A Protocol for Microtransactions

Using Bitcoin With a Medieval Money Contract

Jarl Fransson, Strawpay AB

2015-08-28 - Draft - 6562c51

Abstract

This document presents the *Stroem* payment system which offers instant and efficient payments through an open protocol. Emphasis has been put on consumer simplicity and cost efficiency to make microtransactions possible. Designed as a decentralised network layer on top of bitcoin, Stroem can benefit from bitcoin global settlement efficiencies and constructs like payment channels.

The Stroem layer adds business features to support consumer-merchant interactions for electronic commerce. A payment is executed by transferring a digital form of a *promissory note*, a contractual promise to pay the owner of the note. It is a simple method to represent small *off-chain* transactions. Promissory notes as negotiable instruments have a surprisingly long and interesting history in commerce and finance. In our design they forge a single unit that captures three parts of a payment: its value, what the payment is for, and the party that is obliged to pay. By using basic cryptographic primitives for the notes we can create business functions such as offers, proof of order, purchase receipts, etc.

Efficiency comes from transaction aggregation and a risk model with irreversible transactions: a consumer first acquires a promissory note from an issuer, then transfers it to a merchant. The consumer has no payment liability once the merchant accepted the note as payment. The merchant trusts the issuer rather than each consumer, but in practice the merchant only trusts the issuer indirectly. Instead the merchant relies on its selected redeemer, a party that takes the role of providing advance payments, discounting, and redeeming notes for different issuers.

Contents

1 Introduction

In modern economic transactions where goods or services are transferred from a seller to a buyer, the payment commonly represents half of the transaction, the part used to provide a compensation to the seller.

The cost of doing transactions affects the structure and organisation of the economy. If the cost for doing a particular type of transactions in the market is higher than doing them within an organisation they will not occur in the market. Instead, these transactions will occur within companies or structures that will form and grow from being more efficient than the market. This now established thinking begun with the influential *Nature of the Firm* article by Ronald Coase in 1937 [5, 21, 22].

In *transaction cost economics* various origins of transaction costs are considered such as cost of finding the right product, price, and legal terms before making a trade. When parties engage in large or complex transactions the cost of payment is usually too small to be significant in this context. However, for microtransactions, transactions where the value is small, the cost of performing the payment can represent a major part of the transaction cost. A second, often ignored, cost for small transactions is the mental cost. This is the cognitive effort required by the buyer, and seller, to decide if an offered transaction is beneficial and the effort to perform the steps needed to make the payment [15].

To conclude, if transaction costs are pushed low enough, small internet transactions can become useful and profitable in the open market. New business models will emerge and existing business models will need to change.

1.1 Microtransactions

Although promised since the early days of the web, internet micropayments failed to materialise despite many attempts. Instead, for small transactions, internet users pay with their time and personal information by viewing ads and being tracked online.

1.1.1 Why Early Internet Payments Failed?

It is difficult to explain why information technology has advanced so far in many areas but it is still hard to replicate digitally, the physical act of handing over a few coins for a candy bar. However, there are a few plausible arguments for the current situation:

Cost structure. The existing actors in the financial sector that are providing payments services, have built their products on older systems. Electronic payments evolved out of credit card payments or cheques which were cashed in banks. Credit cards originate from the need to issue credit for retail transactions and even if the card systems have evolved over the 50 years they have existed, the evolution has mostly happened by adding new layers of extra security. The cost structure for these systems, built for typically larger retail transactions, are not suitable to small transactions.

Cognitive load. In the current systems, the extensive user credentials needed to complete retail purchases with debit or credit transactions represent an effort for the user that is larger than the resulting benefit of a small transaction. Also, many of the previous efforts to support micropayments did not successfully address the mental cost for the user taking a decision and action for every transaction.

Market size and maturity. It can be argued that the market value of content and services consumed on the internet in the nineties, when the first micropayment attempts took place, were magnitudes smaller than today [11]. At the same time computation and network resources were expensive.

Proprietary Solutions. Previous attempts proposed proprietary solutions tied to a single company or entity. This required full trust in the success of one company, as failure would make issued tokens worthless.

1.2 What Has Changed?

The question of what is needed to make electronic microtransactions viable is still interesting. The appearance of bitcoin and the blockchain, its underlying technology, has started a new wave of innovation. Even if bitcoin, compared with the well established payment systems, looks like a bold experiment, it represents a global digital currency of sorts. Bitcoin was made for the internet and it is not tied to any country, national currency, or a single commercial entity [10]. Bitcoin and the ecosystem that is emerging around it represents a new take on the concept of digital currency. The currency is independent from the companies and projects that transact in the currency. With bitcoin as a medium of exchange new companies can form and prosper or fail in the competition, but parties holding the currency will not be left with a worthless token.

Money is the ultimate social invention. The existence of a widely recognised digital currency, even with the current volatility seen in bitcoin, represents an important change which will promote new methods of trade in the digital realm.

1.3 What Needs Microtransactions?

Eventually all businesses that deal in information products will turn into a digital form, in part or completely. For all the world's media companies this implies that the business model for digital offerings must support itself without subsidies from physical or printed products. Microtransactions offers a new or complementing way to generate revenue from premium content.

At the same time we see a consumer interest in payment models where users do not need to pass paywalls or manage countless subscriptions. Although there would always some consumers that prefer the "free" ad sponsored version, we believe many would ultimately opt out of ads and big data stalking if they could. Microtransactions represents a simple non-sticky route to premium content for consumers.

2 Internet - the Failed Promise

> *"Free" Internet services: Just because you're not paying money for a service doesn't mean it's free. You always have to give up something and if not money it's your personal data.*
>
> — Mikko Hyppönen, *F-secure*

Internet represented a fantastic revolution for human freedom and development. But, there are serious problems for users of the internet of today. There are three major problems that are connected.

Firstly, many existing news and media organisations are not generating enough income from their published online products to sustain their business. This can lead to a decreasing amount and diversity of the critical investigation and analysis work that plays a vital part in our open democratic societies.

Secondly, lack of revenue alternatives, have led internet companies and media organisations to resort to aggressive ad publishing based on information stalking using various intermediaries that ignore common privacy rules. Many of the internet giant companies of today are basing their entire business on the value of collecting an extreme amount of information about their users and trading it in for money counted in tens of billions USD yearly.

Thirdly, when internet media or news corporations earn revenue exclusively from ads, consumers become products. This could question the integrity and accuracy of the message. It should be a concern that loyalty likely would tilt towards the revenue providers, those companies that pay for the ads.

Users practically have no choice but to agree, explicitly or implicitly, to terms where incredibly much data about their behaviour is collected, processed, and sold, for the the use of marketing. Also, this information easily turns into a liability: both powerful states and cyber criminals make great efforts to get access to all the collected information.

2.1 Subscriptions and Current Payment Options

Until now, subscriptions are the only major monetisation alternative to ads. For most publishers and internet companies this has not been very successful and we can understand this from three major reasons:

- When consumers that are prepared to pay for internet consumption click on something, they have the *intent* to consume a single item rather than enter a subscription. This is the explanation you get when you ask why they would not sign up. The intent is quite different from when signing up for a print magazine. The timing is wrong.

- Subscriptions for internet-only content often turn out expensive for consumers relative to their use. Subscribers of a printed magazine will notice it in their mailbox, but a digital subscription easily gets forgotten while still costing money.

- Subscriptions also represent a mental cost for consumers. It is an ongoing engagement that consumers should keep track of, maybe modify, quit, or replace. Also, paying users now have to put up with more work - log in with their account every time. In general, consumers do not want to have many subscriptions. This goes against the open internet market, offering hundreds of information sources which are linked into the Web. Subscriptions create silos which breaks the concept of links, the very fundamental component of the Web.

Internet media companies may think subscriptions are wonderful and sticky once consumers sign up. However, unless the internet deteriorates to a few companies, we believe this is not attractive to a most consumers in comparison of the alternatives: just stay on the “free” internet, maybe with ad-blockers, or just opting out.

2.2 The "Free" Paradox

The "free" web does not lead to freedom. The wonderful technology that made the cost of communicating information and ideas almost zero has now led to new asymmetries between corporations and consumers:

- Corporations employ ever more efficient technologies, like big data and web tracking technologies, to gain an information advantage over consumers in the market.
- It is almost free for a sender to send information to millions of users, but the aggregated cost of receivers to receive, read, or filter information, is substantial. This is illustrated with the problem of email spam.

Up to now, the idea that media efficiency gains of lower publishing costs by switching to digital form would leave more funds to be spent on the content, has been wrong. Consumers early came to expect internet content to be free, online payments turned out to be extremely hard and cumbersome, and so evolution took this bad path where both merchants and consumers have too few options.

3 Low-Friction Microtransactions

3.1 The True Value Proposition of Bitcoin

Bitcoin and blockchain technology offers a new ingenious way of transferring value between users anywhere on the internet, without a central trusted third party. Bitcoin has been called the internet of money but it is really most impressive as an efficient gross settlement system not restricted by national borders. However, bitcoin does not directly support the massive transaction volumes that would result from turning a chunk of the internet users' daily interactions into financial microtransactions. Also, bitcoin transactions takes time to complete, from around 10 minutes and up depending on the transaction risk profile.

The true value proposition of bitcoin and blockchain technology includes the assumption that it can be connected to the wider economy without encoding every transaction separately in the blockchain. There are clever ways to form a middleware layer containing mechanisms to perform *off-chain transactions* that still benefit from the efficient settlement and the trust model of the blockchain. This is the basis for the scalability and aggregation that are crucial to support microtransactions.

3.2 Payment Systems

There are many different types of payment systems, which are in use today or were used back in history. Today, specie and banknote circulation, i.e. cash based systems, are increasingly replaced with electronic payment systems. These systems can be regarded as an evolution from earlier paper based systems and broadly categorised into two models:

- In the *banking model*, cheques are written and sent from the payer to the payee. The payee must contact his bank to complete the payment. Cheques are cleared via a central entity in a complex process that eventually leads to transfers between the involved banks.

- In the *giro model*, giro transfers are sent by the payer to a giro centre. The transfer happens between accounts at the giro centre. Receipts or updated statements are sent to the payee and payer. The recipient does not have to accept or take any action for the payment to complete.

The different established dominant models in use in different regions of the world reflects legal and historical differences [18]. The giro transfer model has a long history in Europe. Card payment systems originated in USA and they are similar to the cheque based banking model which is the dominant model in North America. In a card transaction the card holder authorises a payment instruction that is later presented to the issuer bank or the card company which completes the payment. To conclude, different payment systems have evolved over a long time and when they transformed into an electronic form they kept many of their original properties.

3.3 The Stroem Payment Network

Transaction cost focus. In this document we present a payment system with the overarching goal to make microtransactions a reality. To succeed, the transaction cost economics needs to improve beyond the current state of the art. In particular the system needs to achieve a low friction experience for consumers to make it worthwhile, and a low overhead cost for merchants to support a profitable business even when each transaction value is just small change for consumers. From the user perspective, transactions must be instant, the failure rate must be low, and if things go wrong there must be a clear way to resolve problems.

Microtransactions are general. Microtransactions are not limited to donations or tips, which are transactions of a special kind that is of unidirectional nature. An archetypal Stroem transaction facilitates the value exchange between two parties in a trade: a merchant provides some service to a consumer and receives a payment as compensation. To support this consumer-merchant interaction, in addition to transfer of the payment value, the system provides business functions such as offer, order, and receipt. This provides the base infrastructure for applications to build specific features like content licensing.

Payment channels avoid deposits. For a consumer to experience that a payment is a small low-risk payment, we do not want to require a larger amount to be prepaid or committed to be spent on a particular service in advance. Classically trusted intermediaries are used when a consumer prefer not to entrust some counterpart with an advance amount. This trust represents a cost and a complication, especially when there are no trustworthy parties around, like in a new industry or in a non-functional economy.

However, bitcoin offers the possibility of using smart contracts that let users make arbitrary small payments to a predefined party. This introduces the concept of intermediaries with limited trust. By letting the consumer acquire a *promissory note* by using a payment channel contract with an issuer, we can extend the reach so the consumer can make a payment to any party, just by transferring the note, which is an instant local off-chain operation. This works without any prearrangement between the consumer and a receiving merchant as long as the merchant party accepts the note as payment.

The promissory note. Stroem uses a digital form of a promissory note to represent the payment value in transit. A promissory note is a written promise to pay the owner of the note. Using a few cryptographic primitives, we can construct a simple time limited digital promissory note, with some additional terms that captures all essentials properties of an unfolding payment.

Value at risk. When a trade occurs between two parties, and where the seller and buyer are not in the same place, for some period of time the value of the trade is at risk for at least one of the parties. If trade terms specify *pay before delivery*, the buyer is at risk, and if the trade terms specify *pay after delivery*, the seller is at risk.

This observation implies it is better doing many small transactions than one large. But more importantly, it points to a rational trade-off between the utility of executing a successful transaction and the risk of loss if the transaction fails. If use of an intermediary improves the speed and convenience, or lowers fees, this constitutes an net utility increase that should be weighed against any increased risk of a failed transaction. For large transactions it is worth spending more effort and money to lower the risk of failure, but for very small transactions some risk may be acceptable for the benefit of more convenience and less overhead costs. The argument can be summed up: for small transactions, losses are limited by proceeding with a transaction only if the previous transaction succeeded.

Pay to play. In the base case, where an internet user access some service, the user pays before delivery, or more exactly the service provider is guaranteed that the payment is secure before the delivery happens. This mode improves convenience as there is no need to perform the identification process of the buyer that would be required in the case where the merchant offers the consumer to pay after delivery. In fact, a merchant can perform payment validation without knowing anything about the consumer, which opens up for merchants to provide attractive login-free or register-free premium services.

Extended function. Stroem extends the basic features of a promissory note with a few functions needed for internet transactions. The owner of the note is given by an indorsement from the previous owner, but here in a digital form. Digital signatures are validated against public keys. Public keys can be pseudonymous and still be used to prove ownership. This means merchants do not necessarily need to collect identity information from consumers to get payed. Also, the constructed notes includes authenticated meta-data that effectively fuses a payment and a purchase order into an atomic transaction, which minimise disputes over what was purchased. Finally, promissory notes can be transferred in aggregated form, used as transaction proofs, sold, and finally redeemed at the issuer.

Open. The system is open in the sense that the protocol is open source, and any party can issue promissory notes for consumers to facilitate payments. This makes it possible for issuers to compete on fees and services. It is natural to think that some issuers will provide currency exchange from national currencies to bitcoin, i.e. issue and sell bitcoin denominated notes in return for local currency and thus provide an on-ramp to a global internet market for consumers that only transact in their local currency. Similarly,

issuers or special redeemers could offer to redeem merchants' notes and exchange into local currency.

History of the note. The use of promissory notes to facilitate trade is very old. It predates modern banking and arguably the promissory note is the prototype of paper money. In medieval times it was used by merchants to avoid the cost and problems associated with using commodity money over long distance trade routes. Money was heavy, could be stolen, taxed and different specie circulated in different cities. It is well documented that promissory notes and bills of exchange [1] were used extensively to improve money transfer for trade networks. The notes would sometimes be sold at discount to special traders, i.e. these notes were prototype negotiable instruments and provided a valuable liquidity mechanism [9]. Another almost identical circulating pattern of notes existed in the Scottish *free banking* period [14], where private banks issued notes that were regularly settled between the banks. In Stroem, issued promissory notes circulate from issuer, to consumer, merchant, and via specialised redeemers back to the issuer. An important distinction from historical use is the time limit of notes in Stroem that makes notes transient.

Liquidity enhancing. In the context of consumer payment systems the bitcoin network is relatively slow and even if payment channels allows instant transactions between two parties, the bitcoin funds are pinned to the two parities until the channel is closed or settled. Adding a decentralised layer of circulating digital promissory notes for micropayments will improve liquidity in a similar way as in medieval trade networks, the main difference being that the notes in Stroem are transferred in milliseconds and expires in days whereas in medieval times notes circulated over months or years.

3.4 Potential to Scale

The scalability benchmark should not be the current transacted volume by VISA, banks, or other payment systems. With lower transaction costs and support for automated transactions in consumer wallets, we should expect an increasing number of transactions from both old and new markets.

A general open protocol for moving monetary value at the micro level has wide applicability. In addition to premium web content and services, we

[1] Bills of exchange are negotiable instruments similar to promissory notes. The difference is that instead of a promise, its an order written by the *drawer*, instructing a *drawee* to pay.

can only imagine all future applications that would be offered. Here are just some examples:

- Attach value to email messages to suppress spam.
- Support of non-profit services like Wikipedia. Pay for the energy consumption of your searches, or sponsor individual writers.
- Pay for computation by the minute. You could pay to solve a large system of equations on a server farm when you suddenly realise you do not want to wait for your laptop to finish.
- Unbundle mobile subscriptions: for each phone call buy the service by the minute from the carrier with the best real-time price.
- Use your phone to rent a bike a few hours in a Berlin, after you just arrived. You see the price and pay in USD, it settles in bitcoin, and the bike company gets euros.
- Pay for that single file conversion you need instead of buying a fancy conversion tool.

We need a system that will scale to many billions of transactions per day. The Stroem protocol was designed with the scaling requirement from start. To achieve a scalable system we provide for transaction aggregation at several layers. The open decentralised nature of the network will create a market of competition and innovation.

3.5 Transaction Flow

To facilitate payments from consumers to merchants for goods and services, intermediaries act as payment hubs that provide the services of issuing and redeeming promissory notes.

The intermediaries specialise in different roles: some issue promissory notes for consumers to make payments, some redeem notes when merchants demand payment, and some trade notes between themselves. When a party redeems other issuers' promissory notes from merchants, it buys promissory notes at a discount representing a compensation for estimated risk and processing cost.

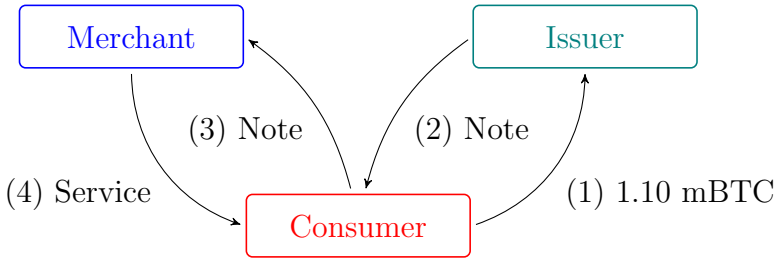

Figure 1: Payment Transaction Flow

3.5.1 Consumer Payment

Figure 1 shows the roles and the interactions taking place when a payment is made. Steps 1 – 4 illustrate a transaction from a user acceptance of an offer to the delivery of a service.

1. When a *consumer* wishes to make a payment to a *merchant* for a service, this is done with the help of an *issuer* of promissory notes. The issuer is selected previously and is someone that must be accepted by the merchant. We say that, the consumer buys a promissory note from the issuer, with the amount of the payment, in this case 1.10 mBTC.

   The system is designed with the assumption that the consumer pays the issuer using a bitcoin payment channel. However, how the consumer pays the issuer is up to the issuer and the consumer and many different methods are possible.

2. The issuer returns a newly issued promissory note to the consumer with the specified attributes, amount etc, according to the consumer's request.

3. The consumer transfers the promissory note to the merchant with added payment information specifying what is purchased.

4. The merchant validates and accepts the payment, and delivers the service to the consumer.

Steps 1 – 4 occur in sequence with no human interaction and will complete in a short period of time. Typically the payment is initiated when a user clicks to accept to pay, and as an immediate consequence the service appears.

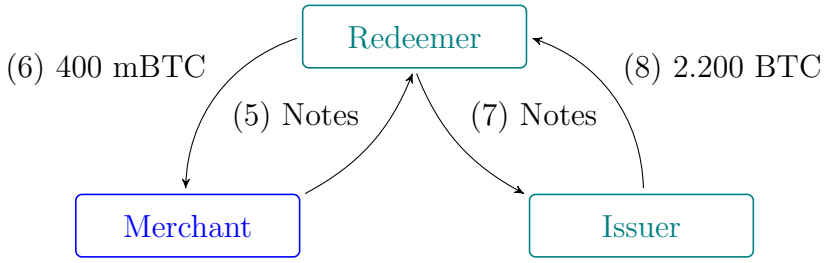

Figure 2: Redeem Transaction Flow

3.5.2 Merchant Redemption

A condition for handling microtransactions efficiently is to achieve high degree of transaction aggregation. This is a design principle employed at different levels to reduce load and improve real-time properties of the system. One example is that merchants can redeem a large number of promissory notes at their *redeemer* in one step, see figure 2.

5. When the merchant wishes to redeem one or more received promissory notes, these are transferred in a single block to the redeemer selected by the merchant. To provide consumer privacy of the goods purchased, the merchant redacts any payment information previously supplied by consumers. The merchant also attaches details about how the redemption payment should be done.

   The redeemer can be the same party as the issuer but in general the redeemer accepts and redeems promissory notes issued by several issuers.

6. The redeemer validates the block and pays the sum of all the notes redeemed (400 mBTC in figure 2) to the merchant according to terms agreed with the merchant, adjusting for discounting note values for various issuers. The terms may stipulate that the payment will be made once a certain amount to be paid has been accumulated at the redeemer.

7. The redeemer will handle the process of transferring batches of promissory notes to the proper issuer.

8. The issuer finally pays the due amount (2.200 BTC in figure 2) according to discretionary terms between the parties.

3.6 Promissory Note Construction

The construction consists of two parts, a base contract and a negotiation list. The base contract consists of a set of attributes shown in Table 1.

Table 1: Base Contract

Attribute	Description
`Legal text`	Specifying the promise of the issuer.
`Amount`	The amount to be paid.
`Currency`	The currency of the amount.
`Issuer name`	The name of the issuer.
`Issuer public key`	The public key of the issuer that can be used to verify an issuer signature.
`Issued date`	The date that the promissory note was issued.
`Validity time`	The redeemable validity time of the promissory note, starting from the issued date.
`Verifiers`	An ordered list of future bearers, that will be required by the issuer before redemption occurs. Each verifier in the list can be assured that it will be the bearer of the note at some point during its life. See section A.1

The second part is a list of negotiation records, where each record in the list represents a transfer of ownership to a new bearer. The result is similar to a bearer instrument as the rights holder is not identified by name or identity. Instead we use a pseudonymous public key and anyone that can create a authentic signature, when verified with the public key, is considered the bearer of the instrument. Each record in the list contains the attributes shown in Table 2. Arbitrary information can be associated with each transfer of ownership. This information can be attached with the promissory note using a record shown in Table 3, and it is authenticated via its hash value and the signature in the negotiation record for the transfer.

3.6.1 Issuance

When the promissory note is issued, the attributes of the base contract and the attributes of the initial negotiation record are given values. The issuer then creates a signature that covers all attributes in the base contract and

Table 2: Negotiation Record

Attribute	Description
`To the order of`	Specifying the public key of the bearer.
`Payment info hash`	A cryptographic hash value of the Payment info attribute.
`Signature`	A digital signature created by the previous bearer of the promissory note.

Table 3: Payment Info Record

Attribute	Description
`Payment info`	Redactable arbitrary information. The use and meaning of this attribute is left to the discretion of the parties involved in the negotiation. This attribute contains some data in the form of a byte string with an added nonce.

negotiation record. The resulting signature is included in the initial negotiation record. At this point the list of negotiations consists of one record value. The issuer would probably ignore the payment info record. See more in section 3.6.2 about attaching information. The bearer of the newly issued promissory note is defined in the `To the order of` attribute.

Even if the identity of the bearer is not part of the promissory note, the identity can optionally be demonstrated with a certificate that associates a public key with an identity. The details are outside of this construction but a certificate is assumed to be a document from some authoritative source, certifying that a particular key is controlled, exclusively, by some party with a given identity.

3.6.2 Negotiation

The bearer has the power to transfer the ownership to a new bearer by filling in and adding a negotiation record to the list of negotiations of the promissory note. To complete the new negotiation record the bearer creates a signature that covers the promissory note, comprising the base contract, all previous negotiation records up to the newly added negotiation record attributes with the new bearer's public key and the hash of any attached information in a payment info record. The point of inclusion of this hash in the signature is to

authenticate any attached information in the `Payment Info` attribute. The `Payment info` is not included in the negotiation record but can be attached when the promissory note is sent. The reason to use the hash instead of the full payment information is to allow the `Payment info` to be redacted at a later stage. Typically only the last negotiation's `Payment info` is attached when a promissory note is sent.

3.6.3 Redemption

Redemption of the promissory note is when its bearer transfers it back to the issuer and demands the issuer to fulfil the promise to pay the amount of the note. This step is performed by the bearer by negotiating the promissory note to the issuer and, in the same step, supplying redeem instructions with the `Payment info` attribute.

A promissory note's signatures can be validated by verifying each signature in a negotiation record according to the public key present in the previous record of the negotiation list, and finally, using the issuer public key in the base contract to verify the signature of the first record of the list. This validation is ultimately done at the issuer when redemption happens, but each new bearer will also validate all existing signatures in order to know if the promissory note can be accepted as payment or not.

3.7 Comparisons With Other Models

> *"Bitcoin isn't currently practical for very small micropayments. Not for things like pay per search or per page view without an aggregating mechanism..."*
>
> — Satoshi Nakamoto, *bitcointalk 2010*

When aiming for ultra-efficient payments, particularly for small amounts, there are a number of things to consider. Section 3.7.1 looks at general choices made in Stroem. For interested readers, sections 3.7.2 - 3.7.11 compare Stroem with a selection of other payment systems.

3.7.1 General Considerations

Aggregation. Aggregation is critical as it improves system efficiency. We argue that a "gross settlement" based system, where transactions are processed individually from payor to payee without any aggregation is not a viable design.

Decentralisation. With an open protocol, competition and actor diversity will drive innovation and efficiency to lower system costs. This is one argument for a decentralised model, even if in theory a centralised model, where a single monopoly bank would process all transactions, looks technically easier.

Bitcoin. Stroem is built on bitcoin, even if issuers and redeemers can accommodate other currencies. A system for micropayments needs a base money transfer system to settle the net flows between participants. There are many payment networks, as we have seen. We think bitcoin offers clear advantages over existing banking networks and other legacy settlement systems, especially in terms of speed, global coverage, and openness.

Consensus Process. Blockchain based systems, or more general, consensus based systems, where many actors need to synchronise state for each transaction, are expensive. Many technical observers who looked at bitcoin early decided it could never work as it could not possibly scale. It is really the past outstanding evolution in computing and networking that has made bitcoin work as well as it has. We do expect bitcoin to scale many orders of magnitude over time. However, when we envision billions of daily microtransactions, it is extremely wasteful to have every node in a global consensus network witness, agree and store, every transaction. Private off-chain transactions that are later aggregated onto a blockchain seems to be a better solution.

Regulation. In Stroem, merchants and consumers make use of issuers and redeemers. These are trusted entities and will need to follow the appropriate laws and regulations where they operate. In a large global market for micropayments, issuers and redeemers that exchange with bitcoin also need to operate with the existing currencies that most merchants and consumers currently expect. In that context, the requirement to operate as a known entity is not new.

Risk. Trade involves risk and risk represents a cost. In Stroem, an effort has been made to minimise the risks for the parties involved. The general assumption is that each transaction is small and the consumer risk is the value of one transaction. The merchant, which has counterparty risk with the issuer, can choose an acceptable level of risk by redeeming payments frequently or infrequently, and paying varying fees according to the agreement

with his redeemer. Redeeming each small payment instantly would cost more than daily settlement.

Fees. When payments, i.e. promissory notes, are bought by another party or redeemed at the issuer, this will happen at a discount to the par value. This represents fees for risk and the cost of processing. The rate of fees will be set on a market where issuers and redeemers compete for consumers and merchants respectively.

3.7.2 Card Payment Networks

The topology of credit and debit card networks looks similar to the Stroem network. Card networks have card owners that use cards, issuers of cards, merchants that accept card payments, acquirers that help process merchant payments. However, there are important differences.

In a card transaction, the owner of the card gives the merchant the right to present a payment instruction to the card issuer, a bank or card company, to pay for the purchase. The actual payment is delayed from the time when the consumer makes a purchase. At the time of the purchase the transaction does not immediately extinguish the debt to the merchant. The payment from the issuer bank could in principle fail, and if so, the consumer still has a liability to the merchant. There are a number of other conditions that affects if the payment is completed or not.

For microtransactions we want to minimise costly disputes. In a Stroem purchase transaction the payment is effectuated by transferring a promissory note to the merchant, which is a transfer *without recourse*. This effectively makes both the payment and the purchase non-reversible. Similarly, there is no recourse for the consumer against the issuer, if the purchased good or service is unsatisfactory in any way, as the transaction of acquiring the note from the issuer is unlinked to the purchase.

3.7.3 Electronic Cash Schemes

In 1982 David Chaum proposed a secure digital cash system [4]. This work originated the field of financial cryptography and it inspired a series of different electronic cash schemes and also commercial ventures [2]. Common for these systems is that an issuing institution issues tokens with a specific value. Chaum's system relied on blind signatures, which he also invented, to achieve the property of untraceability by the bank or the issuer.

Electronic cash schemes have seen a lot of research and many variations and improvements have appeared over the years. Still, commercial success

has failed to materialise both for microtransactions or more general payment use. It can be argued that all commercialisation of these schemes relied on the existing banking infrastructure, and that the untraceable cash-like properties were, and still are, incompatible with the existing legal framework regulating financial institutions.

The promissory notes used in the Stroem protocol have a similar role as tokens in electronic cash schemes, and both systems are local – they do not rely on global consensus. However, there are some important distinctions:

- Issuing of promissory notes in Stroem is currently not performed using blind signatures. However, this could be added if required, but to achieve a degree of untraceability the denomination value of notes, expiry, etc must be standardised to create a large enough anonymity set [13].

- In a typical electronic cash scheme, the solution of the double spend problem is to handle it after it occurred. It is detected by the issuer which then can reveal the identity of the double spender. This requires the full identity of any user in the system to work. We argue that detecting double spending after the fact is not very useful. For global internet interactions, this leads to complicated identification procedures of users before they can participate, as anyone can be liable of breaking the rules. In Stroem we rely on a mechanism that allows a receiver of payments to also act as a verifier: a merchant, or a party that redeems a merchant's payments, can then verify against double spend. The issued note stipulates the verifiers so payment validation can be done offline.

- Stroem uses bitcoin. While the base currency is not a property of electronic cash schemes, which could use any currency, in Stroem, promissory notes, deposits, and settlement use bitcoin. This makes it easy to connect a micro-service in one country with users in any other country.

- Electronic cash schemes are categorised into online and offline systems. In Stroem we assume that it is not acceptable to detect double spending after the fact, as purchase transactions must be irreversible and we do not want to require full identity of users. Thus the models are not exactly comparable. In a typical transaction, for a Stroem purchase, the consumer needs to talk to the issuer to acquire a promissory note. However, as mentioned merchant validation is possible without internet connection.

Also in Stroem, when a series of equal valued payments are expected, for special applications like tokens for rides, games, or paying for a video stream, promissory notes could be downloaded ahead of use into a wallet and used offline.

3.7.4 The Bitcoin Lightning Network

The Lightning Network is a highly interesting proposal, offering scalability for bitcoin and fast secure transactions, using bitcoin hash- and time-locked contracts. It combines multiple point-to-point payment channels into a network, which can use low trust intermediary nodes as these entities cannot steal the transacted funds. The payment channel operations happens off-chain, but if intermediary nodes are uncooperative, the state of all involved channels can be committed to the bitcoin blockchain and no funds are lost [8].

We believe that the Lightning Network potentially offers great features and its development should be pursued. The Stroem design is based on a different model and we think that Stroem and the Lightning Network can be complementary technologies. As the Stroem protocol is focused on the consumer-merchant interaction when a purchase exchange happens, the Stroem protocol could be layered on top of the lightning network, once it is operational. Actors using the Stroem protocol would use the Lightning Network to perform the settlement payments when redeeming notes. However, there are a few interesting points when comparing the systems:

- For commerce, the goal of transacting in a completely trustless way is somewhat misguided as when a consumer makes a purchase from a merchant there is always trust involved, as noted in 3.3. Trust in the world of trade can be gained or minimised by making small transactions repeatedly with the same parties, something which is made easier with micropayments. For large commercial transactions it is obvious buyers want recourse, by means of legal enforcement, disputing and reversing payment etc.

- The two main design goals for the Stroem protocol are, firstly, to support optimal consumer experience, and secondly, to offer high system efficiency, when performing small payments. By letting merchants asynchronously settle transactions after the consumer transaction is completed, we improve latency as seen from the consumer. It also allows for merchants and intermediary hubs to aggregate payments to greatly improve system efficiency. Consumers do not wait for the trans-

action routing through the network. Merchants do not need to send each micropayment separately to the issuer or other redeemer.

What can be said here is that the Lightning Network could allow a merchant to accept a completely unknown issuer, as long as the payment can be immediately secured, i.e. pulled through the lightning network. It would be a reasonable trade-off to use, a synchronised real-time, non-aggregating system [2] when an issuer is unknown or untrusted.

- Although the Lightning Network proposal represents an innovative design and has attractive benefits, there appears to be several complexities that need to be analysed or worked out:
  - The channels lock up bitcoins, and with multi-hop routes to each and every recipient, someone has to provide these. From a merchant recipient's point of view, locked up bitcoins would represent an illiquid currency risk.
  - Channels need to be managed, settled, restored or recharged with bitcoin.
  - Routing methods of payments are needed where, for each payment, an appropriate route can be determined. If nodes can join the network freely, would that open the network to attackers?
  - Also, a few improvements or additions to the bitcoin protocol is needed, notably the malleability problem for these type of transactions needs to be solved.

  Despite these points, we are optimistic that they can be solved. It seems possible that in the future, a majority of all bitcoin transactions could be routed through a network like the Lightning Network, however, much work remains until it is operational.

- If an overlay network with intermediary nodes route a large volume of transactions, it seems it will be difficult for these intermediaries to hide in the long run, unless they can exist as hidden Tor services, which may be is a possibility.

  In Stroem network it is assumed that high volume intermediaries, are indeed known and trusted by those that choose to use them, although the trust is explicit, well defined, and limited.

[2] It may seem that the Lightning Network can from this observation be likened to a "RTGS" - Real-time gross settlement system. However, funds are locked and the bitcoin funds transferred are not available until channels settle, or parties agree to settle using some other means.

3.7.5 Ripple

The Ripple payment protocol evolved from the Ripplepay concept [20], which is based on allowing, tracking, and changing debts between parties that trust each other. The debts can be denominated in any currency or asset. Inspired by bitcoin public blockchain ledger, the *Ripple protocol* implements the concept by using a public ledger, but with a different consensus process. Ripple protocol also employs its own native currency, which is not mined or distributed, but was created at the inception and owned by the founders and Ripple Labs, the company behind the protocol.

In the context of microtransactions, the Ripple protocol is faster than bitcoin due to the different consensus process. However, Ripple still has a cost and scalability problem as it requires global consensus for each transaction that is put into the global ledger. The consensus process is an iterative process that finds a super-majority among the nodes which agrees on a set of valid transactions. These sets are added to the ledger in discrete steps, but they happen faster than in bitcoin as there are no blocks to be mined.

The original Ripplepay concept is an interesting system that has similarities to an economy of circulating promissory notes, which existed in several places back in history as discussed before [9]. However, we believe there are several design choices that are questionable:

- Assets in Ripple held by a user are IOUs from parties trusted by the user, with the exception of the native currency XRP, which does not have a counterparty. These assets do not appear to be debt obligations as they only acknowledge that a debt exist, not when, or indeed if, the debt should be payed off.

- The Ripplepay concept does not really need a global public ledger as used in the implementation by Ripple payment protocol. The act of issue, transfer, or paying of debt, does not need to be validated by parties other than those involved. A global public ledger does not remove any counterparty risk, and debt money is unlimited an can be issued infinitely so there is no conserved quantity to protect like with bitcoin. It seems costly to add a global consensus process when it does not provide any significant improvement. This was also a question put forward by Peter Todd in a published review on Ripple [16].

- The Ripplepay *rippling* concept seems to have a fundamental financial flaw built-in. The idea is that an obligation asset from one party automatically is exchanged *on par* with an obligation asset from another party. If this function is employed it will likely lead to failure as risk

preferences are not expressed in trading prices. We expect that the rippling function will not be used, which means the system essentially becomes a copy of the banking system where only banks hold deposited assets.

- Ripple on a public ledger has a privacy problem. Once you create a global public ledger for transactions, the privacy of users' transactions becomes an issue. The Ripplepay concept relies on extending credit and accepting debts with parties in a spanning network of users. Even if participants can use pseudonymous identity handles, each party would need to use the same pseudonym for most transactions. Users cannot use new pseudonyms for every transaction as there would not be any routes for payments. We believe the Ripple payment protocol with its public ledger has a major privacy problem for its participants.

3.7.6 Altcoins

Altcoins are cryptocurrencies, derived from bitcoin, with varying degree of changes from the core bitcoin source and design. Several altcoins have been proposed to specifically target micropayment transactions. Notably, Dogecoin users were happy to promote tipping [17], and later Neucoin was launched with the description, "designed specifically to tip content creators and make online micropayments" . However, we do not see any design improvement over bitcoin for the application of handling small payments. On the contrary, bitcoin that is vastly more recognised and liquid than other cryptocurrencies, in our view, represents a much better base money for micropayments.

3.7.7 Sidechains

Using Sidechains, or pegged sidechains, is a way to be transfer assets between multiple blockchains. This is an interesting innovation that makes it possible to connect bitcoin with a different blockchain with different rules, and transfer funds between the chains. This can lead to more innovation as it opens up for more experimentation on the sidechain than what would be acceptable to users on the bitcoin blockchain [1].

Relating to microtransactions, a sidechain with new operations that specifically support small payments, may be invented in the future. These operations could be tried out on a sidechain, and if they work well be migrated to bitcoin, or users could keep a certain amount of funds on the sidechain for purpose of doing microtransactions. It should be noted, the sidechain is

not different from bitcoin in that transactions would still represent expensive on-chain transactions that depend on a consensus process, assuming the sidechain is secure, decentralised, and censor resistant in a similar way as bitcoin.

3.7.8 Open Transactions

Open Transactions is an open source library of financial cryptography functions that allows notary services to witness and help with transactions on behalf of users [12]. Parties can use a notary server to issue and transact with many different cryptographic financial instruments, like digital cash, cheques, vouchers, etc. An open transactions notary is a trusted party, but if it fails to follow the rules, users can prove this and hold the server accountable. The system does not use global consensus and transactions are immediate. Counterparty risks, related to financial instruments, would require known identities with PKI infrastructure for parties issuing assets, cheques, etc. for the system to work.

When comparing to the Stroem protocol, Open Transactions seams to issue similar promissory notes, but these are not targeting payments, or micropayments in particular. The Open Transactions library aims to be the base for a general transaction server, supporting all kinds of financial instruments and transactions possible with financial cryptography. The current use and status of the Open Transactions system for payments is unknown to us, but we understand that the company Monetas are building on the Open Transactions system.

3.7.9 Ricardo System

Ricardo is a system for internet payments that lets users interact with servers to perform payments using cryptography with financial assets [7, 6]. The system introduces *the Ricardian Contract* for issued financial digital assets: it is a signed textual version of the contract representing the obligation of the issuer. Any reference in the system to the asset, when trading or making payments, will use a digest of the signed contract as an identification of the particular asset type. This ensures that all involved parties are bound to the exact same obligation contract for a traded asset type.

The Stroem protocol has a narrower scope and only deals with one type of asset contract, the promissory note. In Stroem, contract terms are included in the note and signed by the issuer or previous bearer when transferred. In our case, individual promissory notes are asset *instances* not asset types.

3.7.10 Deposit Accounts Models - ChangeTip, Tibdit

There are several ventures that use a model where users deposit funds with a special central intermediary that can do micropayments between accounts, essentially implementing a giro micropayment system.

One goal with the Stroem protocol was to avoid using accounts with a centralised entity that every participant needs to register with. The use of time limited promissory notes, issued and denominated in bitcoin, means that notes do not work as money substitutes that could circulate indefinitely. If notes are not redeemed within the time limit they become worthless for the merchant. This avoids a growing counterparty risk with account holders. Using an open protocol, where anyone can become an issuer, should support competition and let merchants and consumers select their preferred counterparty to facilitate payments.

3.7.11 ApplePay, GooglePay, and Facebook Payments

The big change in payments, predicted for a long time, is that consumers will use their smart phones for payments instead of plastic cards. Notably, initiatives by the large global internet companies are now driving this change with products like ApplePay, GooglePay and Facebook Payments. Today, all these products are based on the standard card payment networks.

There are differences: ApplePay are claiming to protect the privacy of consumer purchases. Apple cannot make use of consumers' transaction history as they are not an intermediary once a relation is set up. GooglePay, naturally are quite open with that they want to know everything you buy and use it for their core marketing business. Google is then an intermediary for each purchase, which also means that they can support cards by accepting them, not by signing up issuers to their system which is the case with ApplePay.

Using the processing power in today's smart phones and offering an improved purchase experience by using phone features like fingerprint authentication, we expect these payments to grow over time. The current focus for these products seems to be retail payments and maybe online commerce, the type of transactions that the card payments systems were designed for.

In summary, we think these systems are not adapted to handle microtransactions very well, but the important observation is that consumers will soon use their phone for most payments. It is obvious that you can have many different payment apps in your smart phone. The act of payment, for all future different transaction protocols, will evolve into a simple point and confirm with your phone.

4 Benefits

4.1 Application Benefits

Fast and efficient payments. The proposed Stroem payment system works as a middleware on top of bitcoin. Stroem supports aggregation of small payments, represented by short term promissory notes, which eventually settle on the bitcoin blockchain. We think this is a simple way to add a liquidity mechanism for the relatively slow bitcoin network and we believe the system has a number of attractive properties for business.

Transaction proofs. The consumer-merchant interactions are expressed in a series of messages: offer, payment, and receipt. These provide secure proofs that applications can build on and use to provide access according to merchant specific conditions and terms.

Flexible network modes. Merchants receive and validate payments from consumers without waiting for the bitcoin consensus network and without necessarily talking to an intermediary. Consumers are assumed to be online when they make payments but not otherwise, even if, for special applications, this condition can be relaxed.

Flexible transport layer. The actual step of value transfer from the consumer is performed by signing and sending a note directly to the receiver. This allows payments to be low-latency, off-line, sent over any transport, and adapted to special applications like vending machines, POS, and Smart Cards.

Wallet supports automation. The consumer transaction steps of are performed by a wallet application running on a consumer device. This is a key point from a security and privacy perspective, but more importantly, it allows automation in the future. Automation would respect a set of user preferences and take care of repetitive small transactions without interrupting the user. Automation would also keep track of limits, audit contracts, and apply loyalty credits from merchants etc.

Open to service providers and innovation. The benefit of the open protocol is that it allows many issuers and redeemers to form a network. Financial and payment intermediaries could take these roles and provide a connection to the wider economy by providing currency exchange and credit

services. Entities with an existing user base can extend their business by facilitating payments for their users. These could be bitcoin wallet providers, game operators, or even exchanges.

4.2 Business Benefits

Merchants, service providers. With the Stroem payment system, businesses are provided with a new efficient payment solution that can support microtransactions. The ability to make small payments opens the market for new services. It can also work as a complementary solution and generate new revenue for existing services.

As a new payment network, merchants also have a competing solution to the existing payment solutions on the market which can lower total business costs for payments.

Financial intermediaries. To issue and redeem notes to facilitate payments for merchants and consumer represents a business opportunity for financial intermediaries. Intermediaries will earn fees, compete on efficiency and services. We envision that existing players in the payment industry as well as new entities that today have a user base will find this attractive.

Consumers. In addition to the benefits consumers get from having more options to pay for services and premium content, in the longer term we think consumers can benefit financially from microtransactions. Once small efficient payments are possible, it is likely that peer-to-peer services will evolve to reward participants that contribute to the service. Today these systems mostly rely on altruistic behaviours or donations of spare capacity, however, we think efficient microtransactions can expand the scope and size of peer-to-peer systems greatly.

A Stroem Protocol Features

In this section a description is given of the main functions of the Stroem protocol construction. Notably, functions for validation against double spend, aggregation of notes, and authenticated payment information are described.

A.1 Protection from Double Spending

In a digital context where duplication of a data is trivial there needs to be some way for recipients of payments using promissory notes, to assure that a payment is valid. In this context, the fact that the same promissory note can be transferred to any number of different recipients manifests the well known *double-spending* problem for digital tokens. One way to handle this for contracts that represent claims on some party, here the issuing party, is to immediately redeem or verify the validity of the contract. This has various consequences, one obvious being that each transaction needs to be verified by an always online central party, before it can safely be accepted as payment.

Our solution adds a complementary method that allows immediate local validation of payment. The way to achieve this is to issue the base contract with an attribute specifying an ordered list of entities that represent *double spend verifiers*, as shown in Table 1. The payer will request the issuer to include an appropriate list of verifiers according to a merchant's specification. The issuer will only redeem a promissory note that has been negotiated via the entities in the list in the specified order, thus each verifier that receives this negotiated note as payment, can be assured that the promissory note cannot be spent using a different path to the issuer. In other words, the note must pass each and every verifier on the list. If a verifier receives a note as payment, and has not accepted that note before, then it is a valid payment. In addition to expedient delivery of goods and services to consumers, local validation allows a verifier to safely collect a number of payments over time and redeem them in a single operation.

A bearer who accepts a promissory note would verify that the negotiation list contains the verifiers that are expected so far, in the correct order. In the example in figure 1 and 2, the merchant and the redeemer typically would be added to the list of verifiers.

For a verifier to ensure that each promissory note is only used once, records must be kept of accepted notes. In order to keep records limited there is an `Validity time` attribute specified on each promissory note. After expiry, an issuer will not redeem a note so verifiers can safely forget records when they are expired. See section B.1 for a discussion on expired promissory notes.

A.2 Block Negotiation

The construction of the promissory note allows a set of promissory notes with the same bearer to be negotiated in one operation. The purpose is to let merchants aggregate payments before redemption. This is done by assembling a list of the promissory notes and constructing a hash tree [19]. The leaves of the tree are the hash values for each promissory note that would be used to sign and negotiate each promissory note by itself. The root of the hash tree is used as input for a digital signature that is included in a new negotiation record, valid for the whole block. The record structure is the same as is used to negotiate a single promissory note (shown in Table 2). The new negotiation record is attached to the list to form the finished block. The block represents a transfer of value equal to the sum of the values of the promissory notes in the block.

A.2.1 Negotiate a Received Block

If a party becomes the bearer of a block of promissory notes, negotiated in this way, that party can negotiate the block to a new bearer by adding a new negotiation record.

A.2.2 Negotiate a Subset of a Received Block

The way the signature for a block of promissory notes is constructed, using a hash tree, it is possible to remove any leaf component and leave the signature verifiable as long a the hash value of the leaf is present instead of the leaf itself. This operation can be used to select all leaf promissory notes in a block that has the same issuer. This enables the bearer to split a block into a number of blocks, each containing promissory notes issued by the same issuer. The resulting blocks will each be negotiated to the proper issuer for redemption. The used signature must cover the exact subset of leaves an nothing else to be safe. In a similar way a bearer will want to split a block into blocks depending on the next double spend verifier as described in section A.1.

A.3 Authenticated Payment Information

For every transfer of ownership when a promissory note, or a block of promissory notes, is negotiated to a new bearer, it is possible to supply arbitrary payment information that is sent to the new bearer. The semantics of this information can be anything agreed upon by the two parties engaged in the transfer, i.e. the current bearer and the next to become bearer. More

specifically, when a consumer buys something from a merchant, the payment information could specify what is traded, e.g. an URL to some resource, the title of an ebook, the product number, order id, etc.

In the construction of the promissory note, the negotiation records include a cryptographic hash value computed from the concatenation of the arbitrary payment information and a random nonce, cf. attribute `Payment info hash` in Table 2. If the arbitrary payment information together with the used nonce is sent along with a negotiation occurs, the recipient can verify that this information is authentic, or more precisely, the recipient can verify that the same party that made the payment authenticated the payment information. The attribute `Payment Info`, in Table 3, is used when the payment information and the nonce are supplied in this way.

A.4 Redactable Payment Information

The payment information can be redacted from a promissory note at any time without affecting the validity of the signatures as these do not cover the payment information but the computed hash values, which are still present. This enables a bearer of a promissory note to negotiate the note to a new bearer without revealing anything about what payment information was received by anyone up to this point. For anyone having access to original payment information, it is also possible to authenticate this at any later time with the access to the promissory note, even if all payment information was redacted from the note.

A.5 Coupons

Coupons are tokens that entitles the holder to receive a discount or some benefit when purchasing a product, usually exclusively at a specific merchant. The construction of promissory notes presented in this document can be used as coupons by using the double spend verifier list attribute to restrict the value of the note only to be spent at one merchant. In this case the merchant buys coupons from an issuer of choice and distributes them to customers.

It is also possible for the merchant to issue coupons directly. The denomination, or currency, of a promissory note could be generalised to whatever the merchant wants to offer, e.g. “months of internet usage”, and with an `amount` of 1, such a coupon would entitle the consumer with the coupon, one month of internet usage from that particular merchant.

In both cases, at expiry, the value is returned to the merchant if the coupon was not used.

A.6 Refunds

A refund is a consumer payment that is reversed by a merchant. This can be initiated by a consumer and motivated in a claim that something was not delivered as agreed, or by the merchant when an order cannot be completed. Refunds are assumed to only affect a small fraction of all payments.

Refunds are easy to handle once we have the mechanisms of promissory notes in place. A way to handle refunds would be to use the issuing hub of the received payment, create a new promissory note issued to the consumer. No extra consumer credentials are needed if the refund is issued to the same public key that was used in the payment that is being refunded. Alternatively, a consumer refund public key can be supplied in the payment info for each payment. Using our promissory notes it is easy to express refunds of type *money back*, where the refund can be spent anywhere, and of type *right to exchange*, where the refund can be spent exclusively at the merchant, similar to coupons, cf. A.5. Refund promissory notes should be valid for an appropriate duration, expected to be longer than notes used for normal payments.

A.7 Provable Properties

Using digital signatures it is possible to demonstrate the authenticity of a document. To create a digital signature the signing party has access to a secret key that no other party can access. For each secret key there exists a corresponding public key. If we assume that the digital signature scheme used is secure, then under these assumptions anyone with access to the public key can verify if a given signature of a document is signed by the party with access to the corresponding private key.

This is the basic mechanism for forming a set of propositions that we can demonstrate to be true if that is the case. These propositions are constructed to be beneficial for facilitating trade over a digital channel and are described in the following sections.

Any transfer of ownership of a promissory note with some value can be regarded as a payment. In particular, a payment from a consumer to a merchant is in the form of a promissory note, negotiated to the merchant. This operation is completed when the consumer creates and adds a digital signature to the note with the new bearer. Before that step, the consumer first needs to be the bearer of a promissory note with the right amount and other attributes like issuer, validity time, etc. as required by the merchant. One way to fill this requirement is that the consumer acquires an appropriate promissory note directly from an issuer at the time of the payment. The issuer

acts as an intermediary that the user has preselected and that will make the issues requested by the consumer because the issuer has received some funds ahead of time, or is convinced he can bill the consumer at a later time, or gets paid in real-time using some other means of payment.

A.7.1 Proof of Purchase, Consumer to Merchant

To make a payment, the user must negotiate a promissory note to a merchant. This involves creating a digital signature which requires access to a private key. This means that if a user, here called the demonstrator, has made a specific payment to particular merchant, he should be able to demonstrate, to the merchant, that he has access to the private key that made this payment. The steps for the demonstration that a payment was made is shown below,

1. The *merchant* challenges the user with a document containing a nonce.
2. The *demonstrator* creates a new signature of the challenge document, using the private key that was used for the payment to be demonstrated.
3. The *demonstrator* sends the new signature and a copy of the promissory note used as payment, to the *merchant*.
4. The *merchant* verifies if the payment has been accepted before. If this is true, the payment was real and if the signature is valid when verified with the public key in the promissory note, the *demonstrator* has access to the private key that made the payment.

A use case for this is when a merchant will give the right to each user that buys some content to access that content in the future if the user can demonstrate that it has been paid for previously. Other uses are possible like login authentication, claiming rebates, special offers or subscriptions. It should be noted that anyone that has access to the private key and the transaction details can make the demonstration, and the merchant will only be sure that some user made the payment and some user is demonstrating this fact.

It should be noted that sometimes scope is difficult to limit, e.g. if the consumer buys an article, then the consumer can publish the key and everyone can read the article for free. This might seem equivalent to sharing a password but it is easier (less risky) to share since the key is a pseudonym and the user is anonymous. See discussions for a better protocol for access rights in section B.2.

A.7.2 Proof of Purchase, Consumer to Anyone

There are two ways the proof in section A.7.1 can be extended so that the demonstration can be made to outsiders that do not have access to the merchant's records.

In section A.7.1, the merchant can judge if the demonstrated payment in step 3 is valid by verifying, according to his own records, if this payment was previously accepted. If this verification is not performed the demonstrator can construct something that looks like a payment, in the form of a promissory note that was never sent to the merchant. An outside observer cannot know if a payment was made or not, as there is no signature or receipt from the merchant.

Signed receipt. With the addition of an explicit receipt, in the form of a signed document sent to the consumer from the merchant saying that the merchant accepted the promissory note as valid payment, it can be demonstrated that the merchant received a payment and the private key that made the payment can be accessed by the demonstrator.

Collect receipt from issuer. Another way is for the consumer to use the issuing hub to collect the redeemed promissory notes as they are completed. Every promissory note that the consumer sends as payment is eventually redeemed at the issuing hub if they are redeemed. At this stage, they will contain a signature of the merchant as authentication when they were transferred from the merchant. This could be regarded as proof that the payment was accepted.

A.7.3 Proof of Order, Merchant to Anyone.

The payment information that can be included with the payment is signed by the party making the payment as part of negotiating a promissory note to the next bearer.

Assuming that the trade protocol used between a consumer and a merchant, states that the payment information attribute should include the order details of a purchase, then, in a dispute over what goods a particular payment was for, the merchant can use this information as a proof that cannot be refuted by the payer. The payer must show a proof of purchase as described in section A.7.2, but included in this proof is a hash value of the payment information and the merchant can use his records to provide this information and demonstrate what the purchase was.

B Discussion

B.1 Expired Unredeemed Promissory Notes

A promissory note is a promise to pay some amount on demand. The promise will, however, be limited in time for a few reasons.

Firstly, the outstanding risk will not increase to high levels if promissory notes are redeemed and completed rather than circulating for longer time as money. The design philosophy for this system is that each promissory note should be used to execute only one consumer merchant transaction and then be redeemed and completed.

Secondly, the maximum life-time of a promissory note represents a cost for record keeping against double spend attempts. As long as a specific promissory note is not expired, each double spend verifier needs to know if it has accepted that note before or not.

When a promissory note expires, the issuer's promise should not be upheld. Who is entitled to the value of the note is arguably negotiable between the issuer and the initial buyer of the promissory note. For many protocol use cases, the right to get refunded is assigned to the initial buyer at expiry, so it seems practical to have this as a default if nothing else is agreed. Any notification when an expiry has occurred, or what payment method and how the refund is requested is arbitrary and can be specified by the issuer. In any case, a refund request of an expired note should require a signature of the entitled party to authenticate the payment instruction. The is analogous to how the payment is authenticated for normal redemption.

B.1.1 Execute Expiry Action

Any bearer of a promissory note always has the option to let the note expire. What expiry entails is up to the issuer and the first bearer, as discussed in B.1. For protocols that make use of this feature it is beneficial to have a way to add an annotation to the promissory note, send it to the issuer that can execute the expiry action immediately. However, that would bypass any double spend verifiers, and enable double spend attacks. A solution would be to define an annotation so that the issuer will only execute the expiry action on a note with this annotation, and send the note via the path through all double spend verifiers. The verifiers will negotiate the note in the standard fashion until it reaches the issuer, that will execute the expiry action.

B.2 One-time Rights and Enumerated Demonstration

In this section we extend the proof to the merchant in section A.7.1 to a more license like protocol. We note that we can define the right given to the consumer as a one-time right that, when exercised, returns a new one-time right. This makes it possible to offer the user the right to download a file up to a fixed number of times, or to extend the download right indefinitely. Even if there is no way stop the consumer to copy or share the returned one-time right, with this protocol only the last demonstrator will be holder of the remaining right. The merchants need to track and enforce the one-time property of rights, which means the merchant has to update state for each demonstration, in addition to the processing needed to validate that the payment was previously accepted.

B.3 Merchant Consumer Privacy Preserving Messaging Service

The consumer will frequently communicate with its selected issuer and payments made will eventually be redeemed by the issuer. We note that the issuer could act as a communication point from merchant to the consumer. This is practical for receipts, as described in section A.7.2, or for receiving coupons and refunds defined in sections A.5 and A.6.

For a merchant to send a message to a consumer, the merchant would contact the issuer hub used by the consumer. This issuer can be found from any payment made by the consumer. The merchant could send a message consisting of the public key used in the consumer payment, a url where the consumer can retrieve the message, and a nonce. The consumer will connect to its issuer and retrieve the url and the nonce using the public key as an mailbox address. Using the url, the nonce, and a signature as authorisation, the consumer would retrieve the message.

This method has the advantage that a merchant can send messages to any consumer without requiring the registration or collecting identity information from the consumer. The merchant knows about a previous purchase of the consumer and can reward or send targeted messages to the consumer to improve the business. A consumer is expected to use different public keys for every transaction. This means that it is easy to filter out messages from specific merchants, or relating to a specific purchase. It grants consumers the power to receive messages as long as they want or to be forgotten when they want.

B.4 Digital Wallet

For users to get the full benefits of using promissory notes as payments, users need to have a safe and easy way to store and retrieve the items involved. A *digital wallet* is a client program, running on the user's mobile device or desktop computer that performs these tasks. The wallet will also handle the protocols needed for communication with issuers and merchants. Below is an collection of what a typical wallet would need to store for the user.

Keys. Users make transactions by acquiring notes from an issuer, adding payment information, and transferrinf notes to merchants. This is done using digital signatures which means users must handle key pairs consisting of a private key and a public key.

Promissory notes. Users will want to store copies of completed transactions (sent promissory notes). These represent a record of transactions and can also be used to demonstrate proof of purchase to the merchant, as explained in section A.7.1. Any unspent promissory note would also be stored in the wallet.

Receipts, access rights. Users need a place to store receipts received from merchants, cf section A.7.2. Receipts would act as licenses to access content, according to terms offered by merchants.

Coupons and refunds. Users also need a way to store and retrieve refunds and coupons, described in sections A.5 and A.6. Coupons received with rebates and other offers from merchants would be collected, and used automatically whenever possible. The user would control how coupons are received, displayed and accepted by the wallet.

Messages. The wallet would be responsible for accepting and handling of messages from merchants, as sketched in section B.3.

References

[1] Adam Back, Gregory Maxwell, et al. Enabling Blockchain Innovations with Pegged Sidechains. 2014. [http://www.blockstream.com/sidechains.pdf, Retrieved 18 August 2015].

[2] Mira Belenkiy. E-cash. In Burton Rosenberg, editor, *Handbook of Financial Cryptography and Security*. Chapman and Hall/CRC, 2010. ISBN: 978-1-4200-5981-6.

[3] Payment Channels: Payments to a Pre Determined Party. *bitcoin.it/wiki*, 2015. [https://en.bitcoin.it/wiki/Contract].

[4] David Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 199–203. Plenum, 1982.

[5] R. H. Coase. The Nature of the Firm. *Economica*, 4(16):386–405, 1937.

[6] Ian Grigg. The Ricardian Contract, 2004. [http://iang.org/papers/ricardian_contract.html, Retrieved 4 August 2015].

[7] Systemics Inc. Ricardo by Systemics, 2001. [http://www.systemics.com/docs/ricardo/, Retrieved 4 August 2015].

[8] Thaddeus Dryja Joseph Poon. The bitcoin lightning network: Scalable off-chain instant payments. 2015. [https://lightning.network/lightning-network-paper.pdf, Retrieved 4 August 2015].

[9] Sergii Moshenskyi. *History of the Weksel*. Xlibris, August 2008.

[10] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. [https://bitcoin.org/bitcoin.pdf, Retrieved 25 February 2015].

[11] Jakob Nielsen. The Case For Micropayments, 1998. [http://www.nngroup.com/articles/the-case-for-micropayments, Retrieved 25 June 2015].

[12] Chris Odom. Open-Transactions: Secure Contracts between Untrusted Parties. 2015. [http://www.opentransactions.org/open-transactions.pdf, Retrieved 18 August 2015].

[13] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology, 2008.

[14] George Selgin. *The Theory of Free Banking: Money Supply under Competitive Note Issue*. Rowman & Littlefield Publishers, 1988.

[15] Nick Szabo. Mental Transaction Costs and Micropayments. 1999. [http://szabo.best.vwh.net/berlinmentalmicro.pdf, Retrieved 25 June 2015].

[16] Peter Todd. Ripple Protocol Consensus Algorithm Review. May 2015. [https://github.com/petertodd/ripple-consensus-analysis-paper/raw/master/paper.pdf, Retrieved 18 August 2015].

[17] Wikipedia. Dogecoin, 2015. [https://en.wikipedia.org/wiki/Dogecoin, Retrieved 4 August 2015].

[18] Wikipedia. Giro — Wikipedia, the free encyclopedia, 2015. [https://en.wikipedia.org/wiki/Giro#Model, Retrieved 4 August 2015].

[19] Wikipedia. Merkle tree — Wikipedia, the free encyclopedia, 2015. [http://en.wikipedia.org/wiki/Merkle_tree, Retrieved 25 February 2015].

[20] Wikipedia. Ripplepay - early development (2004-2012), 2015. [https://en.wikipedia.org/wiki/Ripple_(payment_protocol), Retrieved 4 August 2015].

[21] Wikipedia. Transaction cost theory, 2015. [https://en.wikipedia.org/wiki/Theory_of_the_firm, Retrieved 25 June 2015].

[22] Oliver E. Williamson. Transaction cost Economics: the Natural Progression. 2009. [http://www.nobelprize.org/nobel_prizes/economic-sciences/laureates/2009, Retrieved 4 August 2015].