Matter Inc. Whitepaper

rev 0.2.1

Motivation

Asset tokenization is a hot topic in a blockchain-community, although at the moment there are economical, technological, regulatory and perceptional roadblocks for tokenization of every possible asset.

The easiest to explain is the economical part. If the blockchain is used for bookkeeping and enforcement of determined rules in the context of the transfer of a value or an asset transfer it brings transaction cost to an order of 1 USD per transaction (based on Ethereum transaction gas limit of 50000 units of gas, gas price 40 GWei per unit of gas and ETH/USD rate of 500 USD/ETH). It's not a problem only for assets with rare high-volume transactions, although such transactions usually happen between high-value institutional traders who already have existing established and well-regulated instruments for such processes.

For asset tokenization to lift-off on a mass scale, one should have a decent plan on what assets should be tokenized first with a full set of underlying infrastructure and technologies in place to guarantee trust into tokenization process and attract more traditional asset holders to the market. In the next sections, we will try to explain these aspects in details.

Tokenization of financial instruments and their derivatives

Here at *Matter* we believe that tokenization of traditional financial instruments and associated money flow is the highest priority and provide enormous benefits for both traditional world and crypto-community. In contrast to other whitepapers, we will try to make as simple example as possible.

Let's have a look at the proposed process of tokenization of the large portfolio of loans. Usually, shares in distinct portfolio are sold to the small (~10) number of large institutional investors that receive a fixed interest and few subordinated investors that bear much higher risk but receive a floating (and much higher) interest.

- In such a portfolio, for instance, every two weeks (time interval may vary) payments are collected from the final borrowers.
- Payments have a part that covers the interest on their loan and a part that covers the body of the loan (principal amount).
- Here we will skip the part regarding the principal and focus on dividends only for a sake of transparency and simplicity.

- Dividends (after some portfolio-related fees) are distributed between the investors (prime and subordinary) in a waterfall process.
- First, the total collected amount of dividends is distributed between primary investors to guarantee them a fixed income. Payments are distributed proportionally to the share of each primary investor.
- If an amount of dividends is greater than a required amount to pay all the fixed income then the remaining amount is distributed ***in full*** between the subordinated investors, thus covering their increased risks with a chance for much higher interest than the fixed rate for primary investors.
- If the amount was not sufficient an interest to primary investors should be either paid from the cash reserves or from the next collected payments. In this case, subordinated investors get zero interest in this payment period.

Although the process is simplified in the above, it illustrates the process of ***origination*** of a potential tokenized asset as a smart-contract could do the bookkeeping of the original shares in a portfolio and make all the calculations necessary to comply with the mentioned waterfall process, and also introduces a derivative of such portfolio - a crypto-equivalent of the dividends, that is, on the one side, there is a identifying number in the smart-contract, and on the other side there are payments expressed in fiat currency collected and stored on a bank account. Here we see two separate events that should be covered in details.

Smart-asset origination

In the example above the first step is a creation of the portfolio equivalent in a form of a smart-contract that does a bookkeeping and covers processes that can be efficiently executed (with reasonable computation costs) in existing smart-contract capable blockchains. One of the limitation in such blockchains in the maximum amount of computations per transaction (e. g. in Ethereum such limit is equivalent to the block gas limit) and if, for example, dividends calculation process is more complicated than one explained above, such computations are just not (efficiently) possible in a smart-contract. Such problems can be solved either by modern cryptography in a form of zkSNARKs/zkSTARKs or a cryptoeconomical approach that will be covered in the corresponding sections. At the moment it’s enough to illustrate two facts:

- it may be trivial to create a smart-contract equivalent of some asset, a full support of the associated business functions can be either expensive or not possible.
- tokenization of the asset gives rise to the derivative (sibling) assets that can be much more functional and interesting for a market and community.

Smart-asset derivatives

The above example of tokenization of the loans gives rise to the very interesting and practical instrument - a crypto-asset that is one-to-one pegged by traditional fiat currency. Economically this is equivalent to a long-though “stablecoin” with low volatility (due to pegging) and high-

liquidity (due to low volatility and trivial price-derivation rules). Such assets can have practical application and can be much more interesting for a crypto-community as they make a bridge between physical and the crypto worlds. All the bookkeeping for such derivative asset can and should be done on a blockchain, while physical world players can be much more willing to accept such a form of payment due to much-reduced volatility risk.

One can go one step further and bring extra benefits to such a derivative asset by introducing a cash-out procedure (which would be backed by the fiat currency that in turn covers the full emitted value of this asset) and making the transactions costs virtually non-existent. The first part will require a KYC/AML procedure that is described in the corresponding section. The latter one is covered by implementation of the Plasma protocol.

In summary, creating an instrument with high liquidity is largely linked to underlying economics (areas of macroeconomics and monetary policy) and in this section, we try to give an easy explainer what can be done and how such tokenization process can be beneficial to many sides.

Open certificates standard

At the moment when blockchain operates with entities that have direct real-world value, it gives rise to all the requirements from traditional financial regulators and institutions in the area of KYC/AML procedures.
In practice, companies such as cryptocurrency trading platforms adjust they procedures according to the law of countries where they are registered. We generally think that such a practice is right and reasonable. However, we should emphasize that such restrictions should only be enforced in points of junction between traditional and crypto-assets, while trading cryptocurrencies and assets between themselves should be non-restricted. We believe that recent personal data leaks have confirmed the need to make sure that the:

- Right to dispose of personal data should belong to the owner of personal data and only to them.
- Right to access to personal data should be granted only to the services authorized by the personal data owner.

Our ambition is to bring best practice from the area of cryptography and data structures to improve existing solutions. Similarly to other projects, we want to make sure that issuance of certificates(the “Certificates”) should do authorized by entities that already been authorized to undertake KYC procedures in the real world.
We are going to provide the format and the description how to work with digital snapshots of Certificates on the blockchain as well as smart contracts that will store them. We think that blockchain technology is applicable and necessary for the following reasons:

- First of all the right should be granted by the mathematics and not by the opinions of particular people.

- The digital snapshot of user data can be stored on immutable public storage - blockchain. That will allow avoiding censorship and fraud. Denial of Service for a person with particular digital snapshot can always publicly be challenged. Right to access the service and manage own personal data will be granted by math.
- Feasibility of only partial disclosure of the data by owner decision.
- With help of zero-knowledge proofs, it's possible to grant access without knowledge of full user identity and only by knowledge of user access level. As an example, it's applicable when we want to provide a service only if customer has sufficient amount of funds but at the same time we can't ask customer to provide exact amout of funds on his account.
- Certificate owner in case of suspicion of unauthorized access to his certificate can revoke it by his own without requesting any specific service at any time.
- Replay attack protection is necessary in the modern world.

Our solution gives the opportunity to the owner to choose where his or her data will be stored - s/he can choose to store s/he data on s/he own or delegate that to someone by s/he will. In general user experience of working with personal data can be improved for the all involved parties.

Since smart-contract is the crucial part of suggested KYC workflow, we can tightly integrate it with our Plasma implementation. One example would be a KYC based whitelist process to withdraw funds from the Plasma network to the parent blockchain for any further purposes such as transfers or cash-outs. Such logic is similar to the existing procedure for crypto-fiat pairs trading on centralized exchanges.

Whether building a KYC standard is economically profitable for us or not is beyond the scope of this paper. Nevertheless, it's an integral part of our proposed system and larger asset-tokenization ecosystem in general.

Plasma protocol implementation

The Plasma protocol was originally introduced by Vitalik Buterin and Joseph Poon in August 2017. The most recent paper can be found on the corresponding web page (https://plasma.io).

Plasma protocol introduces an idea and defines necessary parts to allow the shift of trust and grant a permission to the central party to produce block in a verifiable way. There are various ways how one can build a particular Plasma implementation (sometimes referred as Minimal Viable Plasma, Plasma Cash, More Viable Plasma, Plasma Debit, Plasma XT, etc.) and it's clear that throughput of 10k+ transactions per second can be achieved. Such an enormous transaction speed and degree of centralization that allows bringing the cost per transaction below 1 cent completely closes the need of the cryptocommunity in value (transfers of fungible assets) and asset (transfer of non-fungible assets) transfers. One should understand that smart-contracts are not possible in Plasma at the current level, but a verifiable nature of it allows few interesting ways of conjecture with the smart-contracts functionality.

For the further explanation we should mention two cornerstones of Plasma protocol:

- Verifiability: external observers that do not have a permission to produce blocks should be able to punish a centralized operator for production of invalid blocks. If an operator allows a trivial double-spend -> an external observer should be able to punish an operator by providing a necessary proof. For such purposes there exists a smart-contract resolving such disputes automatically and some form of the security deposit is required from the operator.
- Liveness: there is always someone monitoring the chain for byzantine behavior. It can be either large external observers monitoring the full chain (remember, 10k transactions per second) or small users, each monitoring his/her own transactions and a small random part of the chain.

To summarize, in addition to a protocol-level correctness a particular implementation should offer some incentives mechanism to attract and engage large players and small users into chain monitoring. Discussion of such mechanisms is beyond the scope of this work, although the easiest one is a liquidity provision mechanism on transactions between Plasma and it's parent blockchain (where the smart-contract is).

Implementation of the Plasma protocol will provide an economically efficient mechanism for tokenized assets trading and, even more important, transfer of the derivative assets. In our proposal, a particular Plasma implementation will not only bring low transaction costs but also provide mechanisms described in the following sections.

Decentralized exchange (DEX)

One obvious application of Plasma is to build an efficient exchange. Implementation of the decentralized exchange using Plasma protocol will require some trade-offs, at least at the initial stages. Every transaction in a blockchain requires an owner's signature (an explicit consent to trade) and simultaneously there should exist a simple and efficient proof that trade was either correct or incorrect (to punish an operator in the latter case). Unfortunately, it's not trivial to provide such an efficient proof for matching mechanism used by traditional exchanges and users will have to explicitly agree of a volume and price of the transaction (or leave an offer that will be filled by first come - first served basis). Nevertheless, such a mechanism is already used by decentralized exchanges in Ethereum network and price reduction per one trade will attract more users to such decentralized solutions.

Conjunction with a supervising smart-contract

Any Plasma implementation requires a complicated smart-contract to ensure correctness and automatical dispute resolution. One can bring additional restrictions to this smart-contract, such as limiting access to the trading of particular assets of instruments only to some group of users. While a necessity for such a mechanism will be discussed below, first we should give an example of how it can be implemented:

- There are observers monitoring the Plasma chain.
- There is a smart-contract with access registry for every Ethereum address (for explanation purposes we use Ethereum network as a parent chain for a Plasma implementation) and corresponding permissions. This can be viewed as some form of voluntary KYC.
- If a centralized Plasma operator adds a transaction on a restricted asset from the user that did not have a required permission by that time, an observer can send a transaction that will prove such fact and the smart-contract will automatically slash a security deposit from the operator and potentially halt the Plasma chain.

One should understand, that our philosophy of Plasma implementation is based on a goodwill and complete freedom for users. Any asset that does not have any physical world equivalent and can not be easily cashed-out (without some form of KYC/AML procedure) should be transferred in Plasma without restrictions, thus allowing users in cryptocommunity to transact freely and with low fees. On the other side, any asset that can easily be converted to fiat currency should be traded only by users who voluntary undergone the KYC procedure. While such a requirement can be seen like a limitation, we think that it's a moral solution to prevent money laundering and illegal trading.

One clear benefit of such restriction is the attraction of attention of large traditional institutional players and regulators. Any of such entities can become a full chain observer and ensure correctness and chain validity and liveness for other users. Simultaneously it will also bridge the gap to the physical world if at some point a fully-pegged tokenized assets derivative will emerge in a market and will have high liquidity and will be used for means of everyday payments.

Further improvements of a Plasma implementation

As was explained above there are various ways how one can build a Plasma protocol implementation and what features will be accessible for users. Here we would like to mention just two: one for a short-term and one for a long-term development:

- In the short term, one can implement "Confidential transactions" in Plasma. This feature refers to an implementation of the protocol where users can hide the exact transaction amount, but still prove the correctness of a transaction by proving that there was not value created out of thin air by it. Various particular implementation exists for such feature, for example in Monero and zCash blockchains, and in our implementation, we believe it should be based on original "Confidential transactions" protocol by Greg Maxwell. One should understand that such functionality can be already implemented in Ethereum network, for example, although transaction price becomes very large (gas amount of few millions is required per one transaction. For comparison a normal transaction requires 21000 gas).
- In a long-term, one can try to use modern cryptographical advances, such as zkSNARKs/zkSTARKs to prove a complete validity of a block and reduce a full chain

monitoring burden for users. Although it's highly experimental and further research is necessary to achieve that ultimate goal. A brief explanation of zkSNARKs/zkSTARKs can be found in the corresponding section.

Cryptoeconomics

In this section, we will try to explain what is cryptoeconomics and how it requires a certain paradigm shift in a model of thoughts.

When smart-contracts were implemented in a blockchain (mostly we mean the main Ethereum network) various attempts were made to transfer a traditional transaction logic to those contracts. One of the best examples is ERC20 token standard that defines a ledger for some arbitrary denominated fungible asset. Unfortunately, there are transactions with associated computations where the computation cost can be high or even too high (above the maximum possible limit per transaction). To implement such procedures in a blockchain one can either switch to implementation of provable (and efficiently verifiable) computations, where any user can prove the correct relation between some input and output data and do it in efficient space and computation costs (much lower than computation cost itself), or one can introduce a cryptoeconomic procedure.

In cryptoeconomic procedure users do not need to prove a correctness of some input and output data, they just state this data is correct and provide some collateral and time limit for some interactive or non-interactive challenge procedure. One should understand that not every procedure can be switched to cryptoeconomical paradigm, although it's clearly beneficial for all users of the ecosystem. Here is a simple example:

- Alice wants to do a transaction to Bob based on some input data I and output data of interest O.
- There exists some (interactive or not) procedure that allows to deterministically prove that output data O is a correct value for input data I.
- Alice computes O from I locally (on a home PC or a cellular device) and sends a transaction to the smart-contract that states the following:
  - Alice has based her computation on the input data I
  - Alice has computed an output O
  - Alice ***states*** that O is a valid output for I
  - Alice is willing to support her statement by some collateral X
  - Value of X is higher than the cost of proving procedure described above
  - X is transferred to the contract immediately and can be redeemed by Alice in Y hours
- During this period of Y hours, any observer (Bob) can verify that O is a valid output for I and if finds a discrepancy he/she can engage into dispute procedure with Alice and if the discrepancy is proved Bob takes collateral X and Alice's transaction is deemed invalid.

To ensure the effectiveness of cryptoeconimics a high “liveness” is required in the network with as many as possible users monitoring as many transactions are possible to start the dispute.

Here is when Matter token comes into play. If *Matter* token is used to pay collateral, we can introduce an additional restriction.

- If some user Bob successfully disputed Alice’s transaction he receives the collateral as usual
- *Matter* takes a responsibility to dispute an invalid transaction even if no one is observing the network
- If no one disputes a transaction from Alice during an allowed time period (including us) and at some later point in time Alice’s transaction is still proved to be invalid, the prover of this fact will slash a security deposit from *Matter* that is made for such exceptional situation

By taking an additional burden we try to bring liveness to the network by an economic mechanism. To avoid trivial forms of cheating that “we ourselves prove that we missed a transaction” and take back the security deposit only half of it will be paid to the prover with another half just destroyed.

Direct application of the *Matter Inc.* token

We have a strong position against discriminative access to any product in our ecosystem, although propose the following application of our token for certain procedures:

- Use a token as a mean of fee payment in our Plasma transfer. If a large number of assets with a large volatility is included in our Plasma implementation, we either will have to implement a sophisticated monitoring systems that provide adaptive fees to be paid in a native denomination of the transferred asset (for example, some ERC20 tokens), or use our own token for such fees collection.
- Issuance of KYC certificate requires an interaction with a provider in one (or more) jurisdiction and a payment for a procedure can be paid in different ways.
- If a cash-out procedure is triggered for some derivative assets, a certain commission should be paid either in *Matter Inc.* tokens or the native denomination of the asset.

Brief overview of zkSNARKs/zkSTARKs

One can already be familiar with zkSNARKs and their practical use by zCash blockchain for purposes of anonymous transactions. While it’s an acronym of *zero-knowledge succinct non-interactive argument of knowledge* one can try to explain it in the following form of statements (and Alice and Bob are here again):

- Alice wants to prove that she has correctly done some computation
- Alice wants to convince Bob in it with a high probability (astronomically close to 1)
- While the amount of computations that Alice does can be substantial, she only provides a small amount of all the results involved in a computation process (a full set is usually called

the “witness”, while Alice only discloses some “public” parameters)
- Alice and Bob agree on the nature of computations (up to the last sign!)
- To achieve her goal Alice prepares and provides a proof based on a zkSNARK protocol
- As a bonus, Bob does not learn anything about any intermediate results of her computations. It’s not too important for our purposes, but can be handy in some cases
- Alice’s proof size is always constant and known beforehand, and verification cost is always constant and known beforehand

As one can see utilization of zkSNARKs efficient mainly due to the last sentence about constant size proofs and constant verification costs. Such procedure can be used instead of cryptoeconomical approach for some complicated computations during a transaction at the expense of much higher proof preparation costs than just computing output O from the corresponding input I. It can even be mixed with cryptoeconomics when during the transaction the proof is not checked but only recorded for further dispute resolution purposes!

Conclusion

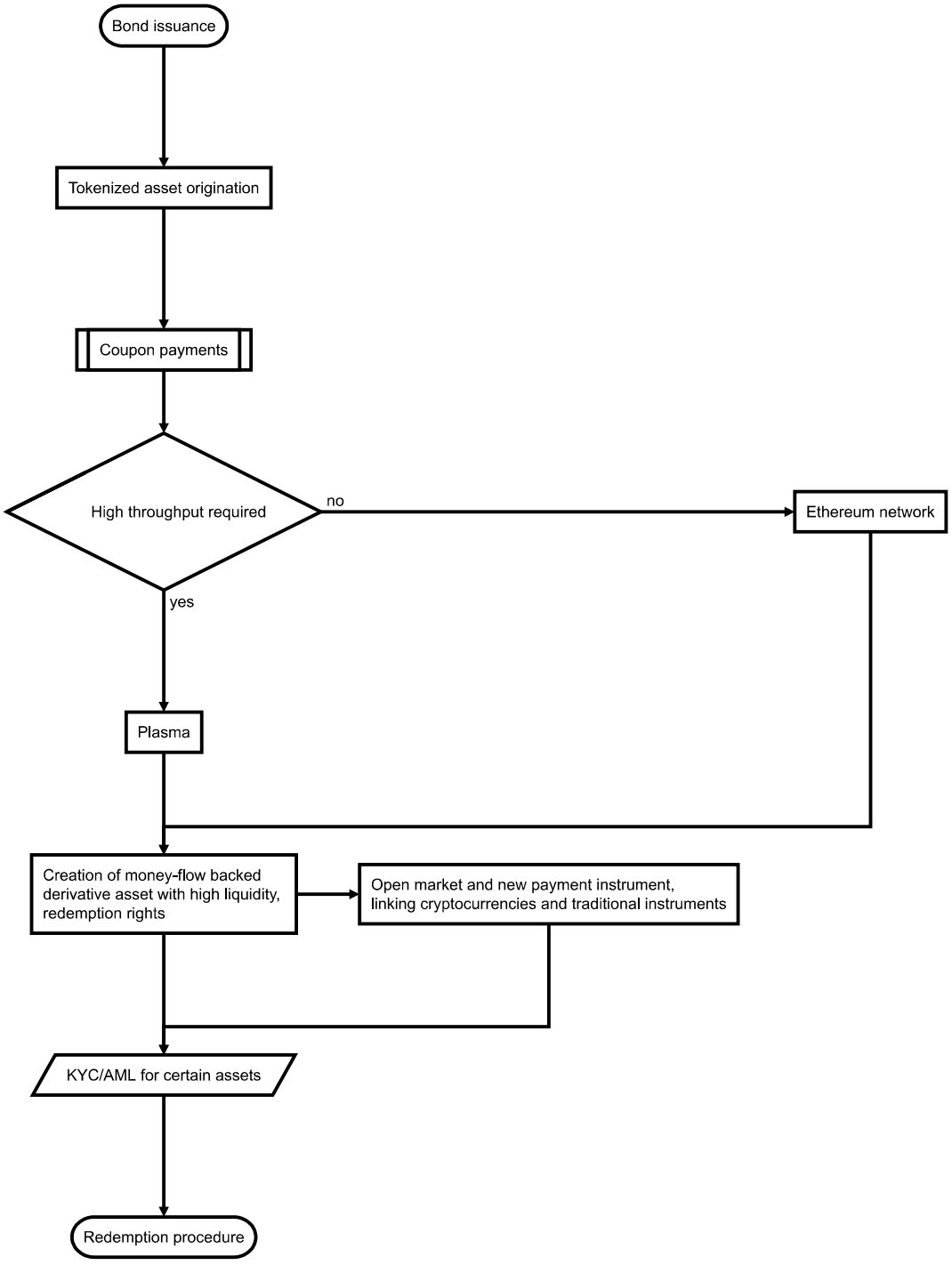
Instead of a long conclusion or a call to action “buy our tokens” we will try to explain our goal.

Matter is a team of engineers mainly, and we believe that systems (crypto-economical too!) should be self-consistent, economically reasonable, practical and have great UX. While many companies try to get involved in the asset tokenization or fast blockchain implementation, we are much more humble and try to build a system, where a Plasma, that has a huge throughput, can be filled with a number of transactions that will utilize this throughput, and do to it we would like to tokenize some classes of assets first. These classes will bring examples and trust to tokenization process and (we hope for it) give rise to the new generation of “stablecoins” with transparent pricing, high liquidity and apparent linkage to the real physical world by a proper cash-out mechanism. To do it we have presented you our ideas in this short (not 300 pages) whitepaper, tried to make all those explanations as easy and trivial as possible, and from the practical side also have experience in both Plasma implementation and a pilot of a portfolio tokenization -otherwise we would never be able to quickly explain it to you.

Alex Vlasov, Artem Vorobev

Appendix

Example of proposed bond tokenization flow chart



Matter Inc. Plasma implementation structure