Table of Contents: P

Part I: Introduction to ThunderCore

ThunderCore is a public, permissionless, Ethereum Virtual Machine (EVM) compatible blockchain running on the worlds' leading Proof-of-Stake consensus mechanism. The ThunderCore blockchain is open to the public, allowing participants from around the world to join in to secure and verify the record to reflect an accurate state of affairs.

ThunderCore isn't just a platform for recording transactions. ThunderCore can execute "smart contracts"– computer programs that directly control asset transfer between parties with objective and fair logic. ThunderCore has been designed to be compatible with Ethereum, the most popular smart contracting platform. Any contract or application deployed on Ethereum can be redeployed on ThunderCore in minutes.

In this introduction, you will be exposed to why blockchains are needed, what they are, how ThunderCore is different. **Part II** is a technical overview of the ThunderCore protocol. Finally, **Part III** discusses where ThunderCore will go from here.

Why Blockchain?

Interactions between any two or more entities require trust–the ability to rely and act on the commitments of others. Trust can be expensive and volatile, and the "cost of trust" in our society cannot be overstated. However, with modern technology trust no longer has to be provided by individuals or institutions. Instead, trust can be provided by and guaranteed in code. Leveraging public/private key pair cryptography and novel incentivization and punishment schemes, new protocols enable individuals and other entities to trust the reliability of records without relying on third-party overseers. The human factor, prone to unclear expectations and only enforceable through costly and convoluted legal agreements or even just veneers of decency, is removed. Operations are more efficient, reliable, predictable and less costly for all parties.

Initial Traction

While the power of this new protocol is only beginning to be understood, blockchain has already found traction in the **finance** industry. Imagine, a global and decentralized financial system without borders or intermediaries, where contracts are tamper-proof and completion guaranteed by code.

Currently about 1.7 billion people around the world do not have access to common financial products, dramatically limiting their options for investments and capital gains. However, with the astonishing low-overhead provided by blockchain, along with proliferation of smart-phones and digital literacy–financial instruments are now available where they had not been before.

Further Applications

Blockchain fundamentally changes assumptions about ownership, reliability, collaboration and trust. Whereas the full spectrum of what is possible with blockchain may not yet even be imagined, its potential to reframe and solve many existing problems is already apparent:

- **Anti-fraud:** The immutable, persistent nature of records on the blockchain enable rapid, inexpensive auditing to take place. This allows for solutions to combat fraud in a simple, straightforward way.
- **Crowdfunding:** Ample successful projects have already been crowdfunded publicly and anonymously using blockchain. The platform provides intrinsic payment processing and investors can be rewarded in tokenized assets.
- **Gaming:** Blockchain-powered games, from casual mobile gaming to dice or card games to lotteries and auctions, can all profit from the transparent and cost-saving nature of smart contacts.
- **Governance:** Both for private and public organizations, blockchains can deploy smart contracts that act as code-based constitutions prescribing actions and performance. These contracts directly collect votes from the stakeholders and execute the winning proposals. Actions to disburse funds or assets can be automated and verified without trust. Public institutions can also benefit from the immutable records of the blockchain by providing greater security, integrity and transparency to the voting process. These immutable records are readily auditable and impossible to censor.
- **Ownership:** Owning assets both in the real world and in the digital world is simpler, cheaper and more trustworthy using blockchain based solutions.
- **Personal Data Ownership:** Personal data such as internet browsing history and other digital footprints can be managed better using blockchain based key sharing techniques. Digital medical records, for example, can be stored on-chain, with access enabled by private keys. These records are tamper-resistant, secure and efficient, thereby improving the work of a physician and quality of care a patient receives.

The truth is, for all of the potential that blockchain technology has, developers have only just begun to unlock its full potential. New ways to store value, automate processes and ensure that trust–previously an expensive and critical component–is guaranteed at a scintilla of the cost are possible now, but there is still much to be done.

A critical mass of developers, practitioners, and users has come into being. The required threshold technology--in the form of internet access, telecommunications, and personal computing devices--has reached the necessary level of market saturation. . Within the next few years, we will see an innovative explosion the likes of which we have not seen in a generation. The blockchains' time has come.

What is a Blockchain?

A blockchain is a series of records, time-stamped and immutable, operated by decentralized servers, also known as **nodes**. A blockchain protocol defines how a network of nodes communicate and how new blocks (containing signed transactions) are added to the chain. There are, of course, various breeds of protocols–each one deploying their own flavor of performance and incentives. Regardless of the flavor, blockchains share invariants that can be used as foundations for building new economic systems.

Immutability

Through cryptographic algorithms, data contained within the blockchain is codified and immutable–meaning that it cannot be changed or deleted. The very nature of blockchain is such that any attempt to change an earlier transaction would create a cascading effect through the rest of the chain, breaking consistency.

This feature provides for a tamper-proof record that can be used to represent anything from bank balances to the results of a political poll. More advanced protocols, like ThunderCore, have "**if-this-then-that**" logic trees to enable programs called **smart contracts**. As mentioned above, these smart contracts can directly control the transfer of assets between parties under specified conditions without an intermediary. The rules of these interactions are set down in code and are followed as though law–hence the common refrain: **code is law**.

Open Source

While some protocols are **permissionless**, allowing any computer (node) to join their network, and others are **permissioned**, maintaining a vetting mechanism to only allow "approved" computers to participate. However, almost all blockchain protocols exist as an open-source software project. Like ThunderCore, these protocols are building economic systems. By nature, robust economic systems are those that allow for a free flow of value into and out of the system.

Furthermore, a blockchain profits from allowing as many developers and participants as possible to access, review, use and develop its code. That way any consumer, developer or business can use the protocol, build on the network and add value to the system by deploying their services on top of it. Open access ensures that the barriers to entry are as low as possible and encourages as many new players and entrants into the field to compete for users–improving the overall value of the system.

Decentralization

A network of computers (nodes) is required to support a blockchain network. This is not because the network requires a massive amount of processing power. Rather, as there are more copies, the blockchain becomes more distributed, robust, and stronger against faults and attacks such as network outage, collusion or corruption.

Arguably, decentralization is the core value that started the blockchain revolution. It has become increasingly apparent that the centralization of the web by governments and corporations comes at an unavoidable cost on privacy, control, and efficiency. Modern techniques in economics and cryptography enable blockchain to challenge the existing status quo and provide alternatives giving full agency to its users and resilience against censorship, corruption and failure.

The ThunderCore Blockchain

As blockchain protocols first came into usage, there were a few significant problems. The first of which has been described as the **blockchain trilemma**. It has been the common understanding that a blockchain could not achieve security, decentralization and scalability all at once–that one of those three would have to be sacrificed in pursuit of the other two. Many blockchains have thus sacrificed scalability, settling for low throughput and slow confirmation speeds.

ThunderCore believes that our consensus protocol is the best of the Proof-of-Stake (defined below) protocols in that we have resolved the trilemma. The PaLa consensus protocol–described in **part II** of this document–permits an open, permissionless network to achieve over 2,000 transactions per second and one second confirmation time. As a public permissionless blockchain, our consensus mechanism allows any token holder to become a proposer or a voter and contribute to security and decentralization of the chain.

Permissionless

As stated above, blockchains can be permissioned or permissionless. That said, ThunderCore believes strongly that permissionless (open participation for everyone, also known as public) is the best way to maximize the value of the blockchain technology.

Another way to look at a blockchain is as an innovative, technology enabled socioeconomic system. Thus we believe that a permissionless system is the only way to achieve the true benefits of a blockchain. No gate-keeper should be responsible for vetting whether or not individuals should have access to services, benefits and financial tools enabled by technological progress. In addition, it is easy to reason, as a platform for the transfer of value, that open access is a fundamental tenet to ensure a healthy growth cycle. This system is better protected through a higher level of decentralization. By deploying a permissionless blockchain network, the level of decentralization increases thereby improving the overall security of the network.

Ethereum Virtual Machine Compatibility

When ThunderCore was being designed, over 200,000 developers had been trained to program on Ethereum. This was a talent pool too rich to pass up. Therefore ThunderCore is made to be fully EVM (Ethereum Virtual Machine) compatible. This means that any application, smart contract or tool built on Ethereum can operate on ThunderCore. Developers can easily migrate their decentralized apps (DApps) in minutes and enjoy an immediate boost in performance, experience, and affordability.

As expected, within the first two weeks after ThunderCore mainnet launched, multiple development teams have ported Ethereum games over to ThunderCore. These games, compared to their Ethereum counterparts, quickly gained magnitude more daily transactions and players due to high speed and throughput of ThunderCore.

Scalable and Fast Consensus

The ThunderCore Proof-of-Stake blockchain relies on the PaLa consensus protocol to drive consensus. Consensus is the process upon which the network agrees on a single canonical and immutable chain of blocks containing user transactions and is the core of any blockchain protocol. The PaLa consensus protocol is the most performant, efficient and simple of its class and has rigorously proven security properties. It is the cutting edge technology that enables our fast and scalable public blockchain. For a detailed explanation of ThunderCore's technology, please see **part II** of this document which is intended for an audience with technical background or interests.

Part II: ThunderCore Technology Overview

The recent and explosive growth of bitcoin has also brought increased attention to distributed byzantine fault tolerant (BFT) classical consensus protocols. These protocols offer increased performance, deterministic finality, and the promise of working around scalability and energy consumption problems that restrict current Proof-of-Work (PoW) blockchains. The tradeoff is that these protocols require more complicated mechanisms and face different obstacles for open and permissionless participation.

Introduction to ThunderCore's Consensus

ThunderCore was originally designed around the Thunderella consensus algorithm that brought together the best of permissioned classical consensus protocols and decentralized Nakamoto consensus. Since then our technology has evolved. We have discovered new consensus mechanisms with exceptional properties. This document introduces the PaLa[1] consensus protocol that we use for ThunderCore's public blockchain. Our consensus protocol is developed by leading researchers in cryptography and distributed consensus and is backed by rigorous proofs that guarantee consistency and liveness.

A typical classical consensus algorithm[2] proceeds in rounds and tolerates up to ⅓ corruptions. In each round, a single proposer will propose new blocks to a set of voters who form a committee. A collection of at least ⅔ of the committee's votes is referred to as a notarization and indicates that a sufficient number of nodes have agreed on a certain state. In normal operation, this process is fast, efficient and safe. However, we need a mechanism to recover in the event that the proposer crashes or misbehaves. This process is referred to as a proposer switch. To complicate matters, voters must agree on the last confirmed state–a responsibility that was previously managed by the proposer. For this reason classical consensus algorithms require two sets of notarizations on a block in order for it to be finalized (included into the blockchain's immutable history). The first one confirms the state and the second one confirms that enough committee members have acknowledged the first confirmation. We will soon see how the PaLa consensus algorithm greatly simplifies the switching mechanism and optimizes the notarization process.

---

[1] "pili-pala" is the sound of thunder in Chinese; it also implies fast, furious and streamlined. PiLi is another consensus algorithm that operates under synchronous network assumptions.

[2] Other classical consensus algorithms use different terms to describe the same or similar concepts described here. The terminology used in this whitepaper is consistent with the terminology used in the PaLa research paper.

The PaLa Consensus Protocol

PaLa is a blockchain consensus protocol based on partially synchronous network assumptions and tolerates up to ⅓ corruptions. Below, we describe a simplified version of the protocol called **Basic Pala** to illustrate its simplicity and effectiveness. Basic Pala is the foundation for understanding the complete version of our protocol which is outlined afterwards. The full details of the PaLa protocol are readily available in the **PaLa research paper**.

Setup

Assume a fixed **committee** of **voters**. How these nodes are chosen is described later. Each node maintains a **local epoch** counter *e* and a local view of the blockchain. Each block contains an epoch number, a list of transactions, and the hash of its parent block. The epoch number of a chain is defined as the epoch number of its last block. Each epoch has a single unique **proposer** which is known to all nodes in the network. In this simplified version, every voter is also a proposer for some epochs.

Consensus proceeds one block at a time. Proposers propose block if they are eligible to propose in the current epoch. Voters vote on blocks if a set of conditions are met. A collection of ⅔ of the committee's votes on a single block is a **notarization** for that block. If there is a notarization for a block, the block is **notarized**. Each block has an epoch which advances monotonically. If an epoch *e* block has an epoch *e-1* parent, the block is a **normal block** otherwise it is a **timeout block**. A block is **finalized** if it is the parent of a notarized normal block. A finalized block is part of the immutable history of the blockchain and indicates consensus has been achieved.

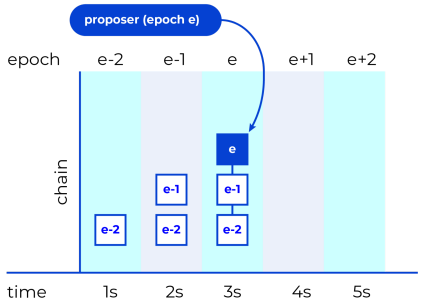
Protocol

Each node keeps their local current blockchain **fresh**. Whenever it sees a valid blockchain that is **fresher** than its current chain–has a higher epoch number than the epoch number of their current chain–they switch to this chain. A valid blockchain should satisfy the following conditions:

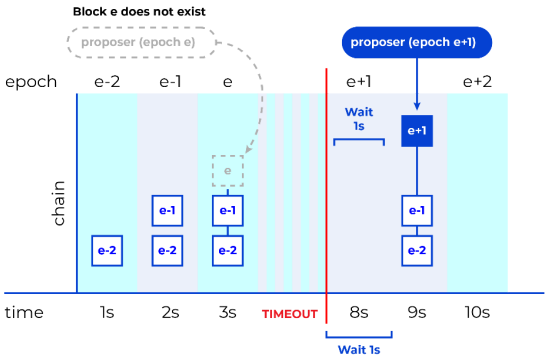
1. The epoch numbers of all blocks should be strictly increasing.
2. Every block in the blockchain is notarized

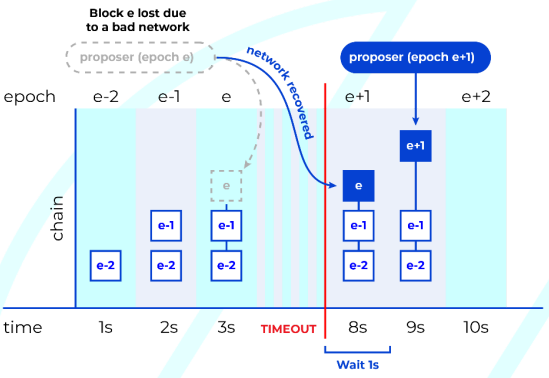
In addition each node will do the following:

- Increase local epoch counter to *e* if
    - Their current local epoch is smaller than *e* AND
        - They see a notarized chain for epoch *e-1* OR
        - They see at least ⅔ committee members' validly signed *clock(e)* messages.
- If they have stayed in epoch *e-1* for more than a fixed amount of time
    - Broadcast the message *clock(e)*.
- If they are the proposer of epoch *e*
    - If their current chain ends with the block of epoch *e-1*, immediately propose a new block for epoch *e* extending their own current chain



- If the epoch number of their current blockchain is *e* (due to a timeout), wait for a fixed period of time (hoping to receive a fresher chain) and propose a new block for epoch *e* extending their own current chain.





- On receiving a block proposal for epoch e from the proposer of epoch *e*, sign the proposal and multicast the signature if the following conditions hold:
    - They have seen the notarization for the parent block of the proposal.
    - The block is at least as fresh as their current chain at the beginning of epoch *e*.
    - They have not signed any other block for epoch *e*.

With these simple rules, the PaLa consensus protocol achieves liveness and consistency. The PaLa protocol is simple and rigorously proven. PaLa is designed based on partially synchronous network assumption. It is inherently partition tolerant and responsive. If there is ever a network partition, there is only a temporary outage in liveness. It will resume progress when the partition heals with no conflict in state.

Committee Reconfiguration and Doubly Pipelined PaLa

In a real world Proof-of-Stake blockchain, we want to periodically switch committees as stake changes hands in the system. The PaLa consensus algorithm supports fast and seamless committee switches. The full mechanism is not detailed here. The basic idea is that the previous committee must finalize a special **reconfiguration block** before the next committee may serve to ensure consistent continuity.

ThunderCore will deploy the full PaLa protocol described in the paper which is also called **Doubly Pipelined Pala** which supports the proposal **pipeline feature**[3]. This version allows for consensus to proceed *k* blocks at a time where *k* is a protocol set parameter. The proposal pipeline allows newer blocks to be buffered while older ones are still being voted on. Based on empirical testing, we found that when network latency is high relative to block times, a higher *k* value can improve throughput volume.

---

[3] Also referred to as doubly streamlined in an unpublished version of the PaLa paper.

Summary

From a performance standpoint, PaLa is a significant improvement on prior classical consensus protocols that require two rounds of voting per block and O(n^2) messages. PaLa makes use of the *pipelined BFT* [4] idea where the second round of voting is 4 piggy-backed on the first round of voting for the next block. The active proposer uses BLS multi-signatures to collect votes and distribute notarizations. Together with a multi hub and spoke network topology PaLa achieves consensus with O(n) messages.

Newer BFT consensus algorithms such as Tendermint, FBFT, Casper FFG and Hotstuff use many of the same innovations as PaLa. However, none of these algorithms are as simple, elegant and optimal as PaLa.

We expect BFT consensus to be equated with PaLa the same way Distributed Hash Table (DHT) is equated with Kademlia.

ThunderCore is also committed to providing the best resources to help others learn more about PaLa. The **reference implementation** of the PaLa consensus protocol is already released and contains thorough documentation and educational material.

Proof-of-Stake [5]

With PaLa we have a mechanism for robust committee reconfiguration and proposer switch. What remains is an incentive compatible Proof-of-Stake (PoS) policy for consensus nodes to be elected. We opt for a simple top-K voter election scheme that progresses in sessions. Each session lasts 3 hours. In each session, voters submit bids to participate as a consensus node in the next session.

---

[4] Pipelined BFT is this first pipeline of doubly pipelined Pala, the second pipeline being the k blocks mentioned above.

[5] Proof-of-Stake is originally a Nakamato consensus algorithm using the hash of miner's stake instead of a randomly chosen nonce for block selection privilege. More commonly, it refers to any blockchain that relies on staking to choose consensus. We follow this ubiquitous terminology appropriation. We still think the etymology PoS is important enough to know in the current state of affairs to mention in this footnote.

A bid includes the following parameters:

1. stake in amount
2. public signing key
3. reward address
4. desired gas price
5. proof of knowledge of signing secret key

At the end of a session, all valid bids are collected and sorted by stake. The top 32 form the new committee for the next session which will take place once the PaLa reconfiguration block is finalized. To limit disk growth and network bandwidth, the gas price is set to the ⅔ ascending median of the desired gas price of the committee such that ⅔ of voters would be happy with the set gas price or a higher one.

Proposer Election

For a fully permissionless protocol, we require a fair proposer election policy. We expect any proposer to also function as a voter because a voter's hardware requirements are a strict subset of the proposers'. To facilitate this, any voter candidate can optionally indicate if they want to be a proposer in their bid.

A weighted list of proposers are thus generated from this subset of committee members.

To be a proposer, a consensus node must bid with the following additional fields:

1. public URI
2. public proposing key
3. optional message

For now, proposers are sorted by stake and chosen based on a round robin policy where in each session a proposer is only allowed to propose a number of blocks proportional to its stake. Thus, every honest well connected proposer will serve at least once per session. Even a majority consortium of byzantine proposers can only stop liveness or censor transactions for a short duration due to these forced proposer switches. Having at least one well-behaved proposer ensures all valid transactions will be included on the chain eventually.

Rewards and Punishment

This section explains ThunderCore's Proof-of-Stake design. At a very high level, we would like a system that achieves the following:

- **Incentive compatibility:** our incentive mechanism should reward honest and high-performance participation.
- **Economically robust to attacks:** in the equilibrium state, there is sufficient stake protecting the blockchain, such that the cost of the attack increases proportionally with respect to the economic importance of the currency.

Thus honest behavior is ensured through rewards and incentives. Our design incurs a very high cost of attack for any node acting byzantine in an attempt to break consistency. For nodes acting faulty or byzantine in an attempt to break liveness, a smaller penalty is incurred and no rewards are given.

Slashing

Staked-in funds are frozen in a smart contract for 24 hours after the voter's session is finished during which time if evidence of bad behavior is recorded, some portion of the staked-in funds will be slashed.

If cryptographic evidence of a voter's votes on two conflicting proposals are submitted to the chain while their stake is still frozen, 10% of their frozen funds are sent back to the reward pool with a small portion given to the reporter as a reward. The remaining funds are jailed for 1000000 blocks. Jailed funds are not available for the voter to stake in with again. To prevent nothing-at-stake attacks, online nodes also must reject blocks that do not extend from blocks within the freezing period.

Given that unintentionally offline nodes will not automatically bid for the next session, ThunderCore will not slash for failure to vote. There is still a high opportunity cost for non-voting due to rewards only being given to participating voters. If we determine the threat of a liveness attack exceeds the opportunity cost, we will add in a slashing condition for non-voting.

Part III: Where do we go from here?

As a technology company, we have many exciting ideas to dive. Below are just some projects we will be looking into in the future:

- **Thunderella** is still a great and unique consensus protocol. It offers asynchronous 1 voting round finality times with synchronous fault tolerance bounds. This is twice as fast as any other protocol. We believe Thunderella may be a great fit for consensus on layer 2 side-chains if not as a standalone protocol for layer 1 blockchains. A production ready reference implementation for Thunderella is still on our future roadmap.
- **Pili**[6] is an incredibly unique synchronous consensus protocol with asynchronous performance. We believe a production ready reference implementation could be of great value to the blockchain community.
- **Trustless random number generation** enables easy to implement unbiasable and unpredictable random behavior on chain. We will be looking into an implementation based on cryptographic multi-party computation using the same assumptions as the PaLa protocol.
- **Trustless cross-chain communication** enables trustless exchange of information across independent blockchains.
- **Paella** integrates slow chain fallback into PaLa for ½ liveness fault tolerance and censorship resistance. The protocol is outlined **below**.
- **Sub second blocktimes** to maximize throughput and leverages the full potential of doubly pipelined PaLa.
- **Storage fees** to address long term blockchain disk usage issues.
- **100,000+ TPS** is possible with EVM optimizations and optimistic concurrent processing together with proposal pipelining to work around the network bottleneck. For now, 2,000 TPS is sufficient for all blockchain usage. ThunderCore will optimize the protocol over time to meet users' demands.
- **Sharding** is a natural extension of PaLa and our PoS scheme. A super committee is elected on a beacon chain and a subset of voters are randomly assigned to each shard. Sharding imposes new challenges and tradeoffs and has the potential to improve scaling by another order of magnitude.

---

6 **https://eprint.iacr.org/2018/980.pdf**

Paella

The PaLa research paper outlines the “stability-favoring” and “democracy-favoring” approaches for electing proposers. The latter approach has more frequent mandatory switches allowing more nodes to take on the proposer role in a meaningful way. ThunderCore will opt for the “stability-favoring” approach in PaLa meaning few scheduled proposer switches. This approach is best for consistent high throughput.

To add more decentralization into the system, we introduce yell messages from the **Thunderella protocol**. Yell messages are transactions on another blockchain with a valid signed ThunderCore transaction in its data field. The signed transactions are expected to be included in the ThunderCore blockchain. Honest voters do not vote for blocks that do not include yell messages (forcing a proposer switch). Honest full nodes reject blocks that do not include valid yell messages.

Yell messages also give a new fallback mechanism to recover from a hung committee should such an event ever occur. If no proposer can collect ⅔ votes to notarize the next block, transactions can still be processed by means of yell messages which are guaranteed to be included in a notarized block at some point in the future as determined by the protocol. In particular, node operators can still send in new bids and elect a new proposer to reconfigure the consensus participating nodes such that it can make progress normally again.

In Thunderella, periodic “alive messages” needed to be posted on the slow chain in order for consensus nodes to coordinate a fallback and recovery. With PaLa, consensus nodes can directly coordinate a proposer switch among themselves so alive messages are no longer required. Alive messages can still be useful for protecting against long range attacks and may be part of the ThunderCore protocol if we determine long range attacks are a real threat to blockchain security.

We affectionately call this modification to PaLa Paella. Paella is a portmanteau of PaLa and Thunderella. It is also the name of a delicious **Valencian rice dish**.

Part IV: The Pillars that Drive Us

ThunderCore is a technology and mission driven organization. In closing, we would like to share the core beliefs that drive us forward.

We believe:

Blockchain will fundamentally change the way humans interact with technology.
We are at the opening chapters of a new epoch of human-machine interaction. Blockchain technology will fundamentally change the way that humans interact with technology. The massive for-profit institutions running centralized technology have hit the limits of their transformative potential but blockchain has only just begun.

People will increasingly rely on, and trust, decentralized authority and service providers.
Individuals are beginning to question the underlying assumption of trust that underline day-to-day interactions with large centralized service providers. As each day goes by, people are beginning to be more and more comfortable putting their trust in code, in decentralized services and authority. Blockchain is built on these fundamental principles and we expect to see massive adoption in the near future.

Unrestricted access to the benefits and value created by technology is a universal right.
Blockchain has incredible implications towards equality on a global scale. Decentralization and openness are required for equality. Unrestricted access to our platform, and services on it, will benefit great numbers of people around the world who will, in turn, bring their own value to the platform.

The future is open, decentralized, and transparent.
The future is not without its challenges. But we stand at a fantastic moment in history where the future is not written and we have the opportunity to be the scribe for the next chapter. The promise of blockchain technology is that the future will be one that is more open, more decentralized and more transparent. This is a future that ThunderCore believes in and is building towards.