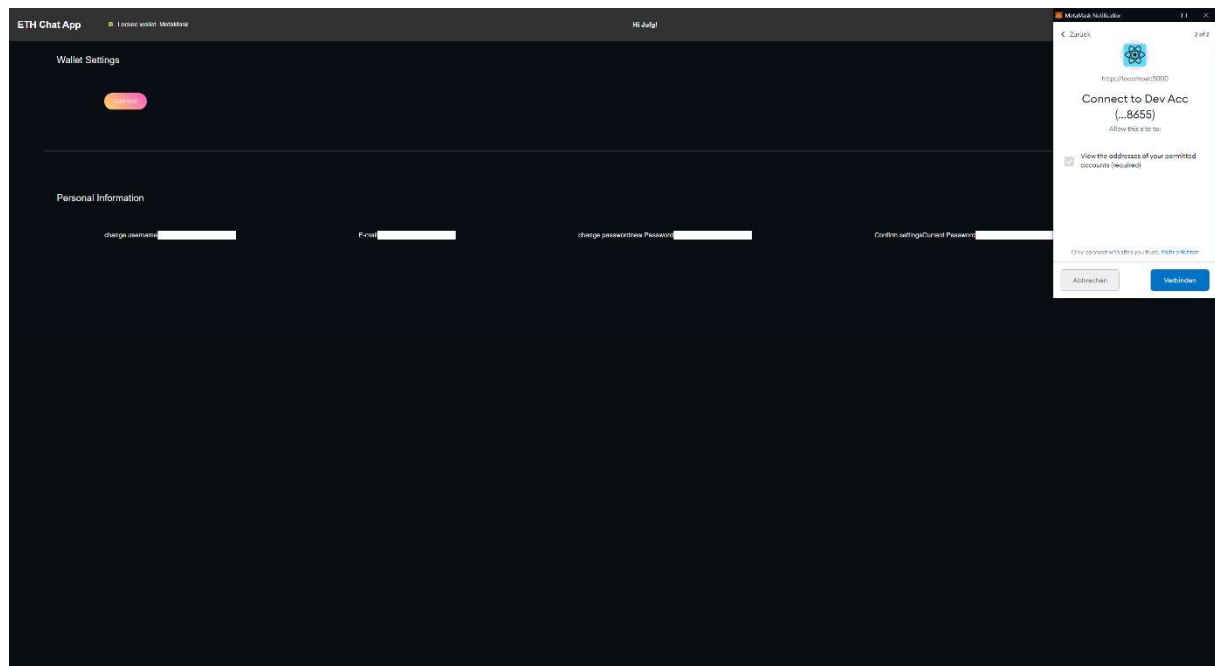


Figure A.16: Linking the wallet, Step 2

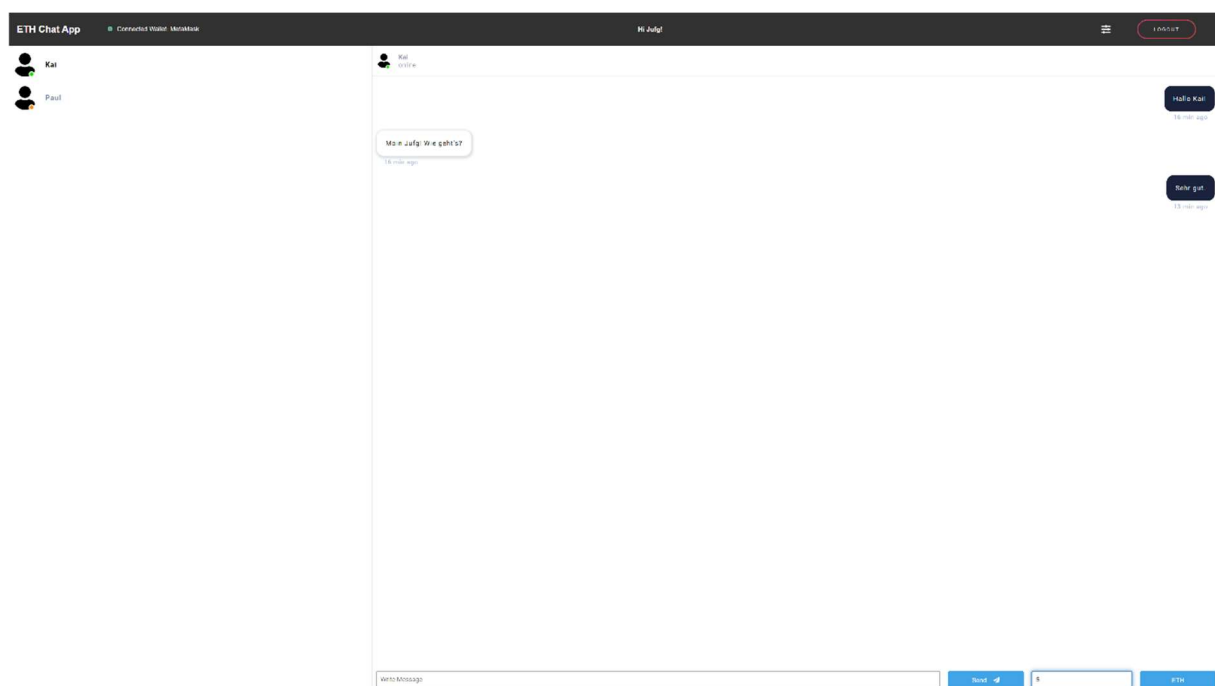


Figure A.17: Chat with linked wallet



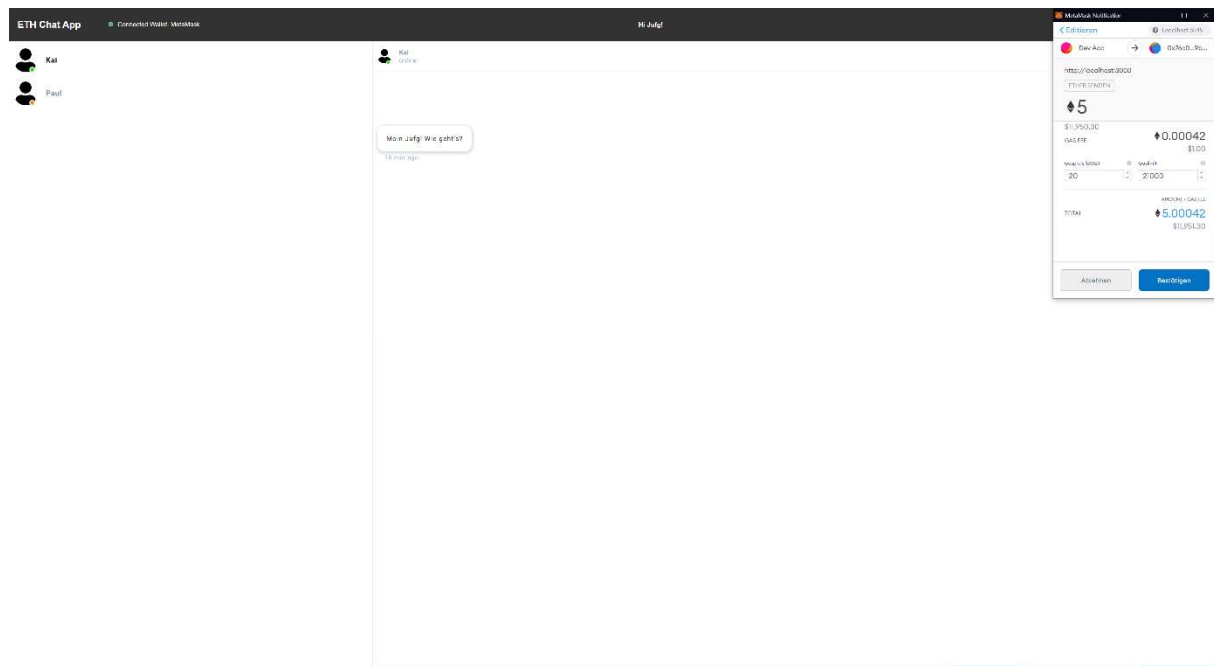


Figure A.18: Transaction to other user

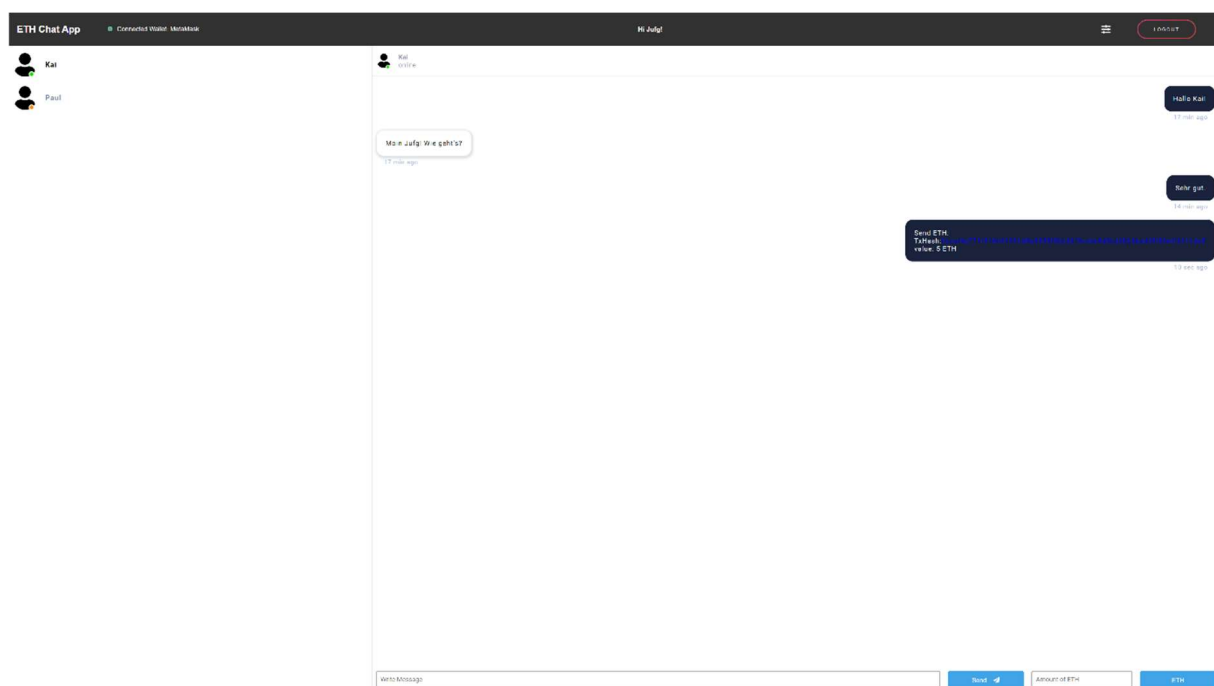


Figure A.19: Chat after the transaction has been executed

All graphics used in this work can be accessed and downloaded in full resolution via the following link:

- <https://ipfs.io/ipfs/QmQr7gpL6BzwKAwGUScBV7n8PBWQP9ZE8pKcRqtW9PCeQr>

## **Declaration of independence**

I certify that I have written this paper independently and without the help of others. I have only used the sources and aids indicated and have marked the passages taken from them, either verbatim or in terms of content. The thesis has not been submitted in the same or a similar form to any other examination authority.

Sundern, 22.04.2021

---

*Juri Kembügler*