GEWEL
NEW AGE CRYPTOCURRENCY

++

White Paper

A Leading Edge Cryptocurrency For A New Monetary System

www.gewel.io

support@gewel.io

Table of Contents

Disclaimer

All cryptographic tokens referred to, in this white paper, refer to cryptographic tokens on a launched blockchain that adopts the Gewel.io software.

This Gewel.io technical white paper is for information purposes only. All Gewel.io documentation, designs and sources are still in the research, developmental and conceptual phase—which are subject to change. Gewel, Inc. does not guarantee the accuracy of or the conclusions reached in this white paper.

Gewel, Inc. does not make, and expressly disclaims, all representations and warranties, express, implied, statutory or otherwise, whatsoever, including, but not limited to:

(i) warranties of merchantability, fitness for a particular purpose, suitability, usage, title or non-infringement

(ii) The contents of this white paper are free from error

(iii) The aforesaid contents will not infringe third-party rights.

Gewel, Inc. and its affiliates shall have no liability for damages of any kind arising out of the use, reference to, or reliance on this white paper or any of the content contained herein— even if advised of the possibility of such damages. In no event will Gewel, Inc. or its affiliates be liable to any person or entity for any damages, losses, liabilities, costs or expenses of any kind, whether direct or indirect, consequential, compensatory, incidental, actual, exemplary, punitive or special for the use of, reference to, or reliance on this white paper or any of the content contained herein, including, without limitation, any loss of business, revenues, profits, data, use, goodwill or other intangible losses.

Introduction

Over the past decades, mankind has made great technological advancements. One of the aforementioned advancements have included the revolutionary step towards an electronic monetary system—which is exactly what Gewel.io aims to be. It is envisioned that over time, new features will be developed internally as well as sourced from the crypto community at large; in order to further enhance Gewel. As new innovations become available, the Gewel.io users are the ones who decide which features can be implemented to improve the utility of the Gewel.io software. Therefore, not only is Gewel.io opening a new chapter in cryptocurrency, but is also allowing its users to partake in what is meant to shape e-currency.

Rather than being static and rigid, the idea of Gewel.io is for the blockchain software to evolve organically through the help of its community. To this end, Gewel, Inc. will look to raise awareness as well as attract talented and likeminded developers, from all over the globe, to help grow, develop and innovate the software.

While much progress has already been achieved Gewel, Inc. intends to speed up development of the Gewel.io blockchain software privacy feature through the integration of a zero-knowledge security layer (zsl). ZSL in Gewel.io will be designed to anonymously settle transactions on the blockchain.

Gewel.io will always endeavor to evolve and adapt by being nimble, open and flexible to new opportunities that may arise. With that in mind the Gewel.io governance system has been setup to quickly adopt and take advantage of new opportunities as they become apparent; without having to deal with some of the decentralized decision-making problems— which are clearly evident with other blockchain software.

The ultimate end-goal of Gewel.io is simple. It is to become the most useful and user-friendly blockchain software for the end user, in order to enable every day and international transactions.

Blockchain & Consensus Algorithms

Distributed consensus, i.e. getting multiple computers to agree on something, is a notoriously difficult problem. The establishment of a digital currency requires this, for all users of the currency must have a way to agree on their wallet balances; in addition to agreeing on the entire transaction history. In practice, a lot of applications lend themselves well to having a Trusted Third Party to solve the problem of consensus. However, the aforementioned procedure is not applicable to digital currencies, as no party can be trusted. The most prominent solution to constructing a digital currency, that has no trust assumptions, is blockchain— along with a Proof of Work or Proof of Stake consensus mechanism.

To be more explicit, a block is a set of transactions, and a blockchain is a tree of blocks representing an entire transaction history. For a blockchain to be useful, the history needs to be eventually consistent among participants, which is to say that one should expect, with very high probability, no bifurcations in the history when going back in time by a fixed amount. In order to achieve this, cryptocurrencies employ consensus algorithms such as Proof of Work or more recently, Proof of Stake.

Problem

Proof of Work is a consensus algorithm that requires nodes to compete in solving an intractable problem; which can only be solved via brute force. By design, the difficulty of the problem to solve will depend on the amount of compute power available in the network. This difficulty is chosen so that the expected time for some node, to solve the problem, is 10 minutes. Nodes with more compute power will tend to be first to solve the problem as well. Thus, this means that those nodes will be the winners of a reward, (the so-called block reward), more often. The biggest issue with proof of work is that as the number of participants increases, the difficulty (of the network) increases— which means that the amount of energy required in order to mine a block increases. For example, as of June 2019, the Bitcoin network, which uses Proof of Work, consumed a total of 70+ Terawatt-hours per year— or enough energy to power the entire Czech Republic. This fact makes Bitcoin wasteful and unsustainable in the long term. That being said, some of the problems with Bitcoin can be summarized into 4 key points:

a) Extreme energy requirements.

b) Low scalability by design— as the expected value of the time to mine a block is fixed at 10 minutes.

c) High costs demanded by miners in terms of infrastructure and overhead.

d) Centralization as a result of miners using ASICs designed to mine the cryptocurrency at speeds unattainable by conventional consumer-grade machines.

The most significant drawback of such consensus is its lack of finality. Finality means once a transaction or an action is performed on the blockchain, it is permanently recorded on the blockchain and impossible to reverse. This is vital to the safety of financial settlement systems as transactions must not be reserved once they are made. In Bitcoin's case, malicious actors can tamper with the transaction history given enough hash power, causing a double-spending attack, provided that there is enough incentive and financial viability to carry out such attacks. Given that mining equipment renting and botnets are currently prevalent world-wide, such an attack has become feasible.

Due to this lack of finality, Bitcoin consensus must rely on extra measures, such as proof-of-work to prevent malicious activities. This impedes the ability of this consensus to scale because a transaction must wait for multiple confirmations before reaching "probabilistic finality". Therefore, safety is not guaranteed by such consensus, and in order to protect the network, each transaction must undergo additional time to process. In Bitcoin's case, a transaction is not considered final until at least six confirmations. Since Bitcoin can only process a few transactions per second, the transaction cost is outrageously high, making it impractical for small payments like grocery shopping or restaurant dining. This greatly hinders Bitcoin's use as a payment method in the real world and on a massive scale.

Attempted Solutions

Many distributed ledger projects have attempted to solve the challenges Bitcoin faces. However, these solutions have yet to resolve the so-called blockchain trilemma: to preserve speed, security, and decentralization at the same time. Often, these solutions must come down to a trade-off among these three vectors.

Direct Methods

The easiest way to solve the transaction speed problem is to reduce the time taken to generate a block. Ethereum generates a new block every 12 seconds. However, as Ethereum is also based on the Bitcoin consensus, it requires more confirmations, usually over 50, for transactions to gain probabilistic finality. Bitcoin Cash introduced bigger block sizes to include more transactions in a single block, but this does not reduce the confirmation time of blocks and can cause network connection problems.

Proof of Stake “PoS”

These shortcomings, along with several others, played a motivational role in the development of superior consensus algorithms such as Proof of Stake. Proof of Stake is a consensus algorithm in which, rather than depending on sheer compute power, participants in the network participate by staking their tokens. In the Proof of Work setting, one need not own any amount at all of the cryptocurrency in order to participate in consensus. Whereas, in Proof of Stake, only participants that own a certain amount of the cryptocurrency may participate. This has a few advantages:

a) It is in the interests of the participants to act in the interests of the network.

b) Malicious actors can have their staked cryptocurrencies “slashed”, meaning dishonesty and adversarial behavior come at a real cost.

On top of this, Proof of Stake does not come with the energy costs and requirements associated with Proof of Work. Hardware requirements associated with a single participant in the Proof of Stake algorithm are fair. One can run node software on a consumer-grade device such as a laptop— where the only requirement is a highly available network connection.

In this setting, for each block, the participants staking their currency are called the validators, and they are said to be validating the block. Among these validators, a single participant will be selected pseudo-randomly— using various characteristics of the state of the blockchain; such as hashes associated with the transaction history, the age of the coins at stake and the amounts at stake. In turn, the participant will become the forger. Therefore, the chosen forger will be the one to create the next block, and he is incentivized to act in the interests of the network. In a typical setting, if a participant owns 1% of the cryptocurrency, they will be, with high probability, chosen 1% of the time.

In addition, it’s important to remark that the so-called 51% attack associated with PoS is significantly harder to achieve since it would require a single participant or a sufficiently large subset of colluding participants to own over 50% of the entire currency.

Gewel

Gewel’s proprietary consensus algorithm overcomes disadvantages of the prior art by providing a distributed transaction system including a group of validator nodes that are known to each other in a network but are indistinguishable to the other network nodes in the network. As used herein, the group of validator nodes may be referred to as a “Committee” of validator nodes. In some embodiments, the system reconfigures one or more validator nodes in the Committee based on the results of proof-of-stake (PoS) challenges. According to some disclosed embodiments, a network node that is not already a validator node in the Committee may be added to the Committee if it successfully completes a POS challenge. In such an event, the network node may become a new validator node in the Committee, replacing an existing validator node.

Gewel, on the other hand, runs its proprietary consensus, anchored by its independent algorithm, and can authentically offer instant finality for its network users. With its algorithm design, the consensus runtime lasts only 50-60 milliseconds (ms). Two-to-three confirmations are all that is required to permanently accept a proposed block into the blockchain, and it only takes 90ms for these confirmations to transpire, making the process significantly faster than any other. Gewel’s consensus does not need to choose between responsiveness and linearity.

Gewel’s dual blockchain structure includes the speeds of a DAG, but its recall for users can happen much simpler and faster, which adds to the availability of transaction information and makes the information more decentralized.

In accordance with some disclosed embodiments, the validator nodes in the Committee may receive transaction requests from other network nodes, for example, in a P2P network. The Committee may include at least one validator node that serves as a “Leader” validator node; the other validator nodes may be referred to as “Associate” validator nodes. The Leader node may be changed periodically, on demand, or sporadically by the members of the Committee. When any validator node receives a new transaction request from a non-validator node in the network, the transaction request may be forwarded to all of the validator nodes in the Committee. Further to the disclosed embodiments, the Leader node coordinates with the other Associate validator nodes to reach a consensus of a disposition (e.g., accept or reject) for a transaction block containing the transaction request and broadcasts the consensus to the entire P2P network. If the consensus is to accept or otherwise validate the transaction request, the requested transaction may be added in a new block of a blockchain that is known to at least some of the network nodes in the network.

Advantageously, the disclosed embodiments provide a distributed-system architecture and related protocols that allow a distributed system, such as a blockchain system, to scale up without incurring an unacceptable increase in decision-making complexity while also preserving the benefit of using a decentralized system. The disclosed embodiments reduce the distributed system's reliance on the stability of any particular node(s), as the validator nodes in the Committee may be changed at a sufficient frequency to remove unreliable, unavailable, or otherwise untrusted nodes. Further, the system and method of the disclosed embodiments provides a scheme that helps ensure the Leader node, as well as the other Committee members, functions properly.

It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the disclosed embodiments or the scope of the inventions as claimed. The concepts in this application may be employed in other embodiments without departing from the scope of the inventions.

Network and Procedures

For blockchain software to operate, full nodes are required. A full node is a computer committing resources to run blockchain software and to honor an agreed upon protocol. Multiple nodes are required in order for the operation to be useful and meaningful. These nodes form a homogeneous Peer-to-Peer network, and send updates to each other in response to events which take place on the network. For a typical cryptocurrency to be effective, a large number of nodes are required and nodes must be able to cope with high volumes of traffic. This means that node operators may shoulder the burden of the high costs associated with the hardware requirements. However, since Gewel is built upon the Proof of Stake concept, the requirements are not as stringent. Cryptocurrencies built upon a Proof of Work concept have witnessed major failures in performance, with block propagation times exceeding 40 seconds. Proof of Stake would allow us to mitigate this problem.

With sustainable and efficient network capabilities, the emphasis is then shifted toward Inodes. An Inode is a full node running Gewel blockchain software (i.e., the aforementioned Leaders, Associates and other validator members of the network). The fundamental role of an Inode is to relay and validate transactions. As such, an Inode requires only two protocol messages to become active on the Gewel network. These two messages are the Inode ANNOUNCE message and the Inode PING message.

The Inode ANNOUNCE protocol message is used to announce the presence of the node, on the network, when it initially starts up. Following this, the PING protocol message will be sent periodically, where the period must be reasonable so as to not overload the network, as part of the proof of service procedure. If this is not carried out at a satisfactory rate, the node will eventually be deactivated. Additionally, the node must have its ports open ready to receive pings, if not, it is deemed to be inactive.

Irrespective of the above mentioned detailing, in practice, the Inode function is more complex and relies upon additional protocols— including private send and instant send; which are both used to ascertain proof of service.

Finding Active Inodes & Inode Arrangements

If the Gewel blockchain software is to be successful, new users must be able to understand quickly and easily which Inodes are currently active. To achieve this, users are sent a known list of Inodes and their status whenever they join the network. Following the receipt of this list, the users cache the list so that when they restart they can just look up the directory; rather than having to ask for the whole list again too often.

As for the Inode arrangement, an algorithm based on the block hash, staked token ages, and wealth of participants is used to order Inodes deterministically and pseudo-randomly.

Transaction Speed & Scalability

By implementing 150-second block times, and an Inode structure that is capable of supporting up to 20-megabyte blocks, Gewel.io provides 80x bitcoin transaction capacity. As a result, Gewel.io ensures transaction fees are kept low. This also implies that the Gewel.io software will have no near-term capacity issues— which, in return, enables the Gewel.io team to research and implement further capacity improvements over time.

How Inodes Are Incentivized

In the Proof of Stake setting, Inodes are incentivized via transaction fees only as the cryptocurrency is neither mintable nor burnable. However, a potential Inode reward program is on our roadmap. This program serves to incentivize the network further, growing the number of full nodes that are operational at any one time.

Individual Inodes are tied to a specific level of service, and this is secured by collateral. This collateral remains protected while the Inode remains in operation—thus, encouraging stability across the network while giving Inode operators the opportunity to earn additional Gewel.io tokens.

Network Checks & Balances

The role of an Inode extends a little beyond relaying and validating transactions. There are other, highly important, benefits which these nodes can provide. As such, it is critical for the network to be able to assess which Inodes are online, which ones are actively responding and which ones are behaving in line with the code of service they have agreed to. This requires proof of service. It is not enough simply to assess which Inodes are functioning, as a certain level of service must also be attained. To ensure this level of service, the Inode network uses quorums; two quorums are chosen for each block; titled quorum A and B, respectively. Quorum A, assesses the service level of quorum B. On one hand, Quorum A is made up of the nodes closest to the hash of the current block. Quorum B, on the other hand, is made up of nodes furthest away from this hash. Using the aforementioned method, the system is kept trustless by randomly selecting nodes through the quorum system which makes the network self-assessing.

Collateral System

It is critical for the security of the system that no one individual, or body, gains control of the entire network of Inodes. To achieve this, the abovementioned collateral system will be implemented. Anyone who wishes to control an active Inode must put down 10,000 Gewel.io tokens as a deposit. These 10,000 Gewel.io tokens will essentially be the stake in the Proof-of-Stake protocol.

Given that Gewel.io tokens will be limited in supply, it means that the price of Gewel.io tokens respond directly and quickly to demand. To gain control of a significant section of the Inodes on the network, the individual would need to purchase a vast amount of Gewel.io tokens units from the open market; hence, driving up the price and making it impossible for them to achieve their goal.

This enables the overall Gewel.io software to utilize the iNode network to carry out sensitive tasks. No trust or overall responsibility is awarded to any one node or group of nodes, and so no one is able to control the network for their own ends.

An inode becomes active when a user sends 10,000 Gewel.io tokens units to a designated address in a designated wallet on the network. Once this action is confirmed, the node will be able to use the ANNOUNCE and PING protocol messages to introduce itself into the network.

The network includes in-built security measures by using a dry mode to prevent malicious activity on the system. For example, if an Inode sends the private key in a message following activation and this is used on a second machine, the system will deactivate the original node, protecting the 10,000 Gewel.io tokens units from theft.

The 10,000 Gewel.io token collateral need not be stored in the actual Onode wallet but rather can be stored remotely in a safe location to prevent the Gewel.io token collateral from being stolen.

Remuneration & Implementation

The Gewel.io software is set up to ensure that each inode receives its due share of the transaction fees. This requires the network to enforce payments between the block in question and the correct inode; which in turn, requires conscientious behavior and practice from the participating validators. If the validator is not able to uphold these standards, the blocks they process will be rejected by the network, in order to discourage cheating.

However, this must be enforced. To achieve this enforcement, inodes create quorums then broadcast their choice of the correct inode; the one which must be paid. The process is completely decentralized and trustless, so there is no way that inodes can collude on the matter and defraud the system. Once a certain number of messages have been received, a voting consensus can be reached and the block will be obligated to pay the chosen Inode.

Reward Program

In our potential reward program, the payments would be derived from the block rewards accrued on the network, with around 45% of the total paid out in tokens to inodes who uphold a certain level of service. Payment per day of inode operation would be calculated thusly;

$n \times r \times s \div t$ in which

$n =$ blocks per day

$r =$ block reward

$s =$ inode share of each block reward

$t =$ total number of inodes

For calculating the reward payments on an inode, the following formula is used, based on the same definitions as below;

$$\text{Rate of return per inode} = (n \times r \times s) \div t \times 365.25 \div 10{,}000.$$

Here, 10,000 is the amount of Gewel.io tokens collateral required to operate an inode.

Governance

Governance & Funding Problem

The conundrum of governance is a tricky one for developers of a blockchain software network to resolve. On the one hand, the network needs to be effective, with decisions taken quickly and effectively to ensure positive development in the short and long term. On the other hand, the decentralized nature of the blockchain software should be protected.

This requires a structured governance system; something which the Gewel.io software implements via a system of self-governance.

Solution

Self-Governance

Self-governance is the governance solution that Gewel.io software uses to allow for quick decision making in a decentralized network. Rather than debating options and decision possibilities, self-governance provides rules which allow for quick resolutions. Taking bitcoin as an example, a debate on this network regarding the size of an individual block has taken years to resolve, while in the Gewel.io software, such a question can be voted on and put to rest in a matter of hours. The result is a far more efficient network.

The self-governance system requires proposals to be put forward to the network as a whole, and then voted on. In practice, this means that inodes are able to cast a vote on important changes to the system or network— as no one can assume control of too many inodes, dominance of the vote is not possible.

Self-Funding

Included in the self-governance system is the way in which block rewards are utilized to provide ongoing funding to the network. For each block that is validated, the participant chosen among the pool of validators for that block receives 20% of the reward— while another 45% is dedicated to the operational cost of the physical inode; which leaves 35% remaining. This 35% is not created until the end of the month. During the month, anyone can make a budget proposal, which is voted on by the network. At the end of the month treasury blocks are created and if 10% of inodes vote for any proposal then that proposal is approved. If no proposals are approved or the 35% of the reward amount is more than needed to cover the cost of the proposal, the excess reward goes to the treasury and is available for funding future proposals. This system allows the network to fund itself and also provides the opportunity to build up assets in the form of Gewel.io tokens which can be used to fund future as well as potentially larger proposals.

Privacy

Fungibilty Problem & Solution

With Bitcoin, it has become apparent that transactions aren't fully private. This leads to a fungibility issue. fungibility just means that my Bitcoin is worth exactly the same amount as yours as they are perfect substitutes. However, by analyzing the public ledger third parties are able to link Bitcoin transactions to people's identities. This can lead to Bitcoins being tainted due to their unfavorable past histories. Gewel.io software will integrate a zero-knowledge security layer (zsl) on top of the Gewel.io software in order to provide users superior transaction privacy and solve the fungibility issue. Our protocol of choice will be the ZeroCash protocol, which makes heavy use of so-called zk-snarks in order to:

a) Obfuscate transaction senders

b) Obfuscate transaction recipients

c) Obfuscate the amounts of a transaction

The idea behind the protocol is that one could convert any non-privacy currency such as Bitcoin into a private one by sending their non-private tokens to the protocol, and receiving a guarantee or commitment by the protocol for an alternative token with an equivalent value. The protocol allows users to move funds for which they have commitments securely and privately (i.e., without revealing to who and how much). Users of the protocol may issue requests to send M different recipients money, and those M recipients may actually just be the sender himself, and the money may be 0 amount! Adversaries would not be able to know. Money can stay as commitments inside the protocol for arbitrarily long periods of time, and the longer it stays, the better for privacy. This is analogous to the concept of mixing. A user of the protocol may withdraw any tokens for which he has commitments by the protocol at any time. We will explain zk-snarks in a bit more detail in the next section. There are settings in which the above setup does not fully guarantee privacy, such as when:

a) The number of peers participating in the network is very low, or

b) The amount of traffic in the network is low.

The danger in having a small number of peers and low traffic is that an adversary could send an artificially large number of peers he owns to the network and generate sufficient traffic, convincing genuine users that the network is safe and private. The adversary would be able to make good guesses about the genuine user transaction destination and correlate users to each other pretty well. In order to mitigate this problem, Gewel.io software will allow peers to generate random noise on the network (such as large amounts of 0-gewel transactions sent to themselves). Users of the network are incentivized to generate random noise because it will strengthen the privacy guarantees. This solves the issues related to low traffic. As for mitigating the issues related to a low number of peers, this issue only exists in the short run, but is easily mitigated via transparency, and via encouraging peers to keep their funds as commitments for long periods of time, until the network has a sufficiently large number of non-colluding peers with high confidence.

Finally, an important aspect of privacy is the off-chain process of opting-in to the Gewel.io ecosystem. Gewel.io, like Bitmex which operates legally in Seychelles, will not require KYC documents from participants. This is an extremely important aspect in enabling a truly private experience for our users. In order to be compliant with laws however, Gewel.io will not allow natural persons or companies who are citizens of the United States to participate unless they submit KYC documents.

Understanding Zk-Snark

The zk-snark – or zero knowledge succinct non-interactive argument of knowledge – proof has been available for some time, but it was first deployed on a widespread scale within the ZeroCash blockchain software.

Via zk-snark, an individual can prove that he has certain information without revealing the information in question, and with no interaction between himself and another user; who are defined as prover and verifier.

With zero-knowledge, the prover is able to convince the verifier that a certain statement is true by only revealing information that proves the validity of the statement but not the statement itself. For example, it could be proved that a hash of a random text exists without revealing what that text is.

Zero knowledge ‘proof of knowledge’ takes this a step further. The prover can convince the verifier that not only does such text exist but that they know what that text is without revealing any information about that text.

It is possible to confirm succinct zero knowledge proofs with only a few hundred bytes and within a few milliseconds even for large statements. Whereas early zero knowledge proof systems required numerous rounds of communication, non-interactive constructions only require a single message be sent between the prover and verifier. At this point the only zero-knowledge proofs that are short enough to publish on a blockchain consist of a setup phase that creates a mutual reference string which is shared between the prover and verifier. These shared reference strings can be referred to as public parameters.

If someone was able to access secret randomness used to create the public parameters, they could produce false proofs and, in turn, create fake Gewel.io tokens which would be indistinguishable from real ones. However, the manner in which the parameters are created makes it impossible for this malicious activity to take place. Public parameters are generated in a sophisticated event involving multiple different users; an event which is known as a ceremony. Each user involved in the ceremony is then forced to destroy their minute piece of the parameters. Even if only a single user destroys their piece, the parameters are unusable, making it highly unlikely that these parameters could exist and fall into the wrong hands

Gewel Layered Network

For a blockchain software network to function properly, there must be a structure in place; a transaction must undergo certain proofs before it can be verified.

In the case of bitcoin, a transaction will be validated after the following three items have been proved;

1. That the bitcoins being used have not been spent by the sender previously. The sender does not need to take any action to show that this is the case as this is ascertained simply by examining the ledger.

2. That the sender has the necessary authority to send the coins to the value agreed upon in the transaction. This is validated by signing the transaction with the secret key which relates to the address where the coins are being sent from.

3. That the transaction is balanced in terms of coins inputted and coins taken out. This should be evident from the transaction, as the amount being transferred is known to all parties.

Gewel.io will use a form of zero-knowledge proofs known as zk-snarks to prove the above points without revealing any additional information. When each transaction is validated, there also exists a zk-snark which can be used to show that Gewel.io tokens exist and have not been spent. The sender has the authority to send the Gewel.io tokens. Thus, the amount of Gewel.io tokens sent equals to the amount of Gewel.io tokens received.

During this process, the information required for spending Gewel.io tokens is attached to the transaction by creating a new zk-snark and is encrypted using the recipient's public key which can only be used by the transaction recipient.

This results in a new distributed ledger which has been called the zero-knowledge security layer.

Instantaneous Transactions

Transactions on the Gewel.io software need to be secure and private, but also quick. Gewel.io uses inodes quorums to provide the ability to send and receive irreversible transactions instantaneously.

When a quorum is formed the inputs of the transaction are locked for spending. This lock takes approximately four seconds to complete. If the inode network achieves a consensus, any conflicting transactions or blocks will be rejected henceforth. Only exact matches on the transaction id will be accepted.

The idea of this is to connect the Gewel.io software with real world usage, for example via a mobile device at the point of sale. If users are able to settle commercial transactions using digitally encrypted blockchain software, with zero delay, then the Gewel.io software could become a serious rival to traditional cash, credit or debit card forms of payment.

Gewel.io Tokens & Gewel.io Software

The supply of Gewel tokens will depend on the balance that is placed into the genesis block by the community. A maximum of 77.7 million Gewel.io tokens will be minted which includes any Gewel tokens that are created in the genesis block. The token will not be mintable, nor burnable, which is to say that the number of Gewel tokens in circulation will be the same *forever*, enforcing scarcity as adoption increases.

Smart Contract

Since the advent of Ethereum, smart contracts have become a standardized feature of later, more advanced blockchains. A smart contract is a set of machine instructions stored on the blockchain ledger and executed in the virtual machine. Smart contracts allow users to create apps (in the usual sense) that are totally decentralized and require no trusted third parties to operate.

Because Ethereum emerged much later than Bitcoin, it has significantly improved the functionality, positioning, and design architecture of smart contract platforms, while avoiding the defects of the Bitcoin script. Vitalik and company have brought our industry's vision of a scalable, enterprise-ready smart contract platform much nearer to reality — especially with their invention of the Ethereum Virtual Machine (EVM). Virtual machines play an integral role in the successful implementation of smart contracts, as they allow any network participant to access and run a smart contract's code while securing the integrity of that contract. In other words, anyone can execute the contract, but no one can violate its design or change its intended outcomes.

Problem

Ethereum has several serious problems that are also commonly found in other smart contract platforms:

1. Missing standard libraries and tool libraries. In the development of their smart contract platform, Ethereum developers realize that no standard library presents in their language, Solidity. They can only copy and paste the code from some open source software. Firstly, the security of these codes is not guaranteed, and people may modify the code unwisely to achieve smaller Gas consumption (i.e., lower transaction/execution fees), which may introduce more serious security issues to their contracts.
2. No data object support. Data objects are nowadays widely used in object-oriented programming. Common data formats such as JSON, XML are used by most APIs to deliver data. The lack of data object support has made Solidity extremely burdensome to process data input and integrate with external APIs.
3. Difficult to debug and test. This problem not only presents a design flaw of EVM, but also relates to its difficulty of implementation. The only exception that EVM can throw is OutOfGas, and there is no debug log or external code available.
4. Floating point and fixed-point operations are not supported.

5. Poor security. Since safe math is not supported by default, there are often “one-hundred-million-dollar code losses” caused by integer overflow vulnerabilities. And because there is no sandbox-style security isolation, it has recently been said that the EVM has been completely broken by a “nuclear bomb” of security exploitation.
6. Unreasonable billing, the high cost of application. EVM not only makes writing good code difficult, it also makes it awfully expensive. For example, storing data on a blockchain requires a lot of Gas. This means that the cost of caching data in a smart contract can be extremely high, so data is often recalculated each time the contract runs. As the contract is continuously executed, more and more gas and time are spent on repeatedly calculating the same data, and after all that, the cost of the code cannot be simulated adequately offline.
7. Only limited storage. No consideration for secure docking of external resources such as IPFS, etc.

Gewel Virtual Machine

The consistent execution of smart contracts must derive from the deterministic nature of its main blockchain’s execution results, which means that all transactions must be handled strictly in chronological order (total order).

Many of the public blockchains that have been launched now support smart contracts, but after careful analysis of their structure, most of them have not overcome the above-mentioned defects of Ethereum, let alone true innovation. Some do not have general purpose functionalities and focus on specific domains only, such as Libra’s MOVE.

By way of introduction, here are some features of the Gewel Virtual Machine (GVM):

1. Based on Java. Gewel adopts the popular Java Virtual Machine (JVM) architecture for its smart contract computation environment. The JVM has the biggest number of device installations in the world. The world’s most popular operating system, Android is a JVM variation. Gewel Virtual Machine (GVM) supports the complete Java instruction set and can seamlessly integrate with other Java APIs, including Java SE, Java EE and Android. The Java language is a platform independent programming language adopted by Gewel smart contracts. Once the Java program is compiled into the Java bytecodes, it can run on any JVM-enabled devices. In addition to being the most widely used, the Java language has the largest developer community in the world. There are a large number of third-party libraries on the network for quick development and use. Gewel’s smart contract also supports data types

such as XML and JSON, which are used by most financial systems. There are already many powerful IDEs available, such as IntelliJ Idea, Eclipse, VS Code, etc. Running on Java allows for high portability, support for floating point operations, and easy CPU optimization. And, of course, there's all those killer apps.

2. Hierarchical calculation. At present, all known blockchains handle smart contracts in unified deployment, with unified computing and unified consensus processing. In real life, because of the different application scopes, the power capacities of all of our computing devices are quite varied. Gewel allows all of these computing devices to run the GVM, without affecting the consensus processing of the blockchain's computation results. Gewel separates the calculation of the smart contract from the consensus of calculation results, which allows the network to categorize computation according to the power of the particular device. Nodes are only responsible for consensus on all computation results to ensure consistency. This design allows the Gewel chain to adapt to a wide range of application scenarios and apply smart contracts at multiple layers. For example, the mainframe can develop complex contracts with huge amounts of computing; PC nodes can develop contracts suitable for PC computing; mobile devices can run smart contracts for mobile devices; and so on.
3. Native 64-bit integer support and fixed-point representation. The GVM has been implementing native acceleration processing on 64-bit integer and fixed-point operations according to different CPU types, thereby improving the computing speeds of all smart contracts.
4. Support for runtime and compile-time security checks. The risks of Ethereum smart contracts come mainly from one of two sources: the vulnerability caused by the flaws in the design of the smart contract language itself, or more importantly, the code loophole caused by the coders' unfamiliarity with the smart contract language. To avoid these problems, the GVM provides an automatic security check mechanism during both compile and execute phases, checking for buffer overflows, stack overflows, excessive computation, excessive memory consumption, and unsafe external requests. Therefore, the reliability and security of the execution of the smart contract are guaranteed, and the robustness of the whole system is ensured to a large extent.
5. Transparent billing mechanism. GVM only bills the computing workload, memory consumption, and storage capacity, and can provide external tools for calculation. Users can clearly calculate the GAS cost without accessing the network. If the network is congested, the user can pay a fixed rating priority fee. The priority fee (temporarily set to an exponent of 2, depending on the appropriate level 1, 2, 3, 4, 5, or 6) is prioritized for queuing, and the fee structure is fixed and transparent.

6. Smart contracts can be updated. In the case where all relevant accounts vote to agree, the smart contract containing breached vulnerability can be deprecated, and the newly deployed contract can inherit the relevant status and data of the prior contract.

7. Fully compatible with all smart contracts written originally for Ethereum. There may be nearly a million smart contracts that have matured on Ethereum. In order to attract these and future users, the GVM provides a fully compatible Ethereum smart contract mechanism that allows users to deploy directly to the Gewel chain without any code changes. The logic and results are exactly the same.

8. Enhanced security. All operations Gewel virtual machines are executed under sandbox isolation, and internal procedures will always provide its own operating status to an external monitor. If unforeseen circumstances are found, such as attacks, infinite loops, the occurrence of astronomically high numbers (often caused by memory overflow), the GVM will immediately halt its operation. This design guarantees the normal operation of a given node and protects the user's on-chain assets from accidental loss. At the same time, the validation committee performs BLS signature processing on the execution result, so that the execution result state data cannot be tampered with during the network communication process, thereby effectively preventing data forgery and ensuring the consistency of our results.

9. Nodes can upgrade new virtual machine functionality without downtime. When designing a virtual machine, the Gewel team built the core functionalities internally, such as our instruction set, consensus mechanism, signature system, and consistency checks. Most of the functional modules, however, are externally linked. When running, the GVM will dynamically search for related modules as needed. This feature makes it easy to upgrade non-core features of the GVM, which becomes especially important for large collaborative computing scenarios. Gewel provides a call stack depth adjustment mechanism that allows miners to make appropriate adjustments based on node performance and user-facing population.

Use Case

Because Gewel is a general purpose blockchain, it supports most common blockchain applications, such as issuance of customized tokens. We hereby only highlight use cases that are unique to Gewel.

Digital Identity

The protection of private data remains a big challenge in modern cybersecurity. Traditional centralized data servers have become easy targets of cyber criminals. In the past few years, massive data leaks of big enterprises including Facebook, Yahoo, Marriott and so on, have resulted in billions of user data records being stolen and insurmountable financial losses. As software and devices become more and more complex, it is practically impossible to eliminate all security vulnerabilities in a system. A new form of identity must be implemented to prevent further data breaches.

Decentralized ledger technology handles identities via shared root of trust instead of centralized authority or a single point of failure. The Decentralized Identifiers (DIDs) is a standard facilitated by the Internet body World Wide Web Consortium (WC3) that allows users to own and manage their personal data. Gewel can fully incorporate with open identity protocols including, but not limited to, the DID standard.

Central Bank Digital Currency

Countries around the world are paying more attention to the concept of Central Bank Digital Currencies (CBDCs), trying to capitalize on this clear and exciting trend, and China is no exception. In fact, China is eager to lead the charge with its DC/EP project. China has been researching and developing the idea of a CBDC as far back as 2014, and its national efforts to adopt digital technologies continue to grow.

CBDCs are positioned to revolutionize the global financial system. These technologies will serve as a new form of fiat currencies, but in their implementation, they will alter the very concept of currency and what it can mean to society. It will do through its five primary design principles:

1. Distributed Payments that eliminate unnecessary intermediaries.
2. Financial Inclusivity for the unbanked and underserved.
3. Efficient Cross-Border Payments for an increasingly globalized economy.
4. Digitized Management of Monetary Policy for more effective and immediate regulation.
5. New Framework to provide for the next generation of financial service innovation.

To achieve this, the People's Bank of China (PBoC) announced that its DC/EP project would take on a two-tiered operating architecture that would maximize resource utilization, foster collaboration, and avoid "disintermediation" or bottle-necking. Together these two tiers allow for consistent, safe financial service–the first-tier being, of course, the government-controlled PBoC, and the second, commercial banks.

The network itself is underpinned by three central authorities: the Certification Center, which manages the relationship between the customers 'identities and their anonymous digital wallets; the Registration Center, which manages records of ownership and transference; and the Big Data Analysis Center, which monitors the entire digital currency environment to support the central bank's monetary policy and macro-prudential supervision.

Gewel is a blockchain capable of fulfilling these technical requirements in order to meet the expectations and needs of the world's first operational CBDCs. Through a number of strategic design choices, Gewel expresses the cutting-edge features of a blockchain without sacrificing the security and scalability required of a national currency. These features include bifurcated identity-authentication (for both practical anonymity and retrievable KYC necessary to combat criminality and terrorism) as well as crucial integration with business applications, legacy transaction systems, and parallel blockchain systems.

A blockchain system will need cross-chain operability to verify its own wallet and central bank digital currency. Gewel can serve as a connector to any type of digital ledger, including RTGS, CBDCs, or other types of virtual currencies such as Bitcoin.

Big Data Analysis

The status of the big data center is unique among the three centers of DC/EP, as it is charged with conducting anti-money laundering, anti-fraud, and other security monitoring. As the data aggregation office of the other two centers, the big data center also needs to analyze and govern all the data being collected and processed, in order to help policy formulation with the output of many indicators.

Due to this important office and sensitive nature of the big data center, in order to ensure transaction security and information security, the center must abide by a special mandate: for all data, the big data center can only read, not modify.

Gewel reserves a monitoring interface for on-chain activities, and can promptly warn against illegal transactions through the big data center.

Clearing System

Considering that there may be problems in directly connecting to the central bank's digital currency system without special permission, Gewel has also established a clearing system architecture, wherein intermediaries (which can be financial institutions with financial strength) perform the final clearing of funds to ensure that user accounts have a relatively smooth experience. The most important parts of the solution are Gewel Connect and Gewel Validator. The two core components are each described below.

Gewel Connect

Gewel Connect is a plug-in module that processes Gewel payment transactions in the banking system (it's a bit similar to the payment front-end system). Between the remittance bank and the receiving bank, Gewel Connect has established an information channel for exchanging KYC / AML, risk control information, handling fees, exchange rates, and other payment-related information. In order to attract banks to join, Gewel has created adaptive designs on KYC / AML that can be personalized by both banks. Before the transaction is initiated, Gewel Connect sends this information to the counterparty of the transaction. Only by confirming OK can one execute the transaction and clear the funds.

Gewel Validator

Gewel Validator is a verification machine. Before the transaction enters the Central Bank and blockchain ledger system, it must be confirmed by Validation. This machine has a strict authentication mechanism, and its verification rules can be customized according to specific system requirements.

Gewel also introduces the scalable collective signature scheme CoSi. All communication parties must jointly sign the key information received and sent. Without increasing communication overhead, the system may always verify the correctness of the communication messages of all parties in the clearing system.

The Gewel chain uses PoS consensus model to solve double-spending, and the ledger nodes also perform additional version checks to ensure data integrity. There is also a queue manager in the system, which packages signed transactions into blocks and broadcasts each block to all participant nodes in the same channel. After receiving the broadcast, the nodes in the channel will verify the transaction again, and then update the transaction block to the ledger.

In order to ensure the privacy of transactions, a two-way channel must be established between every two banks in the system. In each channel, the bank must establish a channel account and allocate appropriate funds in the channel account to ensure that the transaction can be completed. Each transaction must be signed by the two banks in the channel to be verified.

Conclusion and Future Works

This whitepaper has introduced the reader to Gewel, a blockchain platform. Gewel makes a number of advancements in decentralized ledger technology. Firstly, its proprietary consensus mechanism offers a unique solution to the so-called “blockchain trilemma” of scaling, security, and speed with a strong emphasis on applicability. While other third-generation blockchains have abandoned Proof-of-Work, the fundamental innovation of Bitcoin’s Nakamoto Consensus, Gewel marries this older methodology to a more current cutting-edge technology. Gewel achieves this by rethinking the role of mining, breaking the process down into two component parts–minting new coins (with the restriction of a global upper bound on supply) and verifying transactions–and supplying each with its own blockchain.

The reader has also encountered the Gewel Virtual Machine (GVM), a Java-based virtual runtime environment for the execution of smart contracts on the Gewel network. Just as Gewel consensus emerges as a solution from a careful analysis of the pain-points of prior decentralized technologies, the GVM offers original solutions to distributed computing’s most persistent problems. Chief among these attributes is the GVM’s Java compatibility, which grants the Gewel network access to the world’s largest pool of legacy development and computing infrastructure. This serves as a necessary access point between the legacy web and a decentralized future. Other key features of the GVM with which the reader has become acquainted include hierarchical calculation, native 64-bit integer support, fixed-point representation, and increased security, compatibility with other VMs including Ethereum’s EVM, atomic swap support, IPFS docking, and more. The GVM is a key innovation that allows the Gewel network both to support everyday enterprise and economic transactions and to provide a framework for the developments of meaningful, “killer” decentralized applications.

The reader has also learned that Gewel incorporates Zero-Knowledge proof privacy, which in practice allows nodes to communicate about the existence of information without divulging the content of information. This brings the privacy of our public network as it becomes increasingly overrun with sensitive information.

One of the most notable uses of Gewel, as a highly scalable blockchain system, is its use as an interbank intermediary. Through tools like Gewel Connect (a third-party plug-in module for banking systems), and Gewel Validator (a verification machine), our network can link any two banking systems; Gewel can support cross-chain transactions among any two digital currencies. This is a vital feature of our network, particularly given the increasing promise and prominence of Central Bank Digital Currencies (CBDCs). Gewel's architecture makes it an ideal technology to connect all kinds of data systems and enterprises, both public and private, as industries across the board turn to decentralization. We at the Gewel network believe this on-chain migration will be a large part of the economic and technological landscape of the next five to ten years, and Gewel provides the support necessary in this period of change.

In the coming months and years, Gewel plans to adopt several additional features that will become necessary for the network to mature. These include horizontal scaling, increased privacy protections, and smart contract standard libraries. Horizontal scaling, or sharding, is a technique that will help our blockchain scale. In brief, sharding will allow nodes to process and store only a fraction of their data on-chain, with reference to the rest, improving the efficiency of the network. And finally, Gewel is continuing to develop smart contract libraries for the various contracting languages it supports. These standard libraries–repositories that maintain coding definitions for commonly used algorithms, data structures, and mechanisms–will play an important role in the standardization of our nascent industry. They will also foster a common understanding among our developer community, which will be an absolutely necessary foundation for the development of future apps and enterprises in an open system like Gewel's public network.

We are grateful for the interest and support of our community, partners, and advisors, and above all, we are excited to bring our technology to the world.

Appendix

Roadmap

Q2 & Q3, 2020

Gewel Planning and Infrastructure

Gewel will plan the protocols and consensus methods, the structure of nodes, the currency details, validation/mining rewards and distributions, and everything else needed to construct a whitepaper draft.

Q4, 2020

Gewel Blockchain Prototyping

Gewel will work on constructing a minimum viable product for the blockchain, with block reward distribution. The goal is to have a solid framework, and the emphasis is on algorithmic correctness.

Q1, 2021

Gewel ZSL and Wallet

Gewel will integrate a zero-knowledge security layer. This is designed to allow for anonymous and secure transactions on the blockchain. This will keep origins, destinations and amounts private and only known to senders and recipients. Gewel will also develop and release a wallet client for users.

Q2, 2021

Gewel Governance and Funding

The Gewel Governance and Funding System will be finalized in order to help any Gewel Open Decentralized Autonomous Network function efficiently and evolve organically— while utilizing the collective intelligence of the Gewel community. The Gewel governance and funding system will be designed so that the community can vote on funding and governance proposals— that of which are found on any blockchain that decides to use the Gewel software.

Q3, 2021

Gewel Private Messaging

Gewel will integrate a peer-to-peer persistent private messaging system (GewelChat) which will allow for Gewel users to send shielded messages using zero knowledge cryptography. GewelChat will be a completely anonymous communication network, with no third-party intermediaries, that will allow users to communicate privately.

Q4, 2021

Gewel Core JavaScript Library

The Gewel Core JavaScript library will provide the base foundation for the explorer insight API as well as future light wallets. The Gewel Core JavaScript library is a building block for third parties to integrate into Gewel Core.

Q1, 2022

Light Wallet Integration

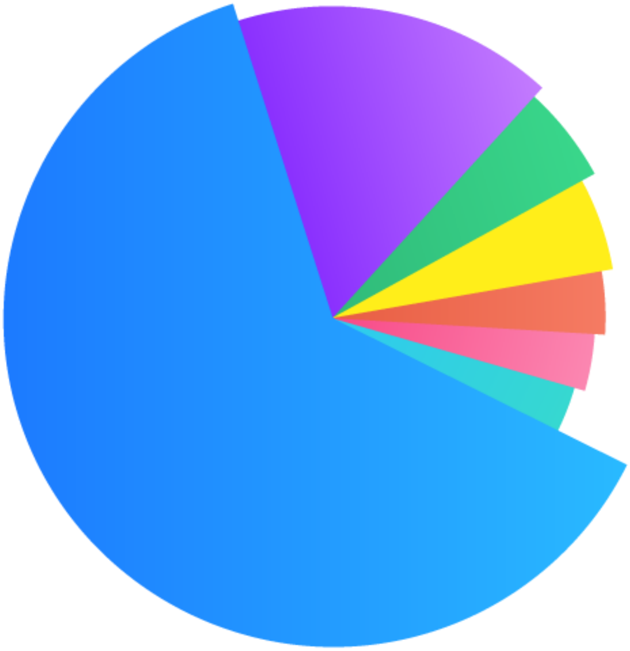
Integrating Gewel into third party light wallet that will allow users to send and receive Gewel tokens easily without the need to run a full node. Users will be able to send and receive Gewel tokens from a mobile wallet in addition to the desktop wallet.

Use of Proceeds

We are looking to raise a significant capital, through the Gewel Token offering, in order to maximize initial launch capabilities and enforce stability. We have allocated the use of these proceeds to build bold and exciting propositions that will empower the daily use of blockchain and cryptocurrency. The proceeds of our ICO will be used to roll out, enable and unlock propositions that can be found below. All of the aforesaid propositions will be fundamental to the Gewel token appreciation and enhanced utility.

Token Sale - 62%

Presale, Private sale and the crowdsale of the token sale.



Founders, Team and Advisors - 18%

Founders, Team and Advisors of the project will be rewarded with 18% of all available tokens.

Ecosystem Partners - 4%

We're allocating 4% of Gewel tokens to remunerate existing and new partners that are providing their expertise and solutions.

Project reserve 6%

We have allocated approximately 6% of tokens to fund any unexpected shortages and for anything else that requires action over the life of the project.

Pre ICO investor - 6%

Early contributors in the Gewel project will be rewarded with 6% of tokens.

Bounties - 1%

Once we're listed on exchanges we want to reward the crypto trading community with bounties so we can spread the news of our token even further.

Token Giveaway Offer, 3%

The first 100,000 Gewel users to participate in our crowd sale will receive a bonus of 100 free Gewel tokens. We will use our social media channels to make the announcement.

This appendix is subject to modifications as it is incomplete.

GEWEL

NEW AGE CRYPTOCURRENCY

WWW.GEWEL.IO