Trustlines Network

the open protocol which enables permissionless mobile payments based on people powered money

Draft Version 0.3
February 2017

Heiko Hees
Gustav Friis
Kristoffer Naerland

Abstract

Present-day virtually mined cryptocurrencies suffer from a number of issues limiting broad adoption. Cryptocurrencies being supply constrained virtual commodity money, are no good fit for our current economic reality, where money is mostly created as debt by banks, and the capital cost incurred when provisioning cryptocurrencies is prohibitive. Moreover, the general practical requirements of upfront deposit money as well as to have accounts with banks and exchanges to acquire and exchange crypto currencies, is hindering adoption.

As a solution we propose Trustlines Network; the decentralized, permissionless, open platform to host *currency networks* in which money is represented as IOUs issued by its participants. The design extends on the original Ripple idea with a strong focus on ease of adoption and smart contract interoperability. The platform comes complemented with a mobile application and the design is such that for participants, there is no need to interact with any centralized services such as banks or exchanges.

Table of contents

On Adoption of Blockchain based Payments

Payments have been seen as the blockchain killer application ever since Bitcoin made its appearance. Sending value from A to B without intermediaries, without permission, without centralized services, was something not seen before. Despite an increasing number of innovations aimed at usability, cryptocurrencies for payments today come with a rather bad user experience. The average user needs to install a wallet, open an account with a fiat-crypto exchange, prove his identity, buy tokens, endure their volatility, secure the wallet, deal with exchange rates when comparing product offers and finally pay, and wait minutes or hours to have the payment confirmed. Even worse, the practical constraint of having to use a bank account when acquiring cryptocurrencies make them non-inclusive for, among others, the 2 billion unbanked people in this world, who would benefit the most in the first place.

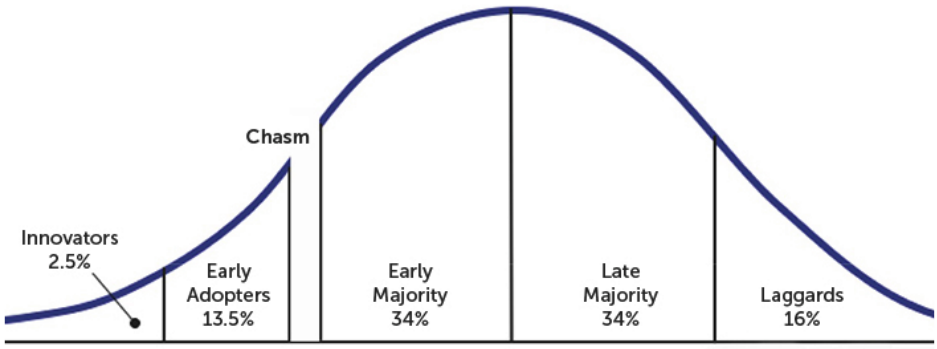
The Problem with Commodity Money

Cryptocurrencies can be seen as virtual commodity money. Coins are minted according to a certain schedule which is independent of the economy's nominal output and money demand. Economists have pointed out widespread usage of cryptocurrencies with an inbuilt deflationary nature would lead to unemployment, since wages generally don't adjust downward. Further, when cryptocurrencies are acquired to be used for payments, they feel more like prepaid accounts or gift certificates than actual money. Pay for them today, eventually use them later. This is quite the opposite of what people are used to. Most of us buy something today using credit we have with our bank or credit card provider and eventually pay off our debt later. People do not need to have money to buy stuff, all they need is credit. The former is scarce while the latter is abundant.

Generally speaking, when people talk about money, it's actually debt they are referring to. This is aligned with the way most of today's money is created by banks. If a bank gives out a loan to a customer it creates commercial bank money. On their balance sheet the created money is a liability, while the customer's promise to pay the loan back is an asset. Definitions may vary a bit, but the essential insight is: Money is debt - and governments, businesses and most people maintain debt relationships, which are also used to facilitate payments. Cryptocurrencies as commodity money might be good for savings, but their supply constraints and the upfront efforts and capital required to acquire them, render them infeasible for day-to-day commerce.

Impact on Adoption

Accounting for a reality - where money is debt - leads to the conclusion that we won't see broad adoption of mined cryptocurrencies for payments in our economy. Furthermore, there is this somewhat paradoxical situation that one in general needs to rely on centralized payment services to get access to their decentralized alternatives.



These conditions restrict most individuals, communities and businesses to benefit from the value offered by blockchain based complementary currencies. Rendering it fair to say blockchain based payments are still a niche application only used by a tiny group of innovators.

We believe broader adoption of cryptocurrencies for payments depend on a value proposition that can outcompete existing centralized payment services, instead of relying on them.

Implementing Money on the Social Graph

The current adoption problem of cryptocurrencies can be mitigated by switching from commodity based money to an IOU based money system, which resembles how banks create money in the real world. The proposed solution is similar by being based on a network of IOUs but diverges in its implementation where the social graph is used to allow for decentralized money issuance by its users. Paths in a network of IOUs between trusting participants (e.g. friends) can be used to transfer money between otherwise non trusting participants.

Modeling money as decentralized issued IOUS solves some main problems of cryptocurrencies. Users don't need a bank account nor upfront capital to participate and money is not scarce but created according to demand and it can be denominated in (a familiar fiat) currency.

IOUs on the Blockchain

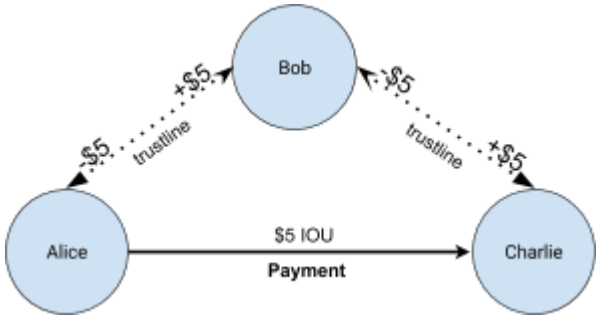
"An IOU (abbreviated from the phrase 'I owe you') is usually an informal document acknowledging debt. IOUs usually specify the debtor, the amount owed, and sometimes the creditor. IOUs may be signed or carry distinguishing marks or designs to ensure authenticity." -Wikipedia

IOUs can be used to model payments notarized on the Ethereum blockchain. As an example, a bank could create an USD denominated IOU-Token contract and promise to feel obligated to owe (and eventually pay out) a certain amount of USD to whomever is a holder of tokens. It's a claim the holder of tokens has against the issuer for whom it's an IOU. Such a bearer instrument is fungible as the traded IOUs have the same properties independent of who the holder is. Therefore such IOUs can be transferred and fiat backed IOUs, which are issued by widely trusted entities, can bring commercial money to the blockchain. IOUs have the potential to solve some of the issues that users have with cryptocurrencies like their volatility, limited supply and exchange rate inconveniences. Bank issued IOUs come with some

drawbacks though, as they are permissioned (i.e. the banks need to know their customers and can freeze their accounts) and holders need to trust in the solvency of the issuer. The privilege of money issuance is given to those central actors which can convince a critical mass of users to trust them. With such fiat-backed IOUs on the blockchain, one still has a decentralized infrastructure for accounting, but token issuance is now rather centralized. What's more, such tokens are prepaid and users need to have an account with the issuing party.

IOU Networks

A system where every user had the privilege to create money by issuing IOUs could solve the prepaid, permissioned and centralization issues - at least if there wasn't the problem of trust.
If a random individual on the internet would issue USD denominated IOUs, it is hard to imagine that they would have value to anyone - except for people who actually know and trust the issuer to feel obligated to settle the IOUs for cash. If instead IOUs are issued between friends the aforementioned trust problem can be mitigated. Let us look at an example; Alice and Bob are good friends and trust each other. They could set up an agreement in order to account for what they owe each other on the blockchain. They also agree that there are limits to the amount they'd lend each other. Further they agree that whatever their netted balance is, accounted for on the blockchain, is representing an IOU in the real world. We call such a bilateral agreement a *trustline*. Now, if Bob also has a *trustline* with his friend Charlie we can facilitate a payment from Alice to Charlie although they don't know or trust each other: Say Alice wants to do a $5 payment to Charlie and there exist *trustlines* Alice-Bob and Bob-Charlie. We then could set this up as an atomic transaction where Alice signs that she owes $5 to Bob and Bob signs that he owes $5 to Charlie.



After this transaction Alice owes $5 to Bob and Bob owes $5 to Charlie. If we model this as accounts, we get that Alice has $5 less on her account, Bob received $5 and spent $5 so his account balance did not change and Charlie has $5 more on his account. Note how Charlie does not depend on the solvency of Alice, but only on his friend Bob to stick to his promises. A default of Alice would directly affect her friend Bob, but not Charlie.

This example shows, how payments between strangers can be implemented by chaining payments on a network of bilateral trust relationships. Users are obligated to the netted balance of all transfers that they facilitated (or helped to facilitate). These chains of friends-of-friends-of-friends allow participants to make fast, cheap and transparent payments to people they don't know through their intermediate trusted social network. Similarly to the theory of six degree of separation, every participant in these chains will not necessarily need to trust more than a few handfuls of connections in order to have access to a path enabling global payments. This basic idea of "money transfer without money

movement" is used in today's Hawala and correspondent banking system. The idea led Ryan Fugger to invent RipplePay back in 2004. It is also the foundation of the Trustlines Network.

The Trustlines Network

The Trustlines Network is designed to solve the issues which hinder broad adoption of blockchain based payments. It's the original Ripple idea built on Ethereum with a focus on interoperability and adoption. The foundation is an open P2P platform to host currency *networks* which is complimented with a mobile payment application to provide easy access to the system.

Permissionless Mobile Payments

Ease of adoption is the outstanding property of the system, when compared to traditional payment services. Joining the payment network will be as easy as:

1. Downloading the mobile application from an app store
2. Connect by creating a creditline with a friend onr trusts
3. Ready: Send and receive payments within the network

The app will hide all the complexity of blockchain technology from regular users. These don't have to deal with Ether, weird hex strings or understand the underlying technology. Most importantly users do not have to register with any exchange to buy crypto tokens, nor do they need to have any money in the first place. All they need is a smartphone and a friend who is already part of the network.

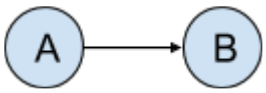
Open Platform

The Trustlines Network is an open platform extendable by smart contracts on the Ethereum blockchain, which in turn enables interoperability and synergy-effects with other Ethereum based Dapps. The open-source nature of the Trustlines Network furthermore enables 3rd party developers to build and monetise specialized applications on top of the platform. The platform can support a variety of possible applications, of which some of these are listed in the 'applications beyond payments' section.

Terminology

This section explains some core concepts used throughout the remainder of this document.

Credit Line

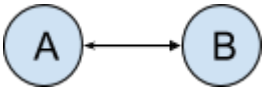


An agreement between two trusting entities (e.g. friends), which allows one party to be indebted to the other. If the *credit line* is used, the granter is the creditor and the one using the *credit line* is the debtor. The agreement is signed by both parties. The creditor signs the maximum amount he is allowing the debtor to be indebted with. The debtor signs that he is obligated to pay off the debt.

A *credit line* agreement can optionally specify an interest rate along with rules regarding changes on the limit and interest rate (effectively enabling loans and other financial products). A *credit line* can be altered by the creditor at any time (unless otherwise agreed on). A *credit line* exists in the context of a *currency network*

and therefore implicitly states the currency in which the accounting is done.

Trustline



A *trustline* is a bilateral agreement between exactly two users which encompasses two *credit lines* (each party giving one to the other party) as well as a *balance* which indicates if and how much one party owes the other one. The given *credit lines* are denominated in the same currency. The *credit lines* can have different limits with the edge case being that one party gives a zero amount *credit line*. A *trustline* can be closed if the balance between both parties is zero.

Trustline Balance

The *trustline balance* accounts for what two parties to a *trustline* agreement owe each other. From the perspective of a user

- positive balance is what the other party to the agreement owes the user
- negative balance is what the user owes to the other party of the agreement

A negative balance is a debt obligation. A positive balance is a debt claim**.** The amount of a user's positive balance equals to a negative balance of the other party of an agreement and vice versa. A user must/can not owe more to the other party of the agreement than the nominal value of the given *credit line* given by that party. Exceptions are, if:

1. The creditor reduces the *credit line* so that it is lower than the balance
2. applied interest can increase the balance beyond the given *credit line*

Currency Network

A *currency network* is the totality of all *trustlines* denominated in the same currency. Becoming a participant in a *currency network* requires agreement on the terms that a user's *trustlines* can be used by the network to facilitate *multi-hop* transfers.

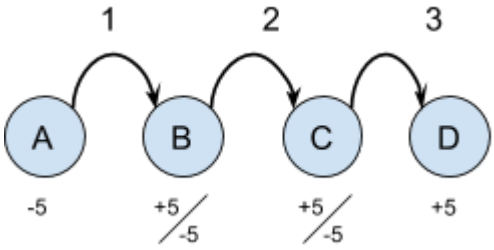
Trustline Money

We use the term *trustline money* to refer to *trustline* imbalances, which are IOUs. Its denomination is defined by the respective *currency network*. It can be denominated in a real world currency, a commodity, a crypto token or any other unit of account. Its main properties are divisibility, durability, portability, uniformity, convertibility and fungibility.

Trustline money can be exchanged for ETH and vice versa on an on-chain exchange. Therefore one can always tell what the value of *trustline* money is compared to other currencies. *Trustline* money has properties very similar to commercial bank money, as it is an IOU. Also its fungibility depends on a network of entities that cooperate when moving debt claims through the network at face value to facilitate payments.

Multi-Hop Transfer (rippling)

A *multi-hop* transfer refers to the mechanism used when multiple users in a *currency network* cooperate (implicitly as part of the agreement with the network) to facilitate a payment between users which have no direct *trustline* agreement.
The path encompasses all involved users and their *trustlines*.

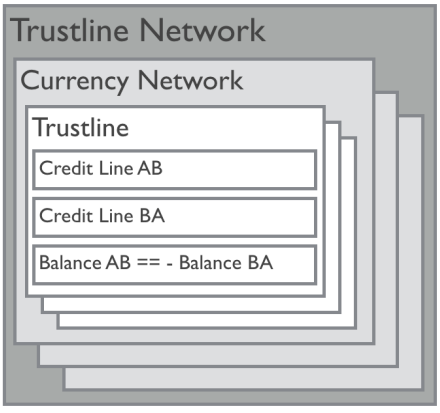


When a *currency network* is pictured as a graph, a chosen path is edges that connect the two parties to a payment. Each edge can also be referred to as a *hop* and a "N hops" transfer would indicate that N edges (and N+1 participants) have been involved in a payment. If the number of hops N is larger than 1, then N-1 participants helped to facilitate the transfer.

As the balances along a path are all updated we also say, that the balance updates are *rippled* through the system. From the perspective of an intermediary the total balance over all its *trustlines* remains unchanged, as it only forwarded what it received.

Trustlines Network

Trustlines Network refers to the blockchain based platform which hosts a network of *currency networks*. It is also the name of the project and as such also refers to the additional tooling like the mobile app, which makes it easy for users to interact with the system.



The Trustlines Network from a User Perspective

The following describes how the system works from the perspective of a regular user, who interacts with the system using his smartphone. The basic concepts are similar for communities or businesses who build accounting systems on the *Trustlines Network*, but we don't go into a detailed explanation of such implementations in this document.

Joining the System

Users join the network by downloading a mobile app which automatically creates a wallet for the user. Next users select a *currency network* they want to join and create a *trustline* with at least one friend, in order to be connected to the network of *trustlines*. Note that users neither have to register with some organization nor do they have to deposit money or link a credit card or bank account. Furthermore, no upfront ETH is required and users can start spending based on the *credit line* given by a friend.

Adding Friends and Trustlines

Adding friends on the *Trustlines Network* is very similar to other applications but with higher privacy standards. Friends can be added in the real world by one phone scanning a QR code from the other. This is also possible by receiving an *add me* link, or by pasting in the address of the friends account. Users can furthermore see who of their friends are using the *Trustlines Network*. Friends (contacts) already participating in the network can be automatically discovered. When adding a

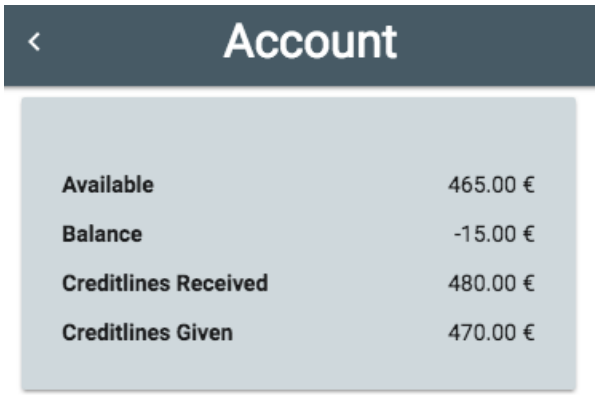
friend a *trustline* is created and both explicitly agree to the mechanics of the system by giving each other a *credit line* up to some limit.

Payments

Once a trustline is set up, users can start sending payments to anyone in the network. For making payments a target address and amount is needed. Addresses are either available in the apps contact list or received by text message or by scanning QR codes (which can also include the value). Before sending a payments a mandatory fee is displayed. Both users get a notification once the transfer is initially confirmed and once it is finalized.

Balances

Users are shown their (total) *available*, next to their (total) *balance* on the main screen of the app. A positive *balance* is the total amount their friends owe them, while a negative balance is the total amount they owe to their friends. The *available* amount, is the sum of all their *credit line* limits plus their balance. Users can also inspect the balances with their friends individually but this is of limited information, because these are changed with every transfer that is routed through their account (i.e. these balances change without user interaction).



Note: The overall balance is not changed when routing transfers through an account, except for tiny amounts which represent interest or earned fees.

Currencies

The system supports payments denominated in many currencies, fiat as well as user created ones. There is a distinct network for each currency. Users can participate in multiple networks.

Cross Currency Payments

Users can send payments to users which only have *trustlines* in other currencies. To initiate a payment either the received payment instructions contain the target address, amount and currency or the user enters them manually. Users are then presented the best conversion rate offer and fee and the total amount the payment would costs them, before confirming the transfer.

Adding / Withdrawing Cash

Adding or withdrawing funds to the *Trustlines Network* can be done by interacting in the real world with any other participant of the network. It is basically implemented as a trustlines network payment for cash received.

If a user handles cash to a receiving user, the receiver sends a payment to the initiating user, thus crediting him the amount in the network. Withdrawing works the other way round. Note, that the parties in the transaction don't need to be friends. Converting cash to *trustline money* is also the method to pay back debt in the *Trustline Network*, which might be necessary if a user on average spent more than he received. Withdrawing cash

is the method to reduce balance surpluses given in the *Trustline Network* (i.e. if a user on average receives more than he spends). Just like with commercial money based bank accounts, adding or withdrawing cash is possible but not necessary for the system to operate and be useful.

Trustline Money Security

Credit relationships are always with trusted friends only. The system guarantees that payments which are routed through untrusted accounts are unaffected if one of these accounts defaults. So a payment from Alice → Bob → Charlie is secure for Charlie even if Alice defaults shortly after. This is because Charlie gets the amount credited from Bob. Bob might suffer a loss, which is “okay”, as he trusted Alice. Thus the default of a payment initiator doesn’t affect the receiver.

Real World Enforcement of Debt Claims

Friends paying back the credit is not guaranteed and is not legally enforceable within the *Trustlines Network*. With social norms effect on informal enforceability, a friend will have no incentive to disobey obligations, due to the negative impact this will have on the friendship as well as the social reputation among the rest of the community. All IOUs on the network are notarized on the Ethereum blockchain. Therefore there is a tamper proof record of all agreements and their balances. In case of a friend not being able to pay back the obligation, there’s also always the option to forgive debt or settle in other ways than first anticipated; a *trustline* payment from the creditor to the debtor would reflect this.

Revoking / Reducing Credit Lines

Users can reduce the set *credit lines* limits, e.g. if they want to reduce their risk exposure to a friend. If their friend has sufficient credit in other *trustlines* - indebtedness between the friends can immediately be reduced to the newly set *credit line* limit. Users can update the *credit line* limits at any time unless they agreed on more restrictive terms.

Leaving the Network

If users want to quit using the *Trustline Network,* they can issue a request to do so by appointing a limited number of friends with whom they want to have their consolidated debt or credit in the network. The result of this action is a log, which attests the balance with a friend after leaving the network. Settlement in cash, goods or services can be done with the friends in the real world.

Interest

Each *trustline* can have optional interest rates for both credit lines. If there is a non zero balance in a *trustline*, every transfer and balance update will reflect for the applied interest since the last update. Users will earn interest if their overall balance is positive and have to pay otherwise. Users can update the interest rates of the given *trustlines* at any time unless they agreed on more restrictive terms.

Fees

Users are required to pay fees for initiated transfers, which are composed of a base fee and optionally a percentage of the transferred value. In order to send transfers to the network, users also need ETH to pay for parts the gas fee. Acquiring the necessary ETH is done in the background by the app and payed for using *Trustline Money*. Note: Users don't need to know about Ethereum or acquire tokens at an exchange.

Capacity Rewards

Users get rewards for transfers that are facilitated using their *trustlines*. The more transfer capacity they provide through their *trustlines*, the more value can be transferred through them and the more rewards they may collect.

ETH Markets

There is constant demand for ETH in the system as it is needed to pay for the operation of the system as well as to facilitate cross-currency transfers or bridge gaps between unconnected partitions in a *currency network*.

Any user can be an ETH Gateway by selling ETH for *trustline* money in the system. This works by sending an offer to an on-chain ETH/Trustline Money exchange. ETH Gateways can earn money on the spread between their ETH price and the market price. They can also receive ETH when being used in a cross-currency transfer and credit money in the *trustline* network in exchange.

The option to sell or buy ETH is on an exchange labeled as an advanced feature in the mobile app and regular users don't have to bother with it.

FIAT-Crypto Exchange

The mobile app will automatically buy ETH necessary to pay for transaction fees. This is done whenever necessary and requires limited user interaction. Regular users can also buy ETH for purposes beyond transaction fee payments.

When entering the exchange function of the mobile application, a user is presented with the cheapest available exchange rate, representing the cheapest of the paths to an exchange offer. They can send limit or market orders to the exchange. Users are free to send acquired ETH from the *Trustlines* wallet to any other external wallets like Mist which allow them to use other DApps.

This is a very convenient way to buy ETH, as one does not have to setup an account with an exchange or meet in person with a seller. The option to buy ETH for *Trustline Money* makes it very easy to get access to the wider blockchain ecosystem, as it allows to interact with all kinds of services and tokens.

Technical Implementation

System Architecture

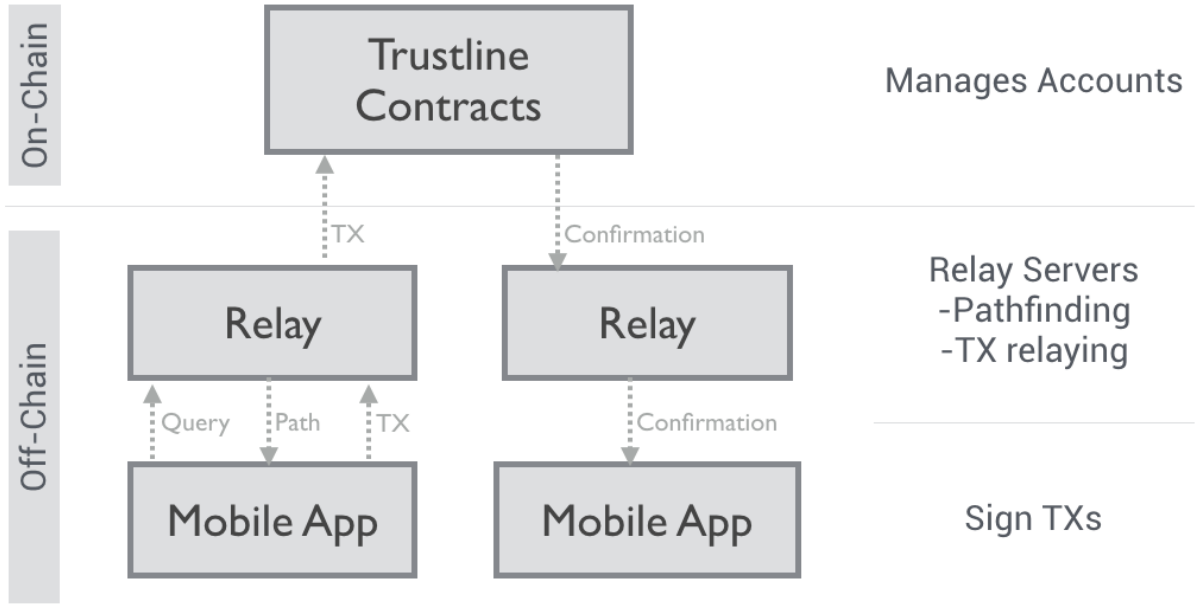
There are three main components that constitute the *Trustline Network* system.

1. The **mobile application** allows regular users to interact with the system using their smartphone.
2. **Relay servers** bridge between the mobile apps and the Ethereum

blockchain and offer services which are not feasible to be implemented on the platform or the mobile app. Most notably the pathfinding required for *multi-hop* transfers.

3. **The platform** is the foundation and implemented as a framework of smart contracts on the Ethereum blockchain.

- A transaction which encodes a call to the transfer function of a smart contract with the sender, value, path parameters is signed on the smartphone and sent to a relay server.
- The relay server forwards the transaction to the blockchain and listens for events from it, which are then forwarded to the smartphone



Basic Operation

The following assumes there is a network of *trustlines* which users set up using their smartphone. Then if a user wants to do a payment the following happen:

- A request to find a path on the network which has sufficient capacity is sent to a *relay server* by the user's smartphone application
- The relay server has a cached graph view of all *trustlines* based on their state as seen on the blockchain. It uses a path finding algorithm to find the path with the lowest fees and returns it to the smartphone
- A transaction which encodes a call to the transfer function of a smart contract with the sender, value, path parameters is signed on the smartphone and sent to a relay server.
- The relay server forwards the transaction to the blockchain and listens for events from it, which are then forwarded to the smartphone apps of the initiator and receiver of the transfer to inform its users of the status of the transaction.

Smartphone Application

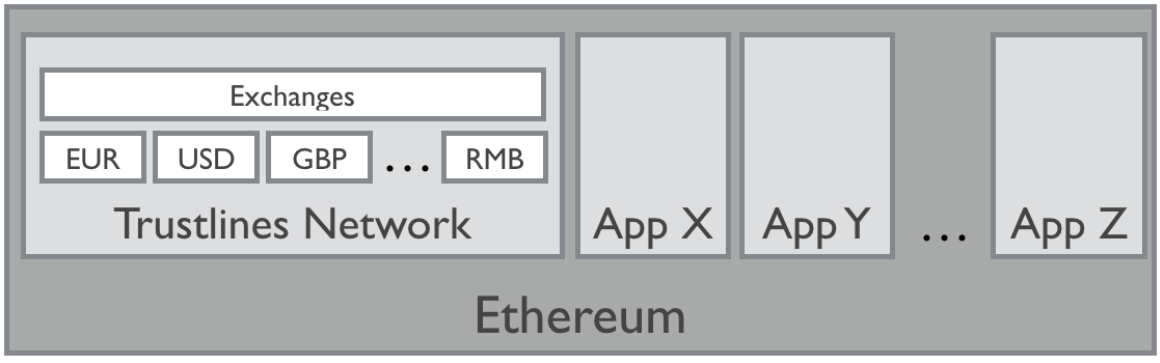
Users interact with the system using a cross-platform mobile application that can be installed from an app store or directly. For payments the user interface resembles interaction patterns known from traditional mobile payment apps. URLs, QR codes or NFC are used to encode payment requests and recipient addresses. The app differs by the ability to manage *trustlines*, i.e. add friends to a contact list and agree on the terms of the granted *credit lines* such as their limit and

optionally an interest rate for imbalances. For contacts in the phonebook that are also using the *Trustlines Network* the establishment of *trustlines* is automatically proposed. The application supports multiple user accounts and the participation in multiple *currency networks*. It also allows to initiate cross-currency payments and as a feature for advanced users, to trade ETH for *Trustline Money*. All transactions are signed locally on the mobile app, which also manages the private keys of the users. There is an option to appoint a set of friends that can cooperate to help registering a new controlling key in case it was lost.

Relay Servers

Relay Servers are a back end service which implement two main features:

1. Discovering a path between a payer and a payee in the graph of *trustlines*
2. Forwarding transactions to the Ethereum blockchain and sending smart contract events to the mobile app.



In theory the system could work without Relay Servers, but this would significantly decrease the user experience. The path finding algorithm depends on knowing all available *trustlines* and a reconstructed graph of the network in memory. This would require the mobile app to sync with the state of the *Trustline Network* whenever it is activated, potentially loading many millions of *trustlines* from the blockchain into the main memory. This introduces significant latency and would be a burden regarding bandwidth and RAM.

Users do not need to trust relay servers with their accounts, as transactions are signed locally on their device and validated by smart contracts on the blockchain. The worst case is, that relay servers fail to provide the cheapest path or give false information on the status of payments and accounts. This can be mitigated by setting up a network of independently operated and incentivised relay servers where the mobile app accesses a subset of them for every query and checks for consistency of their responses.

When relay servers forward a transaction to the blockchain they are payed by the app for their service using a probabilistic micropayments method. The relay server protocol is open and there is competition between relay server operators to provide the best service regarding quality and price. Clients do an internal accounting of the relay servers service quality and therefore develop preferences for the better services.

The Platform (Smart Contract Framework)

The platform is at the core of the *Trustlines Network*. It implements a secure, tamper proof, decentralized, permissionless, environment to host *currency networks*. It has an open protocol and API which supports interoperability with other smart contract based services on the Ethereum platform.

The platform can host multiple *currency networks* which can be denominated in real world fiat currencies (e.g. EUR), commodities (e.g. Gold) or user defined units (e.g. Beers). User defined *currency networks* can be customized in various ways, e.g. to restrict access, impose interest rate limits etc. All *currency networks* come with an on-chain exchange that allows to trade their respective *Trustline Money* to and from ETH. Therefore all currencies networks on the platform are connected via ETH, which allows for cross-currency payments.
The following explains how the backend of the *Trustlines Network* is implemented on the Ethereum platform as a set of smart contracts.

Smart Contracts

Currency Network Token

This contract is at the core of the system. One instance of it implements a currency by tracking user issued IOUs and the network thereof.

The API is compatible with the [ERC20 Token] standard.

The contract records the name and symbol of a currency, all its users and their *trustlines*. In its simplest version the *trustlines* are modeled as "Accounts" which track the *credit lines* two parties agreed on and their current balance:

In reality the data model is more complicated as also interest rates, last modification date and accumulated system fees need to be tracked. A design goal is, to fit all account data into 32 bytes which is the word size of the EVM, so that we only have the gas cost of one sload/sstore per access to a *trustline*. This is important as transfers will usually use multiple *trustlines* to facilitate a transfer. Going beyond the 32 bytes would therefore basically result in doubling the gas cost of a transfer.

```

struct Account {
    uint80 creditlineAB; // creditline given by A to B
    uint80 creditlineBA; // creditline given by B to A
    int80 balanceAB; // balance, value B owes to A (if positive)
}

mapping (bytes32 => Account) accounts;

```

```
transfer(value=5, path=[Bob, Charlie])
```

When transferring 5 tokens from Alice to Charlie via Bob, the contract's "transfer" method needs to be called by Alice with the value and the path, where the recipient is the last address in the provided path. As a result the accounts of Alice/Bob and Bob/Charlie are updated according to the call parameters and a Log is generated

which records the payment event on the blockchain. The requirement to provide a path, when doing *multi-hop* transfers breaks the compatibility with regular ERC20 tokens. This can be worked around by a method to temporarily register the path for a payer-payee combination with the contract before calling the transfer method.

Currency Network Token Factory

This contract can be used to create and register a new *currency network* on the platform. It generates a new *currency network* instance with properties according to a supplied configuration.

Identity Proxy

In order to support recovery of lost accounts, we use a proxy contract which is called with transactions signed by the users and forwards the actual transaction payload to its target, which authenticates the request based on the address of the calling proxy contract. This indirection is used to allow users to change the private key they are using to control the system. This is helpful in case a key is compromised. We are working on implementing a multisig scheme with uPort to allow to appoint a set of friends or other trusted entities, that can cooperate to update the controlling key. The contract can also be used to implement spending limits and give other contracts access to the account of the user.

Probabilistic Relay Payment

Relay servers are paid for their service with ETH. It would be expensive (10k additional gas per tx) to have ETH microtransactions on the blockchain. Therefore they are paid using a probabilistic payments scheme. Basically every payment is set up in a way, that it can only be redeemed with a very low probability P on the blockchain, which would then result in a redeemable amount which is 1/P of the actual fee per request. This is similar to a lottery, where one has only a tiny chance to win but can win a multiple of the price of the ticket.

Users need to put a certain ETH amount into escrow with this contract. When paying a Relay Server (which they do for every tx they send), they sign a message with the recipient address, the eligible amount and a threshold value which encodes the probability that the relay can redeem the specified amount in N blocks from now if the hash of a block in the future is lower than the threshold.

ETH Exchange

This contract allows to exchange ETH for credit in the *Trustline Network*. Users who want to sell ETH can register an offer with the contract which indicates the number of *trustline* currency tokens they want to receive in return for the ETH deposited with the offer. There is one such contract instance for every *currency network*. Users who have a path with sufficient capacity to the seller can pick the offer, pay with *Trustline Money* and receive ETH in their account. The same contract can be used to offer to credit *Trustline Money* for ETH.

Fees

There are multiple fees necessary to setup proper functioning of the system. For users only a single fee is though quoted before confirming a transaction. Fees are payable with *Trustline Money*, except for the gas and relay fees which are payable in ETH. Note that it was chosen to use *Trustline Money* for fees whenever possible to avoid unnecessary dependence on ETH supply in the system.

Relay Fee

This fee goes to the relay servers to reward them for their service. The fee is payable in ETH and is a constant amount for every transaction.

Ethereum Transaction Fee

This fee is payable in ETH and necessary to pay for the gas used when a transaction is processed by the system. This fee is the predominant one, for transfers less than $10 it accounts for more than half of the fees of a transfer.

Imbalance Fee

This fee aims to incentivise debt cancellation in the network. It works by making paths which include hops where the account imbalance of a *trustline* is increased, more expensive than paths where it is decreased (or increased less). The beneficiary of the fee is the sending side of a *trustline* because he might have to pay a higher interest due to the added imbalance. The fee is proportional to the imbalance added to a *trustline* and its interest rate.

Capacity Fee

This fee aims to reward providing transfer capacity by maintaining *trustlines*. The beneficiary is every sending side of a *trustline* along a path. The fee is proportional to the transferred amount.

Interest

Users can decide to set an interest rate on tapped *credit lines*, which is applied if there is an imbalance (i.e. the other part owes them money) in a *trustline*. The occurred interest is calculated and applied when the *trustline* is updated, but at most once a day.

Automatic Acquisition of ETH

As the system is based on the Ethereum blockchain it is necessary to pay for transactions in its native currency ETH. Relay servers also need to be paid in ETH. The *trustline* system aims to hide this complexity from users. The mobile application will automatically buy and maintain a certain amount of ETH tokens for the user. Therefore, when necessary it trades ETH for *Trustline Money* using the *ETH Exchange* contract.

Cross-Currency Transfers and bridging Gaps in the Graph

ETH can also be used as a bridge if there is a gap in the graph, i.e. when no path with sufficient transfer capacity can be found between two users. It works by finding a path from the sender to a seller of ETH (which has an offer at the exchange) and finding a path from someone trading ETH for *Trustline Money* to the recipient. Such a transfer is split into two transfers and two ETH exchange interactions but can be executed atomically in one transaction. These kind of transfers will be more expensive though, as the gas fee is higher and also because there will be a spread between bids/asks on the ETH exchanges.

Smart Contract Interoperability

The core value of Ethereum is synergy.

The TLToken contract is compatible with the ERC20 token standard. There are some things to keep in mind though, as for example `balanceOf` represents the spendable amount of a user, while the actual total balance of a user can be negative. Also the methods `transfer` and `transferFrom` only work if a path between sender and recipient was

temporarily registered with the contract. The methods to delegate and check transfer rights `approve` and `allowance` work as expected.

Further extensibility is provided by "Governance Hooks", which allow to delegate the exclusive right to update *credit line* and interest rates to smart contracts. As the parties in a *trustline* can also be smart contracts, this allows to model more sophisticated logic and enables applications like *trustline* based loans.

Friends Discovery

An important feature of all social network based applications is a mechanism which makes it easy to discover and connect existing friends in the application. Unfortunately most of the current systems work by uploading (leaking) the address book to a central server where it is matched against all other known users. This practice is unacceptable and was ruled out for the *Trustlines Network*. Therefore a scheme where no contact lists are leaked and being befriended with a certain person can be plausible denied was developed. The system is implemented using IPFS, public key encryption and memory hard hashing phone number combinations with a high probability of collisions. The details are beyond the scope of this paper and will be published in a different document.

From a user perspective it works quite similar to existing methods. Users are notified if one of their contacts is "most likely" also using the system and then given the option to send a *trustline* initiation message to his friend. False positive notifications are possible but infrequent.

Scaling the System

Supporting 100 Million active users on the system would exceed the transaction throughput of the current Ethereum platform by a factor of ~100. *Currency networks* and clustering users by connectivity are obvious partitioning options which may allow for efficient sharding which aligns with the Ethereum 2.0 design.

We are working closely with the Raiden Network team on their efforts to build a generalized off-chain state framework, which will be the foundation for scaling the system by moving most of the state transitions off-chain.

Applications Beyond Payments

Loans

Credit lines are at the core of the system and effectively allow users to lend money. A loan is different as it also comes with a promissory note which describes the terms of the agreement, i.e. usually there are guarantees on the interest rate and and a repayment schedule. Loans can be implemented on *trustlines* by delegating the right to update interest rate and *credit line* limits to a smart contract, which can encode the terms of the agreement.

Savings

Saving works similar to saving with a bank today. If users have spare cash, they can trade it with any other participant for *Trustline Money*, which (depending on the *currency network* configuration) yields an interest.

Custom Complementary Currencies

Complementary Currencies are monetary systems that operate in complement to a national currency. A popular variant are LETSystems which are used in more than 1500 communities worldwide. Another is time-based currencies where the unit of account/value is the person-hour or some other time unit. *Trustlines* has multiple benefits over existing complementary currency systems as it is permissionless, decentralized and tamper proof. Most importantly it solves the problematic issue of money issuance. Many complementary currencies are limited to be local in their scope and not interoperable with those of other communities as they could not trust each other's issuance mechanic.

The *Trustlines* Network allows communities to set up customized complementary currencies according to their preference. For example communities can in compliance with Islamic banking decide to not enable the feature of setting interests. Communities can also customize inbuilt mechanisms for how to deal with inflation as well as decide whether the currency should be interoperable with other currency-networks.

Conclusion

Cryptocurrencies are an important building block of public blockchain systems, as they provide the foundation for economic consensus protocols. But their commodity money nature and the efforts to acquire them, render it unlikely for them to become widely adopted everyday payment solutions. If one can admit to a view that money is a tool to account for who owes what to whom with the additional properties of fungibility and convertibility, we've demonstrated how money based on *currency networks* can be built on the Ethereum blockchain and how they can provide the current bank's main services. We've also explained how such a system can effectively solve the adoption problem of blockchain based payments. Especially as it has onboarding properties superior to traditional payment systems.