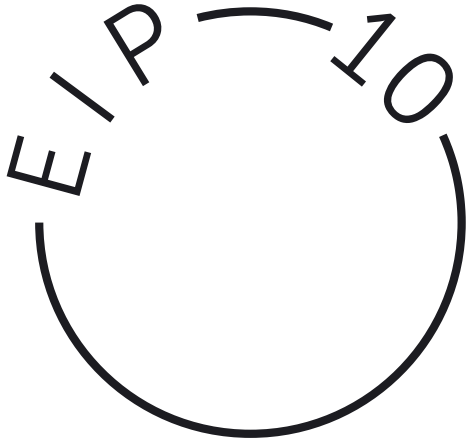


WHITE PAPER

NICHEWORK™: MULTISERVICE DIGITAL WALLET ECOSYSTEM WITH EXCHANGE/CONVERSION FUNCTIONALITY

EIP10-020 REVISION

http://eip10tech.com/

ABSTRACT

Digital [mobile] wallet concept when provides safe mechanisms for storage and exposition of digital assets and combined with cryptographically-secured transactions has demonstrated its utility through many projects, Google Wallet, Apple Pay, Samsung Pay and others to name. Each such a project is a simple application representing the services from wallet stakeholders within an appropriate ecosystem.

Same kind of applications with a social network [of wallet owners] behavior, especially peer-to-peer capabilities without third party authority supervision, has substantial value added user benefits and wide potential of future evolution. The most attractive from a user [of a such wallet] perspective set of functions are the exchange/conversion capabilities of assets (values, credentials), which are stored in that wallet.

Nichework™ platform implements this concept in a generalized manner. It provides a plurality of resources, each with a distinct state and operating code but with an ability to interact through a message-passing framework with others. Having as the execution environment based on the blockchain technology framework (e.g. Ethereum) Nichework™ allows securely, effectively, and transparently execute transactions of exchange/conversion functionality for participants [wallet owners and service providers (SP)] for a proposed digital wallet ecosystem.

In addition to all other advantages of blockchain technology we are focusing on the idea of smart contracts within proposed [Ethereum] implementation of blockchain technology. Smart contracts are computer protocols intended to facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts help you exchange money, property, shares, or anything of value in a transparent, conflict-free way while avoiding the services of a middleman.

To avoid scale and time execution issues of root blockchain framework (e.g. Ethereum) we suggest using the Plasma framework to do scalable computation on the blockchain with the structure of creating economic incentives to autonomously and persistently operate the chain without active state transition management by the contract creator. The nodes themselves are incentivized to operate the chain.

Plasma is a proposed framework for incentivized and enforced execution of smart contracts which is scalable to a significant amount of state updates per second (potentially billions) enabling the blockchain to be able to represent a significant amount of decentralized financial applications worldwide. These smart contracts are incentivized to continue operation autonomously via network transaction fees, which are ultimately reliant upon the underlying blockchain (e.g. Ethereum) to enforce transactional state transitions.

We discuss Nichework™'s design, implementation issues, the opportunities it provides and the future hurdles we foresee.

1. INTRODUCTION

With ubiquitous smartphone penetration and total digitalization of more and more services in the human world, mobile applications have turned into incredibly in-demand and even irreplaceable means of using that services for consumers. Technology-rooted movements like Apple Pay, Samsung Pay, Google Wallet and others have demonstrated, through the power of safe storage and convenient presentation [to use] for the digital assets (payment, loyalty, id etc.), that mobile applications with wallet functionality is becoming universal day-to-day working instrument for millions of people in the world. The past few years have seen the emergence of increasingly capable digital wallets that integrate payments, offers, loyalty programs, enhanced product information, and identity management. Delivering contextually relevant services to customers' fingertips, digital wallets will change how customers shop, and not only shop, behave. Successful digital wallet operators threaten to come between banks and their consumer and merchant customers. This a [mobile] Digital Wallet paradigm is the first element of a disrupting FinTech Concept, named as [mobile] Multiservice Digital Wallet with Social Network capabilities.

At the same time phenomenon of social network with its default, consensus mechanisms and voluntary respect of the social contract, coupled with an incredibly cheap global information transmission due to pervasive internet connections allow to consider such ecosystem [of wallet owners] as a peer-to-peer decentralised value-transfer system, shared across the world and virtually free to use. This system can be said to be a specialized version of a cryptographically secure, transaction-based state machine. It is the second element of the Concept [Multiservice Digital Wallet with Social Network facilities].

And, finally, the third element, is the Blockchain Technology that performs one key goal - facilitating transactions between consenting individuals who would otherwise have no means to trust one another. This may be due to geographical separation, interfacing difficulty, or perhaps the incompatibility, incompetence, unwillingness, expense, uncertainty, inconvenience or corruption of existing legal systems. By specifying a state-change system through a rich and unambiguous language, and furthermore architecting a system where an agreement is enforced autonomously. Transactions in this Blockchain based system would have several attributes that are not often found in the real world. The incorruptibility of judgement, often difficult to find, comes naturally from a disinterested algorithmic interpreter. Transparency, or being able to see exactly how a state or judgement came about through the transaction log and rules or instructional codes, never happens perfectly in human based systems since natural language is necessarily vague, information is often lacking, and plain old prejudices are difficult to shake.

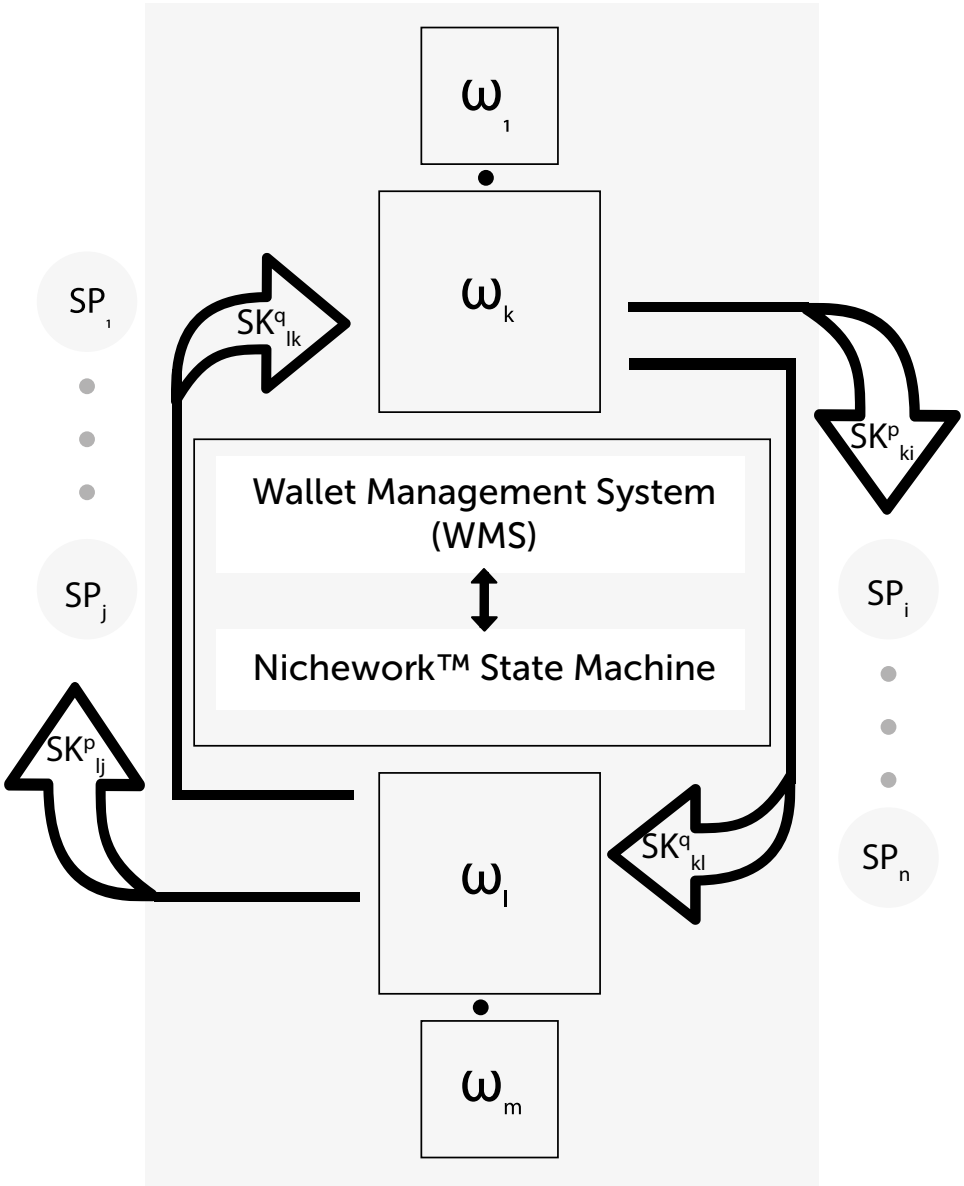
Overall, it is proposed to provide a system such that users can be guaranteed that no matter which other individuals, systems or organizations they interact with, they can do so with absolute confidence in the outcomes and how those outcomes might come about.

The Nichework™ platform utilizes the blockchain paradigm coupled with cryptographically-secured transactions and smart contract possibilities, particularly on the Ethereum message-passing framework of a transactional singleton machine with shared-state.

2. MULTISERVICE WALLET WITH EXCHANGE/CONVERSION FUNCTIONALITY

The functionality of exchange/conversion the assets which are stored in the digital (mobile) wallet is central to the Nichework™ concept. It is assumed that the wallet (**ω**) is loaded into terminal devices of **m** of users, and **n** of service providers (**SP**) is connected to the platform. The basic principles of such functionality are examined in the example, where the **user**$_k$ (ω_k) and **user**$_l$ or (ω_l) exchange assets that are stored in their wallets that are in the Nichework™ and are governed by its rules.

The **user**$_k$(ω_k) wants to exchange as asset, for example, points of the **SP**$_i$ loyalty scheme, for the **user**$_l$**'s**(ω_l) asset, points of the loyalty scheme **SP**$_j$. For simplicity, we consider the case when, that these points are exchanged completely without a remainder, and of course both users are the registered participants of the loyalty programs for both schemes.



To further explain the process flows in the Nichework ™ framework, we introduce the concepts of blockchain technology forming the basis of Nichework ™ framework, and adopt the following notation:

- $account_k$ – account in blockchain network, uniquely matched to ω_k
- sk^p_{kj} – smart contract of type p, triggered by a transaction between $account_k$ and $account_j$
- ω_k (α_1, ..., α_i, ..., α_j, ..., α_n) – wallet **k** with values (credentials) from SP_1, ..., SP_i, ..., SP_j, ..., SP_n

The process of exchange of assets α_i and α_j between wallets ω_k and ω_l respectively consists of two opposing actions: the transfer of value α_i from wallet ω_k to wallet ω_l and the transfer of value α_j from wallet ω_l to wallet ω_k in the opposite direction.

Note that the an asset α_i transfer process from wallet ω_k to wallet ω_l is initiated at the WMS (Wallet Management System) level, but is performed in an execution environment built on blockchain technology (see section 4 of this document). In other words, each multiservice wallet (not to be confused with other wallets, for example in the Ethereum framework) corresponds to an address (account) in the blockchain environment, and the transaction is triggered by a message from $account_k$ to $account_l$.

Actually the transfer of values from ω_k to ω_l occurs when executing a smart contract, in this case sk^q_{kl}. An important point is that a smart contract has a type, that is, in fact there are different smart contracts in the system. This implies that a smart contract, which is responsible for the transfer of an asset between addresses (accounts), also on the scheme in the designation of this contract there has no index i, which should demonstrate that the asset of α_i was transferred. This is done because this particular asset is one of the input parameters for the executable smart contract and does not persist in the smart contract body.

In fact two smart contracts are started simultaneously. One, described above, and the other using a message that triggers a transaction between $account_k$ and $account_{SPi}$ to write off the value from wallet ω_k to wallet ω_l. This fact suggests that SPs should also be "loaded" into the blockchain execution environment. In other words if SPs agree to participate in the multiservice wallet system then they must connect to the Nichework™ framework using blockchain technology, in particular, they will agree to the execution of smart contracts when performing transactions in the system. Generally speaking transferring their operations to a blockchain environment brings a number of benefits to SPs (see section 3 of this document). Here such a smart contract is denoted as sk^p_{ki}. This is a contract of another type (p), and with respect to the notation, the remark similar to the previous remark related to the absence of the index i is valid.

As a result the first part of the exchange is realized by performing two smart contracts:

$account_k$ ----------> (α_i) to $account_l$: sk^q_{kl} (1)
$account_k$ ----------> (α_i) to $account_{SPi}$: sk^p_{ki} (2)

similarly the second part of the exchange is realized:

$account_l$ ----------> (α_i) to $account_k$: sk^q_{lk} (3)
$account_l$ ----------> (α_i) to $account_{SPj}$: sk^p_{ij} (4)

This example is offered only for a high-level description of the idea of exchanging/converting the assets/values in the blockchain execution environment. Due to the limits of the current document, this example does not demonstrate the following:

- other business cases when wallets exchange assets of different nature, for example, points of the loyalty scheme are sold/bought; partial exchanging of assets between parties; a case when one of the exchanging users is not a participant in the SPi loyalty scheme, and the other one is a member of that scheme; etc.
- interaction issues of WMS and blockchain execution environment, for example, synchronization of operations between non-blockchain world and blockchain execution environment; parameter exchange issues between the worlds of blockchain/non-blockchain; implementations of a repository for smart contracts; etc.
- issues of using other mechanisms of the blockchain environment, for example, the fact that values in this environment are circulated in the form of special type tokens (not to be confused with the concept of a token in the WMS system); issues of synchronization of operations in the blockchain environment, when a simultaneous execution of several contracts logically representing a single set of actions is initiated; etc.
- the purpose and operation of the Nichework™ State Machine which manages the core idea of the Nichework™ concept, namely wallet states, the aggregate of which constitutes the state of the Nichework™ network at a moment of time.
- other implementation issues of the concept of Nichework™.

3. BLOCKCHAIN FOR CUSTOMER LOYALTY REWARDS PROGRAMS

In this document the concept of a multiservice wallet is based on the product developed by the Prime Numbers company [Pri-Num] [1]. The main features are the universality of the representation (invariance) of values/assets in the wallet from various SPs, and the exceptionally high level of safety of values/assets stored due to the unique proprietary tokenization technology. Wherein by now it is obvious that the presence of payment (financial) instruments, bank cards, electronic money, crypto currency, etc. is a necessary condition for the success of any wallet on the market.

Nevertheless the real success driver of the digital wallet applications is the idea of putting all the tools (values) into the wallet customer uses in his/her daily life, that is, loyalty cards, transport cards, identification cards etc. In other words, literally the values from the customer's leather wallet are digitized and moved to the digital wallet. And the primary role in this process is played by the services of various loyalty schemes. Availability in digital wallet of loyalty points, coupons, vouchers, etc. is a sufficient condition for the success of this wallet.

Apart from benefits of the blockchain technology for the exchange/conversion features of the multiservice wallet there are many benefits which can be obtained from the placement of loyalty programs in the same blockchain environment. The blockchain technology can lead to a synergy between these two sets of functionality: an exchange of values between owners of digital wallets and the loyalty schemes. Moreover this synergy is achieved through the same mechanism in blockchain technology, namely: system of smart contracts.

The main benefits that are achieved by placing loyalty schemes in blockchain technology environment are discussed in detail in the Deloitte report [2]. Here are just a few important considerations on this.

Connecting a disconnected world

Blockchain will allow instantaneous and secure creation, redemption, and exchange of loyalty reward points across programs, vendors, and industries through a trustless environment using cryptographic proofs in lieu of trusted third parties and administrators. Through a rigorous online protocol, well-programmed building blocks, and smart contracts, blockchain has the capability to operate without intermediaries. The key elements of such a blockchain solution are a loyalty network platform (hereon referred to simply as a loyalty network), reward applications, and loyalty tokens.

Reducing costs

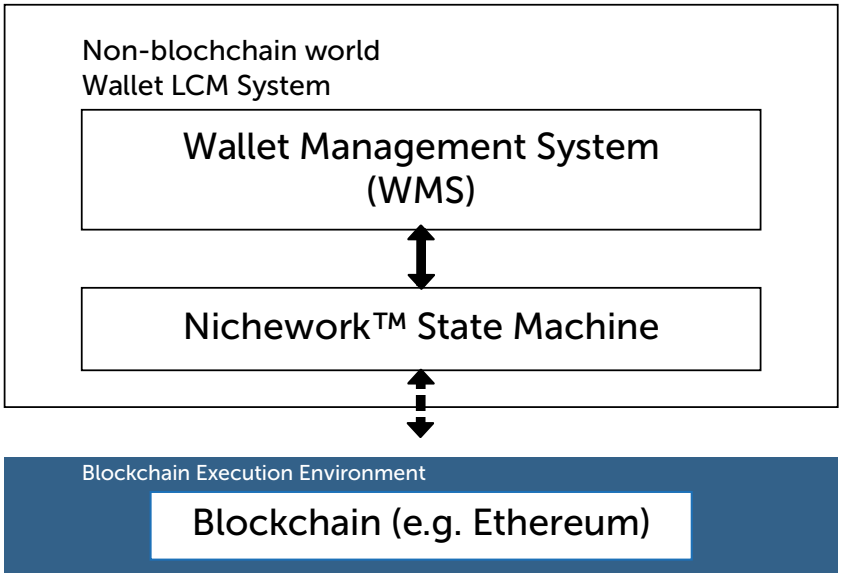
Many managers of loyalty reward programs are hesitant to incur the cost of implementing new technology. This hesitation is understandable, given that they are being asked to switch to a fundamental overhaul of how transactions, customer acquisition, and systems management are executed. But this "overhaul" should be put into perspective. Despite the fundamental changes it promotes, blockchain is a system facilitator, not a replacement for an existing system. One of blockchain's attractive aspects is that it interacts with legacy systems through smart contracts that transmit transaction records accessible to the users with access who integrate them into their systems. These legacy systems continue to perform functions outside of what they specifically need blockchain for - to enhance or improve transaction process. An existing loyalty rewards management system, for example, will still hold sensitive Personally identifiable information (PII), or sensitive personal information (SPI) of a customer, as that data will not reside on the blockchain.

And finally we believe that loyalty schemes, which are considered by their users to be the most valuable, can in fact significantly increase user loyalty (user retention), to acquire new users (attract new users) and increase their economic performance (increase in the number of transactions for goods/services) precisely through participation in an open ecosystem of loyalty. This vision is based on the latest results of economic behavior research, primarily the work of the Nobel Prize winner in economics for 2017 by Richard H. Thaler [3].

On the practical side, these results mean that in an open loyalty ecosystem there is a place for successful win-win strategies in which all members of the ecosystem benefit, and even direct competitors.

4. BLOCKCHAIN EXECUTION ENVIRONMENT

The design of the proposed architecture consists of two main parts: wallet Life Cycle Management System and the execution environment based on the blockchain technology framework.



The choice of the blockchain technology as a basis for the proposed solution is inspired by the desire to take advantage of the basic principles of the technology that include:

- **Decentralization**. There is no central authority, with no single point of vulnerability or failure (P2P nature of transactional environment).
- **Trustlessness.** A blockchain does not require trust in any authority or any participant.
- **Consensus network**. A process allows participants to come to an agreement over what is true or false. For a cryptocurrency, it would typically concern the validity of a transaction.
- **Transaction transparency**. The validity of all transactions is available to everyone on the network.
- **Transaction immutability**. Once added to the blockchain, a transaction cannot be changed or manipulated.
- **Pseudonymous**. Transactions are anonymous (in that they do not require personal information) but can be traced back to a public key.
- **Interoperability**. Interoperable ready-made environment.

With this approach, we can consider a multiservice wallet with the exchange/conversion functionality of assets stored in it, a distributed application executing in a safe and self-regulating environment.

5. ETHEREUM FRAMEWORK

Despite the fact that the design of the architecture assumes the use of any implementation of blockchain technology as the execution environment, it is suggested to select the Ethereum framework as such environment. This choice is dictated primarily by those specific features that are inherent to Ethereum, primarily the ability to implement the required functionality of digital wallet through the concept of smart contracts. Below we describe the features of Ethereum, which are important for the implementation of the proposed architecture.

The intent of Ethereum is to create an alternative protocol for building decentralized applications, providing a different set of tradeoffs that will be very useful for a large class of decentralized applications, with an emphasis on situations where rapid development time, security for small and rarely used applications, and the ability of different applications to efficiently interact, are important. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met, can also be built on top of the platform, with vastly more power than that offered by Bitcoin scripting because of the added powers of Turing-completeness, value-awareness, blockchain-awareness and state.

The concept of **Smart Contract** plays the main driving role in the proposed multiservice wallet design. Note that "contracts" in Ethereum should not be seen as something that should be "fulfilled" or "complied with"; rather, they are more like "autonomous agents" that live inside of the Ethereum execution environment, always executing a specific piece of code when "poked" by a message or transaction, and having direct control over their own ether balance and their own key/value store to keep track of persistent variables.

The author of the idea of smart contract Nick Sczabo wrote "What is the meaning and purpose of "security"? How does it relate to the relationships we have? I argue that the formalizations of our relationships -- especially contracts -- provide the blueprint for ideal security." In other words, smart contract is not only a matter of convenience, transparency and making the transaction cheaper, it is primarily a matter of security.

Szabo's 1994 description was as follows: "A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitrations and enforcement costs, and other transaction costs." Smart contracts today have now developed in terms of technological innovation and legal construction but the definition of a "smart contract" is still materially the same as Szabo defined it over 20 years ago.

So, if blockchain is a "distributed ledger technology" and if two parties can have a shared distribution ledger that governs the whole process of creation, movement and management of money or cryptocurrency, and if those two parties have a shared business logic they can also create a "smart contract" between themselves.

Under a smart contract, if an actor from one party proposes a transaction and it is validated against the contract by a blockchain network, when there is consensus from the network then the transaction under the contract is finalized. Therefore the rules for how the contract is being evaluated are the same for both sides.

Smart contracts aim to provide security superior to traditional contract law and to reduce other transactional and administrative costs associated with contracting. Whilst it is unlikely that smart contracts will fully replace the traditional legal contract (in the near future at least) they can reduce the burden and the complexity of writing a new contract each time as the smart contract technology can be used to execute a number of terms of a contract between two parties automatically.

Thus the terms of a legal contract are then written in a programming code and this code is used to define the rules and consequences in the same way that a traditional legal document would, stating the obligations, benefits and penalties which may be due to either party in various different circumstances. This code is then automatically executed by a distributed ledger system without the further onerous input from either party.

Ethereum is an open source smart contract protocol and is decentralised – representing a cultural shift of some of its predecessors (just as Bitcoin also was). Ethereum uses the "ether" to motivate a network of peers to validate transactions, secure the network and achieve consensus about what exists and what has occurred – thus enabling a smart contract to self-execute.

The "Ethereum Virtual Machine" (EVM) is where the smart contracts run in Ethereum. It provides a more expressive and complete coding language than Bitcoin for scripting and is also a Turing Complete programming language – meaning that it can encode any computation that can be conceivably carried out. The Ethereum blockchain records transfers of native cryptocurrency called "Ether."

One of the main benefits of using Ethereum is the tighter security. Where every participant is a client and a server at the same time, this allows Ethereum to increase its network's security and resilience – arguably far beyond that of competitors such as Bitcoin. This is because in other systems, the entire network is handled by a single server entity and thus it becomes a weak point and far more able to be exploited by potential attacks and hacks.

As Ethereum is a decentralised network, it is very resistant to such hacker attacks and has, potentially, zero downtime – even if some parts of the network go down. The transaction log becomes robust as the integrity of the data is verified, stored and protected. Records can be accessed by anyone on the network, are easily traceable and are virtually unalterable – therefore, Ethereum has inbuilt checks and balances to ensure that transactions are near 100% accurate.

Ethereum has become the best way to ensure that applications work efficiently and correctly. As the blockchain network behind the application, using Ethereum, executes an order or transaction by itself, verifies the output(s) by itself and distributes the value between participants by itself there is no need to have separate blockchains for each application or to have costly central administrative processes for monitoring and execution.

6. PLAZMA FRAMEWORK [18]

With blockchains, the solution for enforcing correctness has generally been having every participant validate the chain themselves. To accept a new block requires one to fully validate the block to ensure correctness. Many efforts to scale blockchain transactional capacity (e.g. Lightning Network) requires using time commitments to build a fidelity bond, (an assert/challenge agreement) so that the asserted data must be subject to a dispute period for participants on the blockchain to enforce the state. This assert/challenge construction allows one to assert a particular state is correct, and if the value is incorrect, then a dispute period exists where another observer can provide a proof challenging that assertion before a certain agreed time. In the event of fraudulent or faulty behavior, the blockchain can then penalize the faulty actor. This creates a mechanism for participants to be encouraged to enforce if-and-only-if the incorrect state is asserted. By having this assert/challenge-proof construction, interested participants can be able to assert ground truths to non-interested participants on the root blockchain (e.g. Ethereum).

This structure can be used not only for payments, but extended to computation itself so that the blockchain is the adjudication layer for contracts. However, the presumption would be that all parties are participants in validating the computation. In Lightning Network, for example, the construction makes it so that one can establish commitments to computing contract state (e.g. with pre-signed tree of multisignature transactions of conditional state).

These constructions allow for highly powerful computation at scale, however there are some issues which require the summation of a lot of external state (i.e. summation of entire systems/markets, computation of a large amount of shared/incomplete data, large number of contributors). This form of commitment to multiparty off-chain state ("state channels") requires participants to fully validate the computation, or else there are significant amount of trust established in the computation itself, even in single-round games.

Additionally, there is usually a presumption of "rounds" whereby the execution path must be completely unrolled before contract initiation, which gives participants the opportunity to exit and force expensive computation on-chain (as it is not possible to prove which party is halting).

Historically, many people believe that the blockchain is best applied towards transactional payments as a gross settlement system. However, it's understood that a gross settlement system is difficult to scale. Net settled designs such as the Lightning Network, a payment channel network, changes the structure to allow for nearly unlimited payments between participants. Transactional capacity is increased dramatically as channels are net-settled on the blockchain. Payments can be routed across a network of these channels.

This structure additionally allows for effectively instantaneous payments. This is instrumental for not only payments which require a high degree of time sensitivity, but also for contracts as well.

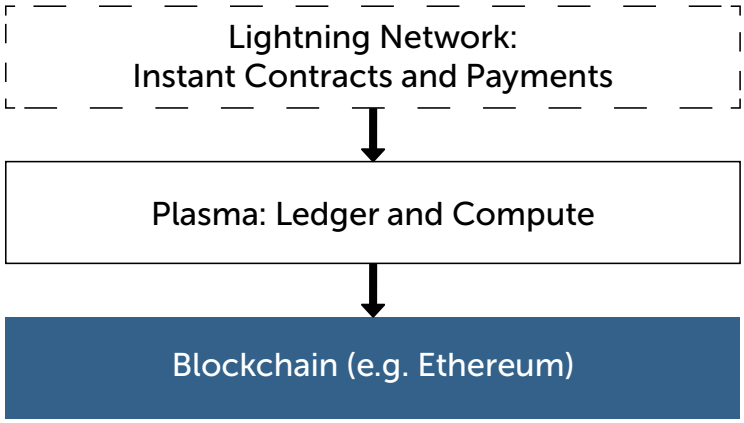
Plasma is not designed to reach assured finality rapidly, even though transactions are confirmed in the child chains rapidly, it requires it to be finalized on the underlying root blockchain. Channels are necessary to be able to have rapid local finality of payments and contracts (enforceable on-chain).

In smart contracts, there is an issue of the "free option problem" whereby the receiver (second or last signer) of a smart contract offer is needed to sign and broadcast the contract to enforce it - during that time the receiver of the contract may treat it as a free option and refuse to sign the contract if the activity does not interest them. This is exacerbated as smart contracts are most effective when dealing with counterparties who are untrusted (as that creates minimization in counterparty risk and thereby information costs).

Plasma does not solve this problem on its own, as there are no guarantees of atomicity with the first and second signature step for interactive protocols in blockchains.

With Lightning (including Lightning on top of Plasma), it's possible to do incredibly rapid updates with reasonable sense of localized finality. Instead of having a single payment which gives optionality to the last party, a payment can instead be split into many small payments. This minimizes the free option to the amount per split fraction. Since the second party of the smart contract only has the free option on the amount in the split fraction, the value of the free option is minimized.

Within the above use cases, it's possible that Lightning may be a primary interface layer for rapid financial payments/contracts on top of Plasma, as Plasma allows for ledger updates with minimal root chain state commitments.



At the root is the blockchain, which is the adjudication layer for contracts and payments. The contracts themselves are located on the root blockchain. The Plasma chain contains the current ledger state which can be settled and redeemed on the root blockchain. Fraud proofs exist to allow for funds to be redeemed. Plasma represents a nested set of Plasma chains to create venues to withdraw funds in a scalable way with minimal blockchain transactions. On top is the Lightning Network, which allows for instantaneous payments across Plasma and Block Chains.

7. LIGHTNING NETWORK: SCALABLE OFF-CHAIN INSTANT PAYMENTS [20]

"If a tree falls in the forest and no one is around to hear it, does it make a sound?"

The above quote questions the relevance of unobserved events - if nobody hears the tree fall, whether it made a sound or not is of no consequence. Similarly, in the blockchain, if only two participants care about an everyday recurring transaction, it's not necessary for all other nodes in the bitcoin network to know about that transaction. It is instead preferable to only have the bare minimum of information on the blockchain. By deferring telling the entire world about every transaction, doing net settlement of their relationship at a later date enables blockchain users to conduct many transactions without bloating up the blockchain or creating trust in a centralized counterparty. An effectively trustless structure can be achieved by using time locks as a component to global consensus.

Currently the solution to micropayments and scalability is to offload the transactions to a custodian, whereby one is trusting third party custodians to hold one's coins and to update balances with other parties. Trusting third parties to hold all of one's funds creates counterparty risk and transaction costs.

Instead, using a network of these micropayment channels, blockchain framework can scale to billions of transactions per day with the computational power available on a modern desktop computer today. Sending many payments inside a given micropayment channel enables one to send large amounts of funds to another party in a decentralized manner. These channels are not a separate trusted network on top of blockchain. They are real blockchain transactions.

Micropayment channels create a relationship between two parties to perpetually update balances, deferring what is broadcast to the blockchain in a single transaction netting out the total balance between those two parties. This permits the financial relationships between two parties to be trustlessly deferred to a later date, without risk of counterparty default. Micropayment channels use real blockchain transactions, only electing to defer the broadcast to the blockchain in such a way that both parties can guarantee their current balance on the blockchain; this is not a trusted overlay network payments in micropayment channels are real blockchain communicated and exchanged off-chain.

The Lightning Network solves these problems [24] [25]. It is one of the first implementations of a multi-party Smart Contract (programmable money) using bitcoin's built-in scripting. The Lightning Network is leading technological development in multiparty financial computations with bitcoin.

Instant Payments. Bitcoin aggregates transactions into blocks spaced ten minutes apart. Payments are widely regarded as secure on bitcoin after confirmation of six blocks, or about one hour. On the Lightning Network, payments don't need block confirmations, and are instant and atomic. Lightning can be used at retail point-of-sale terminals, with user device-to-device transactions, or anywhere instant payments are needed.

Micropayments. New markets can be opened with the possibility of micropayments. Lightning enables one to send funds down to 0.00000001 bitcoin without custodial risk. The bitcoin blockchain currently enforces a minimum output size many hundreds of times higher, and a fixed per-transaction fee which makes micropayments impractical. Lightning allows minimal payments denominated in bitcoin, using actual bitcoin transactions.

Scalability. The bitcoin network will need to support orders of magnitude higher transaction volume to meet demand from automated payments. The rise of the number of internet-connected devices needs a platform for machine-to-machine payments and automated micropayment services. Lightning Network transactions are conducted off the blockchain without delegation of trust and ownership, allowing users to conduct nearly unlimited transactions between other devices.

How it Works. Funds are placed into a two-party, multisignature "channel" bitcoin address. This channel is represented as an entry on the bitcoin public ledger. In order to spend funds from the channel, both parties must agree on the new balance. The current balance is stored as the most recent transaction signed by both parties, spending from the channel address. To make a payment, both parties sign a new exit transaction spending from the channel address. All old exit transactions are invalidated by doing so.

The Lightning Network does not require cooperation from the counterparty to exit the channel. Both parties have the option to unilaterally close the channel, ending their relationship. Since all parties have multiple multisignature channels with many different users on this network, one can send a payment to any other party across this network.

By embedding the payment conditional upon knowledge of a secure cryptographic hash, payments can be made across a network of channels without the need for any party to have unilateral custodial ownership of funds. The Lightning Network enables what was previously not possible with trusted financial systems vulnerable to monopolies—without the need for custodial trust and ownership, participation on the network can be dynamic and open for all.

REFERENCES

1. pri-num.net/Pri-Num's Brochure
2. www.finextra.com/finextra-downloads/newsdocs/us-fsi-making-blockchain-real-for-loyalty-rewards-programs.pdf
3. www.nobelprize.org/nobel_prizes/economic-sciences/laureates/2017/popular-economicsciences2017.pdf
4. www.fastcompany.com/3035723/app-economy/smart-contracts-could-be-cryptocurrencys-killer-app
5. www.blockchaintechnologies.com/blockchain-smart-contracts
6. www.coindesk.com/making-sense-smart-contracts/
7. blogs.lexisnexis.co.uk/futureoflaw/2016/09/what-makes-a-smart-contract-smart/
8. cointelegraph.com/ethereum-for-beginners/what-is-ethereum
9. www.pcworld.com/article/3086211/a-blockchain-smart-contract-could-cost-investors-millions.html
10. dupress.deloitte.com/dup-us-en/focus/signals-for-strategists/using-blockchain-for-smart-contracts.html#endnote-14
11. uk.businessinsider.com/smart-contracts-pose-enforceability-issues-2016-11
12. www.coindesk.com/three-smart-contract-misconceptions/
13. www.ibtimes.co.uk/barclays-gets-into-nuts-bolts-smart-contract-templates-1596874
14. re-publica.com/en/dub16/session/blockchain-smart-contracts-future-law
15. ethereum.org
16. github.com/ethereum/wiki/wiki/White-Paper
17. bravenewcoin.com/assets/Whitepapers/Ethereum-A-Secure-Decentralised-Generalised-Transaction-Ledger-Yellow-Paper.pdf
18. plasma.io/plasma.pdf
19. lightning.network/lightning-network.pdf
20. lightning.network/lightning-network-paper.pdf
21. lightning.network/lightning-network-paper-DRAFT-0.5.pdf
22. lightning.network/lightning-network-presentation-sfbitcoinsocial-2015-05-26.pdf
23. lightning.network/lightning-network-presentation-time-2015-07-06.pdf
24. lightning.network/lightning-network-summary.pdf
25. lightning.network/lightning-network-technical-summary.pdf