SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Development and Evaluation of a Blockchain-Based Application for Mobile Social Payments

Lucas Grabmaier, B.Sc.

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Development and Evaluation of a Blockchain-Based Application for Mobile Social Payments

Entwicklung und Evaluation einer Blockchain-Basierten Application für Mobile Social Payments

Author:	Lucas Grabmaier, B.Sc.
Supervisor:	Professor Pramod Bhatotia, Ph.D.
Advisor:	Michael Fröhlich, M.Sc.
Submission Date:	December 15, 2022

I confirm that this master's thesis in information systems is my own work and I have documented all sources and material used.

Munich, December 15, 2022 Lucas Grabmaier, B.Sc.

Acknowledgments

While writing this thesis, I received generous support from many people. First, I would like to express my deep gratitude to my advisor Michael Fröhlich for his unconstrained support and thoughtful feedback. I would also like to thank Prof. Dr. Pramod Bhatotia for making this thesis possible by supervising it. Moreover, I am grateful for the Technical University of Munich and the CDTM that provided me with the education and tools that have made this thesis possible. Finally, I would like to thank my family and friends for their endless support during the writing of this thesis, I wouldn't be here without you.

Abstract

Social connections between humans build the fundamental structure of our modern society. From these connections emerges a network used for various possible applications - one being sending and receiving currency to transfer value between individuals. Through the digitalization of everyday life in recent years, these value transfers are often conducted through new payment infrastructures that combine social connections and value exchange. *Mobile social payment applications* are peer-to-peer (P2P) payment services incorporating social media aspects such as following friends' transactions within a feed. At the same time, cryptocurrencies like Bitcoin and Ethereum are experiencing significant adoption and struggle to scale with the rising demand. New, so-called Layer 2 protocols have emerged and promise to solve these scaling challenges. However, as most HCI research focuses on Bitcoin and Ethereum, the effectiveness of these new protocols, which promise to enable cryptocurrencies to be usable as a means of everyday payment, is yet to be verified. Therefore, we implemented a P2P social payment application using Layer 2 scaling solutions and evaluated it in a mixed methods study. Our results show that it is possible to build a usable payment system on the Layer 2 protocol Polygon and that privacy concerns arise from social aspects incorporated in the application. We discuss quantitative usability scoring as well as qualitative insights and derive recommendations for building social payment applications using cryptocurrencies.

Outline

Chapter 1: Introduction
The first chapter describes the motivation behind this thesis and provides the problem statement. The proposed solution and the methodology used are outlined.

Chapter 2: Related Work
The second chapter discusses related work and elaborates on important concepts regarding cryptocurrencies as a means of payment and social awareness streams.

Chapter 3: Requirements Elicitation
The third chapter elicits the requirements for the proposed system. It identifies actors and scenarios in order to derive functional requirements and use cases. Additionally, the chapter outlines nonfunctional requirements for the proposed system.

Chapter 4: System Design
The fourth chapter presents the system design for the proposed system and the design choices that led to this architecture.

Chapter 5: Implementation
In the fifth chapter, the process for the prototypical implementation of the proposed system is described.

Chapter 6: Evaluation
The sixth chapter evaluates the built system. A quantitative analysis of the system's usability is conducted based on survey results from a user study. Furthermore, a qualitative analysis of the system's usability is conducted based on user interviews. Finally, the implementation is contrasted with the set requirements.

Chapter 7: Conclusion and Future Work
The final chapter discusses the system critically and derives key learnings for building blockchain-based social payment applications. Additionally, we contrast our findings to those of related studies and provide directions to improve the system further and conduct further research.

Acronyms

ANOVA Analysis of variance

API Application Programming Interface

DeFi Decentralized Finance

ERC-20 Ethereum Request-for-Comments-20

EVM Ethereum Virtual Machine

HCI Human-Computer Interaction

IaaS Infrastructure as a Service

P2P Peer-to-Peer

PoS Point of Sale

REST Representational State Transfer

RPC Remote Procedure Call

SAS Social Awareness Streams

SDK Software Development Kit

SUS System Usability Scale

UEQ User Experience Questionnaire

UI User Interface

Contents

1 Introduction

The following chapter introduces the research described in this thesis and gives an overview of the proposed solution and its evaluation.

1.1 Motivation

Cryptocurrencies have experienced substantial growth since the introduction of Bitcoin as a "Peer-to-Peer Electronic Cash System"[1] in 2008. The increasing adoption rates [11] and reports that more than 300 million people own cryptocurrencies[2] led to comparisons with the early days of the Internet[3]. This growth spiraled into a combined market capitalization of all cryptocurrencies of $2.9 trillion in November 2021[4] and a prevailing understanding of cryptocurrencies as a speculative investment class [1] [45].

The high volatility in the market capitalization that has since decreased by 72% and brought the discourse back to the fundamental applications and use-cases of this novel technology[5]. Decentralized governance through DAOs, proof of identity concepts like ENS, and decentralized finance (DeFi) with prediction markets and synthetic assets are all exciting applications of cryptocurrencies. Nevertheless, using cryptocurrencies as money and a means of payment is still the number one use case according to Ethereum founder Vitalik Buterin[6].

In western countries, wider adoption of cryptocurrencies as a means of payment has not happened yet. There, the dominant means of payment for Point-of-Sale (PoS) transactions are still cash and credit cards [11]. However, what stands out is that for Peer-to-Peer (P2P) transactions of daily life, payment services such as Venmo, CashApp, and PayPal, have gained significant adoption as a means of payment in the western world [63]. Their simple value proposition of enabling users to send instant

[1] https://bitcoin.org/bitcoin.pdf (last accessed: 12.12.2022)

[2] https://blog.crypto.com/global-crypto-owners-near-300-million-predicted-to-hit-1-billion-by-the-end-of-2022/ (last accessed: 12.12.2022)

[3] https://archive.nytimes.com/dealbook.nytimes.com/2014/01/21/why-bitcoin-matters/ (last accessed: 12.12.2022)

[4] https://coinmarketcap.com/charts/ (last accessed: 12.12.2022)

[5] https://techcrunch.com/2022/10/28/crypto-bear-markets-are-a-great-time-to-launch-startups-industry-execs-say/ (last accessed: 12.12.2022)

[6] https://vitalik.ca/general/2022/12/05/excited.html (last accessed: 12.12.2022)

digital transactions without transaction fees between individuals convinced consumers who were annoyed by slow bank transfers with potentially high fees [48]. Another differentiating factor from traditional banking is the social aspect that P2P payment providers incorporate into their services. Initially emerging in social networks, social awareness streams (SAS) display a stream of content from friends or other related contacts with the intent of increasing engagement and retention with the product in network-driven systems [10]. Venmo, introduced one year after Bitcoin in 2009, is a dominant player in the US market, with a total volume of $230B in 2021[7], features such a social awareness stream. Many argue their success stems from them adopting the concept SAS into their application which led to their payment application, in some instances, being perceived as a form of social media instead of a pure payment provider [2].

Established cryptocurrencies like Bitcoin and Ethereum[8] fulfill many properties of money [45] but struggle to find adoption as a means of payment [39]. Primarily due to technical constraints leading to slow transaction speeds [54], high fees [65], and volatility in the transferred currency [24]. But also due to sustainability concerns originating from the energy consumption of proof-of-work (PoW) blockchains like Bitcoin [27] [66]. So-called Layer 2 networks, emerging in recent years, arguably address these issues [30]. Blockchains like Bitcoin Lightning[9] or Polygon[10] lead with the promise to solve the scalability challenges of their Layer 1 pendants, making them more suitable for everyday payments. This promise originates in providing transaction settlements at near real-time speeds, and low transaction costs [23].

However, to this date, these technological capabilities could not be used to build a product that can substantially compete with established P2P payment solutions. Systems like Venmo or PayPal still provide a better value proposition supported by strong network effects [23] [46].

1.2 Problem Statement

Layer 2 solutions, like Bitcoin Lightning or Polygon, promise to solve the technical limitations of Bitcoin and Ethereum, which are constrained in their applicability to P2P payment use cases because of these limitations [54]. Solving them would enable Layer 2 solutions to offer comparable transaction speed and fee performance and additionally to leverage cryptocurrency ecosystems to build additional functionality like SAS on top

[7] https://www.statista.com/statistics/763617/venmo-total-payment-volume/ (last accessed: 12.12.2022)
[8] https://ethereum.org/en/whitepaper/ (last accessed: 12.12.2022)
[9] https://lightning.network/lightning-network-paper.pdf (last accessed: 12.12.2022)
[10] https://polygon.technology/lightpaper-polygon.pdf (last accessed: 12.12.2022)

of it. With that, cryptocurrencies could become a viable medium for P2P transactions in daily life [32]. However, while technically possible on paper, such Layer 2 protocols have yet to be explored within the use case of a payment system. There is no empirical evidence of whether solving the technical limitations of Bitcoin and Ethereum enables a system to be used as a means of payment in practice. Furthermore, it is unclear whether such a system would prove to be a viable alternative to established payment systems like Venmo regarding ease of use and reliability. Previous research exploring Lightning as a payment system is only focussed on the customer-to-merchant interaction of PoS use cases, not taking into account the social aspect of P2P transactions [23]. Therefore, this thesis aims to suggest and develop a reference implementation of a social payment system and evaluate it in a real-life context. We aim to leverage the technical capabilities of Layer 2 protocols to provide a blockchain-based P2P payment solution. This approach addresses the current challenges users have with using cryptocurrencies as a means of payment in everyday situations and aims to explore the appeal of social aspects that make social payment systems like Venmo unique.

1.3 Proposed Solution

We propose a reference implementation for a peer-to-peer social payment system using the Layer 2 protocol Polygon as the settlement layer. The system enables users to create a non-custodial cryptocurrency wallet on their smartphones. Further, it provides the functionality to send and receive transactions using a cryptocurrency without being subject to significant volatility. Additionally, the system generates a feed of the transactions of all contacts that a user specifies as their social circle. Our proposed system, which will be referred to as "wallet" from here on, consists of:

- a mobile application acting as the user interface to engage with the wallet's functionality
- a transaction engine allowing the user to interact with the blockchain to send and receive transactions
- a feed engine generating the wallet feed for users

Besides the user interface, the mobile application entails storing the cryptographic keys that make up the wallet. The transaction engine interacts with the blockchain by communicating with a blockchain node and manages any data stored off-chain in the cloud. The feed engine generates the wallet feed on the device, which is possible due to using pre-computed blockchain data gathered by interacting with a blockchain querying service.

1.4 Research Goal

The purpose of this thesis is to show the feasibility of building a mobile payment application using Layer 2 blockchain technologies and further evaluate whether this application provides a viable alternative to other social payment applications, such as Venmo, for everyday use.

This results in the following research questions:

1. Is it possible to build a mobile payment app using Layer 2 blockchain technologies as a viable alternative to established applications?
2. How do users interact with such an app when using it as a payment service in their everyday life?
3. Do "social payment features" such as a wallet feed add value to a mobile payment app?

To test the assumption that it is possible to build such an application, the reference implementation proposed is implemented in the form of a prototype. The system design and choices made for implementing the system are described in chapters four and five. The evaluation done to assess the usability of the system and viability as a social payment application is described in chapter six.

1.5 Methodology

We make use of an empirical approach to develop and evaluate the proposed solution. Figure 1.1 displays the process that we used to develop and evaluate the wallet over a time frame of six months. The rows represent the respective activities done in a phase. We started with a literature review that spanned over two months. At the same time, the requirement elicitation, and analysis and design phases were started to initiate the development process. Afterward, the system was implemented. The evaluation of the prototype was conducted using two studies. In the field study, the prototype was deployed to a class of students during two weeks of a course at the Center for Digital Technology and Management. In the lab study, six think-aloud interviews were conducted over the course of two weeks. After all studies were completed, the final phase encompasses an analysis of the gathered data and documenting the results.

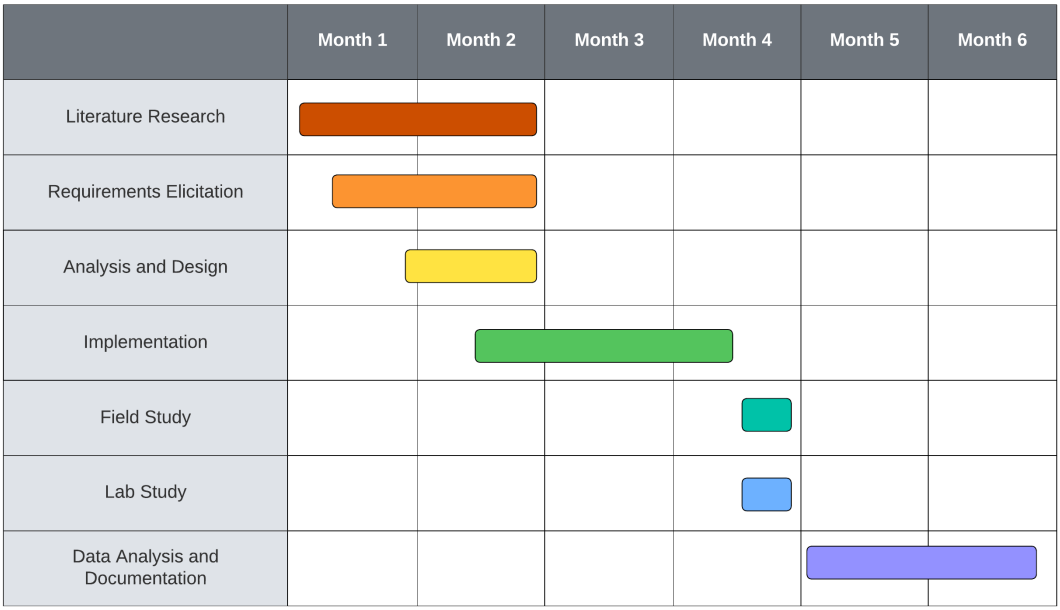

Figure 1.1: The process used to develop the payment system

2 Related Work

This section discusses selected literature and related projects.

2.1 Cryptocurrencies as a Means of Payment

This thesis builds on the findings of previous research on cryptocurrencies. The HCI community has been interested in the concept of cryptocurrency and published various findings in recent years. Exploring the intent of cryptocurrency users shows that, for most, the primary motivation to engage with cryptocurrency are the investment opportunities associated with it [28] [35] [65]. However, users expressed the wish to use their cryptocurrency as a means of payment [26]. In their perception, cryptocurrencies fulfill many properties of money [45], but they do not find any convenient opportunity to use it as such [26]. For daily Point of Sale (PoS) purchases, there is a lack of adoption by merchants [28], and also in personal peer-to-peer (P2P) transactions, traditional services, like PayPal, dominate[1]. Despite this growing ask for crypto payments by users, there are only very few studies exploring cryptocurrencies as a means of payment, especially in the personal P2P space. Those that exist focus on Bitcoin or Bitcoin Lightning, not other Layer 2 scaling solutions.

2.1.1 Bitcoin

This focus on Bitcoin stems from the fact that it was the first cryptocurrency and combined the technological concepts necessary to prevent the double-spend problem. Initially, Bitcoin was introduced as *"peer-to-peer electronic cash"*[2]. However, in recent years it did not establish itself as a means of transaction or payment method in any significant way [26] [39]. The reasons are, on the one hand side, Bitcoin's technical constraints, causing slow transaction speeds and high transaction costs. On the other hand, its subpar usability for daily transactions stems from a complicated transaction process and volatility in the transacted asset. Lastly, the increasingly scrutinized energy consumption of proof-of-work (PoW) blockchains like the Bitcoin network is causing climate concerns and the call for more efficient networks [27] [66].

[1] https://www.statista.com/forecasts/997123/peer-to-peer-payments-in-the-us (last accessed: 08.12.2022)
[2] https://bitcoin.org/bitcoin.pdf (last accessed: 08.12.2022)

The technical constraints causing slow transaction speeds and high fees are rooted in the comparably long block time of about 10 minutes. Resulting that is a slow transaction speed, as multiple block confirmations, i.e., multiples of the 10 minutes block time, are required to confirm a transaction, depending on the recipient [24] [32] [54]. Compared to solutions from the traditional finance space, this lack of speed is even more evident. Bitcoin can handle about 7 transactions per second (TPS) whereas PayPal handles on average 170 TPS and Visa on average 2000 TPS with a maximum of 56000 TPS [29]. The high transaction fees originate in its network approaching the maximum possible TPS as transaction fees increase with the utilization of the network. These surging fees make small transactions, such as P2P transfers or PoS exchanges with a merchant, economically unfeasible [65].

Froehlich et al. argue that Bitcoin's usability issues hold back its adoption further [23]. Its perceived usability is worse than traditional means of payment such as credit cards or PayPal [3]. Users expect the payment process to be "as easy as with PayPal" and are unwilling to engage with a payment process perceived as complicated and manual [24]. The volatility of Bitcoin further worsens its usability as a means of payment [24]. Bitcoin is perceived as a speculative investment asset class rather than a currency [38]. Thus, its volatility[3] is higher than in other asset classes such as stocks and much higher as currencies like the Euro or US-Dollar [44]. This volatility is detrimental to the user experience as funds received from another party could be worth substantially less already a day later.

In conclusion, Bitcoin's slow transaction speeds, high fees, and poor usability make it unsuitable to fulfill its intended use case as *"peer-to-peer electronic cash"*[4]. Moreover, while some usability challenges can be solved through future iterations and more user-friendly interfaces to the underlying technology, the core technical properties cannot be changed and make Bitcoin ill-suited for real-time purchases [24].

2.1.2 Ethereum

Ethereum[5], the cryptocurrency which introduced the opportunity to deploy smart contracts on the blockchain, solves some of the stated issues. The smart-contract capability enables the creation of ERC-20 tokens, which are used to create so-called stablecoins. According to Mita et al., a stablecoin can be defined as a "cryptocurrency whose value aims to be pegged with a given underlying asset" [47]. The peg to an underlying asset, such as the US dollar, acts as a stabilization mechanism to offer minimal volatility making stablecoins better suited as a currency [47]. They also lower

[3]https://buybitcoinworldwide.com/volatility-index/ (last accessed: 08.12.2022)
[4]https://bitcoin.org/bitcoin.pdf (last accessed: 08.12.2022)
[5]https://ethereum.org/en/whitepaper/ (last accessed: 08.12.2022)

the barrier to entry for new users, according to Voskobojnikov et al., as cryptosystems using stablecoins mimic established payment systems better than systems with a volatile currency like Bitcoin [65]. While Ethereum's capabilities and the resulting possibility to transfer stablecoins arguably solve the volatility issue of Bitcoin itself deals with similar scalability issues [15]. These result in transaction speeds and fees just as unsuitable for daily payments as Bitcoin's with 30 - 40 TPS [49].

2.1.3 Scaling Solutions

To address the scalability issues of Bitcoin and Ethereum, so-called Layer 2 networks have emerged over the past years [30]. Chains like Bitcoin Lightning[6] or Polygon[7] lead with the promise to solve the scalability challenges of their Layer 1 pendants, making them more suitable for everyday payments. This promise originates in providing transaction settlements at near real-time speeds and low transaction costs [23].

Lightning enables Bitcoin transactions to settle off-chain through a layer of P2P transaction channels built on top of Bitcoin. Any number of transactions can be done through such a channel, whereas only the first and the last transaction are written to the blockchain [68]. This significantly improves the speed and costs of any transaction within those channels. Yet, whenever the funds in such a channel should be sent to another party, as it would happen for personal transactions, a transaction on the Bitcoin main layer happens, which again falls under Bitcoin's scalability issues. This would happen regularly for everyday transactions between friends as a person would most certainly want to use the funds received for another objective other than sending it back to the same person at some point in the future. Furthermore, Waugh et al. criticize the reliability of the Lightning network, finding that transactions "fail much too often, in particular when sending larger payments in excess of USD 50" [68]. Finally, Lightning invoices pose another usability challenge for P2P transactions. By design, the recipient of a Lightning transaction must create a so-called Lightning invoice specifying the amount before the sender can send a transaction by filling said invoice.

Similar Layer 2 scaling solutions also exist for the Ethereum blockchain. There are many different approaches, from rollups such as Arbitrum[8], Optimism[9], or ZK-Sync[10], to Layer 2 sidechain solutions like Polygon[11]. All of them enable a scale sufficient for

[6] https://lightning.network/lightning-network-paper.pdf (last accessed: 08.12.2022)
[7] https://polygon.technology/lightpaper-polygon.pdf (last accessed: 08.12.2022)
[8] https://developer.offchainlabs.com/intro/ (last accessed: 08.12.2022)
[9] https://community.optimism.io/docs/how-optimism-works/ (last accessed: 08.12.2022)
[10] https://v2-docs.zksync.io/dev/fundamentals/zkSync.html (last accessed: 08.12.2022)
[11] https://polygon.technology/lightpaper-polygon.pdf (last accessed: 08.12.2022)

everyday P2P transactions with, e.g., 7000 TPS in the case of Polygon[12], which is well above the average 170 TPS of PayPal [29].

As they all offer different benefits and drawbacks and are essential for the implementation of this thesis, the different Ethereum scaling solutions will be evaluated in detail in chapter 5.1 [30]. In addition to arguably solving the scalability problems of Ethereum, they do not deal with the usability challenge of invoices nor the volatility of Bitcoin as Lightning both does. The technical approaches to scaling differ from Lightning's. This enables all mentioned solutions to send transactions without any invoice generated beforehand, but just with the recipient's address, like on Ethereum. Furthermore, they support stablecoins just like Ethereum, eliminating any volatility [47].

2.1.4 Existing Applications of Layer 2 Scaling Solutions

The technical improvements around scaling solutions were only built in recent years. Therefore, most human-computer interaction (HCI) research evolves around Bitcoin [25]. As these developments fundamentally change the opportunities application range of cryptocurrencies as a means of payment, further research should be done on implementations of Layer 2 technologies [30].

Froehlich et al. contribute a reference implementation for a PoS payment system using Bitcoin Lightning to the research body [23]. Their evaluation of the system in a mixed methods study shows that "Bitcoin Lightning is a usable alternative to traditional solutions" but "setting up the wallet and initially acquiring Bitcoin was, however, prone to different challenges" [23]. While the system proved usable with an acceptable experience in terms of transaction speed and fees, Froehlich et al. confirm Lightning's drawbacks of volatility and usability issues in their study. The participants note that they find Lightning's invoice process cumbersome, and there is no benefit to Lightning at the UX level [23]. Especially compared to services like PayPal, which do not charge any transaction costs, and offer wide acceptance, fiat currencies without volatility, and buyer protection. The mentioned volatility was a 15 percentage point difference between the highest and the lowest transaction versus the mean transaction price over the two-week study [23]. Some participants mentioned they perceive the volatility negatively and that "not knowing how much I can buy in the future" stresses them out [23]. Additionally, there are some commercial implementations, such as the Tips feature of the social media platform Twitter[13].

Once set up, users can tip other users with Bitcoin Lightning. The discussed usability issues, such as invoices, are here abstracted by Twitter and the Strike Wallet, which

[12]https://polygon.technology/solutions/polygon-pos/ (last accessed: 04.12.2022)
[13]https://help.twitter.com/en/using-twitter/tips (last accessed: 09.12.2022)

recipients must hold to use the feature. According to an unverified article by Coingeek, the feature is being shut down at the time of this writing due to low demand, i.e., only being used by 1348 out of Twitter's 200 million users, amounting to a total transaction volume of $8500[14].

Besides this study of Lightning and the shut-down Twitter Tips feature, we could not find other HCI research evaluating payment systems' implementations based on Layer 2 scaling solutions. This requires further research to understand, first, whether Layer 2 scaling solutions based on Ethereum, such as Polygon, can deliver the stated facts of low volatility through stable coins and a scalable settlement layer. Second, whether, if viable in practice, these scaling solutions provide an alternative to established payment systems.

2.2 Social Payment Applications

The broad adoption of digital payment solutions like PayPal, Venmo, or CashApp has fundamentally changed how people conduct their personal finances [63]. These P2P payment applications enable users to send instant digital transactions without transaction fees between individuals [48].

Existing research evaluating people's experiences with such apps focuses primarily on factors other than the social dynamics behind the transactions [14] [16]. Aspects that promote their use, the functional design of such applications [2] [13] [37], or the technical concepts supporting it [36] [60], are discussed extensively by the research community.

2.2.1 Social Awareness Streams

Social aspects are a topic well worth exploring in the context of digital payments, as financial activities influence the social dynamics between people and how they approach their relationships. Motivation and personal behavior towards others improve when a financial transaction is involved in the interaction [64]. However, financial transactions can either impede or reinforce the development of interpersonal relationships. Potential reasons are that people are less engaged when using digital means of conducting a financial transaction [53], and as Wherry et al. show, there is possible awkwardness and potential negative reciprocity associated with lending or borrowing behaviors resulting in worsened personal relationships [69].

[14] https://coingeek.com/twitter-tips-on-lightning-network-fizzles-out-after-1-year-due-to-low-demand/ (last accessed: 09.12.2022)

Wagner et al. define social awareness streams (SAS) as an aggregation of multiple personal awareness streams which "allow users to post short, natural-language messages as a personal stream of data that is being made available to other users" [67]. For mainstream social networking applications like Instagram and Twitter, SAS are the primary product users interact with. Nevertheless, they are often also incorporated in task-based systems such as music streaming, e.g., Spotify, fitness tracking, e.g., Strava, and also P2P payment systems, e.g., Venmo [12].

SAS are predominantly designed to foster the habitual use of a system and help people to share and connect with each other. E.g., by following the life of friends through a stream of photos and posts. According to Burke et al., implementing a SAS in network-driven systems can increase user engagement and retention [10].

Because using applications of SAS often involves disclosing personal data, like, e.g., one's location, there is often a conflict for users of providing content to others and securing one's privacy [19]. Furthermore, this desire for privacy "is often in tension with a desire to present one's self honestly and accurately" according to Zhao et al. [71]. Silfverberg et al. present the example that on Last.fm, some users stream music they would not otherwise, to give a particular image of themselves [59]. In contrast, this privacy concern "typically fades as people become used to disclosing personal data" according to [70].

2.2.2 Existing Applications of Social Awareness Feeds

Findings in the research of SAS can be used to shape the design of social features in domains where they are not as common, such as in the payments space [12]. Evaluating people's use and perception of social features can provide developers with crucial information to build an experience worthwhile for users and with the potential of high engagement through SAS [31].

One already existing application of SAS in the payments space is the P2P social payments app Venmo. It is one of the most popular payment applications, with a transaction volume of $230B in 2021[15], and prevalent amongst people in their teens, 20s, and early 30s[16]. Traditionally, services featuring a SAS like, e.g., social media applications, are ritualistic, meaning users open the application to browse and are directed by the content without a concrete intent [51]. Hiniker et al. discovered that, in contrast to that, Venmo is primarily used instrumentally with a clear intent or task in mind, such as sending a specific friend a specific amount of money [31]. Consequently, contributing to the SAS of Venmo is only possible by actually using the application's

[15] https://www.statista.com/statistics/763617/venmo-total-payment-volume/ (last accessed: 08.12.2022)
[16] https://www.washingtonpost.com/news/the-intersect/wp/2015/02/26/why-would-anyone-in-her-right-mind-use-venmo/ (last accessed: 08.12.2022)

core functionality: sending and receiving transactions. Additionally, a core difference between Venmo's and other SAS is that the content of the feed is financial transaction data that people traditionally find too private to share with strangers [34]. The data in this feed is publicly shared by default. This can be changed by the user and creates three distinct feeds: A private feed that only the sender and recipient have access to, showing additional details like the amount of a transaction. A feed with friends' transactions and the completely public feed do not show transaction amounts. Each feed features engagement functionality, such as commenting and liking transactions.

In their study "Friends Don't Need Receipts: The Curious Case of Social Awareness Streams in the Mobile Payment App Venmo" Caraway et al. evaluate the application of SAS within mobile payment applications by surveying users of such and interviewing users of Venmo specifically.

They find that the uses and experiences of Venmo echo findings on SAS in, e.g., applications of ritual use such as social networks. Another study from Acker et al. examining Venmo explicitly in the context as a form of social media confirms this finding [2].

Similar to other SAS in Johnson's study [33], "many participants were indifferent to their settings for who could see their transactions" [12]. In contrast, most participants indicated they had at least some privacy concerns. These specifically deal with others having access to a persistent history of their transactions. Some pointed out the paradox that they enjoy following their friend's transactions but would not want their friends to see their own transactions. In the end, even the participants concerned about privacy pointed out that they would still choose the payment app that is most convenient to them and, if necessary, adjust any available privacy settings.

Like in social networks, participants of the study cared a lot about how others perceived them [71]. They aimed "to perform in the SAS, taking time to craft transaction description to appear as clever." [12].

People new to Venmo were observed to experience instances of social learning. Meaning they learned about the norms of using Venmo, such as, e.g. often obfuscating the transaction description by only using emojis by observing others using Venmo through the SAS [4].

One of the primary goals of SAS in social networks, making people feel more connected, was observed only in a few cases where people described that the incidental encounters in the SAS made them "feel more connected to people they care about" [12].

In that context, it was discovered that participants use the application differently for transacting with close friends or family and mere acquaintances. With strangers or acquaintances, the SAS is used as a record to document transactions with very descriptive notes. On the other hand, the transaction notes were more playful for family

and friends, with hidden messages through emojis, and resembled typical social media posts [12].

Resulting from these findings are insights that could inform the requirements of future payment applications featuring SAS. Caraway et al. recommend that "designers should consider what content adds to the SAS and what is unnecessary or might even hinder use" [12]. In detail, they mention the example of the transaction amount not being displayed for anyone other than the sender and recipient, which according to them, results in relatively low privacy concerns, considering the feed contains financial data. Further, they recommend developers to "consider what tone they want to set in their app, and how the tone augments the functional purpose or detracts from it". Concretely, they mention the prominent use of emojis in the application to make the app fun and less tense, considering the economic context, for a young demographic. However, they caution against blindly following these recommendations as users of different cultures might perceive those design decisions differently, and privacy norms differ strongly from culture to culture [12].

3 Requirements Elicitation

In this chapter, we define the requirements for the system to be built. Requirements engineering aims to develop a shared understanding of the desired outcome between all participating engineers and third parties such as potential clients. A requirement in this context can be described as a feature a system must have to be accepted by the client [9].

We introduce the actors using the system and their respective roles. Furthermore, the scenarios in which the actors interact with each other will be defined. Finally, we elicit the functional and non-functional requirements for the system and identify its use cases.

3.1 Actors

Figure 3.1 shows the identified actors interacting with the system: Sender, Receiver, and Observer.

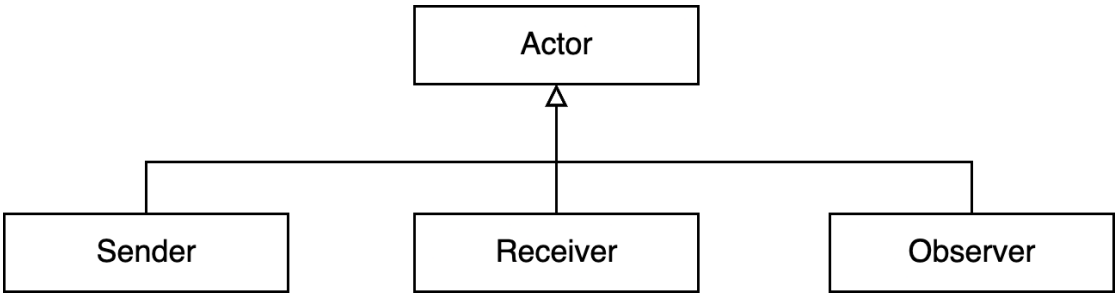

Figure 3.1: A taxonomy showing the three identified actors of the system

The *sender* wants to send a monetary transaction from his wallet as a private individual. He cares about the transaction arriving safely and with the amount he specified. Furthermore, he should be able to attach a note to the transaction to explain it.

The *receiver* can be a private individual or a business. They are interested in being able to receive transactions into their wallet and have them stored there securely. They also want an overview of their current balance after receiving a transaction.

The *observer* wishes to follow the transactions of their friends. She cares about what her friends are currently doing and is interested in the social dynamics the transactions might suggest. She should be able to get an overview of those transactions and more detailed information, such as the date the transaction was sent and if the sender attached any other information.

3.2 Scenarios

Scenarios are an illustration of interactions between actors using the future system. Developers use them to describe concrete events as a tool to communicate with potential users and the client [9]. We contrast an as-is scenario with multiple visionary scenarios to outline the differences between current solutions and a future system to be developed.

3.2.1 As-Is Scenario

This section describes two As-Is Scenarios to capture the current state of solutions for the actors without our system.

As-Is Scenario 1: Sending transactions with Bitcoin Lightning

Scenario Name	Sending transactions with Bitcoin Lightning
Participating Actors	Verena: Sender, Gerhard: Receiver
Flow of Events	1. Verena is a student and has lunch with Gerhard. As he paid for both of their meals, Verena wants to pay him back 10€ for the food. 2. Gerhard is very into new tech and proposes to try cryptocurrencies to transfer the money. He has a Bitcoin Lightning wallet for fast and cheap transactions and sends Verena his wallet address. Verena also has a wallet, so she is happy to try it. 3. Unfortunately, Verena cannot directly send a transaction to Gerhard's address using Lightning and needs a concrete invoice from him. She could use the Bitcoin Mainnet, but the transaction would cost more than 10% of the funds she wants to transfer. 4. She asks Gerhard to create an invoice using his wallet. Gerhard sends the code to Verena, who now has 24h to pay the invoice before it expires. She pastes the invoice code into her wallet. The code is recognized, and Verena can send the transaction. 5. The funds arrive in Gerhard's wallet instantly. Yet, the amount is displayed in Bitcoin instead of Euros, so he is not quite sure how much Verena sent him. 6. A week later, Gerhard wants to use the amount to pay back another coworker for lunch. Unfortunately, Bitcoin crashed by 40%, and he is left with only 6€ in his wallet. He feels like peer-to-peer payments using cryptocurrencies should be easier and pays his colleague with PayPal.

Table 3.1: As-Is Scenario 1: Sending a transaction between individuals

As-Is Scenario 2: Observing friend's transactions

Scenario Name	Observing friend's transactions
Participating Actors	Lauren: Observer
Flow of Events	1. Lauren, a student from the US, is very interested in the social dynamics around campus. Besides traditional social media, she follows her friends on a payments app called Venmo. There she can observe all transactions her friends send between each other and the little notes and emojis they attach to them. These act as information between the students on which new relationships are forming and what else her friends are up to. 2. During her semester abroad in Germany, she wants to quickly integrate herself into the social life at her new university. When a fellow student bought her lunch, she offers to pay them using Venmo, also partly to be included in that part of social media on campus. 3. The other student explains that Venmo does not exist in Germany and that they use PayPal. PayPal works similarly to Venmo, but the transactions and little notes of her friends that Lauren enjoys observing are only visible between the individuals sending and receiving the transaction. 4. Lauren proceeds to use PayPal but is wondering why this feature that she enjoys so much in the US is not offered by the payment application used in Germany.

Table 3.2: As-Is Scenario 2: Observing friend's transactions

3.2.2 Visionary Scenarios

This section describes three Visionary Scenarios to give an outlook on a future system with a non-custodial Layer 2 wallet to help the actors achieve their goals more efficiently.

Visionary Scenario 1: Sending transactions with the wallet

Scenario Name	Sending transactions with the wallet
Participating Actors	Verena: Sender, Gerhard: Receiver
Flow of Events	1. Verena and her friends go out to eat at her favorite pizza place. Because she forgot her wallet Gerhard picks up the check. 2. After their unpleasant experience with Bitcoin losing its value, Verena looked into other cryptocurrency payment apps as she is still interested in the technology in general. She found the wallet online. 3. Gerhard can easily download the wallet and set it up in minutes as it is non-custodial and does not require any lengthy signup process. He shows Verena his wallet QR code which she scans with her wallet. She adds Gerhard as a contact and sends him a transaction of 10€. Because she liked the pizza so much, she adds the note "Best pizza ever!" and an emoji to the transaction for Gerhard to see. 4. Gerhard instantly receives the transaction with minimal fees, just like with Lightning. This time he does not have to worry about losing his money as the wallet works with the stablecoin USDC, which is always pegged to the US Dollar.

Table 3.3: Visionary Scenario 1: Sending transactions with the wallet

Visionary Scenario 2: Receiving money from friends

Scenario Name	Receiving money from friends
Participating Actors	Gerhard: Receiver
Flow of Events	1. Gerhard and his friends are going to a concert together. To simplify the organization, he bought the tickets for the whole friend group. 2. He wants them to send him the money for the tickets with the wallet payment app, so he sends them his wallet address and the amount in their group chat. As was the case with Bitcoin Lightning, he does not need to create an invoice for every one of his friends individually. 3. His friends enter the amount on the wallet home screen and paste his address. They add a note such as "Thanks for organizing the tickets!" and send him the money for their ticket. 4. Gerhard receives the funds instantly, with minimal fees, and without setting up the payment with every friend individually. They have a great time together at the concert.

Table 3.4: Visionary Scenario 2: Receiving money from friends

Visionary Scenario: 3: Observing friend's transactions via the activity feed

Scenario Name	Observing friend's transactions via the activity feed
Participating Actors	Lauren: Observer
Flow of Events	1. Lauren's semester abroad comes to an end soon. She enjoyed the time very much and had a great experience during her time in Germany. Before she goes home, her birthday is coming up. 2. Her friends from the US know she loves to ski in the German alps. Accordingly, they decide to gift her a ski trip for her and a +1. As they cannot be there to deliver the gift in person, they all send her $20 with the wallet. The wallet, unlike Venmo, is non-custodial and can be used by people in any country. They add heartwarming congratulatory notes to the transactions explaining what the money is for. 3. As they are all friends on the wallet, they can see everyone's wishes and notes to Lauren in the wallet feed. Lauren loves the gift and is excited to go on the ski trip. 4. Through the wallet Feed, Lauren can stay connected to her friends no matter in what country they are and what the common banking app there is. She is looking forward to meeting her friends again in the US.

Table 3.5: Visionary Scenario: 3: Observing friend's transactions via the activity feed

3.3 Functional Requirements

Functional Requirements (FR) describe a system's functionality and interactions with its environment. They are independent of any concrete implementation. A system's

environment can be its users, but also any other third-party system that is interacted with [9]. We identified the following functional requirements based on the visionary scenarios and insights gained from evaluating related work.

FR1: Cryptographic Generation and Storage of Wallet Keys

To create a wallet, the application should be able to generate the cryptographic keys that represent a wallet. This entails storing the keys safely on the hardware device and generating the secret phrase users would use to back up their wallet.

FR2: Interaction with the Blockchain

The system needs to be able to interact with the chosen Blockchain. Publishing new transactions to the Blockchain to enable users to send funds, as well as monitoring addresses for any changes that might need to be reflected in our systems user interface.

FR3: Mobile User Interface

Users should be able to get an overview of all information gathered from the Blockchain through a mobile user interface. This interface should feature the balances of the wallet, any transactions sent, and transactions of friends saved as contacts.

FR4: Handling of Note Storage

In order to add notes to each transaction and preserve the users' privacy, the system needs to be able to store these notes in the form of text or Unicode, respectively, for emojis, separately from the Blockchain.

FR5: Quick Response Code

The system should feature QR code functionality for the user to share wallet addresses easily. The system must generate QR codes containing the wallet's address to enable that. Furthermore, the system needs to scan those codes with the smartphone's camera and derive the ingrained address.

3.4 Use Cases

Use cases represent generalized scenarios. They are an abstraction describing all possible interaction cases from the actors' perspective using the system. Through this,

they determine the scope of the system to be developed [9]. This section describes our system's six most relevant use cases.

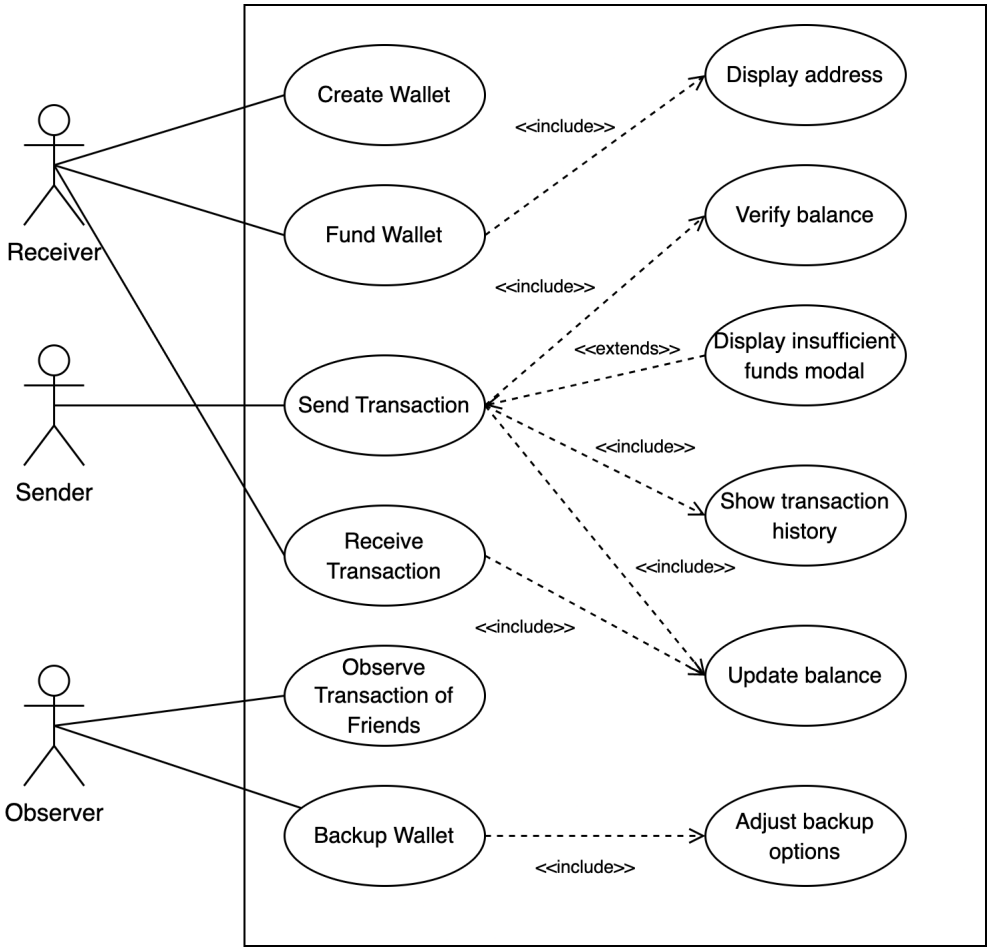

Figure 3.2: The Use Case Model of the Wallet system

Figure 3.2 gives an overview of the use cases for the system to be built, derived from the functional requirements and scenarios. It features the involved actors and their relationship with the use cases.

Use Case 1: Create Wallet

Use Case Name	Create Wallet
Participating Actors	Sender, Receiver, Observer
Entry conditions	The Sender/Receiver/Observer has installed the wallet app on their smartphone.
Flow of Events	1. The Sender/Receiver/Observer opens the wallet app on their smartphone. 2. The Sender/Receiver/Observer press "Get a new wallet" to create a new non-custodial wallet. 3. The Sender/Receiver/Observer is greeted with some guidance instructions on how to fund the wallet.
Exit Conditions	The Sender/Receiver/Observer have created a new non-custodial wallet with the wallet app.
Special Requirements	-

Table 3.6: Use Case 1: Create Wallet

Use Case 2: Fund Wallet

Use Case Name	Fund Wallet
Participating Actors	Receiver
Entry conditions	The Receiver has created a wallet with the wallet app.
Flow of Events	1. The Receiver opens the wallet app on his smartphone. 2. The Receiver reads the guiding instructions on how to fund the wallet from the home screen. 3. The Receiver swipes to the right to access the profile screen and copies his wallet address. 4. The Receiver sends the necessary cryptocurrency from a crypto exchange to his wallet address as described on the home screen.
Exit Conditions	The funds have arrived and are displayed on the wallet's home screen.
Special Requirements	-

Table 3.7: Use Case 2: Fund Wallet

Use Case 3: Send Transaction

Use Case Name	Send Transaction
Participating Actors	Sender, Receiver
Entry conditions	The Sender has a wallet funded with the necessary cryptocurrency.
Flow of Events	1. The Sender opens the wallet app on her smartphone. 2. The Sender enters the EUR amount she wants to send with the keypad on the home screen. 3. The Sender selects the wallet of the Receiver by either 1) entering his wallet address, 2) entering his corresponding ENS address, or 3) selecting the previously saved contact of the Receiver. 4. The Sender adds a note to the transaction consisting of text and an emoji. 5. The Sender reviews the transaction amount that was converted into the corresponding cryptocurrency and the Receiver's wallet address. 6. The Sender authenticates herself with the device via FaceID or passcode and sends the transaction. 7. The Sender can see the confirmed transaction in her transaction history.
Exit Conditions	The transaction arrived successfully in the Receiver's wallet.
Special Requirements	The transaction time is < 5 seconds.

Table 3.8: Use Case 3: Send Transaction

Use Case 4: Receive Transaction

Use Case Name	Receive Transaction
Participating Actors	Receiver, Sender
Entry conditions	The Receiver has created a wallet with the wallet app.
Flow of Events	1. The Receiver opens the wallet app on his smartphone. 2. The Receiver presses the "Receive" button on the home screen. 3. The Receiver either shares his wallet address via a third-party app with the Sender or shows her the displayed QR code that the Sender can then scan with her wallet. 4. The Sender proceeds with sending a transaction as described in Use Case 3. 5. The Receiver can see the received amount added to his wallet balance on the home screen.
Exit Conditions	The sent funds are displayed in the Receiver's wallet.
Special Requirements	The transaction time is < 5 seconds.

Table 3.9: Use Case 4: Receive Transaction

Use Case 5: Observe Transactions of Friends

Use Case Name	Observe Transactions of Friends
Participating Actors	Observer, Sender, Receiver
Entry conditions	The Observer installed the wallet and added the Sender and the Receiver as contacts.
Flow of Events	1. The Sender and the Receiver send transactions back and forth. They add text and emoji notes to their transactions. 2. The Observer opens the wallet app on her smartphone. 3. The Observer swipes to the left to navigate to the wallet feed screen. 4. The Observer follows the transactions of her friends, the Sender, and the Receiver, in the wallet feed. She sees all their transactions ordered chronologically with the time that passed since the transaction was sent and the added note.
Exit Conditions	The Observer has seen all sent transactions by the Sender and the Receiver.
Special Requirements	The Observer only sees transactions of added contacts.

Table 3.10: Use Case 5: Observe Transactions of Friends

Use Case 6: Backup Wallet

Use Case Name	Backup Wallet
Participating Actors	Sender
Entry conditions	The Sender has created a wallet with the wallet app.
Flow of Events	1. The Sender opens the wallet app on her smartphone. 2. The Sender opens the settings by swiping to the right to access the profile screen and pressing the cog button. 3. The Sender backs up their wallet by pressing the Backup button and authenticating herself via FaceID or passcode. She writes her secret phrase on a piece of paper and stores it securely. 4. The Sender confirms that she wrote down the secret phrase, and the backup modal disappears.
Exit Conditions	The Sender has written down her secret phrase and stored it in a safe place.
Special Requirements	-

Table 3.11: Use Case 6: Backup Wallet

3.5 Non-Functional Requirements

Non-Functional Requirements (NFR) describe attributes of a system unrelated to the functional behavior of the system. They complement the functional requirements by defining quality aspects the system needs to provide [9]. We identified the following non-functional requirements based on the visionary scenarios and insights gained from evaluating related work.

NFR 1: Usability

The wallet application should provide good usability. This requirement is based on the fact that systems which are challenging to use will be used less, and current payment systems built on Layer 2 protocols have been shown to provide below-average usability [23]. It encompasses that the system should be user-friendly in the sense that it is straightforward to utilize the application's functionality, especially for novice cryptocurrency users.

NFR 2: EVM Compatibility

The system should be compatible with the Ethereum Virtual Machine (EVM). The EVM is the computation engine responsible for deploying and executing smart contracts on the Ethereum Blockchain, the default development platform and the most established smart-contract-capable Blockchain. EVM Compatibility enables a Blockchain to interact with the EVM and use its infrastructure tooling and developer documentation. This compatibility is essential to ensure the extensibility and maintainability of our system. As most crypto developers are familiar with the Ethereum stack, this will enable further development.

NFR 3: Transaction Speed

The time that elapses until a transaction is received by the recipient's wallet should not exceed five seconds. Slow transaction speeds are perceived as negative as users of established P2P payment applications are used to instant transactions [24]. This threshold is set arbitrarily to ensure a speed that an average user would perceive as fast enough.

NFR 4: Platform Independence

The system's implementation should be independent of the development platform. In detail, the system needs to support both prevailing mobile operating systems, iOS and Android. The underlying reason is that due to network effects, users are not willing to switch to a new system when all their relevant contacts cannot use the same system [46].

NFR 5: Security

The system should ensure secure transmission of user input between the different subsystems and to third-party services. As financial transactions are sensitive data,

external parties must not be able to modify any variables of a sent transaction or read its payment reference [39]. Therefore, data regarding these transactions must be transmitted and stored securely.

4 System Design

This chapter describes the system design of the proposed system. We provide an architecture diagram to give an overview of the system's components.

4.1 Proposed Architecture

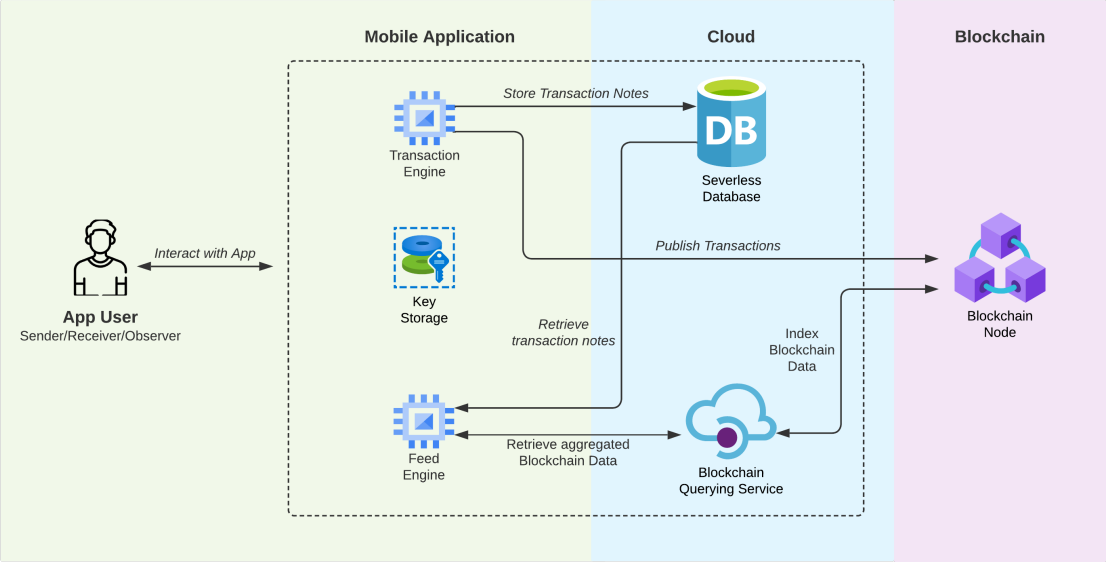

Figure 4.1: Architecture Diagram of the system

The requirements gathered in chapter 3 were considered to determine all components necessary to build the specified system. Figure 4.1 shows these components and indicates how they and the actors interact with each other. The system spans three environments. First, the mobile application running on the *App User's* smartphone (green background). Second, the cloud containing the storage solution and Blockchain interface (blue background). And third, the actual Blockchain (purple background).

The user interacts with the mobile application on his smartphone. This app has three core responsibilities: 1) Safely storing the cryptographic keys that make up the

cryptocurrency wallet in the *Key Storage*. 2) Publishing all transactions sent by the user to the Blockchain and storing the transaction notes sent by the user in the *Transaction Engine*. 3) Generating the social feed comprised of every contact's transactions in the *Feed Engine*. Three dedicated cloud services are used to fulfill these responsibilities. The Transaction Engine publishes transactions directly to a *Blockchain Node* and stores the user's transaction notes in a *Serverless Database*. The Feed Engine uses a third-party *Blockchain API* to query aggregated Blockchain data. The dashed line marks the border of the system.

4.2 Design Choices

Additionally, we describe the architectural design choices made for implementing the prototype. As our system's business logic is all included in the mobile application, we do not need dedicated application servers on our backend. We require a database with basic CRUD functionality, access to a Blockchain node, and a Blockchain API provider. By using a serverless two-tiered architecture, the cloud services are accessed directly from the client, and there is no need to run our own dedicated servers.

This architecture allows us to build and run the system without having to manage the underlying infrastructure, which comes with a set of advantages in comparison to other means of abstraction, such as containerization:

- **Cost:** Cloud service providers offer the flexibility to manage the allocation of storage or compute resources dynamically as needed. This translates into cost savings as the developer is only billed for the actual usage instead of pre-set units of capacity, not any virtual machines that might be unused if there is no system usage [52]. These cost savings enable the development of a prototype without high startup costs.
- **Productivity:** As there is no need to build our own APIs to access the database or manage the infrastructure, productivity is significantly improved. The development time savings originate in enabling the developers to focus on the business logic, in this case, part of the mobile application, instead of engineering a separate backend and managing its deployment or infrastructure maintenance [57]. These productivity increases allow for rapid prototype development and fast iterations.
- **Scalability:** The scaling of the backend services is entirely handled by the cloud provider based on the current need of our system. Compared to running our own virtual machines, we do not need to maintain any deployed containers or scale

them via an orchestration system like Kubernetes[1]. These scalability capacities enable continued feature development should the software gain users quickly instead of focussing on infrastructure problems.

Serverless architectures also come with downsides that result in the need to trust the cloud service provider. Developers have little control over the software stack or hardware that the backend runs on. Also, servers are mostly not dedicated to one customer but shared. This could result in security consequences, and we need to trust the provider to enact the necessary configurations and measures to prevent any security issues. Lastly, there is a vendor lock-in as it is harder to transfer the backend to another provider without significant refactoring effort than containerization and running a dedicated backend[2].

However, the described advantages significantly outweigh these downsides, especially for our requirements and the goal of developing a prototype. Therefore, we will use a serverless two-tier architecture for this system. In the following, we describe each component in more detail.

Mobile Application

The *Mobile Application* acts as the user interface and contains the business logic. We chose a mobile application as this would most serve NFR1 - "Usability". Users want to use means of payment independent of location, especially on the go [26]. Therefore, a desktop application was not applicable. A web-based application would not fulfill FR1 - "Cryptographic Generation and Storage of Wallet Keys" as the wallet then could not be non-custodial but would need to trust a service provider to store the keys. With a mobile application, the keys representing the wallet can be stored natively on the user's hardware device. Besides the *Key Storage*, the application also houses the Transaction Engine and the Feed Engine. The *Transaction Engine* is responsible for publishing transactions to the Blockchain and storing the corresponding notes in the database. The *Feed Engine* is responsible for generating the wallet feed for the user. To do that, it retrieves the transaction notes from the database and queries aggregated Blockchain data via a Blockchain API service provider.

[1] https://kubernetes.io/ (last accessed: 03.12.2022)

[2] https://aws.amazon.com/de/blogs/startups/migrating-your-startup-from-firebase-to-aws/ (last accessed: 03.12.2022)

Blockchain Node

The *Blockchain Node* that is necessary to enable the Transaction Engine to publish the transactions to the Blockchain is accessed through an Infrastructure-as-a-Service (IaaS) provider. A Blockchain development environment consists of multiple layers. The Network Layer contains the actual Blockchains and their nodes which validate transactions[3]. These can be EVM-compatible, such as Ethereum or Polygon, and non-EVM- compatible such as Solana or NEAR. Based upon that is the Blockchain Interaction Layer that enables developers access to the data of different Blockchains. We chose to use a node managed by an IaaS provider instead of running our own node as it allows for the system to carry out the serverless architecture described and leverage the benefits of lower cost, increased productivity, and lower maintenance effort.

Storage

The only data published to and stored on the Blockchain are the transactions themselves. All other data of the application is stored off-chain for two reasons. First is privacy, as everything stored on the Blockchain is irreversibly public. Second, the cost of on-device or cloud storage is substantially cheaper than storing on the Blockchain[4]. Thus, we use a document-based NoSQL cloud database to store personal data like the notes users add to transactions and contact relationships. When a user sends a transaction, the Transaction Engine publishes it to the Blockchain via the Blockchain Node. Simultaneously, it persists the transaction note added by the user in the database via the native Software Development Kits (SDK) for both mobile operating systems that the database service provides. The SDK sends an HTTP call to the Application Programming Interface (API) exposed by the cloud database provider to persist the string that contains the user's transaction note. To display the transaction notes in the wallet feed, the Feed Engine can retrieve the notes based on the hash of a transaction. This process will be specified in chapter 5. Storing static content is currently not a requirement of the system, as all needed static content is part of the application bundle. Nevertheless, most serverless database providers also offer static content storage using object storage if that requirement changes in the future.

Blockchain Querying Service

The Blockchain Querying Service provides the Feed Engine with the necessary data to generate the wallet feed. It allows querying pre-processed data, such as all transactions

[3]https://www.web3.university/article/web3-stack (last accessed: 02.12.2022)
[4]https://ethereum.org/en/developers/docs/gas/ (last accessed: 02.12.2022)

of an address, segmented for only transactions of only a specific cryptocurrency. Because of that, there is no need for a dedicated application server, as the querying service already does most of the necessary computation. Therefore, the Blockchain Querying Service enables the system to work with the proposed serverless two-tiered architecture, as the smartphone can do the remaining computation on-device. The Feed Engine only has to match the data from the database and the querying service, which is easily in the realm of the computation power of modern smartphones. Furthermore, all benefits like lower cost, increased productivity, and lower maintenance effort versus doing the computation in our system on a dedicated server also apply to this part of the system.

5 Implementation

This chapter describes the implementation of the proposed system. We built the mobile application based on a fork of the Ethereum wallet Rainbow and used the cloud services Firebase Firestore, Covalent, and Infura. In the following, the prototypical implementation of these components is described in more detail.

5.1 Blockchain

The Polygon Blockchain, formerly called MATIC, is used for the prototypical implementation of the system as it best fulfills the requirements laid out in chapter 3 while providing extensive developer resources and documentation.

Scaling Approaches

There are multiple different approaches to solving the scaling problem of Ethereum. From alternative Layer 1 networks such as, e.g. Solana, over Layer 2 rollup solutions on Ethereum, such as, e.g., Arbitrum, Optimism, or ZK-Sync, to Layer 2 sidechain solutions like Polygon, they all offer a different set of benefits and drawbacks.

Layer 1 networks, like Solana, aim to offer an alternative to the Ethereum network. By making a different set of tradeoff decisions in the blockchain trilemma regarding scalability, security, and decentralization, they offer better scalability for less security and decentralization [50]. As we defined EVM-Compatibility as requirement NFR2 of the system to leverage Ethereum as the most established developer platform with the most extensive infrastructure tooling and developer documentation, we do not use another Layer 1 blockchain to implement the system.

Rollups on Ethereum aim to solve the scalability problem directly on Ethereum. They keep the transaction state and execution on a sidechain. Multiple sidechain transactions are then bundled into a single transaction on the Ethereum network. This bundling happens by generating a cryptographic proof, also referred to as SNARK, that the transactions were executed and then only storing this proof on the Ethereum main chain [58]. At the time of development, usable rollups like Optimism, called optimistic rollups, were significantly more expensive than Polygon transactions with 0,45€ versus

0,002€ per transaction[1] [2]. ZK-Rollups based on zero-knowledge proofs are currently developed to minimize the cost of rollups [58]. Popular implementations such as ZK-Sync were not live yet at the time of development and also offer much more limited development resources and documentation due to their novelty. Therefore, Rollups are not used as the Layer 2 blockchain to implement the system.

Protocols like Polygon aim to solve the scalability problem by running sidechains alongside a "main" blockchain like Ethereum. These sidechains have their own consensus mechanism and PoS validators to process transactions enabling more transactions per second. They link to and communicate with the Ethereum main chai to allow assets to be transferred between both chains.

Polygon

Table 5.1 compares Polygon, Rollups, and the Layer 1 Solana. It shows that Polygon offers the best compromise as it does well in all relevant criteria without missing any like Solana. As a sidechain of Ethereum it is EVM-compatible like rollups are, as Ethereum smart contracts can be deployed directly to the Polygon chain as well. This contract compatibility enabled quick adoption by developers and is the reason for the extensive tooling and documentation that exists for the Polygon ecosystem.

	Polygon	**Rollups**	**Solana**
EVM compatibility	Yes	Yes	No
Transaction Cost	0,002€	0,45€	0,00025€
Transactions per Second	7k	40k	710k
Project Status	Active	In Testing	Active
Developer Resources	Plenty	Developing	Sufficient

Table 5.1: Blockchain evaluation criteria

This is also why popular stablecoins like the USDC coin are also available on Polygon. They rely on ERC-20 smart contracts on Ethereum, which can be used directly on Polygon as well. Froehlich et al. argue that free-market dynamics which create, e.g., volatility in the assets transferred, complicate everyday use for payment app users [23]. By having access to a stablecoin like USDC[3], this can be prevented as USDC is pegged to the US Dollar. Furthermore, it provides an easily identifiable currency with units familiar to the user [21].

[1] https://l2fees.info/ (last accessed: 04.12.2022)
[2] https://polygon.technology/solutions/polygon-pos/ (last accessed: 04.12.2022)
[3] https://www.circle.com/en/usd (last accessed: 04.12.2022)

Another result of EVM compatibility is that many third-party services are available on Polygon, e.g., the Blockchain Querying API we use for the wallet feed. Access to such services enables a fast prototype development as it allows the implementation of the serverless architecture described in chapter 4.

Furthermore, transactions are cheap, with 0,002€ per transaction, and fast, with up to 7000 transactions per second versus Ethereum's 15 transactions per second[4]. This fulfills requirement NFR3 - Transaction Speed which is essential for a payment application to provide near-instant transactions for the users.

The main drawback of using Polygon is that its benefits require compromises in the network's decentralization. Often referred to as the Blockchain Trilemma is the theorem that scalability, security, and decentralization are on each other dependent features of blockchains [17]. Polygon prioritizes scalability and security over decentralization by using its own validators that are less distributed than Ethereum's. This reduced decentralization might be a drawback for an established system, but it is not one for implementing this prototype.

As a result, Polygon is chosen as the Blockchain for the prototypical implementation of this system. How the system interacts with the Polygon blockchain is described in chapters 5.4 and 5.5.

5.2 Mobile Application

The system is built as a fork of the popular Ethereum wallet Rainbow[5] as this allows to best fulfill the requirements laid out in chapter 3 and set the focus on the usability and social aspects of the application. In general, the application must be available for iOS and Android to fulfill the requirement NFR4 - Platform independence. Therefore we implemented the system with React Native, an open-source JavaScript framework, to build mobile apps natively rendered on both platforms.

Building upon an established Wallet

Forking an existing wallet versus building out basic wallet functionality, like generating and storing private keys, was chosen to shorten the development time. As these foundational features are most often built with established libraries like ethers.js[6], this would not have resulted in new research output. This enabled us to implement the full

[4]https://polygon.technology/solutions/polygon-pos/ (last accessed: 04.12.2022)
[5]https://rainbow.me/ (last accessed: 04.12.2022)
[6]https://docs.ethers.io/v5/ (last accessed: 30.11.2022)

prototype and conduct two studies focussing on the main scope of this thesis: Wallet usability and the impact of social features. Finally, forking an existing wallet has the benefit that the study works with a realistic baseline of the user experience currently available for non-custodial wallets.

Building upon a production-level application with over 100.000 users, maintained by over 60 engineers, also has drawbacks. A substantial amount of the prototype development time went into understanding the complex codebase, which used complicated custom-built UI components and multiple backup solutions for every used API. Furthermore, only the application itself, but not its backend, was open-source. This resulted in the need to reverse engineer a part of it to enable transactions on Layer 2 networks such as Polygon. Conclusively, the benefits of faster prototype development time were canceled out by the additional time needed to work with the complex codebase. Nevertheless, the wallet proved to be a good foundation for the prototype of our system because of the benefits elaborated on in the following.

Rainbow Wallet

We examined eleven open-source Ethereum wallets in detail. Six were non-custodial, and only two were built with React Native. Between those two, we chose Rainbow because it offers the best UI and usability of any non-custodial wallet supporting Layer 2 transactions, as per the authors' assessment. Besides the UI, it offers two key concepts aiding the goal of building a usable social wallet. First, it provides an abstraction of not just measuring amounts in native cryptocurrency but also fiat currency, such as the Euro, in its user interface. Froehlich shows that one major usability issue in today's cryptocurrency applications is confronting new users with too many new concepts [21]. Therefore, abstracting the cryptocurrency by using the USDC stablecoin and using Rainbow's capabilities to display every amount in fiat currency is expected to improve the usability of the payment system. Second, Rainbow offers personalization of the wallet by naming it and choosing a profile picture or emoji for the wallet. This is said to improve the usability of a crypto application by better serving the needs of a heterogeneous user base [1].

5.3 Key Storage

The Key Storage safely stores the cryptographic keys used to sign transactions and prove the ownership of a cryptocurrency wallet. The keys are generated with the ethers.js

library[7], encrypted, and stored via Keychain[8] on the user's device. The implementation of the Key Store is already part of the open-source wallet forked for the development of the prototype and, therefore, not the scope of this thesis.

Key Storage User Interface

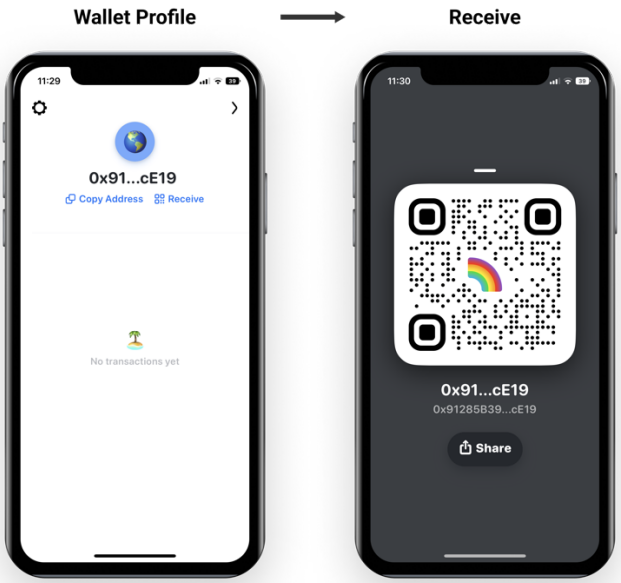

Figure 5.1: Key Storage User Interface

The user interface to display the public key, also referred to as address, is shown in Figure 5.1. The profile section of the application shows the user's address and offers the functionality to copy this address or display it as a QR code and share it via a third-party service. Users can also rename their wallet from their address to any other name for display purposes. This will, however, not change the underlying address.

[7]https://docs.ethers.io/v5/api/signer/#Wallet (last accessed: 30.11.2022)
[8]https://github.com/oblador/react-native-keychain (last accessed: 03.12.2022)

5.4 Transaction Engine

The Transaction Engine publishes all transactions sent by the user to the Polygon blockchain using the ethers.js library and stores the transaction notes added by the user in the serverless Google Firestore NoSQL database.

Publishing Transactions to the Polygon Blockchain

Transactions sent by the users are published to the Polygon blockchain using the ethers.js library[9]. Ethers.js sends the transaction via a remote procedure call (RPC) to the managed node service Infura[10] which will then validate the transaction. The implementation of this part of the Transaction Engine is already part of the open-source wallet forked for the development of the prototype and, therefore, not the scope of this thesis.

Storing Transaction Notes on Firestore

Users can add notes in the form of text or emojis to each transaction. These notes are stored in a Firestore NoSQL database by Google Firebase[11].

Firestore is very performant with up to 1M concurrent client connections and highly available with a service level agreement (SLA) of 99,99%. It allows implementing the presented serverless architecture as all APIs are accessible via an SDK for both iOS and Android and a wrapper for React Native[12].

Firestore is a document-based NoSQL database where the data is stored as key-value pairs in documents. These documents are grouped into collections.

Figure 5.2 describes the system's data model. The database consists of one single collection *Addresses*. It holds documents of the entity *Address* that each represents one wallet address. Each of these addresses is identified by its address, or public key, and holds a subcollection *Transactions* which represents all transactions that this wallet has sent. Each subcollection holds *Transaction* documents that represent one single transaction. A transaction consists of a timestamp and the note that the user added. It is identified by the transaction hash that the blockchain generated when the transaction was sent. The contact management in the application does not work with friend requests in this prototype iteration, as the participants of this study are considered to be one social circle of friends. Therefore, the relationships between

[9] https://docs.ethers.io/v5/api/signer/#Signer (last accessed: 03.12.2022)
[10] https://docs.infura.io/infura/ (last accessed: 03.12.2022)
[11] https://firebase.google.com/docs/firestore/ (last accessed: 03.12.2022)
[12] https://rnfirebase.io/ (last accessed: 03.12.2022)

individuals and their wallets are not part of the data model. More details on this topic are clarified in chapter 5.5.

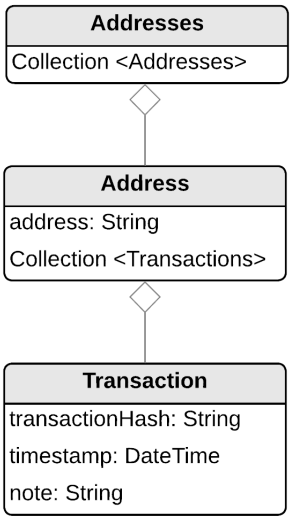

Figure 5.2: Data Model of the System

The application sends an API call to Firestore containing the wallet address, transaction hash, and the user-added transaction note to store a transaction note when the Polygon network confirms a transaction. This data is then stored by Firestore using the data model described above.

Firestore offers powerful querying functionality for specifying which documents should be retrieved from a collection or collection group. Collection group queries allow the retrieval of documents from multiple different collections in one single query. Furthermore, Firestore allows ordering data during retrieval, which the system uses to order the notes by timestamp. This saves the system computation during the generation of the wallet feed, which is happening on the device. Therefore, only one query is needed to retrieve the transaction notes, e.g., to generate the wallet feed. The application does this by sending an API call to Firestore and receiving an array of JSON objects representing one note. Each note consists of the transaction hash, timestamp, and note string.

Transaction Engine User Interface

The Transaction Engine user interface comprises two components, as shown in Figure 6.4. First, the home screen with the keypad to display the wallet balance and start a

transaction. Second, the send flow specifies the details of a transaction and a note to the transaction. The wallet balance is displayed in Euro and the cryptocurrency unit. In the case of this prototype, the stablecoin USDC. The keypad allows to enter the amount that should be sent quickly, and a tap on Send starts the flow to specify the recipient further and add a custom note to the transaction. The note can be any string of text, including emojis. After reviewing the transaction checks and pressing send, the transaction is published via the Infura node. Once confirmed, the note is persisted in the Firestore database with the previously described API. All interface is implemented with React Native and partly using the custom UI components available from Rainbow.

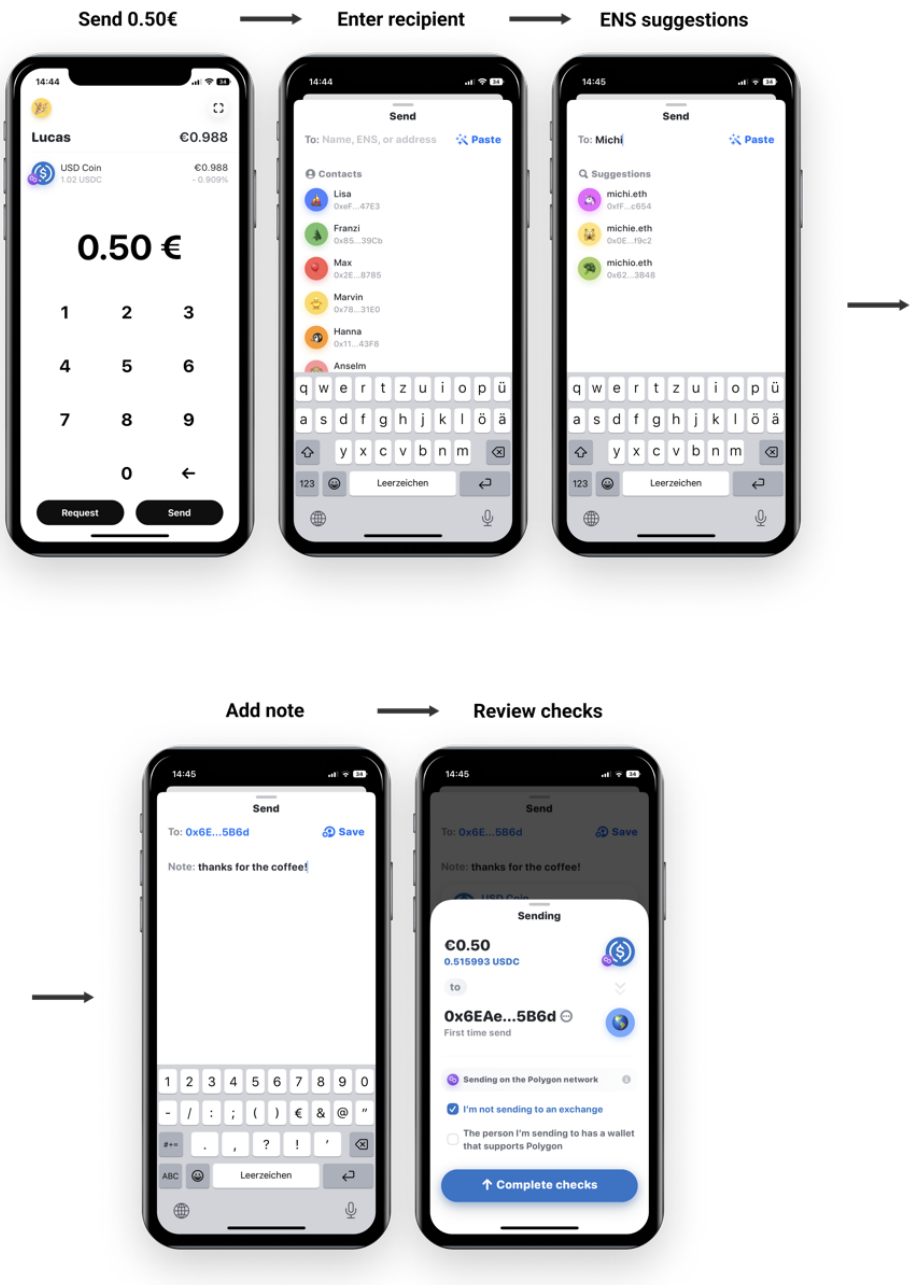

Figure 5.3: Transaction Engine User Interface

5.5 Feed Engine

The Feed Engine generates the wallet feed, a social feed consisting of transactions from all contacts and the notes they added. The feed is generated on the device with two APIs to supply the necessary data. First, the Firestore API described in 6.4 queries the transaction notes. Second, the Blockchain Querying Service for which the Covalent Blockchain API[13] is used supplies the transaction data of every contact saved by the user.

Covalent as Blockchain Querying Service

The Blockchain Querying Service uses the Covalent API to access aggregated blockchain data, reducing the compute necessary to generate the wallet feed. Covalent runs its own blockchain nodes, indexes the data, and allows to retrieve various aggregations via its RESTful API. Possible endpoints are, e.g., the balances for an address or the portfolio value over time. It supports all major blockchains and runs on a delay of 2 blocks, which in the case of Polygon is six seconds.

The Feed Engine uses the `/transactions_v2` API to get all transactions for a specific address. Suppose a generation of the wallet feed is triggered. In that case, the application sends an API call via HTTPS to Covalent, specifying the chainID 137 of Polygon, the address of which to get the transactions, and the currency in which the transactions should be displayed. The API returns a JSON with all transactions of the specified address on the Polygon chain.

Generating the Wallet Feed

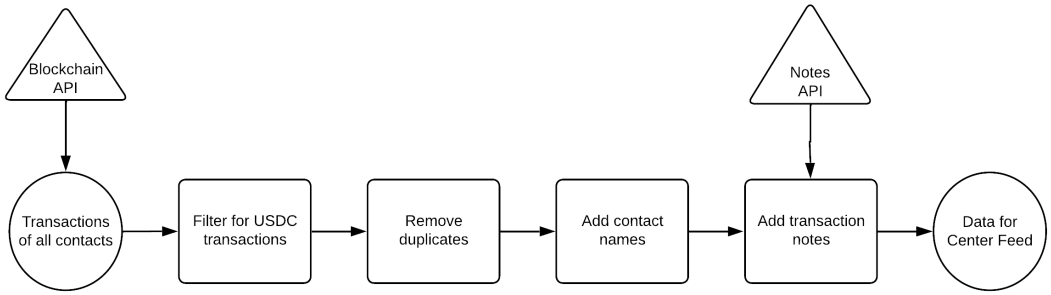

Figure 5.4: Feed Generation Flow

[13]https://www.covalenthq.com/docs/api/ (last accessed: 03.12.2022)

The Feed Engine computes the wallet feed for one individual wallet user, as shown in Figure 5.4. First, the Blockchain Querying Service queries all transactions on the Polygon chain for all user contacts. Then any transactions not using the USDC stablecoin are filtered out using the `to_address` attribute of each transaction. This is possible as any transaction sending an ERC-20 token like the USDC stablecoin is sent to the smart contract of said token executing the transaction then. Therefore, any transaction not sent to the recipient `'0x2791bca1f2de4661ed88a30c99a7a9449aa84174'` is not a USDC transaction. Following, any duplicates in the transactions are removed. Duplicates arise when both, the sender and recipient of a transaction, are a contact of the user, as then the transaction is queried once for each contact. Based on the addresses of all contacts and the user's own wallet, the clear names of the contacts, Unknown in the case of an external party or "Me" in the case of the user, are added to each transaction. Finally, the system queries the notes for the transactions and matches them based on the transaction hash. The resulting data represents the transactions that should be displayed in the wallet feed UI for the user.

Feed Engine User Interface

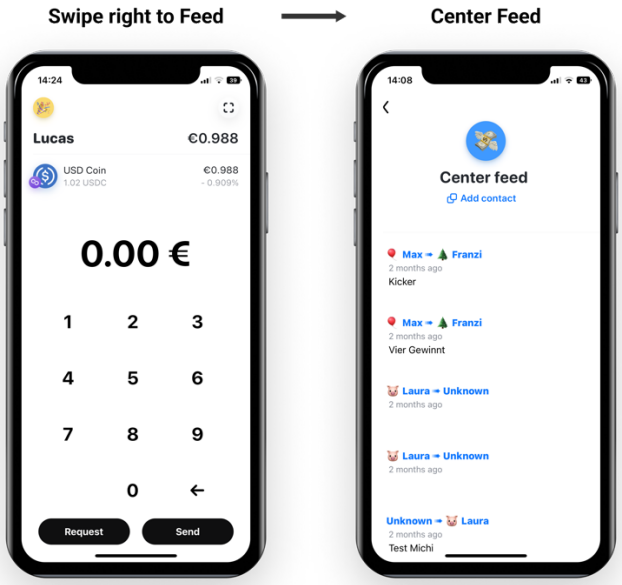

Figure 5.5: Wallet Feed User Interface

The user interface built to display the wallet feed is accessed by swiping on the home screen as shown in Figure 5.5.

At the top, the user has the option to add more contacts. In this prototype implementation, any wallet address can be added as a contact. This decision was made as all participants of the conducted study belong to one social circle of friends and enabled the development effort to fit the scope of this thesis. In future iterations, a friend request system should be implemented that allows users to decline being added to another user's contacts. This will be discussed in more detail in chapter 7.2.

Below the header with the add contact button is a list with each row representing one transaction of a contact. The data for this feed is generated as described and rendered on scrolling. Besides the sender's name, the recipient's name, and the note, users can also see how long ago the transaction was sent. The transaction's timestamp is converted into this format using the date-fns library[14]. The data for the wallet feed is refreshed on each navigation to the feed, a change in the user's contacts, or manually by the user using the pull-to-refresh motion. For future iterations of this prototype, WebSockets could be implemented to automatically add new transactions into the feed when they are published. This feature is currently not supported by the Covalent API used for the querying service but will be offered in the future as per their product roadmap.

[14]https://date-fns.org/v2.29.3/docs/formatDistanceToNowStrict (last accessed: 03.12.2022)

6 Evaluation

This chapter describes the evaluation done to the built application. It reiterates the research goal of this thesis and how the study was conducted as depicted in Figure 6.1. The central part is what results we can derive from the research and whether this fulfills our requirements. Finally, we highlight the limitations that this research might have.

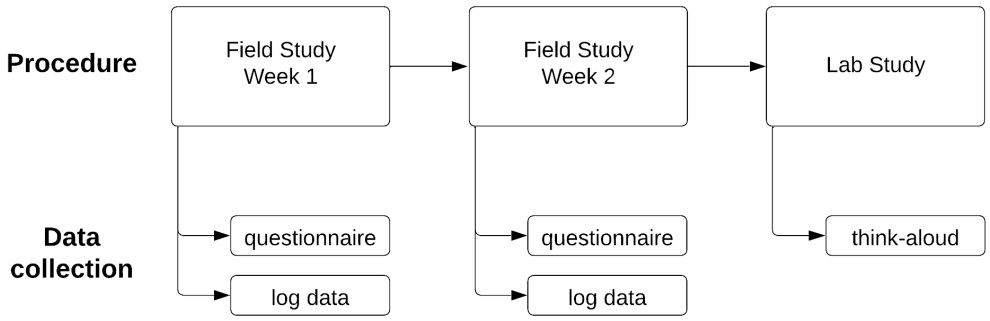

Figure 6.1: Overview of the study procedure and the collected data

6.1 Goal

This thesis aims to answer three research questions that arise by combining the notion of technical advances in the blockchain space and the trend of social payment apps in recent history.

These research questions are:

1. Is it possible to build a mobile payment app using Layer 2 blockchain technologies as a viable alternative to established applications?
2. How do users interact with such an app when using it as a payment service in their everyday life?
3. Do “social payment features” such as a wallet feed add value to a mobile payment app?

To provide a detailed answer to these research questions, we built a mobile social payment app as described in previous chapters. This app was evaluated in the study that will be described in the following.

6.2 Study Design

The study to evaluate the built application entails two different parts that both answer multiple research questions by looking at them from different angles. This two-week-long mixed-method study was conducted in September 2022. 29 people participated in the study, of whom 23 were in the field study and 6 were in the lab study. The participants of the field study and the lab study did not overlap. The field study focuses on evaluating if the requirements for the payment application were fulfilled in a real-life setting. The lab study dives deeper into the usability of the application.

6.2.1 Field Study

Goal

The field study evaluates whether the built application meets the requirements in a setting of real-life usage as a payment app over two weeks.

Context & Participants

We deployed the developed payment app in a cohort of an educational institute associated with a German university. In this environment, the cohort worked full-time and in person at the institute's facility. This closed community ensured every participant had the app available, and we could focus on the evaluation in this controlled environment.

N=23 participants used the application in a timeframe of two weeks, sending 64 transactions. Their age varied from 21–29 (mean: 23,9) years. 60,9% of the participants were male and 39,1% female. All participants are Bachelor or Master students with a major distribution of 56,5% CS & EE, 26,1% Business, and 17,4% Other. 52,2% of participants noted they have no experience with cryptocurrencies, while 13,0% own one currency and 34,8% own several cryptocurrencies.

Methods

We collected data from various sources and evaluated them using several methods. The primary research method is questionnaires with open and closed questions. The

participants at the end of each week completed these. Furthermore, ethnographic methods were used. We observed participants while naturally using the applications and asked about their experiences. This also leads to informal feedback and feature requests outside of the questionnaires. Third, usage logs contributed to the data collected for this study. The backend system used for storing user data, public blockchain, and analytics service implemented in the mobile wallet application provided data that is used to set the results of the questionnaires into context.

Apparatus

Our apparatus comprises the implemented system described in section 6, a setup presentation, and multiple questionnaires. The setup presentation introduced the study to the cohort of students, laid out the research procedure for the following weeks, and helped students set up their personal wallets by installing the application. The demographic questionnaire collected structured data like age, gender, educational background, and the participant's familiarity with cryptocurrencies. The weekly questionnaires featured single ease questions as a proxy for the difficulty users faced with the application [55]. This was followed by an open question asking for the reasoning for choosing this particular score, as recommended, to collect qualitative data as well[1]. To quantitatively determine the perceived quality of user experience and usability of the application, we used two established scoring systems, the User Experience Questionnaire (UEQ) and the System Usability Scale (SUS) [8][40]. The complete questionnaires can be found in the appendix.

Procedure

After installing the wallet application on their personal smartphones and submitting their wallet addresses, every participant received $20 of cryptocurrencies in their wallet. $18,5 of USDC stablecoin and $1,5 of MATIC, the native token of the Polygon network. Participants were encouraged to use the wallet app as their default peer-to-peer payment app with each other for a timeframe of two weeks in lieu of traditional services like, e.g. PayPal. They were free to use the funds as desired but given some pointers of potential use cases like paying for each other's meals and reimbursing the other party. After each week, the participants were asked to complete the above-mentioned weekly questionnaire to describe their experience in that particular week.

[1] https://measuringu.com/seq10/ (last accessed: 30.11.2022)

6.2.2 Lab Study

Goal

The lab study evaluates the application in-depth from a usability perspective to detect concrete flaws in the user journey that participants struggled with [41]. Particular focus was placed on understanding the participant's reasoning for their experience with the Social features of the application and its privacy implications.

Context & Participants

The interviews for the lab study were conducted individually in October 2022. The participants selected to feature diverse backgrounds in terms of educational background, age, and familiarity with modern applications and smartphones.

N=6 participants were interviewed for the lab study. Their age varied from 24-61 (Mean 36,2) years. 66,7% of the participants were male and 33,3% female. 50% were Bachelor students, 16,7% working in engineering, 16,7% working as a government clerk, and 16,6% were retired. All participants had minor or non-existent experience with cryptocurrencies and especially non-custodial wallets to represent a realistic user base of non-crypto services like, e.g., PayPal.

Methods

The primary research method in the lab study was the think-aloud approach [41]. Participants were asked to share their thoughts while completing a set of tasks provided to them using the application. This approach was chosen to explore flaws in the user journey that the participants encountered in-depth. After completing the tasks, semi-structured interview questions were used to clarify comments made by the participants while using the application and to get a more detailed perspective on the implications of the social features of the application.

Apparatus

The apparatus for this study consisted of the implemented payment system, an introductory explanation of cryptocurrency wallets, and the tasks given to the participants. All participants were given an explanation of basic crypto concepts before they started using the application. This contained an introduction to public and private keys, custodial and non-custodial wallet and their implications, and the purpose of USDC and MATIC, the two cryptocurrencies used in the study. This was done to level the playing field between the participants regarding their cryptocurrency experience. Said level

playing field was crucial to accurately compare the findings about necessary guidance and the difficulty of applying cryptocurrency concepts like public/private keys while using the application. The list of tasks given to participants modeled a full user journey from setting up a wallet, sending and receiving transactions, and using social features to securing the wallet via a backup. These tasks were chosen to collect data on features rarely used during the field study, such as the wallet backup mechanism.

Procedure

Every interview started with introductory questions about demographics and their previous experience with payment apps. Afterward, every participant received an explanation of crypto basics and was able to ask any remaining questions. Then we provided the participants with a smartphone with the developed wallet application installed and a list of instructions for six tasks they had to complete using the app. Participants were nudged to use the think-aloud approach to share their thoughts while completing the tasks [41]. Finally, semi-structured interview questions were posed to clarify comments made by the participants while using the application and to get a more detailed perspective on the implications of the social features of the application. The interviews were audio-recorded, transcribed, and coded using an inductive approach.

6.3 Results

During the field study, the system was provided to 23 participants who sent 64 transactions over a time frame of two weeks. The average transaction count per participant was 2,78, with a minimum of 1 and a maximum of 4 transactions. We collected 105 qualitative statements from the surveys. Furthermore, 46 data points from 23 participants were used to compute the System Usability Scale (SUS) and User Experience Questionnaire score (UEQ). The lab study yielded 175 relevant coded statements from the six think-aloud interviews. We present the findings of both studies per research question to evaluate the system from three distinct perspectives:

1. The technical perspective, evaluating whether we succeeded in satisfying the requirements of building a social mobile payments application using Layer 2 blockchain technologies.

2. The usability perspective, evaluating how users interact with the application and what challenges they might face when using the system as their main payment service.

3. The social perspective, evaluating whether social features add value to payment applications and what other consequences might arise from that.

Interviewee statements are denoted with "P" and statements from field study surveys with "S" [20]. Interview statements were translated from German into English.

6.3.1 RQ1: Building a Payment App on L2

The first question we want to answer in this thesis is whether it is possible to build a satisfactory mobile payment app using Layer 2 blockchain technologies. In this case, satisfaction is defined by first, users being able to use the application as a means of payment, and second, the requirements we elicited in chapter 3 being fulfilled.

Evaluating the fulfillment of requirements

We start by evaluating whether the requirements elicited in chapter 3 are fulfilled by the proposed system.

- **FR1 - Cryptographic Generation and Storage of Wallet Keys:** All lab and field study participants were able to generate a new and individual wallet successfully. The industry standard open-source library ethers.js[2] handles all cryptographic details.
- **FR2 - Interaction with the Blockchain:** All lab and field study participants were able to send transactions with their wallets. Publishing those transactions to the Polygon blockchain is handled with the ethers.js library. Furthermore, all participants could observe their friend's transactions in the wallet feed. The querying of balances for multiple wallets is handled through the Covalent blockchain APIs [3].
- **FR3 - Mobile User Interface:** All lab and field study participants were able to get an overview of all information gathered from the Polygon chain in a mobile user interface. The participants perceived the UI as easily understandable (S19, S23). Further details on the UI can be found in chapter 6.3.2.
- **FR4 - Handling of Note Storage:** All lab and field study participants were able to add notes comprised of text and emojis to their transactions and retrieve them via the wallet feed. To preserve the privacy of this data, the notes are not written on the blockchain but stored separately using Firebase Cloud Firestore[4], a scalable NoSQL database.

[2]https://docs.ethers.io/v5/ (last accessed: 30.11.2022)
[3]https://www.covalenthq.com/docs/api/ (last accessed: 30.11.2022)
[4]https://firebase.google.com/docs/firestore (last accessed: 30.11.2022)

- **FR5 - Quick Response Code:** All lab study participants were able to show a QR code representing their wallet address and scan a QR code when shown to them with the wallet. The QR code is generated natively by the system. The RN-camera library was used to access the camera and scan the shown code[5]. The QR code feature was mentioned as their highlight feature by 33,33% of the lab study participants (P1, P4).
- **NFR 1 - Usability:** The usability of the system partly fulfilled this requirement. A detailed account of the reasons can be found in chapter 6.3.2.
- **NFR 2 - EVM Compatibility:** The system generates an Ethereum wallet, and all transactions run on the Polygon blockchain, which is EVM compatible[6].
- **NFR 3 - Transaction Speed:** The system meets the set threshold of five seconds for the transaction speed in 82% of all cases. During the lab study, the average transaction speed was three seconds with peaks of six seconds when the system was used during peak hours. Then the participant had the option to reduce that time by 50% by paying an, on average, 30% higher transaction fee.
- **NFR 4 - Platform Independence:** All lab and field study participants were able to install and use the application, irrespective of what mobile platform they used. The system was built with the cross-platform- framework React Native and thus is available on both iOS and Android[7]. It was installed on thirteen different types of iPhones and six different types of Android devices during the evaluation period.
- **NFR 5 - Security:** The security of the data storage is ensured by Google. The Firebase database solution used for the prototype is certified by the industry-recognized ISO 27001 and SOC 1-3 security standards[8] [18]. There is never an absolute guarantee for the security of an application. However, the Rainbow Wallet, which the prototype for this thesis is based on, is an open-source wallet. This enables a community beyond the original developers to monitor the security of the application and fix bugs. This monitoring and fixing is done regularly, as evident via their GitHub repository[9].

The system completely fulfills all requirements except NFR1 and NF3. NFR1 Usability and NFR3 Transaction Speed are partly fulfilled. The reasons for the only partial

[5]https://react-native-camera.github.io/react-native-camera/ (last accessed: 30.11.2022)
[6]https://polygon.technology/ (last accessed: 30.11.2022)
[7]https://reactnative.dev/ (last accessed: 30.11.2022)
[8]https://firebase.google.com/support/privacy (last accessed: 30.11.2022)
[9]https://github.com/rainbow-me/rainbow (last accessed: 30.11.2022)

fulfillment of the Usability requirement will be elaborated on in chapter 6.3.2. The cause for the varying transaction speed is, as discussed, the current utilization of the Polygon blockchain. Currently, the maximal speed is only one second over the set threshold. This issue should be monitored, and if necessary, the system could switch to a better scaling blockchain.

Evaluating the system usage

	Week 1	**Week 2**	**Overall**
Total Nr. of Transactions	30	34	64
Total Nr. of Sessions	89	81	170
Avg. Nr. of Sessions / Day	8,60	7,00	7,80
Avg. Nr. Sessions / User / Day	1,53	1,37	1,45
Avg. Session Length (Min)	1	3	2

Table 6.1: Usage metrics of the system

Table 6.1 describes the system's usage metrics over the field study's timeframe. When testing with ANOVA, no metrics differences were statistically significant between the two weeks. The participants sent 64 transactions and used the application in 170 sessions, confirming that the participants were able to use the application regularly throughout the two-week study. The average number of sessions per day and per user per day further verifies this observation. The fact that the average session length is two minutes shows that the 170 sessions were sessions of real-life usage, not just opening the application by accident.

These usage metrics confirm that the participants were able to use the developed system during the entire duration of the study. Most of the requirements elicited for the system were fulfilled, with two only being partly fulfilled.

6.3.2 RQ2: User Interaction Observation

After establishing that it is possible to build a social, mobile payments application with Layer 2 blockchain technologies, we focus on how users interact with the wallet application when using it as a payment service. Despite 50,2% of participants being inexperienced in using cryptocurrencies and never owning one, they were confident of being able to use and assess a cryptocurrency wallet. The self-reported vulnerability to wallet loss score is low at 1,85, and the self-efficacy of being able to prevent unauthorized access to the wallet is medium at 3,13. Both on a scale from 1 (low) to 5 (high).

Therefore, the perception of the users participating in the field study can be considered very relevant and a good proxy for the overall usability of the application.

Quantitative Usability Scoring

We used two scoring approaches to offer a quantifiable measurement of the system's usability.

Brooke introduced the System Usability Scale in 1995 as a "quick and dirty" approach to evaluating a system's usability [8]. Over the years, it established itself as the go-to measurement solution and proved reliable [6]. All statements are evaluated by the participant on a Likert scale from 1: Strongly disagree to 5: Strongly agree. This results in a potential score from 0 to 100. It is important to point out, though, that this score is not to be understood as a percentage but rather in terms of a percentile rating in comparison to other scores [5].

The User Experience Questionnaire was introduced by Laugwitz et al. in 2008 and offers even more detail into the user's experience with a solution. It provides a granular breakdown into a hedonistic, pragmatic, and attractiveness perspective [40]. All statements are evaluated by the participant on a Likert scale from 1 to 7. This results in a potential score from -3 to 3. It is important to point out though, that because of the averaging between many questions and collected data points scores outside of -2 to 2 are extremely rare. To control for participants randomly filling out the survey or not understanding the question at hand the UEQ asks multiple questions per usability dimension. We consider dimensions that surface a difference of more than 3 points between multiple questions problematic. If more than two dimensions per participant stand out in such a way we exclude this data point. In our analysis, we did not find any contradictory statements and therefore all submitted data points were considered in computing the score. We collected both scores after each week of usage from the participants.

Evaluating the SUS scores

	Week 1	**Week 2**	**Overall**
SUS	67,72	59,34	63,53

Table 6.2: SUS scores over time

Table 6.2 describes the collected usability scores in both weeks individually as well as their average, the overall score. A SUS score of 63,53 or grade C- can be interpreted as below average compared to other mass-market consumer software, which show an average score of 74,00 or grade B- [5]. It is comparable to other custodial wallets [22]

[24]. Data for non-custodial wallets other than the one collected in this study could not be found.

Segmentation	**Week 1**	**Week 2**	**Overall**
-	67,72	59,34	63,53
Uses / week > 0	67,72	59,34	63,53
Uses / week > 2	86,25	57,50	71,88

Table 6.3: SUS scores with Transaction Usage

Next, we explore how the SUS score develops with higher usage of the system. Table 6.3 shows the score baseline and segmentations by transaction uses per week. The value of Uses / week > 0 is the same as the baseline score as every participant sent at least one transaction per week with a score of 63,53. Participants who used the system more regularly with more than two transactions per week showed a higher usability score of 71,88 or grade C+ [5]. This is still considered below average to mass-market consumer software, but better than public-facing websites or business-to-business software [43]. Testing with ANOVA both, the differences in the week to week scores (F=2,43; p=0,13), as well as the difference between normal and high transaction usage (F=1,09; p=0,30), are not significant at $p < 0,05$.

The cause for the slight improvement in usability score might lie in several reasons. First, the possible development of a user's habit towards using the system over other payment applications. Second, a learning curve users go through when adopting the usage of a new system where they get more value by using the application as they become more familiar with its features by using it more. 43,49% of participants who did not use the system extensively mentioned an existing habit of using another payment application like e.g. PayPal as the main reason why they did not use the wallet app. Why this habit might have been very strong and how the UX of the wallet could be improved to provide an incentive to change this habit will be explored in the qualitative reasoning part at the end of this chapter.

Evaluating the UEQ scores

UEQ Dimension	**Week 1**	**Week 2**	**Overall**
Attractiveness	0,84	0,45	0,65
Perspicuity	1,21	0,56	0,89
Efficiency	0,30	0,27	0,29
Dependability	0,46	0,19	0,33
Stimulation	0,26	0,30	0,28
Novelty	0,45	0,62	0,54

Table 6.4: Detail attributes of the UEQ Scoring

Table 6.4 describes the collected UEQ scores per dimension in both weeks individually as well as their average, the overall score. The significant week-to-week differences presumably lie in a higher usage in the second week, which enforced the tendency to the lower score. There is no single score output for this scale since there is no framework to interpret such a score as it spans multiple dimensions. UEQ scores between -0,8 and 0,8 can be interpreted as neutral [56]. Scores > 0,8 describe a positive experience, and scores < -0,8 describe a negative experience [56]. The dimension of perspicuity evaluates slightly positively with a score of 0,89. All other dimensions describe a neutral experience, with Stimulation being the lowest scoring dimension with 0,28. Perspicuity evaluates how understandable and accessible to use the experience is. Stimulation evaluates whether the experience is exciting or rather inferior in comparison to other possible options. The main driver for this low score of 0,28 is users disagreeing with the notion that the app provides value to them with a score of -0,4.

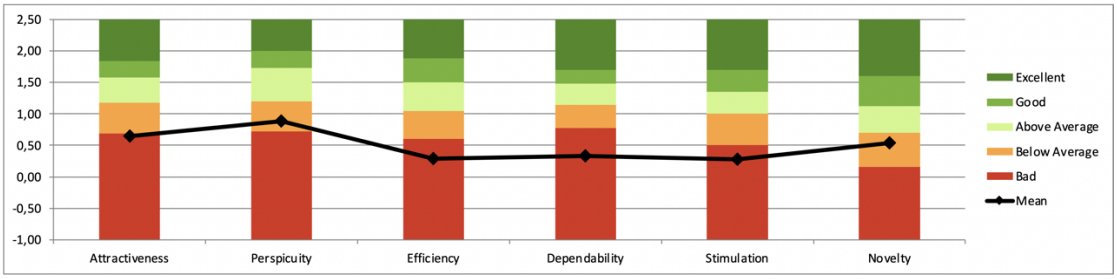

Figure 6.2: Benchmark of UEQ scores

To better understand what these values mean, we set them into context to the results of 468 studies evaluating several digital products like business software, web applications, and social networks [56]. As expected, Figure 6.2 shows the application ranks bad, meaning in the worst quartile of all results, for Attractiveness, Efficiency, Dependability, and Stimulation. Perspicuity and Novelty rank slightly below average. This is in line with our expectation of a prototype cryptocurrency app that aims to be user-friendly. Novelty and Perspicuity are comparably high as it is a new type of product that people might be interested in trying, and we focussed on delivering a user-friendly application that is easy to understand. Yet the challenges that arise with using non-custodial wallets and cryptocurrencies lead to a worse experience than what users are used to from traditional and established payment services like, e.g. PayPal.

This results in overall bad scores for the application's user experience, with Stimulation and Efficiency being the lowest-rated dimensions.

Evaluating the qualitative aspects

To dive deeper into the reasons for the comparatively bad usability scores of the applications, we look at the qualitative part of the user surveys and the think-aloud interviews of the lab study. In the surveys, the participants were asked follow-up questions to explain their reasoning for rating the applications with the score they did. The interviews allowed us to examine the user's interaction with the system in more detail by watching them perform specific actions with the app.

Overall, there are three main factors for the weak usability scores. First, the current default solution, PayPal, works fine for most participants, and the added features of the wallet provide no significant benefits to them. 100,00% of the field-study participants use PayPal, and 73,91% of them report it as their primary payment application. Second, the peculiarities of working with cryptocurrencies in your own non-custodial wallet pose additional challenges to users in comparison with traditional finance apps. Third, a majority of the participants did not see a big value-add in the wallet's social features, and many even mentioned them as the reason for a low usability score. As the social capabilities constitute a significant part of the system, this third factor will be discussed separately in chapter 6.3.3.

No significant benefits for users

43,49% of all study participants appreciated the user interface and overall design of the application, which made "the usage [...] super intuitive" (S19). This explains the disproportionately high Perspicuity score in the UEQ, as this dimension focuses on how easy to learn and understandable a system is. Yet, interview participants added that this is mostly the case in other payment apps (S1, S4, S6). Transaction speed which is often discussed in relation to cryptocurrency systems, especially in the payments context, was perceived as fast. (P2, S23, S13) To enable instant-feeling transactions was a big factor in deciding on the current implementation. Yet, while this was achieved, only most participants see this as a baseline feature that does not add significant value as this is standard in traditional payment apps. Also, key benefits of non-custodial crypto wallets, such as independence from financial institutions and governments, were not valued by 91,30% of participants in this context. The ones who did reference not sharing their data and staying anonymous as a benefit of the wallet versus traditional payment apps. Furthermore, most participants requested additional features for the wallet already part of PayPal. 27,59% of all participants asked for a bank connection to quickly transfer funds from and to their traditional bank without an intermediary step of converting the USDC to Euro with an exchange before transferring them. Other features requested by individual participants, such as FaceID, account statistics, a web

app, or the functionality to not only send funds between people but also pay in online shops, are also all already included in PayPal. While valuing certain aspects of the wallet, 56,52% of all participants concluded that they perceive no significant benefit compared to PayPal, some adding that they consider payments a "solved problem" for themselves (S4, S13, S20).

Usability challenges

In three steps of the user journey, usability challenges emerged during the analysis of the lab-study interviews. While all participants could complete all six tasks during the lab study, some steps posed challenges to most participants. Challenge is defined here as the participant needing additional input or help from the interviewer to complete the task at hand. These are the tasks of the lab study. The ones where participants encountered challenges are bolded.

1. **Create a new wallet with the wallet app.**
2. Fund your wallet with $2 USDC Coins and 2 MATIC Coins.
3. **Send a transaction of 0,5 € to the interviewer with the note "Test your name".**
4. You paid for my lunch and want me to pay you back using the wallet app. How can I send you your money?
5. You use the wallet app with your friends and have saved them as contacts. Take a look at the transactions of their friends.
6. **You want to protect your wallet. Create a backup of your private keys.**

All of the challenges are specific to a non-custodial cryptocurrency wallet and could potentially be solved by implementing additional guidance mechanisms in the application. This is especially relevant for inexperienced wallet users like those in this study.

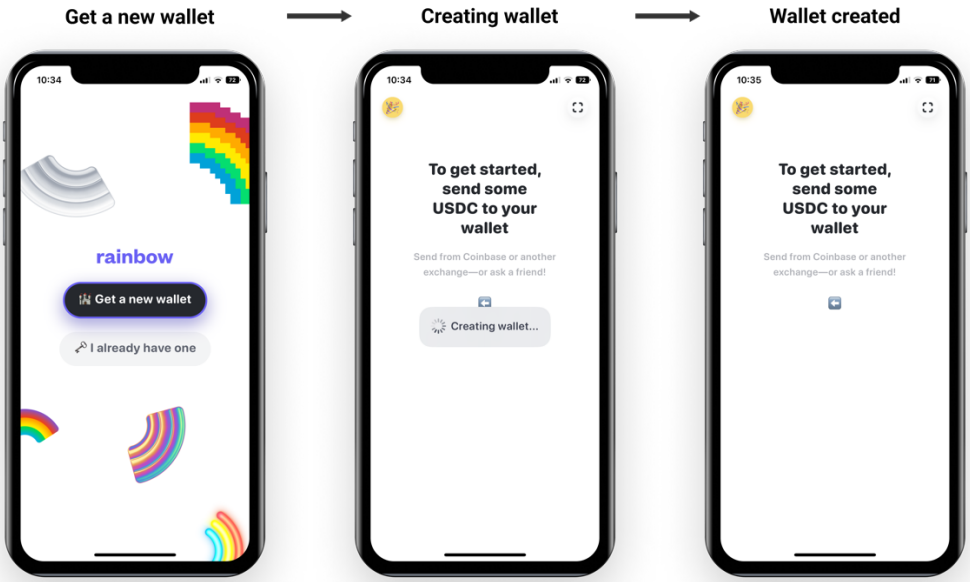

Figure 6.3: User flow to create a wallet

During the creation of the wallet, 100,00% of all participants were confused about whether they had already completed the creation of the wallet. Some wondered if they did something wrong as the creation displayed in Figure 6.3 took only a few seconds. They expected a long process to set up a wallet where personal information like, e.g., name, E-Mail address, and date of birth are requested, which is a usual practice for custodial wallets that require KYC processes or traditional regulated payment applications. This difference in setup time and required personal details led users to believe there was an error on their side and that the wallet was not yet created. Most participants mentioned they would appreciate an intermediary step with more guidance, such as a message like “Welcome! Your wallet was created!” before seeing the instructions on how to fund their wallet. After a hint from the interviewer that their wallet was indeed already created, the participants had no problems following the guidance instructions on the home screen and were able to fund their wallet successfully.

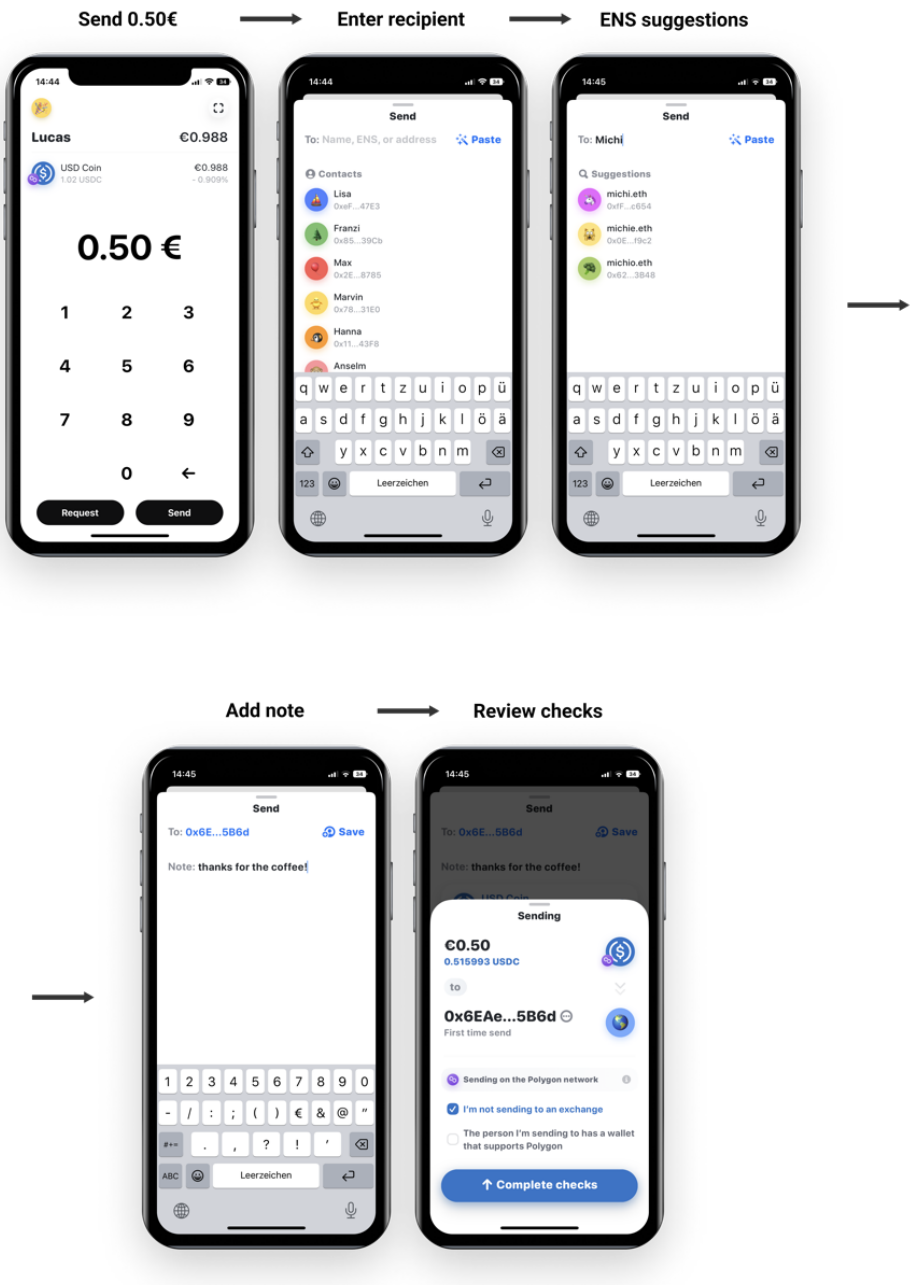

Figure 6.4: User flow to send a transaction

The second usability challenge the participants encountered was identifying the right recipient they intended to send their transaction to. The user flow shown in Figure 6.4 starts by entering the amount on the home screen and pressing send. Then a user will see a list of their contacts and select a recipient by typing in the name of a contact or an ENS domain or pasting a public-key address. When there are no contacts, the system will suggest ENS addresses. This crypto-native concept of ENS domains allows participants in the Ethereum ecosystem to link the public key of their wallet to an alphanumerical domain like, e.g., lucas.eth. During the lab study, the participants tried searching for the recipient by their name like, e.g., Michi in Figure 6.4. As they did not have any contacts with that name yet, the application suggested ENS domains starting with "Michi". The source of confusion then was who the people behind these ENS domains were and who the correct recipient that they were looking for was. All participants ran into this challenge. After a short introduction to ENS domains and mentioning that their recipient has not registered an ENS domain, they all quickly realized they needed to enter the recipient's public-key address instead of their name. Additionally, using the public-key address as a means of identity was rated as tedious by 21,74% of the field-study participants. With the feature to save addresses as contacts, this would only be necessary once, but still, the participants preferred other means of identity like a phone number or E-Mail address. One specifically frustrating situation cited was where one party did not have their phone on them to share their address which led to using other services like PayPal with an E-Mail address as the identifier (S13). Others inquired about a feature to share a link others could use to pay them or a feature to directly request a payment from another party to solve this problem for them (P1, P6, S5). These are all features that traditional payments offer, but this implementation would go against the independent and anonymous nature of non-custodial wallets. ENS domains address those problems by providing an easier-to-remember "clear name" for a wallet. However, they are not easily accessible for novice crypto users as acquiring an ENS domain is challenging [7]. Until this process is more accessible, the concept of ENS domains should be either removed from the developed system or introduced with more guidance.

Another usability challenge the participants encountered is part of the last step in sending a transaction, the review modal, shown in Figure 6.4. All participants managed to send a transaction, but 66,67% needed help clarifying this screen, and 33,33% just checked the boxes without understanding their purpose. This review modal shows an overview of each transaction before sending it to allow the user to check the amount and recipient. In the case of a first-time transaction to the recipient, it additionally asks the user to check two conditions. First, the recipient is a privately held wallet, not a cryptocurrency exchange like e.g., Coinbase. Second, the wallet the transaction is sent

to also supports the Polygon Layer 2 network. These two conditions are essential to ensure the funds arrive as intended and don't get lost [42]. While some participants remembered the Polygon network from the crypto basics introduction they got before the interview, none of them were aware of the responsibility to verify the conditions of the checks themselves (P2, P6). If they were, they needed clarification on what the checks meant and how to verify them with their limited crypto knowledge. Typical questions were why it did matter if they sent to an exchange and if there was a way to check if the other person's wallet supports Polygon without asking them directly (P3, P5). 33,33% of the participants tried to find answers to their questions by pressing the i-button and looking at the guidance modal. While it provided helpful context to understand the Polygon network deeper, it ultimately did not answer how to verify the conditions for them (P2, P5). Another 33,33% of the participants did not try to understand or verify the conditions or wondered how they could but just checked the boxes as they recognized that would enable the send button (P1, P3). After the interviewers clarified all open questions, the participants mentioned that they would need more detailed guidance to understand the need for these checks, their consequences, and how to verify them. Furthermore, most of them added that these checks, especially the feeling that they have to double-check everything themselves, are detrimental to their confidence and sense of security while using the app. They would want the validation done automatically by the application, e.g., a bank verifies that an IBAN is of the correct format.

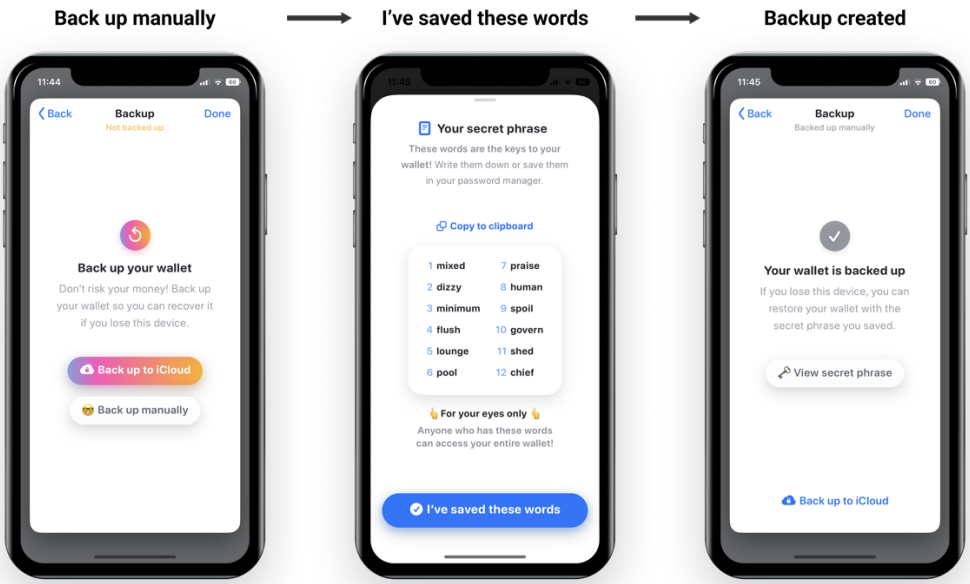

Figure 6.5: User flow to backup the wallet

Figure 6.5 shows the user flow to back up the wallet manually. Half of all participants seamlessly were able to back up their wallets and understood how to recover them in case of loss. Of the other half who encountered challenges, all those issues were rooted in the participants choosing the manual backup instead of the more user-friendly iCloud option and skipping the provided guidance text. The issue was not finding the backup mechanism in the application or it being too complicated. Instead, participants thought they created a backup but, in fact, did not because they only looked at the 12-word keyphrase without writing it down or saving it in a password manager, as the instructions suggest. This challenge might be caused by the label of the main action button, "I've saved these words" which, according to one of the participants, is misleading with its checkmark and suggests that the backup was already done. Furthermore, the participants who suggested the most that the application should provide more guidance were the ones who did not read any of it. A possible solution for the manual backup could be making the instructions more prominent or adding a pop-up after the button is pressed to confirm that the user really has saved their keyphrase.

Usability Challenges Conclusion

The general learning here is that ample guidance is necessary to overcome the usability challenges that non-custodial wallets currently pose. During the interviews, we identified two preferences among new cryptocurrency users. One group expects much guidance in the application; the other wants all complexity abstracted, so no guidance is necessary.

The group expecting guidance wants to understand a process as detailed as possible. They mention it is a difficult balance as they feel uncomfortable if there is not enough information, but if there is too much detail, they would not be able to understand it, which would defy the purpose (P2). The problem with that approach is that if users ignore the provided guidance, there would be no safety net, and issues like lost funds through mistakes in the backup process could arise. This is especially relevant since the participants who requested more guidance the most were the ones who skipped the already existing guidance regularly (P4, P6).

The other group would be overwhelmed because they would not understand that level of detail. They want all complexity abstracted by the application to focus on their goal, e.g., sending a transaction without thinking about verifying that the other party can securely receive that transaction with their wallet without the funds getting lost. The benefit of this approach is that it does not rely on the user reading and following particular guidance in the application. On the other hand, it might be hard to realize this approach without restricting the possibilities a non-custodial cryptocurrency wallet provides. In the case of the backup challenge, this might mean not offering a manual backup with guidance but purely the more user-friendly cloud approach.

Overall, both approaches have their benefits and drawbacks. In this study, with new cryptocurrency users, both preferences were equally represented, and each developer needs to make an informed assessment based on their target group of users. As this application focuses on high usability and acts purely as a payments service, not a crypto wallet for advanced use cases like lending and staking, future iterations of this application should focus on abstracting as much complexity as possible.

6.3.3 RQ3: Social Features

The social features of the application are a crucial part of the developed system and core to the research done with this thesis. The evaluation of the reasoning behind the participants' usability ratings in the field study showed that most did not see a significant value-add in the wallet's social features. Many even mentioned them as the reason for a low usability score. Therefore we focus this chapter on the research question of whether social features such as the wallet feed add value to a mobile

payments application. We explore the impact the social features had on the usability scores and evaluate the system's social aspects in-depth via the lab study's think-aloud interviews.

Before the studies, the participants reported using traditional social media such as Instagram, Facebook, or BeReal with average frequency for their age group. They were critical of the value-add social features would provide to a payments app disagreeing with the notion that a feed displaying transactions their friends send would add value to a payments app. Concern was stated regarding the privacy of such applications agreeing with the notion that financial transactions are personal and should be private. This resulted in the participants' interest in the social features of the application being low and them disagreeing with the notion of being interested in having more context about the payments inside of their circle of friends.

Usability Scores with Social Usage

Criteria	Week 1	Week 2	Overall
-	67,72	59,34	63,53
Notes sent > 0	-	53,00	53,00
Contacts added > 0	65,83	55,19	60,51

Table 6.5: Usability Scores with Social Usage

First, we explore whether the social features of the application impacted the usability scores negatively, as suggested by the participants' perception reported in the demographics survey and their qualitative reasoning for the overall low usability scores. Hence, we look at how the SUS score develops with higher usage of social features. Table 6.5 shows the score baseline and segmentations by sending notes with transactions and adding contacts. These act as a proxy to evaluate whether people who used the features necessary to have a populated wallet feed enjoy the application more. For Week 1, there is no value for Notes sent > 0, as no notes were added to transactions that week. Participants who used the system's social features showed lower usability scores of 53,00 and 60,51 or grade D. This is still considered much below average to mass market consumer software and about the same score as Microsoft Excel [43]. Testing with ANOVA, neither the differences between normal and usage with sending notes ($F=1,68$; $p=0,20$) nor the difference between normal and usage with adding contacts ($F=1,79$; $p=0,19$), are statistically significant at $p < 0,05$.

Even if the difference to the baseline is not statistically significant, the scores are still considerably lower. The segmentation criteria "Contacts added" and "Notes sent" focus on ensuring the participants used the wallet feed as intended and were able to follow their friends' transactions. Therefore, we know the lower usability scores originate

not from an incomplete experience but from the experience itself being subpar from a usability perspective. The fact that the scores even decreased with frequent social usage emphasizes the need for an in-depth exploration of the reasons for that in the lab study interviews.

Usability of the Wallet Feed

The social features of the system were a central topic during the interviews. 49 out of 175 (28,00%) of all coded statements relate to them. Only 9 of those 49 statements (18,37%) were positive remarks.

Overall, the interviews showed that the usability of the social features, specifically the wallet feed, was good. The feed was understood well, and the participants requested additional features they would theoretically like to see for a further iteration of the application. The core issue, though, and the origin of most of the unsatisfactory usability ratings across all studies, are the privacy aspects of the current wallet feed implementation. Those led all participants to the conclusion that they would like to deactivate the social features or use another payment application.

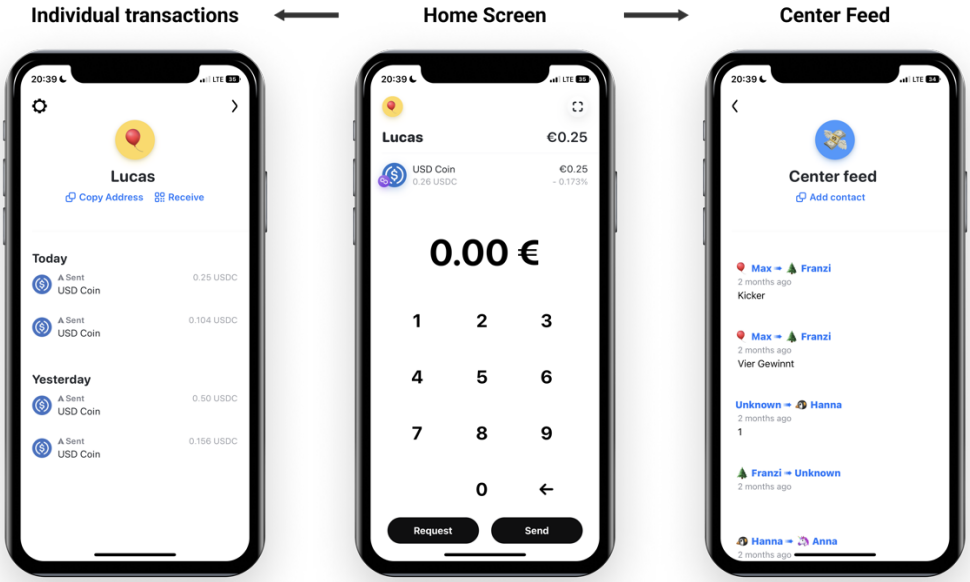

Figure 6.6: User flow to view the Wallet Feed

The wallet feed UI, as displayed in Figure 6.6, was rated positively by all participants for its simplicity. The structure of who sends funds to whom, how long ago, and the notes added to transactions were well understood by all participants.

For 66,67% of participants, finding the wallet feed when using it for the first time posed a challenge (P3, P4, P5, P6). They mentioned it would be great to have a clearly identifiable icon on the home screen to navigate to the wallet feed instead of just being able to reach it while swiping to the left on the home screen. Furthermore, multiple participants asked for incoming transactions from their friends and their notes to be displayed on the transactions screen to the left of the home screen as well. This could be realized by merging the personal transactions of a user and the wallet feed into one screen. Including the transaction amount in the wallet feed and the ability to send pictures or GIFs instead of just text, as the transaction notes were features that participants were missing from the current implementation (P2, P5, P6). Finally, one participant who rated the social features positively suggested engagement features such as displaying particular emojis next to the name of users who, e.g., sent the most funds. Popular social media apps, such as Snapchat, currently offer this feature.

Privacy concerns

There was no consensus regarding the interest in the content that the wallet feed provides the participants. 50,00% had no interest in the information and stated they would not check the feed if it were part of their primary payment app (P1, P4, P5). The other 50,00% stated it would be interesting to follow the personal relationships through the transactions happening and that seeing the feed piques their curiosity regarding the story of a particular transaction (P2, P3, P6). Use cases where they could imagine the feed to be particularly interesting are getting inspiration of where to shop, acting as an influencer by showing where they shop publicly, or generating peer pressure to pay back personal debts between friends. However, while interested in the information or mentioning particular use cases, these are edge cases, and the participants added that they would not be willing to share their transaction history in return. They prefer to use certain social features like adding notes to transactions privately with the transaction recipient without the wallet feed public between friends.

The discrepancy is that while some would theoretically be interested in following their friend's transactions with the wallet feed, nobody is willing to share their personal transactions. The notion that financial transactions are very personal and should be kept private that was agreed upon in the field study was concurred with by every participant of the lab study. Reasons cited for this belief were feeling exposed and not wanting to justify one's spending to others (P4, P6). The most common concern was

fostering negative emotions like envy between friends (P6). Especially the amount of a transaction being shown as requested by some participants would lead others to view them negatively as self-promoting and showoff (P3). These concerns apply primarily to shopping transactions like buying goods from a merchant versus personal payments between people, which is the central use case of the application. Two other participants mentioned that the reason for their concern was that seeing their friends' transactions might make them feel excluded. E.g., not being included in social activities and later seeing the corresponding transactions would impact their mental health negatively (P2, P3). Aside from not wanting their friends to see all their transactions, there is also the concern that the sensitive data that some transactions present could be shared outside of that known target group, e.g., by taking a screenshot (P3).
The general pattern in all concerns is that all participants would rather keep their transactions between the sender and receiver. Thus, 66,67% of them concluded that they would still use the application if they could turn off the wallet feed feature in general or at least on the transaction level to better control who the transactions are visible to (P2, P3, P4).

6.4 Limitations

This thesis provides a first reference implementation and evaluation of a system using Layer 2 blockchain technologies for social payments. The main goal was to evaluate the prototype built, find potential usability challenges in the experience and examine the value of social features in this context. Therefore, all findings are limited to this specific system and not payment applications in general. As a result of this, certain limitations to this study arise:

- **Short Study Duration:** The evaluation period of the field study was limited to two weeks. This timeframe is too short to fully evaluate the application's value, especially the social aspects, as the need for payments between individuals in a social circle mostly does not arise daily and cannot be forced. Potential network effects and their benefits take longer to develop. Furthermore, the participants used the application during a stressful time when they had many other academic and professional obligations, which limited their time to use and evaluate it during those two weeks. Therefore, a prolonged study is necessary to evaluate the social aspects of the application in more detail.
- **Limited Number of Observations:** Only 23 participants completed the field study. This leads to a relatively low number of 64 observed transactions and 46 observed statements for the quantitative usability scoring. Especially with

segmenting those observations by, e.g., usage, the distinct individuals making up the usability scores were very limited, with just four individuals in the case of the social usage segmentation. Six participants completed the lab study. However, this does not pose a significant limitation, as Turner et al. describe that most usability challenges are already detected with circa five study participants in the used research method [62].

- **Language Barrier:** Two participants of the lab study possess only rudimentary English skills. While all instructions and tasks they had to complete were in German, the application and its guiding text were in English. This language barrier might have created additional challenges for those individuals and impacted their perception of the application.
- **Cultural Bias:** While the cultural backgrounds of the field study participants were very diverse, the backgrounds of the lab study participants were mostly German and only European. This fact is a relevant limitation to the perception of the system's social features as social norms, especially regarding data privacy and financial transparency, might be very different in other cultures.

7 Conclusion and Future Work

The following chapter concludes this thesis and discusses its results critically. We contrast the developed prototype with existing solutions and derive learnings to improve this or other systems. Additionally, we recommend directions for further research.

7.1 Discussion

In this thesis, we present the implementation of a P2P payment system built on Polygon and its evaluation in a mixed-methods field study. The evaluation shows that the participants were able to use the system as their daily P2P payment solution throughout the two-week study. However, the usability of the application and privacy concerns regarding the social feed leave room for improvement. This discussion reflects on these results, compares them to the findings of related work, and derives learnings for further improvement of the system.

Findings in the Context of Related Work

Compared to related work proposing reference implementations using different Layer 2 scalability solutions such as Bitcoin Lightning, we can confirm some observations but also found different results in other areas because of the different technical approaches. Like Froehlich et al., we observed that the system sent transactions reliably, fast, and with low fees. In contrast to their Bitcoin Lightning implementation, we did not encounter any volatility in the transacted asset as we used the stablecoin USDC as the underlying asset. Furthermore, the usability challenge that the invoice flow of Bitcoin Lightning holds, perceived as complicated in Froehlich et al.'s study, is non-existent in the Polygon implementation.

In comparison to related work evaluating traditional social payment applications like, e.g., Venmo we cannot confirm most of their observations. The reason for that could be the different cultural circumstances in which the studies were conducted. As Venmo is not available in Europe, all research evaluating it was conducted in the USA, which is traditionally less privacy focussed than Europe [61]. More detail and references to the precise interviews where these observations were made can be found in chapter 6.3.3.

Acker found that Venmo is perceived as a form of social media by the participants [2]. Furthermore, Venmo made participants feel more connected in a few cases, according to Caraway et al. [12]. We cannot confirm either observation as in this study, participants did not particularly care about the information displayed in the social feed and mentioned they would not check it regularly like other social media. We share Caraway et al.'s observation that participants had privacy concerns regarding publically sharing a persistent history of their transactions [12]. Additionally, our interviews confirm that participants would be interested in following the transactions of their friends in some cases but don't want to share their own. However, we can't confirm the observation that participants were indifferent to the offered privacy settings and, in the end, would use the most convenient application, no matter the privacy implications [12]. This prototypical application did not offer such settings, yet participants asked for them and mentioned they would want to use another application if such settings would not exist.

We followed the design recommendations Caraway et al. gave for the development of SAS applications for our prototype implementation and agree with them [12]. The recommendation to strongly consider what content is added to the SAS and what might hinder its use led us not to display the transaction amount in the social feed. This was perceived as positive by participants, who mentioned that displaying the amount would have worsened their privacy concerns. The recommendation to consider what tone the application sets and which implications this has for the application led us to design the application in a playful and modern way using emojis, which was perceived as very positive by the participants.

Learnings from the Usability Study

During the usability study, we observed a comparably low, but for a prototype expected, System Usability Scale score of 63,53. With frequent usage of the payment functionality (transactions/week> 2), the score increased to 71,88. However, with usage of the social functionality (notes sent > 0), the score decreased to 53,00. We derived three key learnings for blockchain-based mobile social payment applications. They are based on the usability challenges identified in chapter 6.3.2. Approaches to implement these learnings and solve the underlying problems for this specific prototype implementation to improve the score to a better rating are described in chapter 7.2.

Payment applications need to offer a distinct benefit. Compared to established P2P payment apps like, e.g., PayPal, crypto-based payment applications need to offer a distinct benefit to incentivize users to switch to a new system. Though there is an interest to try novel payment solutions by users, in the long run, the additional

usability challenges posed by blockchain-based payment applications and potentially missing features keep users from changing their habits. Additionally, participants often based their answers on a comparison with systems they are already familiar with when rating a new system, e.g., with the SUS. Therefore, the perceived value of any new system depends not only on its functionality but also on whether it offers any distinct benefit to established players who possibly benefit from network effects [46]. A possible solution is to develop a system further until it reaches feature parity with the established competition and invest in creating distinct benefits through, e.g., new features that users cannot find in other solutions.

Crypto-based usability challenges need to be abstracted. Blockchain-based payment applications present various usability challenges to users rooted in using cryptocurrencies as a means of payment instead of traditional fiat currencies. Concepts like wallet addresses instead of phone numbers as identifiers or mnemonic phrases instead of customer support to recover wallets differ from what users expect based on their experiences with other systems. Additional guidance is a possible solution for technically inclined users who want to understand the system's technical details. However, instructions such as e.g., why ERC-20 tokens cannot be sent to any Ethereum wallet and the sender needs to validate with the recipient if they can receive such tokens are often complicated. Users often ignore them, as we discovered in our study (P6). Most users do not care about the underlying technological implementation and prefer the solution that provides the best usability to them. Participants in our study mentioned they just want to achieve their goal of, e.g., sending a transaction instead of needing to worry about whether a transaction can be received by the other party (P6). Therefore, those concepts and the UX challenges they create need to be abstracted for the user to make using a blockchain-based payment system a viable alternative to fiat-based systems.

Privacy Features are Key for Social Payment Apps. Privacy concerns are the main reason the SUS score went from a baseline of 63,53 to 53,00 for users who used the social features of the application. In our study, participants differed in their interest in the information provided by a SAS in a payment application. However, they all had privacy concerns, as they deem financial transactions a personal topic that should be kept private. The main concerns were getting judged for one's spending and fostering negative emotions such as jealousy. This issue is faced by many applications featuring SAS and can lead to users completely disqualifying a product [33] (P2, P3, P4). Therefore, privacy features are fundamental for social payment applications. This way, the users interested in the social aspects of an application can participate in them, and the users who are not have the opportunity to opt-out instead of being forced to use

another application that suits their preferences better. Privacy features should therefore be part of every social payment application to accommodate various users' needs and preferences.

These learnings can provide valuable insights for scholars looking to build or improve their own blockchain-based payment solutions. However, as they are based on the findings of the conducted studies, they have their own biases and limitations, as detailed in chapter 6.4. Therefore, the recommendations are especially applicable to users in continental Europe who are unfamiliar with cryptocurrencies and their underlying technologies.

7.2 Future Work

This section describes future work that could be done to improve the described system further as well as contribute to the HCI research body to a greater extent. We base these suggestions on the findings of the studies described in chapter 6 and an assessment of the current state of other payment applications.

7.2.1 Evaluate on Larger and Representative Sample

The number of participants for both studies was limited in size. We interviewed six participants in the lab study, and 23 people participated in the field study. We interviewed a group diverse in gender, age, and background. Yet, the field study participants all fall in the same 20-30 year-old, academically educated age group. Conducting the studies with a larger and more diverse set of participants could result in additional insights into usability challenges and privacy.

7.2.2 Evaluate Over Longer Time Frame

While interviewing five users over one hour is said to surface most usability issues[1] it would benefit the field study to be conducted over a longer timeframe than two weeks. This would help to evaluate whether users would actually develop a habit of using our system or switch back to another solution.

[1] https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (last accessed: 08.12.2022)

7.2.3 Conduct Study in Region With Different Cultural Norms

Caraway et al. argue that the cultural norms around privacy and finance differ strongly from culture to culture [12]. Western Europe, where this study was conducted, has strict regulations and cultural norms regarding privacy. Therefore, conducting in a region with a different approach to privacy would help to clarify the relationship between the usability scores of the application and the privacy concerns of the participants [61].

7.2.4 Achieve Feature Parity

As described in chapter 7.1, the usability ratings in the conducted studies largely depend on a comparison of the system and other established payment applications such as PayPal. Therefore, the prototype should be developed further to reach feature parity with those applications to evaluate whether and how much the missing features impact the usability scores of the system. To have an immediate positive impact on the user experience of the application, we propose five features that solve pressing issues of the prototype mentioned by the study participants:

- **Fiat on-ramp:** Participants want a connection to their bank to exchange Euros for cryptocurrencies and vice versa easily. To easily fund the wallet with USDC, a fiat on-ramp should be added to the application where users can buy USDC coins directly in the app with the OS-native payment system Apple Pay or Google Pay. While converting USDC to Euro requires access to a regulated cryptocurrency exchange and is difficult to implement, the application should also feature guidance on how to do that.
- **Shareable URL:** Participants want an easier way to share their wallet address. This could be done via a shareable URL hosted by the system, allowing the recipient of the URL to start a transaction from their wallet directly. As an alternative, the system could leverage crypto-native ENS profiles inside the application to provide users with a more straightforward way to share their wallet addresses.
- **Security:** Participants want face authentication and passcode to secure their wallet application.

7.2.5 Provide Distinct Benefits

As recommended in chapter 7.1, blockchain-based payment apps need to offer a distinct benefit compared to established payment applications to attract users in the long term. Therefore, the prototype should be developed further to provide unique and useful features to users that established payment applications don't offer. We propose three

features that payment applications such as PayPal currently do not offer in Germany. Two are for immediate iteration on the prototype, and one requires more development effort:

- **Engagement features:** Currently, social payment applications like Venmo do not exist in Germany. Adding engagement features such as likes and comments to the wallet's social feed would allow users to engage with their friends besides purely following transactions on the feed.
- **Payment insights:** The public nature of all transactions on the blockchain allows for analytics features to provide insights to the user that other apps can't for regulatory reasons, e.g., how high a user's spending was compared to other users.
- **Liquidity market:** The wallet could offer a lending and borrowing feature providing credit or interest to users. This could be implemented with an open-source protocol such as Aave[2] and offer users liquidity services based only on their wallet without any regulatory screening.

7.2.6 Offer Privacy Controls

Privacy concerns were one of the key issues the participants had with the prototype implementation of the wallet, and the third learning in chapter 7.1 is that privacy features are key for social payment applications. To resolve these privacy controls, the prototype should be developed further by offering additional privacy controls to users. First, a friend request system should be introduced, where users must accept when somebody adds them as a contact. This was not necessary for the prototypical implementation of the study as all participants belong to the same social circle, but it is crucial for a real-world application. Second, users should be able to opt out of sharing their transaction history with friends completely. Third, users should be able to control on a transaction level who the transaction they are about to send can be seen by: Public, friends, or just the recipient.

[2]https://aave.com/ (last accessed: 08.12.2022)

List of Figures

List of Tables

Bibliography

[1] S. Abramova, A. Voskobojnikov, K. Beznosov, and R. Böhme. "Bits Under the Mattress: Understanding Different Risk Perceptions and Security Behaviors of Crypto-Asset Users." In: Dec. 2021, pp. 1–19. DOI: 10.1145/3411764.3445679.

[2] A. Acker and D. Murthy. "Venmo: Understanding Mobile Payments as Social Media." In: *SMSociety '18: Proceedings of the 9th International Conference on Social Media and Society*. Dec. 2018, pp. 5–12. ISBN: 9781450363341. DOI: 10.1145/3217804.3217892.

[3] A. Alshamsi and P. P. Andras. "User perception of Bitcoin usability and security across novice users." In: *International Journal of Human-Computer Studies* 126 (2019), pp. 94–110. ISSN: 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2019.02.004.

[4] A. Bandura. "Social Learning Theory of Aggression." In: *The Journal of communication* 28 (Dec. 1978), pp. 12–29. DOI: 10.1111/j.1460-2466.1978.tb01621.x.

[5] A. Bangor, P. Kortum, and J. Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale." In: *J. Usability Stud.* 4 (Dec. 2009), pp. 114–123.

[6] A. Bangor, P. T. Kortum, and J. T. Miller. "An Empirical Evaluation of the System Usability Scale." In: *International Journal of Human–Computer Interaction* 24.6 (2008), pp. 574–594. DOI: 10.1080/10447310802205776.

[7] D. P. Bauer. "Ethereum Name Service." In: *Getting Started with Ethereum : A Step-by-Step Guide to Becoming a Blockchain Developer*. Berkeley, CA: Apress, 2022, pp. 103–106. ISBN: 978-1-4842-8045-4. DOI: 10.1007/978-1-4842-8045-4{_}9.

[8] J. Brooke. "SUS: A quick and dirty usability scale." In: *Usability Eval. Ind.* 189 (Dec. 1995).

[9] B. Bruegge. *Object Oriented Software Engineering Using UML, Patterns, and Java*. Prentice Hall International; Auflage: 3rd revised edition., 2009. ISBN: 0138152217.

[10] M. Burke, C. Marlow, and M. Lento. "Feed Me: Motivating Newcomer Contribution in Social Network Sites." In: *CHI 2009*. Dec. 2009. DOI: 10.1145/1518701.1518847.

[11] K. Busse, M. Tahaei, K. Krombholz, E. von Zezschwitz, M. Smith, J. Tian, and W. Xu. "Cash, Cards or Cryptocurrencies? A Study of Payment Culture in Four Countries." In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 2020, pp. 200–209. DOI: 10.1109/EuroSPW51379.2020.00035.

[12] M. Caraway, D. A. Epstein, and S. A. Munson. "Friends Don't Need Receipts: The Curious Case of Social Awareness Streams in the Mobile Payment App Venmo." In: *Proc. ACM Hum.-Comput. Interact.* 1.CSCW (Dec. 2017). DOI: 10.1145/3134663.

[13] B. Carminati, E. Ferrari, and N. H. Tran. "Enforcing Trust Preferences in Mobile Person-to-Person Payments." In: *2013 International Conference on Social Computing*. 2013, pp. 429–434. DOI: 10.1109/SocialCom.2013.67.

[14] S. Centellegher, G. Miritello, D. Villatoro, D. Parameshwar, B. Lepri, and N. Oliver. "Mobile Money: Understanding and Predicting Its Adoption and Use in a Developing Economy." In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.4 (Dec. 2018). DOI: 10.1145/3287035.

[15] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor. "Blockchain and Scalability." In: *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. 2018, pp. 122–128. DOI: 10.1109/QRS-C.2018.00034.

[16] C.-W. Chiang, C. Anderson, C. Flores-Saviaga, E. J. Arenas, F. Colin, M. Romero, C. Rivera-Loaiza, N. E. Chavez, and S. Savage. "Understanding Interface Design and Mobile Money Perceptions in Latin America." In: *Proceedings of the 8th Latin American Conference on Human-Computer Interaction*. CLIHC '17. New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 9781450354295. DOI: 10.1145/3151470.3151473.

[17] G. Del Monte, D. Pennino, and M. Pizzonia. "Scaling blockchains without giving up decentralization and security: a solution to the blockchain scalability trilemma." In: Dec. 2020, pp. 71–76. DOI: 10.1145/3410699.3413800.

[18] G. Disterer. "ISO/IEC 27000, 27001 and 27002 for Information Security Management." In: *Journal of Information Security* 04 (Dec. 2013), pp. 92–100. DOI: 10.4236/jis.2013.42011.

[19] D. A. Epstein, A. Borning, and J. Fogarty. "Fine-Grained Sharing of Sensed Physical Activity: A Value Sensitive Approach." In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 489–498. ISBN: 9781450317702. DOI: 10.1145/2493432.2493433.

[20] C. Fiesler and N. Proferes. ""Participant" Perceptions of Twitter Research Ethics." In: *Social Media + Society* 4.1 (2018), p. 2056305118763366. DOI: 10.1177/2056305118763366.

[21] M. Froehlich. "Usable Cryptocurrency Systems." PhD thesis. 2022.

[22] M. Froehlich, C. Kobiella, A. Schmidt, and F. Alt. "Is it Better With Onboarding? Improving First-Time Cryptocurrency App Experiences." In: Dec. 2021, pp. 78–89. DOI: 10.1145/3461778.3462047.

[23] M. Froehlich, J. A. Vega Vermehren, F. Alt, and A. Schmidt. "Implementation and Evaluation of a Point-Of-Sale Payment System Using Bitcoin Lightning." In: Association for Computing Machinery (ACM), Oct. 2022, pp. 1–12. ISBN: 9781450396998. DOI: 10.1145/3546155.3546700.

[24] M. Froehlich, M. Wagenhaus, A. Schmidt, and F. Alt. "Don't Stop Me Now! Exploring Challenges Of First-Time Cryptocurrency Users." In: Dec. 2021, pp. 138–148. DOI: 10.1145/3461778.3462071.

[25] M. Froehlich, F. Waltenberger, L. Trotter, F. Alt, and A. Schmidt. "Blockchain and Cryptocurrency in Human Computer Interaction: A Systematic Literature Review and Research Agenda." In: *Designing Interactive Systems Conference*. DIS '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 155–177. ISBN: 9781450393584. DOI: 10.1145/3532106.3533478.

[26] M. Fröhlich, F. Gutjahr, and F. Alt. "Don't Lose Your Coin! Investigating Security Practices of Cryptocurrency Users." In: *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. DIS '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1751–1763. ISBN: 9781450369749. DOI: 10.1145/3357236.3395535.

[27] U. Gallersdörfer, L. Klaaßen, and C. Stoll. "Energy Consumption of Cryptocurrencies Beyond Bitcoin." In: *Joule* 4 (Dec. 2020). DOI: 10.1016/j.joule.2020.07.013.

[28] X. Gao, G. D. Clark, and J. Lindqvist. "Of Two Minds, Multiple Addresses, and One Ledger: Characterizing Opinions, Knowledge, and Perceptions of Bitcoin Across Users and Non-Users." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1656–1668. ISBN: 9781450333627. DOI: 10.1145/2858036.2858049.

[29] J. Göbel and A. E. Krzesinski. "Increased block size and Bitcoin blockchain dynamics." In: *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*. 2017, pp. 1–6. DOI: 10.1109/ATNAC.2017.8215367.

[30] K. Grauer, W. Kueshner, E. McMahon, and H. Updegrave. *The Chainalysis State of Web3 Report*. 2022.

[31] A. Hiniker, S. N. Patel, T. Kohno, and J. A. Kientz. "Why Would You Do That? Predicting the Uses and Gratifications behind Smartphone-Usage Behaviors." In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 634–645. ISBN: 9781450344616. DOI: 10.1145/2971648.2971762.

[32] H. Jang, S. H. Han, and J. H. Kim. "User Perspectives on Blockchain Technology: User-Centered Evaluation and Design Strategies for DApps." In: *IEEE Access* 8 (2020), pp. 226213–226223. DOI: 10.1109/ACCESS.2020.3042822.

[33] M. Johnson, S. Egelman, and S. M. Bellovin. "Facebook and Privacy: It's Complicated." In: *Proceedings of the Eighth Symposium on Usable Privacy and Security*. SOUPS '12. New York, NY, USA: Association for Computing Machinery, 2012. ISBN: 9781450315326. DOI: 10.1145/2335356.2335369.

[34] J. J. Kaye, M. McCuistion, R. Gulotta, and D. A. Shamma. "Money Talks: Tracking Personal Finances." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 521–530. ISBN: 9781450324731. DOI: 10.1145/2556288.2556975.

[35] I. E. Khairuddin, C. Sas, S. Clinch, and N. Davies. "Exploring Motivations for Bitcoin Technology Usage." In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 2872–2878. ISBN: 9781450340823. DOI: 10.1145/2851581.2892500.

[36] S. Khandekar, J. Liang, A. Razaque, F. Amsaad, and M. Abdulgader. "Security Research of a Social Payment App and Suggested Improvement." In: *Communications on Applied Electronics* 4 (Dec. 2016), pp. 14–21. DOI: 10.5120/cae2016652059.

[37] A. Khanna. "Venmo'ed: Sharing Your Payment Data With the World." In: 2015.

[38] T. Klein, H. Pham Thu, and T. Walther. "Bitcoin is not the New Gold – A comparison of volatility, correlation, and portfolio performance." In: *International Review of Financial Analysis* 59 (2018), pp. 105–116. ISSN: 1057-5219. DOI: https://doi.org/10.1016/j.irfa.2018.07.010.

[39] K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl. "The Other Side of the Coin: User Experiences with Bitcoin Security and Privacy." In: Dec. 2017, pp. 555–580. ISBN: 978-3-662-54969-8. DOI: 10.1007/978-3-662-54970-4{_}33.

[40] B. Laugwitz, T. Held, and M. Schrepp. "Construction and Evaluation of a User Experience Questionnaire." In: *USAB 2008*. Vol. 5298. Dec. 2008, pp. 63–76. ISBN: 978-3-540-89349-3. DOI: 10.1007/978-3-540-89350-9{_}6.

[41] J. Lazar, J. H. Feng, and H. Hochheiser. *Research Methods in Human-Computer Interaction*. Dec. 2017, pp. 1–560.

[42] R. Leonhard. "Decentralized Finance on the Ethereum Blockchain." In: *SSRN Electronic Journal* (Dec. 2019). DOI: 10.2139/ssrn.3359732.

[43] J. R. Lewis. "The System Usability Scale: Past, Present, and Future." In: *International Journal of Human-Computer Interaction* 34.7 (July 2018), pp. 577–590. ISSN: 15327590. DOI: 10.1080/10447318.2018.1455307.

[44] Z. Li, H. Dong, C. Floros, A. Charemis, and P. Failler. "Re-examining Bitcoin Volatility: A CAViaR-based Approach." In: *Emerging Markets Finance and Trade* 58.5 (2022), pp. 1320–1338. DOI: 10.1080/1540496X.2021.1873127.

[45] J. Mattke, C. Maier, and L. Reis. "Is Cryptocurrency Money? Three Empirical Studies Analyzing Medium of Exchange, Store of Value and Unit of Account." In: *Proceedings of the 2020 on Computers and People Research Conference*. SIGMIS-CPR'20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 26–35. ISBN: 9781450371308. DOI: 10.1145/3378539.3393859.

[46] A. Milne. "What is in it for us? Network effects and bank payment innovation." In: *Journal of Banking & Finance* 30.6 (2006), pp. 1613–1630. ISSN: 0378-4266. DOI: https://doi.org/10.1016/j.jbankfin.2005.09.006.

[47] M. Mita, K. Ito, S. Ohsawa, and H. Tanaka. "What is Stablecoin?: A Survey on Price Stabilization Mechanisms for Decentralized Payment Systems." In: *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*. 2019, pp. 60–66. DOI: 10.1109/IIAI-AAI.2019.00023.

[48] C. Newlove. "The rise of peer-to-peer (P2P) payments on mobile." In: *Medium* (2018).

[49] S. Paavolainen and C. Carr. "Security Properties of Light Clients on the Ethereum Blockchain." In: *IEEE Access* 8 (2020), pp. 124339–124358. DOI: 10.1109/ACCESS.2020.3006113.

[50] G. A. Pierro and R. Tonelli. "Can Solana be the Solution to the Blockchain Scalability Problem?" In: *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 2022, pp. 1219–1226. DOI: 10.1109/SANER53432.2022.00144.

[51] J. Raacke and J. Bonds-Raacke. "MySpace and Facebook: Applying the Uses and Gratifications Theory to Exploring Friend-Networking Sites." In: *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society* 11 (Dec. 2008), pp. 169–174. DOI: 10.1089/cpb.2007.0056.

[52] R. A. P. Rajan. "Serverless Architecture - A Revolution in Cloud Computing." In: *2018 Tenth International Conference on Advanced Computing (ICoAC)*. 2018, pp. 88–93. DOI: 10.1109/ICoAC44903.2018.8939081.

[53] R. Rice and G. Love. "Electronic Emotion: Socioemotional Content in a Computer-Mediated Communication Network." In: *Communication Research* 14.1 (1987), pp. 85–108. DOI: 10.1177/009365087014001005.

[54] C. Sas and I. E. Khairuddin. "Design for Trust: An Exploration of the Challenges and Opportunities of Bitcoin Users." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 6499–6510. ISBN: 9781450346559. DOI: 10.1145/3025453.3025886.

[55] J. Sauro and J. Dumas. "Comparison of three one-question, post-task usability questionnaires." In: *Conference on Human Factors in Computing Systems - Proceedings*. Dec. 2009, pp. 1599–1608. DOI: 10.1145/1518701.1518946.

[56] M. Schrepp, A. Hinderks, and J. Thomaschewski. "Construction of a Benchmark for the User Experience Questionnaire (UEQ)." In: *International Journal of Interactive Multimedia and Artificial Intelligence* 4 (Dec. 2017), pp. 40–44. DOI: 10.9781/ijimai.2017.445.

[57] M. Sewak and S. Singh. "Winning in the Era of Serverless Computing and Function as a Service." In: *2018 3rd International Conference for Convergence in Technology (I2CT)*. 2018, pp. 1–5. DOI: 10.1109/I2CT.2018.8529465.

[58] C. Sguanci, R. Spatafora, and A. M. Vergani. "Layer 2 Blockchain Scaling: a Survey." In: (July 2021).

[59] S. Silfverberg, L. A. Liikkanen, and A. Lampinen. ""I'll Press Play, but I Won't Listen": Profile Work in a Music-Focused Social Network Service." In: *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*. CSCW '11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 207–216. ISBN: 9781450305563. DOI: 10.1145/1958824.1958855.

[60] B. Singh and J. K.s. "Comparative Study on Various Methods and Types of Mobile Payment System." In: Dec. 2012, pp. 143–148. ISBN: 978-1-4673-1869-3. DOI: 10.1109/MNCApps.2012.44.

[61] G. Steinke. "Data privacy approaches from US and EU perspectives." In: *Telematics and Informatics* 19.2 (2002), pp. 193–200. ISSN: 0736-5853. DOI: https://doi.org/10.1016/S0736-5853(01)00013-2.

[62] C. Turner, J. Lewis, and J. Nielsen. "Determining Usability Test Sample Size." In: *International Encyclopedia of Ergonomics and Human Factors*. Vol. 3. Nov. 2006.

[63] C. Unger, D. Murthy, A. Acker, I. Arora, and A. Chang. "Examining the Evolution of Mobile Social Payments in Venmo." In: *International Conference on Social Media and Society*. SMSociety'20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 101–110. ISBN: 9781450376884. DOI: 10.1145/3400806.3400819.

[64] K. D. Vohs, N. L. Mead, and M. R. Goode. "The Psychological Consequences of Money." In: *Science* 314.5802 (2006), pp. 1154–1156. DOI: 10.1126/science.1132491.

[65] A. Voskobojnikov, S. Abramova, K. Beznosov, and R. Böhme. "Non-Adoption of Crypto-Assets: Exploring the Role of Trust, Self-Efficacy, and Risk." In: *ECIS*. 2021.

[66] A. de Vries, U. Gallersdörfer, L. Klaaßen, and C. Stoll. "Revisiting Bitcoin's carbon footprint." In: *Joule* (Dec. 2022). DOI: 10.1016/j.joule.2022.02.005.

[67] C. Wagner and M. Strohmaier. "The Wisdom in Tweetonomies: Acquiring Latent Conceptual Structures from Social Awareness Streams." In: *Proceedings of the 3rd International Semantic Search Workshop*. SEMSEARCH '10. New York, NY, USA: Association for Computing Machinery, 2010. ISBN: 9781450301305. DOI: 10.1145/1863879.1863885.

[68] F. Waugh and R. Holz. "An empirical study of availability and reliability properties of the Bitcoin Lightning Network." In: (June 2020). DOI: 10.48550/arxiv.2006.14358.

[69] F. Wherry, K. Seefeldt, and A. Alvarez. "To Lend or Not to Lend to Friends and Kin: Awkwardness, Obfuscation, and Negative Reciprocity." In: *Social Forces* 98 (Dec. 2019). DOI: 10.1093/sf/soy127.

[70] X. Zhang, S. Tang, Y. Zhao, G. Wang, H. Zheng, and B. Zhao. "Cold Hard E-Cash: Friends and Vendors in the Venmo Digital Payments System." In: *Proceedings of the International AAAI Conference on Web and Social Media* 11.1 (May 2017), pp. 387–396. DOI: 10.1609/icwsm.v11i1.14873.

[71] X. Zhao, N. Salehi, S. Naranjit, S. Alwaalan, S. Voida, and D. Cosley. "The Many Faces of Facebook: Experiencing Social Media as Performance, Exhibition, and Personal Archive." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 1–10. ISBN: 9781450318990. DOI: 10.1145/2470654.2470656.