LGCY NETWORK
WHITE PAPER V1.0

INTRODUCTION

ABSTRACT

LGCY Network is a decentralized application (dApp) based blockchain using the Libertas Protocol to engage community participation.

INTRODUCTION

In July 2017 TRON designed an ingenious solution to the scalability issues of Ethereum. TRON was able to solve the energy waste problem by adopting a Proof of Stake (POS) consensus rather than a Proof of Work (POW) consensus like previous blockchain iterations. TRON, however, upgraded the original concept of a POW consensus to a Delegated Proof of Stake (DPOS) system which utilizes 27 Super Representatives (SRs) to produce blocks for the network. Since in a POS system, parties are able to hoard large amounts of the network's currency to gain heavy influence in the ecosystem, the original concept of decentralization quickly reverts back to a centralized model. The DPOS consensus is designed to spread votes among the 27 SRs in order to level the playing field among token holders and thus keep the network decentralized.

We believe that DPOS is the best model for decentralizing a blockchain while still giving the network mutability as technology advances. Unfortunately, TRON's DPOS consensus is not without its problems. Since vote strength is determined by how many tokens a voter has, larger accounts continue to influence the network more than people who own very few. The problem of centralization remains.

LGCY Network will solve this issue with DPOS by implementing the Libertas Protocol to the SR system of governance. By splitting up the 27 SRs into three branches, limiting the power of each, and introducing terms to the governance system, no single large token holder will be able to gain an unbalanced amount of power. Also, by splitting the power up among three branches, we are further utilizing the power of a digital democracy by giving voice to smaller token holder and community led dApp projects.

FORK/PURPOSE

LGCY will begin as a fork of the TRON blockchain. The essential base and functions of TRON will remain for LGCY with the major differences lying in the system of governance and the focus on community. Things will remain the same for the first year while the fork testnet and mainnet is tested. When LGCY Mainnet is enacted we will implement the new governing and energy systems (the Libertas Protocol).

The value of a dApp blockchain resides in the strength of its community. Without dApps and happy project leaders, LGCY would be worthless. The major focus will therefore be aimed at supporting the projects that build on our chain and directing a portion of our resources toward them. By reworking the system of governance to give voice to smaller projects and token holders, dApps will be incentivized to build on LGCY.

A major focus in supporting our community will be a push toward developing more tools and making sure we keep up with what is available on Ethereum. We believe we have created the best blockchain there is; the better the tools and development, the more reason developers will have for coming to LGCY Network.

FOCUS ON COMMUNITY

We believe that for a blockchain to be truly decentralized, it must be led by the community. At the beginning of any venture, centralization is required to get off the ground. As the network grows, however, the community will begin to take over the leadership of LGCY Network. It is not our intention to remain in voting control forever. We, in fact, have set a series of milestone (outlined in the roadmap) in which we will slowly hand over the reins of government to the community. At that point, LGCY Network will be the most decentralized blockchain in existence, allowing the community which runs on the network to be in control.

TEAM

Justin Wai'au - *Project Manager*
Blockchain & Business experience
http://www.linkedin.com/in/justinwaiau
https://twitter.com/waiaujustin

Wes Wade - *Team Leader*
10+ Years of Management experience
MDiv, The Cornerstone Seminary
http://www.linkedin.com/in/weswadejr
https://twitter.com/weswadejr

Craig Mackie – *Business Developer and Marketing*
https://www.linkedin.com/in/craig-mackie-638199

Vinod Kumar - *Lead Blockchain Developer*
Founder and CEO at TscHybrid
https://twitter.com/answervinod

Abhilasha Singh – Blockchain Developer
Co-founder & COO TSCHybrid
Operations Management, Strategy Planning & Designing
Twitter: https://twitter.com/Abhilasha07

Umair Hussain - Art Director/Graphics Designer
Karachi School of Art
https://www.linkedin.com/in/umair-hussain-2a977b106/
https://twitter.com/omihussain

Thomas Roelofs - *Marketing Advisor*
7+ Years in Digital & Growth Marketing

Alan Crawley – *Advisor*
Twenty Four years of management experience in financial markets primarily in proprietary derivatives trading and market making. Former Director of Timber Hill and Interactive Brokers in Hong Kong and Australia.

Anurag Rao - *Legal consultant*
Legal experience in the FinTech and Blockchain industry with Reserve.
https://www.linkedin.com/in/anu-rag-rao-6a4236111/

DEVELOPMENT

TOKEN RELEASE SCHEDULE

- 33% of the supply released with Private Sale series A and B and Exchange (20% Private Sale, 5% Exchange, 5% Team, 3.33% Dev Fund)
- 16.66% token release for year 2/bring team control to 50%
- 16.66% token release for year 3 and burn remaining 33% token supply as a symbol of the team handing over the project to the community and helping the investors.

LGCY NETWORK ROADMAP

- **UNISWAP** — 1 MONTH
- **EXCHANGE LISTING** — TBD
- **TRON FORK TESTNET** — 7 MONTHS
- **TRON FORK MAINNET** — 12 MONTHS
- **DEX ADDED** — 12 MONTHS
- **LGCY TESTNET** — 1.5 YEARS
- **LGCY MAINNET** — 2 YEARS

TOKEN RELEASE SCHEDULE

33% of the supply released with Private Sale series A and B and Exchange (20% Private Sale, 5% Exchange, 5% Team, 3.33 Dev Fund)

16.66% tokens release for year 2/bring team control to 50%

16.66% tokens release for year 3 and burn remaining 33% token supply as a symbol of the team handing over the project to the community and helping the investors

MILESTONE

16TH AUG - 15TH SEP

TscHybrid team will be engaged in research work pertaining to building a clone of Tron Blockchain as requested by client.

END OF SEPTEMBER

Initial fork of the existing Blockchain and addition to the LGCY GitHub repository starts from this milestone, all documentation would continue to be added as and when progressed.

OCTOBER

Java Tron Compilation and bug rectification, possibility of using it outside TVM (Tron Virtual Machine)

NOVEMBER - MARCH

LGCY Grid, LGCY Scan, Super Representative, LGCY Box, inbuild DEX, Contract Compilation & Development Tools.

APRIL - MAY

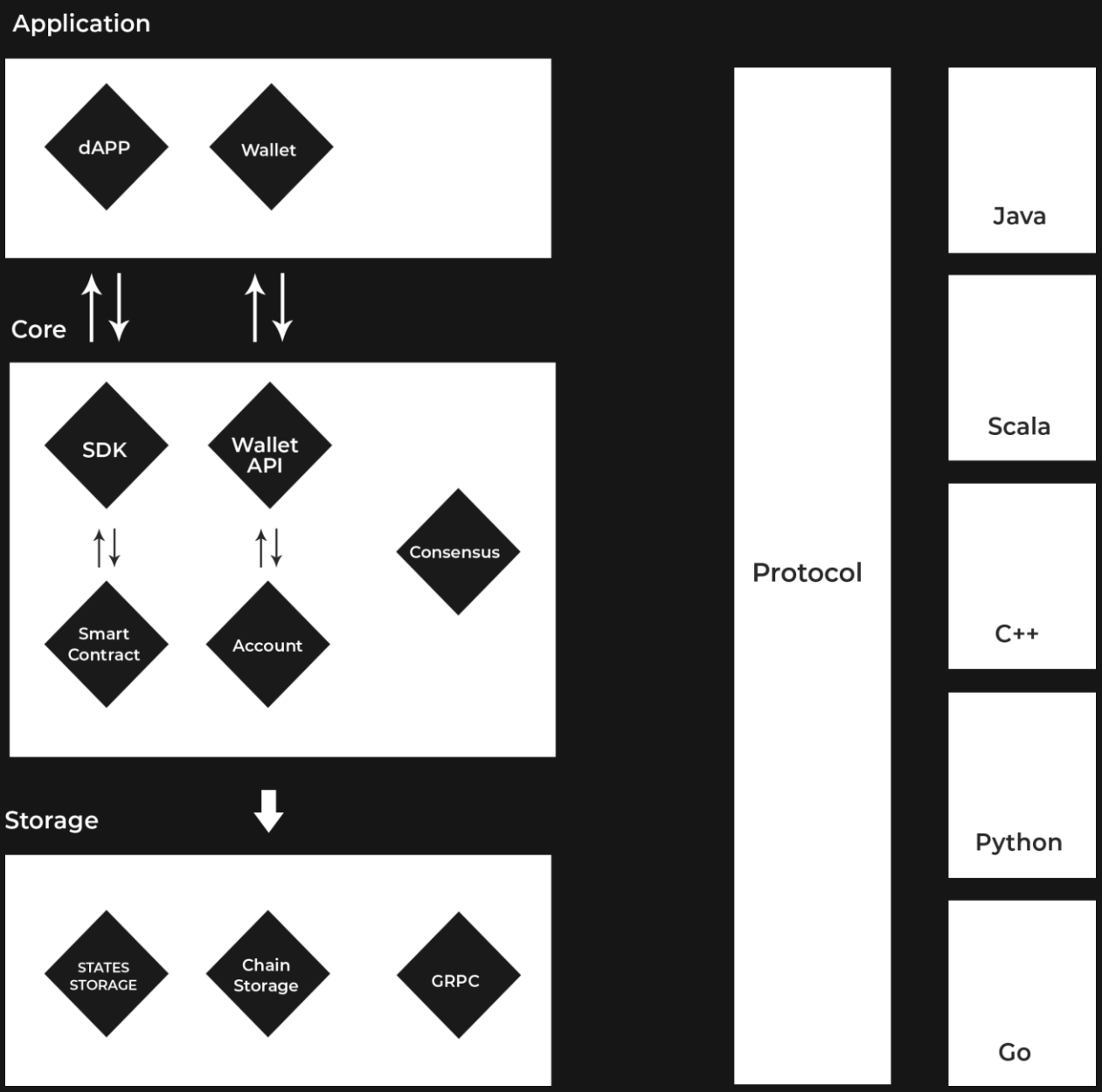
Test Net testing and debugging

JUNE - AUGUST

Main Net Launch, debugging and complete Handover including Source Codes.

ARCHITECTURE

LGCY utilizes a 3-layer architecture which is separated into Storage Layer, Core Layer, and Application Layer. We adhere to Google Protocol Buffers which are language-neutral, platform-neutral, for structuring our data.



CORE

The core layer includes smart contracts, account management, and consensus. The LGCY Network will have a stack-based virtual machine implemented within and will use an optimized instruction set. Solidity is the smart contract language—chosen in order to better support dApp developers.[1] This will be followed by future support of other advanced languages. LGCY's consensus mechanism is based on DPOS but because of the nature of this blockchain there are some special changes.

STORAGE

LGCY utilizes a distributed storage protocol which consists of Block Storage and State Storage. The real world has a need for many types of data storage. A graph database is therefore implemented into the design of the storage layer.

BLOCKCHAIN STORAGE

LevelDB was chosen as our blockchain storage. It is developed by Google and has a proven track record. It supports arbitrary byte arrays as both

[1] Please see the offical Solidity documentation: https://solidity.readthedocs.io/

keys and values, singular get, put and delete, batched put and delete, bi-directional iterators, and simple compression using the Snappy algorithm.

STATE STORAGE

In the full-node memory LGCY has a KhaosDB supports witnesses to switch from their own active chain into a new main chain. It can also store all the newly forked chains created within a certain period of time. Blockchain storage is also protected by making it more stable from being terminated abnormally in an intermediate state.

APPLICATION

One of the shining points of the LGCY network is the ability for developers to create dApps and customized wallets. The best part is that the applications and uses are limitless because of the ability for smart contracts to be deployed and executed on our chain.

PROTOCOL

LGCY protocol adheres to Google Protocol Buffers.

PROTOCOL BUFFERS

Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler. You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams and using a variety of languages.[2]

Google Protocol Buffers currently support generated code in Java, Python, Objective-C, and C++. With our new proto3 language version, you can also work with Dart, Go, Ruby, and C#, "with more languages to come.[3]" There is also the option to use third-party implements. Protobuf unifies the API definitions, which makes development easier, and also optimizing data transfers. Users can simply take the API .proto from LGCY's protocol repository and integrate through the automatically-generated code libraries.

Protocol Buffers is 3 to 10 times smaller and 20 to 100 times faster than XML, with less ambiguous syntax. Protocol Buffers generate data access classes that are easier to use programmatically.[4]

---

[2] "What are protocol buffers?" https://developers.google.com/protocol-buffers
[3] "Pick your favorite language" https://developers.google.com/protocol-buffers
[4] Adapted from 2.4.1 "Protocol Buffers" in https://tron.network/static/doc/white_paper_v_2_0.pdf.

HTTP

A RESTful HTTP API is provided on the LGCY Protocol as an alternative to the Protobuf API. While the interfaces are the same, the HTTP API can be used in javascript clients.

LGCY VIRTUAL MACHINE (LVM)

The LVM is a lightweight, Turing complete virtual machine. The LVM connects seamlessly with the existing development ecosystem to provide millions of global developers with a blockchain system that is efficient, convenient, stable, secure, and scalable.[5]

DECENTRALIZED EXCHANGE (DEX)

The LGCY network natively supports decentralized exchange functions. A decentralized exchange consists of multiple trading pairs. A trading pair (notation “Exchange”) is an Exchange Market between LRC-10 tokens, or between an LRC-10 token and LGCY. Any account can create a trading pair between any tokens, even if the same pair already exists on the LGCY network. Trading and price fluctuations of the trading pairs follow the Bancor

[5] Adapted from 2.5 “TRON Virtual Machine (TVM) in https://tron.network/static/doc/white_paper_v_2_0.pdf

Protocol.[6] The LGCY network stipulates that the weights of the two tokens in all trading pairs are equal, so the ratio of their balances is the price between them. For example, consider a trading pair containing two tokens, ABC and DEF. ABC has a balance of 10 million and DEF has a balance of 1 million. Since their weights are equal, 10 ABC = 1 DEF. This means that the ratio of ABC to DEF is 10 ABC per DEF.[7]

IMPLEMENTATION

The LGCY blockchain code is implemented in Java and was originally a fork from TRON 4.0.

[6] Read about the Bancor Protocol: https://about.bancor.network/protocol
[7] Adapted from 2.6 “Decentralized Exchange (DEX)” in https://tron.network/static/doc/white_paper_v_2_0.pdf

CONSENSUS

DELEGATED PROOF OF STAKE (DPOS)[8]

The earliest consensus mechanism is the Proof of Work (PoW) consensus mechanism. This protocol is currently implemented in Bitcoin[9] and Ethereum.[10] In PoW systems, transactions broadcast through the network are grouped together into nascent blocks for miner confirmation. The confirmation process involves hashing transactions using cryptographic hashing algorithms until a merkle root has been reached, creating a merkle tree:

Cryptographic hashing algorithms are useful in network attack prevention because they possess several properties:[11]

- ***Input/Output length size*** - The algorithm can pass in an input of any length in size, and outputs a fixed length hash value.
- ***Efficiency*** - The algorithm is relatively easy and fast to compute.
- ***Preimage resistance*** - For a given output *z*, it is impossible to find any input x such that *h(x) = z*. In other words, the hashing algorithm *h(x)* is a

[8] We borrow heavily from TRON's Whitepaper in this section as they have done the ground breaking work on the DPOS system. See 3.1 Delegated Proof of Stake (DPoS): tron.network/static/doc/white_paper_v_2_0.pdf

[9] Bitcoin whitepaper: https://bitcoin.org/bitcoin/pdf

[10] Ethereum whitepaper:https://github.com/ethereum/wiki/wiki/White-Paper

[11] PAAR, C., PELZL, J., *Understanding Cryptography: A Textbook for Students and Practitioners*, 2010 ed.
Springer-Verlag Berlin Heidelberg, 2010.

one-way function in which only the output can be found, given an input. The reverse is not possible.

- ***Collision resistance*** - It is computationally infeasible to find any pairs $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$. In other words, the probability of finding two different inputs hashing to the same output is extremely low. This property also implies second preimage resistance.
- ***Second preimage resistance*** - Given x_1, and thus $h(x_1)$, it is computationally infeasible to find any x_2 such that $h(x_1) = h(x_2)$. While this property is similar to collision resistance, the property differs in that it is saying an attacker with a given x_1 will find it computationally infeasible to find any x_2 hashing to the same output.
- ***Deterministic*** - maps each input to one and only one output.
- ***Avalanche effect*** - a small change in the input results in an entirely different output.

These properties give the cryptocurrency network its intrinsic value by ensuring attacks do not compromise the network. When miners confirm a block, they are rewarded tokens as a built-in incentive for network participation. However, as the global cryptocurrency market capitalization steadily increased, the miners became centralized and focused their

computing resources on hoarding tokens as assets, rather than for network participation purposes. CPU miners gave way to GPUs, which in turn gave way to powerful ASICs.

To solve the energy waste issue, the Proof of Stake (PoS) consensus mechanism was proposed by many new networks. In PoS networks, token holders lock their token balances to become block validators. The validators take turns proposing and voting on the next block. However, the problem with standard PoS is that validator influence correlates directly to the number of tokens locked up. This results in parties hoarding large amounts of the network's base currency wielding undue influence in the network ecosystem.

The LGCY consensus mechanism uses a Delegated Proof of Stake system in which 27 Governing Branches (GBs) produce blocks for the network. Every 6 hours, LGCY account holders who freeze their accounts can vote for a selection of GB candidates, with the top 27 candidates deemed the GBs. Voters may choose GBs based on criteria such as projects sponsored by GBs to increase LGCY adoption, and rewards distributed to voters. This allows for a more democratized and decentralized ecosystem. GBs' accounts are normal accounts, but their accumulation of votes allows them to produce blocks. With the low throughput rates of Bitcoin and Ethereum due to their PoW consensus mechanism and scalability issues, LGCY's DPoS system offers

a mechanism resulting in 2000 TPS compared to Bitcoin's 3 TPS and Ethereum's 15 TPS.

For the first year before the Libertas Protocol is implemented along with our revolutionary LFUEL gas token, the LGCY protocol network will generate one block every three seconds, with each block awarding 32 LGCY to Governing Branches. A total of 336,384,000 LGCY will be awarded the first year to the 27 Governing Branches. Each time a GB finishes block production, rewards are sent to a sub-account in the super-ledger. GBs can check, but not directly make use of these LGCY tokens. A withdrawal can be made by each GB once every 24 hours, transferring the rewards from the sub-account to the specified GB account.

The three types of nodes on the LGCY network are Witness Node, Full Node, and Solidity Node. Witness nodes are set up by GBs and are mainly responsible for block production and proposal creation/voting. Full nodes provide APIs and broadcast transactions and blocks. Solidity nodes sync blocks from other Full Nodes and also provide indexable APIs.

ACCOUNT[12]

TYPES

The three types of accounts in the LGCY Network are regular accounts, token accounts, and contract accounts.

1. Regular accounts are used for standard transactions.
2. Token accounts are used for storing LRC-10 tokens.
3. Contract accounts are smart contract accounts created by regular accounts and can be triggered by regular accounts as well.

CREATION

There are three ways to create a LGCY account:

1. Create a new account through API
2. Transfer LGCY into a new account address
3. Transfer any LRC-10 token into a new account address

An offline key-pair consisting of an address (public key) and a private key, and not recorded by the LGCY Network, can also be generated. The user address generation algorithm consists of generating a key-pair and then

[12] Since LGCY Network is a fork of TRON, this section will remain nearly identical to the TRON Whitepaper: tron.network/static/doc/white_paper_v_2_0.pdf

extracting the public key (64-byte byte array representing x, y coordinates). Hash the public key using the SIN3-256 function (the SIN3 protocol adopted is KECCAK-256) and extract the last 20 bytes of the result. Add 41 to the beginning of the byte array and ensure the initial address length is 21 bytes. Hash the address twice using SIN3-256 function and take the first 4 bytes as verification code. Add the verification code to the end of the initial address and obtain the address in base58check format through base58 encoding. An encoded Mainnet address begins with L and is 34 bytes in length.

STRUCTURE

The three different account types are Normal, AssetIssue, and Contract. An Account contains 7 parameters:

1. **account_name**: the name for this account – e.g. BillsAccount.
2. **type**: what type of this account is – e.g. 0 (stands for type 'Normal').
3. **balance**: balance of this account – e.g. 4213312.
4. **vote**: received votes on this account – e.g. {("0x1b7w...9xj3",323), ("0x8djq...j12m",88),...,("0x82nd...mx6i",10001)}.
5. **asset**: other assets expected LGCY in this account – e.g. {, }.

6. **latest_operation_time:** the latest operation time of this account.

Protobuf data structure:

```
message Account {
   message Vote {
      bytes vote_address = 1;
      int64 vote_count = 2;
   }
      bytes accout_name = 1;
      AccountType type = 2;
       bytes address = 3;
      int64 balance = 4;
      repeated Vote votes = 5;
      map<string, int64>  asset = 6;
      int64 latest_operation_time = 10;
}



enum AccountType {
   Normal = 0;
   AssetIssue = 1;
   Contract = 2;
}
```

BLOCK

A block typically contains a block header and several transactions.[13]

Protobuf data structure:

```
message Block {
   BlockHeader block_header = 1;
   repeated Transaction transactions = 2;
}
```

BLOCK HEADER

A block header contains raw_data, witness_signature, and blockID.

Protobuf data structure:

```
message BlockHeader {
   message raw {
      int64 timestamp = 1;
      bytes txTrieRoot = 2;
      bytes parentHash = 3;
      uint64 number = 4;
      uint64 version = 5;
      bytes witness_address = 6;
   }
   bytes witness_signature = 2;
   bytes blockID = 3;
}
```

Raw Data

Raw data is denoted as **raw_data** in Protobuf. It contains the raw data of a message, containing 6 parameters:

1. **timestamp**: timestamp of this message – e.g. 1543884429000.

[13] This section remains nearly identical to TRON's "5. Block" in the TRON Whitepaper. The bandwidth model will change once the Libertas Protocol is inacted. Tron.network/static/doc/white_paper_v_2_0.pdf

2. **txTrieRoot**: the Merkle Tree's Root – e.g. 7dacsa...3ed.
3. **parentHash**: the hash of the last block – e.g. 7dacsa...3ed.
4. **number**: the block height – e.g. 4638708.
5. **version**: reserved – e.g. 5.
6. **witness_address**: the address of the witness packed in this block – e.g. 41928c...4d21

WITNESS SIGNATURE

Witness signature is denoted as witness_signature in Protobuf, which is the signature for this block header from the witness node.

BLOCK ID

Block ID is denoted as blockID in Protobuf. It contains the atomic identification of a block. A Block ID contains 2 parameters:

1. **hash**: the hash of block.
2. **number**: the hash and height of the block.

TRANSACTION

Signing

LGCY's transaction signing process follows a standard ECDSA cryptographic algorithm, with a SECP256K1 selection curve. A private key is a random number, and the public key is a point on the elliptic curve. The public key generation process consists of first generating a random number as a private key, and then multiplying the base point of the elliptic curve by the private key to obtain the public key. When a transaction occurs, the transaction raw data is first converted into byte format. The raw data then undergoes SIN-256 hashing. The private key corresponding to the contract address then signs the result of the SIN256 hash. The signature result is then added to the transaction.

BANDWIDTH MODEL

Ordinary transactions only consume bandwidth points, but smart contract operations consume both energy and bandwidth points. There are two types of bandwidth points available. Users can gain bandwidth points from freezing LGCY, while 5000 free bandwidth points are also available daily.

When a LGCY transaction is broadcast, it is transmitted and stored in the form of a byte array over the network. Bandwidth Points consumed by one transaction = number of transaction bytes multiplied by bandwidth points rate. For example, if the byte array length of a transaction is 200, then the transaction consumes 200 bandwidth points. However, if a LGCY or token transfer results in the target account being created, then only the bandwidth points consumed to create the account will be deducted, and additional bandwidth points will not be deducted. In an account creation scenario, the network will first consume the bandwidth points that the transaction initiator gained from freezing LGCY. If this amount is insufficient, then the network consumes the transaction initiator's LGCY.

In standard LGCY transfer scenarios from one LGCY account to another, the network first consumes the bandwidth points gained by the transaction initiator for freezing LGCY. If that is insufficient, it then consumes from the free 5000 daily bandwidth points. If that is still not enough, then the network consumes the LGCY of the transaction initiator. The amount is calculated by the number of bytes in the transaction multiplied by 10 DRIPS. Thus, for most LGCY holders who may not necessarily freeze their LGCY to participate in GB voting, the first step is automatically skipped (since LGCY balance frozen = 0) and the 5000 daily free bandwidth powers the transaction.

For LRC-10 token transfers, the network first verifies whether the total free bandwidth points of the issued token asset are sufficient. If not, the bandwidth points obtained from freezing LGCY are consumed. If there are still not enough bandwidth points, then it consumes the LGCY of the transaction initiator.

FEE

LGCY Network generally does not charge fees for most transactions, however, due to system restrictions and fairness, bandwidth usage and transactions do take in certain fees.

Fee charges are broken down into the following categories:

1. Normal transactions cost bandwidth points. Users can use the free daily bandwidth points (5000) or freeze LGCY to obtain more. When bandwidth points are not enough, LGCY will be used directly from the sending account. The LGCY needed is the number of bytes * 10 DRIPS.
2. Smart contracts cost energy but will also need bandwidth points for the transaction to be broadcasted and confirmed. The bandwidth cost is the same as above.

3. All query transactions are free. It doesn't cost energy or bandwidth.

LGCY Network also defines a set of fixed fees for the following transactions:

1. Creating a witness node: 9999 LGCY
2. Issuing a TRC-10 token: 1024 LGCY
3. Creating a new account: 0.1 LGCY
4. Creating an exchange pair: 1024 LGCY

TRANSACTION AS PROOF OF STAKE (TAPOS)

LGCY uses TaPoS to ensure the transactions all confirm the main blockchain, while making it difficult to forge counterfeit chains. In TaPoS, the networks require each transaction include part of the hash of a recent block header. This requirement prevents transactions from being replayed on forks not including the referenced block, and also signals the network that a particular user and their stake are on a specific fork. This consensus mechanism protects the network against Denial of Service, 51%, selfish mining, and double spend attacks.

TRANSACTION CONFIRMATION

A transaction is included in a future block after being broadcast to the network. After 19 blocks are mined on LGCY (including its own block), the transaction is confirmed. Each block is produced by one of the top 27 Governing Branches in a round robin fashion. Each block takes ~3 seconds to be mined on the blockchain. Time may slightly vary for each Governing Branch due to network conditions and machine configurations. In general, a transaction is considered fully confirmed after ~1 minute.

STRUCTURE

Transaction APIs consist of the following functions:

```
message Transaction {
   message Contract {
      enum ContractType {
         AccountCreateContract = 0; // Create account/wallet
         TransferContract = 1; // Transfer LGCY
         TransferAssetContract = 2; // Transfer LRC10 token
         VoteWitnessContract = 4; // Vote for Governing Branch (GB)
         WitnessCreateContract = 5; // Create a new GB account
         AssetIssueContract = 6; // Create a new LRC10 token
         WitnessUpdateContract = 8; // Update GB information
         ParticipateAssetIssueContract = 9; // Purchase LRC10 token
         AccountUpdateContract = 10; // Update account/wallet information
         FreezeBalanceContract = 11; // Freeze LGCY for bandwidth or energy
         UnfreezeBalanceContract = 12; // Unfreeze LGCY
         WithdrawBalanceContract = 13; // Withdraw GB rewards, once per day
         UnfreezeAssetContract = 14; // Unfreeze LRC10 token
         UpdateAssetContract = 15; // Update a LRC10 token’s information
         ProposalCreateContract = 16; // Create a new network proposal by any
GB
         ProposalApproveContract = 17; // GB votes yes for a network proposal
         ProposalDeleteContract = 18; // Delete a network proposal by owner
         CreateSmartContract = 30; // Deploy a new smart contract
         TriggerSmartContract = 31; // Call a function on a smart contract
         GetContract = 32; // Get an existing smart contract
         UpdateSettingContract = 33; // Update a smart contract’s parameters
```

```
        ExchangeCreateContract = 41; // Create a token trading pair on DEX
        ExchangeInjectContract = 42; // Inject funding into a trading pair
        ExchangeWithdrawContract = 43; // Withdraw funding from a trading
pair
        ExchangeTransactionContract = 44; // Perform token trading
        UpdateEnergyLimitContract = 45; // Update origin_energy_limit on a
smart contract
      }
   }
}
```

LGCY VIRTUAL MACHINE (LVM)

INTRODUCTION

LGCY Virtual Machine (LVM) is a lightweight, Turing complete virtual machine. Its goal is to provide a blockchain system that is efficient, convenient, stable, secure and scalable.

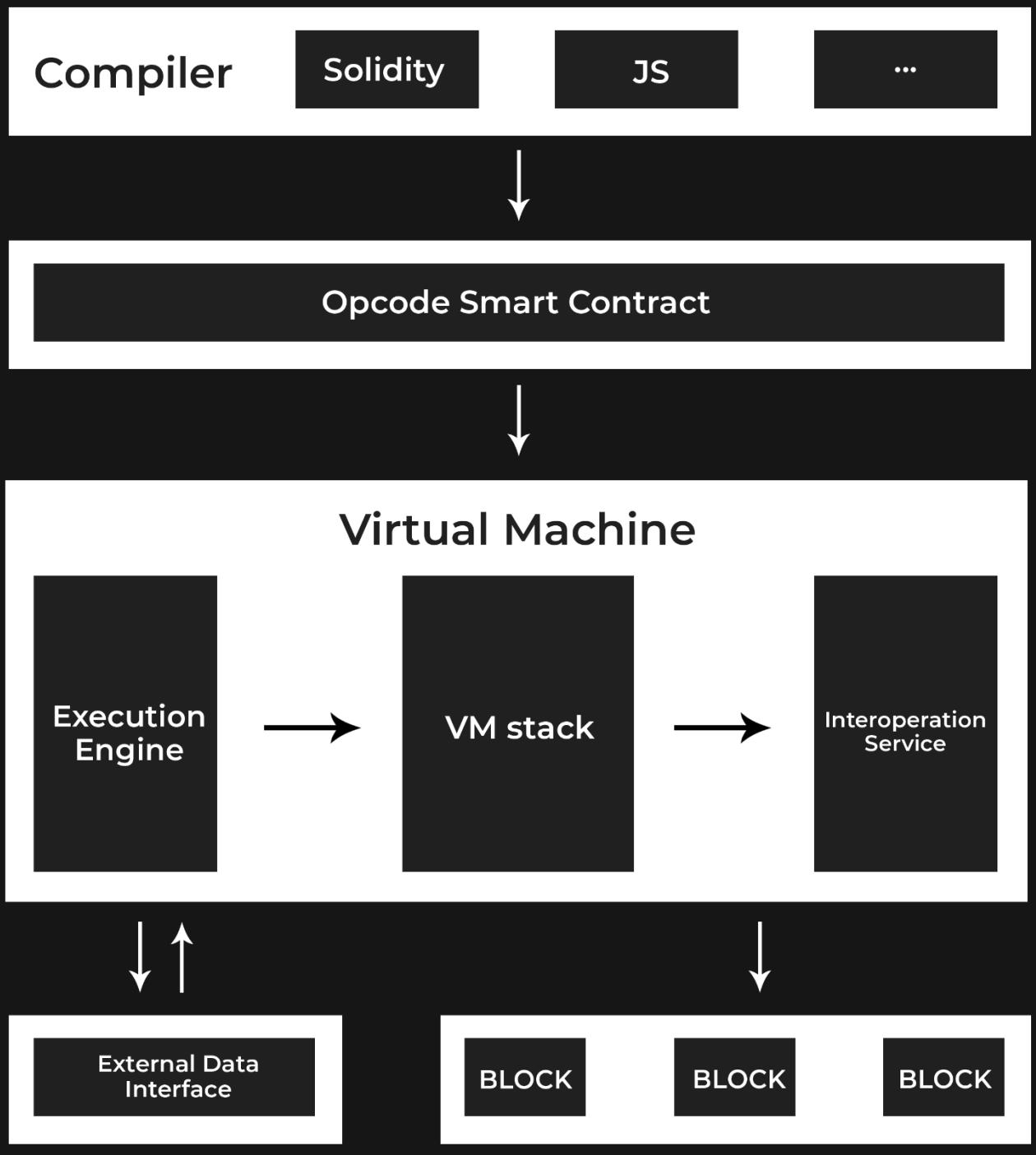
LVM initially forked from TVM[14] and can connect seamlessly with the existing Solidity smart contract development ecosystem. Therefore, LVM also supports DPoS consensus.

LVM employs the concept of Energy. Since LGCY will begin as a fork of TRON, the Energy system will be the same in the beginning. However, after LGCY Mainnet is enacted, energy will be paid out in a tradeable stable coin called LFUEL. The Energy system is different from the Gas mechanism on EVM,[15] operations of transactions and smart contracts on LVM are free, with no LGCY consumed. Technically, executable computation capacity on LVM is not restricted by total holding number of tokens.

[14] TVM: https://developers.tron.network/docs/virtual-machine-introduction
[15] EVM: Ethereum Virtual Machine (https://github.com/ethereum/ethereumj)

WORKFLOW

The compiler first translates the Solidity smart contract into bytecode that is readable and executable on the LVM. The LVM then processes data through opcode, which is equivalent to operating the logic of a stack-based finite state machine. Finally, the LVM accesses blockchain data and invokes External Data Interface through the Interoperation layer.


Compiler
Solidity
JS
...
Opcode Smart Contract
Virtual Machine
Execution Engine
VM stack
Interoperation Service
External Data Interface
BLOCK
BLOCK
BLOCK

PERFORMANCE

LIGHTWEIGHT ARCHITECTURE

LVM adopts a lightweight architecture which reduces resource consumption to guarantee system performance.

ROBUST

LGCY transfers and smart contract execution cost bandwidth points only, instead of LGCY, which exempts LGCY from being attacked. Bandwidth consumption is predictable and static since each computational step cost is fixed. This will become LFUEL consumption after LGCY Mainnet is enacted.

HIGH COMPATIBILITY

LVM is compatible with EVM and TVM, and will be compatible with more mainstream VMs in the future. This way, all smart contracts on EVM and TVM are executable on LVM.

LOW COST

Due to LVM’s bandwidth setup, development costs are reduced, and developers can focus on the logic development of their contract code.

Because of our major focus on supporting developers—especially startup projects who often lack funds—the low cost model LGCY offers will greatly help establish new and many dApp projects on the blockchain. LVM also offers all-in-one interfaces for contract deployment, triggering and viewing to offer the convenience for developers.

SMART CONTRACT

INTRODUCTION

A smart contract is a protocol that digitally verifies contract negotiation. They define the rules and penalties related to an agreement and also automatically enforce those obligations. The smart contract code facilitates, verifies, and enforces the negotiation or performance of an agreement or transaction. From a tokenization perspective, smart contracts also facilitate automatic funds transfers between participating parties should certain criteria be met.[16]

LGCY smart contracts are written in the Solidity language. Once written and tested, they can be compiled into bytecode, then deployed onto the LGCY Network for the LGCY Virtual Machine. Once deployed, smart contracts can be queried via their contract addresses. The contract Application Binary Interface (ABI) shows the contract's call functions and is used for interacting with the network.

[16] Adapted from "Smart Contract" in tron.network/static/doc/white_paper_v_2_0.pdf

ENERGY MODEL

The maximum energy limit for deploying and triggering a smart contract is a function of several variables:[17]

- Dynamic energy from freezing 1 LGCY is 50,000,000,000 (Total Energy Limit) / (Total Energy Weight)
- Energy limit is the daily account energy limit from freezing LGCY
- Remaining daily account energy from freezing LGCY is calculated as Energy Limit - Energy Used
- Fee limit in LGCY is set in smart contract deploy/trigger call
- Remaining usable LGCY in the account
- Energy per LGCY if purchased directly (10 LGCY = 1 Energy) = 100,000, GBs can vote on adjustment

There are two consumption scenarios to calculate for maximum energy limit for deployment and trigger. The logic can be expressed as follows:

```
const R = Dynamic Energy Limit
const F = Daily account energy from freezing LGCY
const E = Remaining daily account energy from freezing LGCY
const L = Fee limit in LGCY set in deploy/trigger call
const T = Remaining usable LGCY in account
const C = Energy per LGCY if purchased directly

// Calculate M, defined as maximum energy limit for deployment/trigger of
smart contract
if F > L*R
```

[17] Note: Since LGCY began as a TRON fork, these variables remain the same.

```
   let M = min(E+T*C, L*R)
else
   let M = E+T*C
```

DEPLOYMENT

When a LGCY Solidity smart contract is compiled, the LGCY Virtual Machine reads the compiled bytecode. The bytecode consists of a section for code deployment, contract code, and the Auxdata. The Auxdata is the source code's cryptographic fingerprint, used for verification. The deployment bytecode runs the constructor function and sets up the initial storage variables. The deployment code also calculates the contract code and returns it to the LVM. The ABI is a JSON file that describes a LGCY smart contract's functions. This file defines the function names, their payability, the function return values, and their state mutability.

TRIGGER FUNCTION

Once the LGCY smart contracts are deployed, their functions can be triggered individually either via LGCYStudio or through API calls. State-changing functions require Energy while read-only functions execute without Energy.

LGCY SOLIDITY

LGCY Solidity is a fork from TRON's Solidity language. LGCY adopted TRON's project to support LGCY and DRIPS units (1 LGCY = 1,000,000 DRIPS). The rest of the language syntax is compatible with Solidity ^0.4.24. Thus the LGCY Virtual Machine (LVM) is almost 100% compatible with EVM and TVM instructions.

TOKEN[18]

LRC-10 TOKEN

In the LGCY network, each account can issue tokens at the expense of 1024 LGCY. To issue tokens, the issuer needs to specify a token name, the total capitalization, the exchange rate to LGCY, circulation duration, description, website, maximum bandwidth consumption per account, total bandwidth consumption, and the amount of token frozen. Each token issuance can also configure each account's maximum daily token transfer Bandwidth Points, the entire network's maximum daily token transfer Bandwidth Points, total token supply, locking duration in days, and the total amount of tokens locked.

LRC-20 TOKEN

LRC-20 is a technical standard used for smart contracts implementing tokens supported by the LGCY Virtual Machine. It is fully compatible with ERC-20 and TRC-20.

The interface is as follows:

```
contract LRC20Interface {
   function totalSupply() public constant returns (uint);
   function balanceOf(address tokenOwner) public constant returns (uint
balance);
```

[18] This token model remains the same as TRON.

```
   function allowance(address tokenOwner, address spender) public constant
returns (uint remaining);
   function transfer(address to, uint tokens) public returns (bool success);
   function approve(address spender, uint tokens) public returns (bool
success);
   function transferFrom(address from, address to, uint tokens) public
returns (bool success);

   event Transfer(address indexed from, address indexed to, uint tokens);
   event Approval(address indexed tokenOwner, address indexed spender, uint
tokens);
}
```

From a developer's perspective, there are several differences between LRC-10 and LRC-20. Some of the key differences are that LRC-10 tokens are accessible by APIs and smart contracts while LRC-20 tokens allow for interface customization but are only accessible within smart contracts.

From a cost perspective, LRC-10 tokens have transaction fees that are 1000 times lower than LRC-20, but carry bandwidth costs for API transfers and deposits. Transfers and deposits in smart contracts for LRC-10 tokens cost both bandwidth and energy.

BEYOND

Since LGCY uses the same Solidity version as Ethereum and TRON, more token standards could be readily ported to LGCY.

TOKEN ISSUANCE

LGCY Network has a total supply of 100,000,000,000 tokens ($LGCY). We will begin with a 33,333,333,333 (33.3%) token release split up between our Private sale series A, Private sale series B, team, and our exchange liquidity. 20,000,000,000 of that number will go to the private sales A and B at a value of $5,000,000. 10,000,000,000 will go to the private sale series A at $0.0002, 10,000,000,000 will go to the private sale series B at $0.0003, and we will then launch liquidity on Uniswap. The remaining 13,333,333,333 will be held back from the release for the exchange (5,000,000,000), team (5,000,000,000) and a Developer Fund (3,333,333,333, explained in the next section). Nothing will change for the circulating supply during the first year.

At the second-year mark, we will begin with our quarterly token releases. Each year we will unlock 16,666,666,666 (16.66%) split up over four quarters (4,166,666,666 a quarter). Of that 16,666,666,666 each year, 1,666,666,666 (10%, 416,666,666 a quarter) will go to the Developer fund. This will create less sell pressure from token releases and a healthier dApp ecosystem.

At the end of the second year after 4 token releases, the team will be down to 50% voting control of the network. This trend will continue until at

the end of 3 years where we will burn the remaining 33,333,333,334 tokens. This final burn will be a good faith symbol of the LGCY team handing over the project to the community, allowing the network to be fully decentralized and increase the value of each remaining token for every investor.

DEVELOPER FUND

One of our biggest strengths will be our focus on our developer community. In order to foster a good relationship between the LGCY team and the developers building on our network, we will choose 12 projects each year to gift some funds as a thank you. This developer fund will only last for the last 2 of our 3-year voting control of LGCY Network before we hand it over to the community. Over the course of the 2 years we will choose 24 projects to gift 277,777,777 LGCY tokens to help with costs and to show appreciation for our community.

In order to qualify for this gift, a project must be actively developing on the LGCY Network. The application process will be made available on our website at www.lgcynetwork.com

GOVERNANCE

GOVERNING BRANCHES

GENERAL

Every account in the LGCY Network can apply and have the opportunity to become a Governing Branch (denoted as GB). Everyone can vote for GB candidates. The top 27 candidates with the most votes will become GBs with the right and obligation to generate blocks. The votes are counted every 6 hours and the GBs will change accordingly.

To prevent malicious attacks, there is a cost to becoming an GB candidate. When applying, 9999 LGCY will be burned from the applicant's account. Once successful, such account can join the GB election.

When LGCY Mainnet is enacted we will begin to split up the 27 GBs into three branches of government (the Libertas Protocol). The Executive Branch (EB) will have no ability to write proposals, that power will lie solely in the Legislative Branch (LB). The EB will only have the power to veto what is proposed by the LB. The Judiciary Branch (JB) will have the power to overrule any vetoes issued by the EB.

The top 9 GBs will function as the EB, the middle 9 as the JB, and the bottom 9 as the LB. This division of power will help limit the authority of the

bigger GBs, while giving a major voice to the smaller ones. This system creates a harmony amongst the powers that will ensure no single individual is able to gain control.

ELECTION

LGCY Power (denoted as LP) is needed to vote and the amount of LP depends on the voter's frozen assets (LGCY).

LP is calculated in the following way:

1 LP = 1 LGCY frozen to get bandwidth

Every account in the LGCY Network has the right to vote for their own GBs.

After the release (unfreeze, available after 3 days), users won't have any frozen assets and lose all LP accordingly. As a result, all votes become invalid for the ongoing and future voting round unless LGCY is frozen again to vote.

Note that the LGCY Network only records the most recent vote, which means that every new vote will negate all previous votes.

REWARD

For the first year or so LGCY will operate like TRON while mainnet and the Libertas Protocol in the Governing Branches are being developed. Once the Libertas Protocol is implemented, the bandwidth, energy, and reward payout system will change.

The reward payout for being a GB in the LGCY Network ecosystem is called LGCY Fuel (LFUEL). We will be combining the LGCY reward and the TRON energy model into LFUEL which will be a stable coin. This will help alleviate the sell pressure from miners selling LGCY tokens and facilitate a growing decentralized stable coin ecosystem. Paying out GBs in LGCY is not a sustainable model, so changing it to a stable coin both adds value and increases the longevity of the network. Also, changing the energy model from a non trade-able energy to a trade-able fuel token both brings value to the ecosystem and provides a second avenue for collecting energy.

COMMITTEE

GENERAL

The committee is used to modify LGCY dynamic network parameters, such as block generation rewards, transaction fees, etc. The committee

consists of the 27 GBs in the current round. Only the LB will have the right to propose and vote on proposals. When a proposal receives 5 votes or more (simple majority), it is passed to the EB for veto or approval (simple majority to veto or approve). If the proposal is vetoed it is passed to the JB to rule (again, simple majority). The proposal is either accepted or rejected at this point. If the proposal is accepted the new network parameters will be applied in the next maintenance period (3 days).

CREATE PROPOSAL

Only the GB accounts have the rights to propose a change in dynamic network parameters.

VOTE PROPOSAL

Only committee members (GBs) can vote for a proposal and the member who does not vote in time will be considered as a disagree. The proposal is active for 3 days after it is created. The vote can be changed or retrieved during the 3-days voting window. Once the period ends, the proposal will either succeed or fail.

In the first year the proposal will either succeed with 19+ votes or fail. After the implementation of the Libertas Protocol, the Legislative Branch will vote with a simple 5-4 majority needed to pass the proposal.

CANCEL PROPOSAL

The proposer can cancel the proposal before it becomes effective.

STRUCTURE

GBs are the witnesses of newly generated blocks. A witness contains 8 parameters:

1. **address**: the address of this witness – e.g. 0xu82h...7237.
2. **voteCount**: number of received votes on this witness – e.g. 234234.
3. **pubKey**: the public key for this witness – e.g. 0xu82h...7237.
4. **url**: the url for this witness – e.g. https://www.noonetrust.com.
5. **totalProduced**: the number of blocks this witness produced – e.g. 2434.
6. **totalMissed**: the number of blocks this witness missed – e.g. 7.
7. **latestBlockNum**: the latest height of block – e.g. 4522.

8. **isjobs**: a boolean flag.

Protobuf data structure:

```
message Witness{
   bytes address = 1;
   int64 voteCount = 2;
   bytes pubKey = 3;
   string url = 4;
   int64 totalProduced = 5;
   int64 totalMissed = 6;
   int64 latestBlockNum = 7;
   bool isJobs = 8;
}
```

DAPP DEVELOPMENT

APIS

The LGCY network offers a wide selection of over 60+ HTTP API gateways for interacting with the network via Full and Solidity Nodes. Additionally, LGCY.Web is a comprehensive JavaScript library containing API functions that enable developers to deploy smart contracts, change the blockchain state, query blockchain and contract information, trade on the DEX, and much more. These API gateways can be directed towards a local privatenet, the Sinai testnet, or the LGCY Mainnet.[19]

NETWORKS

LGCY has both a Sinai testnet as well as a Mainnet. Developers may connect to the networks by deploying nodes, interacting via LGCYStudio, or using APIs via the LGCYGrid service. The LGCYGrid service consists of load balanced node clusters hosted on AWS servers worldwide. As dApp development scales up and API call volumes increase, LGCYGrid successfully fields the increase in API traffic.

[19] Each of these tools were adapted from TRON, please see the original work: tron.network/static/doc/white_paper_v_2_0.pdf

TOOLS

LGCY suite offers everything a developer needs to get their unique dApp up and running. LGCYStudio is a IDE or Integrated Development Environment that allows developers to deploy, compile, and debug their Solidity smart contracts. It also contains an internal full node where you can create a private local environment for smart contract testing, prior to development. LGCYWeb API library connects developers to the network via a wide selection of HTTP API calls wrapped in JavaScript. LGCYBOX is a framework that uses LGCYWeb API to allow developers to test and deploy smartcontracts. LGCYGrid is a load balanced and hosted API that lets developers access the LGCY, without the need for their own node. LGCYGrid allows access to Sinai testnet and LGCY Mainnet.

RESOURCES

The LGCY Developer Hub is a comprehensive API documentation site tailored towards 12 developers wishing to build on the LGCY network. The Developer Hub provides a high-level conceptual understanding of LGCY and walks users through the details of interacting with the network. The guides

walk developers through node setup, deployment and interaction with smart contracts, API interaction and implementation, building sample DApps, and using each of the developer tools. Additionally, developer community channels are available through Discord.

CONCLUSION

LGCY seeks to decentralize the governing system of the traditional DPOS model by adding the Libertas Protocol to ensure fair representation. The Libertas Protocol in conjunction with purposeful support of developers on our network will ensure that LGCY will be the best dApp blockchain in existence.

Legal Disclaimer

No regulatory authority has examined or approved of any of the information set out in this Whitepaper. No such action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this Whitepaper does not imply that the applicable laws, regulatory requirements or rules have been complied with. Prospective purchasers of LGCY tokens (as referred to in this Whitepaper) should carefully consider and evaluate all risks and uncertainties associated with LGCY, the Distributor and their respective businesses and operations, the LGCY tokens and all information set out in this Whitepaper prior to any purchase of LGCY tokens. If any of such risks and uncertainties develops into actual events, the business, financial condition, results of operations and prospects of LGCY and/or the Distributor could be materially and adversely affected. In such cases, you may lose all or part of the value of the LGCY tokens. No information in this Whitepaper should be considered to be business, legal, financial or tax advice regarding LGCY, the Distributor, the LGCY tokens. You should consult your own legal, financial, tax or other professional adviser regarding LGCY and/or the Distributor and their respective businesses and operations, the LGCY tokens. You should be aware that you may be required to bear the financial risk of any purchase of LGCY tokens for an indefinite period of time.