WHITEPAPER

teaproject.org

CONTENTS

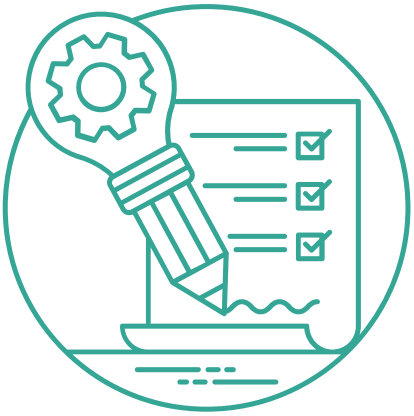
teaproject.org

EXECUTIVE SUMMARY

teaproject.org

The TEA Project is a decentralized cloud computing platform that hosts rich, decentralized applications that run completely serverless inside of contributor's compute nodes. The TEA Project builds upon web3 technologies like blockchain as well as trusted hardware to unlock a decentralized compute layer that runs on par with traditional cloud apps. TEA leverages hardware security modules (HSM) within decentralized compute nodes to ensure a secure app execution environment. The TEA Project's state machine runs decentralized and continuously using time as a root of trust to order transactions.

The TEA Project aims to become an ideal entry point for developers looking to migrate from web2 to web3. We don't require devs to learn a new programming language or adapt to blockchain concepts within their workflow. Developers can deploy TEA Project apps (known as TApps) using programming languages and methodologies they're already familiar with. Our platform includes the same three-tier architecture that's prevalent in web2: a front-end, a back-end, and a database. And developers can use any number of popular programming languages as our output format (WebAssembly) has wide compatibility. We believe that the initial blockchain-based forays into decentralized compute has revealed the limitations of smart contract-based decentralized networks.
TEA has used elements of blockchain along with other roots of trust to make an evolutionary leap bringing decentralization into the realm of cloud computing. By using non-traditional consensus, the TEA Project is able to create a decentralized app platform that's unstoppable, censorship free, and respects user privacy.



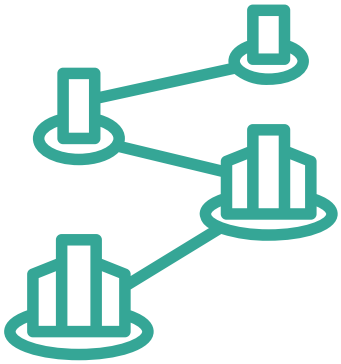
THE PROBLEM

teaproject.org

Blockchains are an incredible invention that brought the benefits of decentralization to a wider audience. Layer-1 blockchains run dApps written as smart contracts and have gained a strong toehold in niches like DeFi. These blockchains are further augmented by web3 innovations like IPFS, a peer-2-peer file system that stores files decentralized. But smart contracts by themselves cannot currently run complex algorithms. Attempts to do so have shown smart contracts to be too slow or too expensive as they lack the processing power compared to modern cloud computers. IPFS is useful for serving static files but isn't capable of serving dynamic content by itself. Valuing decentralization has up to now meant dealing with the limitations of consensus and the limited computing capabilities of smart contracts. Developers wishing to deploy to web3 currently have to deal with the limitations of technologies like blockchain and IPFS. For decentralization to gain a larger marketshare among app developers, dApps will need to approach the speed and complexity of web2 cloud-based apps.

A logical question is whether smart contract-based decentralized systems can ever begin to compete with general computing for marketshare? Smart contract-based blockchains like Ethereum have always been affected by scalability issues. Ethereum and other chains have dealt with this issue by either:

- **Attempting to increase scalability by tweaking the consensus mechanism.**
- **Utilizing a layer-2 to clear transactions off-chain.**

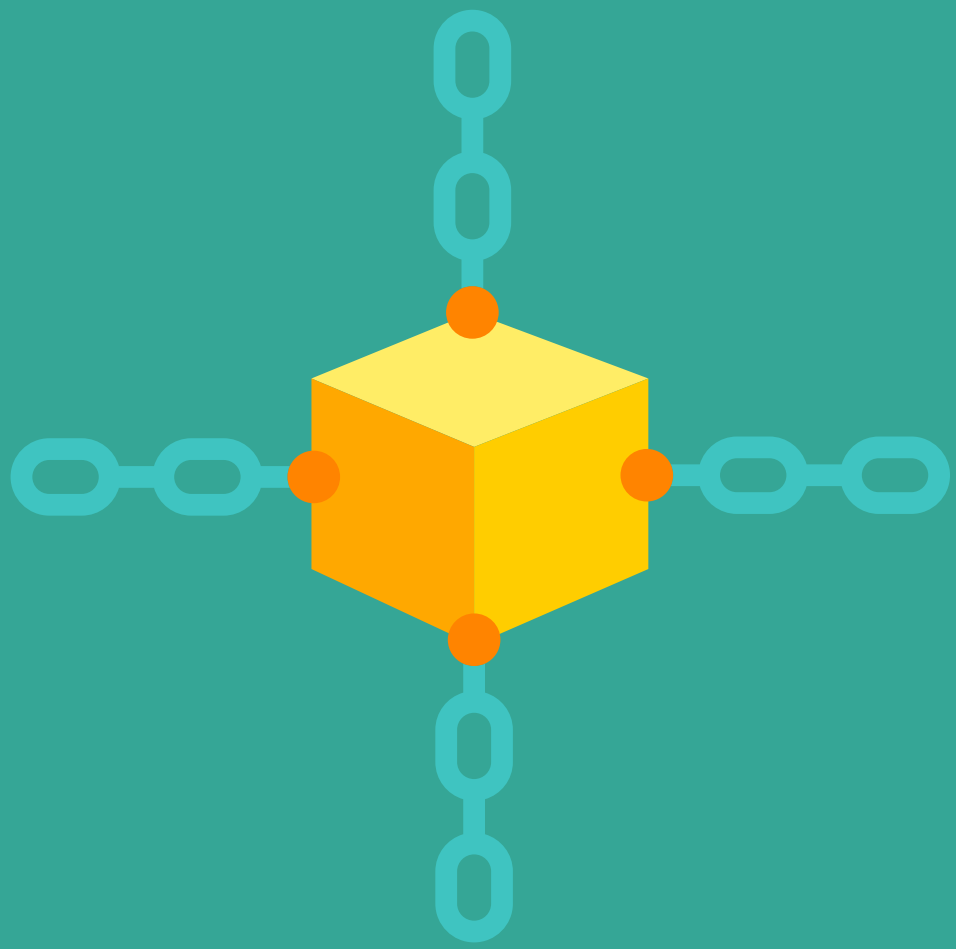


THE PROBLEM

teaproject.org

But tweaking the consensus mechanism can only go so far when lots of time is wasted waiting on consensus being reached among peer nodes. The fundamental problem is that the decentralized nodes don't trust each other and have to find clever ways to work around that limitation.

Although layer-2s can speed up transaction throughput, these introduce an extra verification step that's computationally expensive. Layer-2 also isn't sufficient for privacy as transactions are rolled up back to layer-1. And even after scalability tweaks, these chains still have the limitations inherent in smart contracts. For decentralized computing to capture web2 market share, we need a new infrastructure that goes beyond the ledger-bound smart contract.

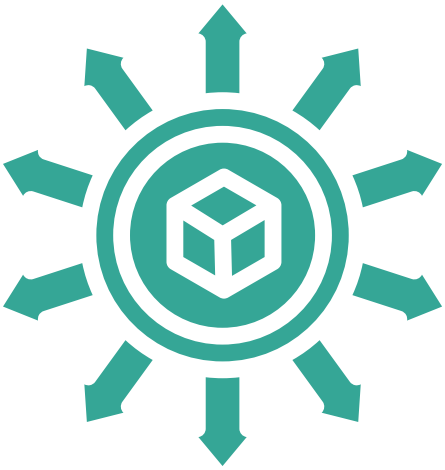


THE SOLUTION

teaproject.org

The TEA Project moves decentralized computing from its current focus (on blockchain and smart contracts) and into cloud-speed app execution. Smart contracts run like stored procedures of a simple database whereas the TEA Project unlocks decentralized front-end and back-end server tiers in addition to a decentralized database. The TEA Project builds upon blockchain by incorporating it as one of its roots of trust but combines them with two other roots of trust to unlock full-speed decentralized computing.

One of the enduring problems of decentralized systems is how to circumvent the long wait times for consensus inherent with decentralized nodes. The TEA Project avoids slower forms of consensus through the use of three roots of trust: blockchain, trusted hardware, and time. These three roots of trust are used to create two separate consensus algorithms:



1-Proof of Time:

The consensus mechanism for a decentralized state machine is ultimately trying to get its participants to agree on the same sequence of transactions. To this end, the TEA Project's state machine runs a separate consensus based on time. Specifically, TEA uses the timestamps as reported by the compute nodes' atomic clocks (or GPS) units to order transactions coming to its state machine. This is the same algorithm used by the Google Spanner database. Since time is trustable in our universe, TEA uses it as the basis of transaction ordering in its state machine

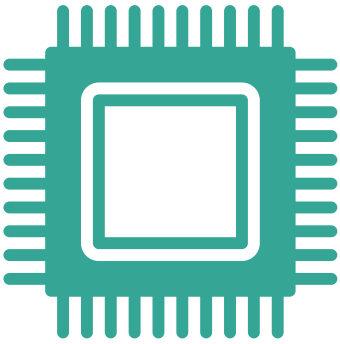
INTRODUCING THE TEA PROJECT

teaproject.org

The TEA Project presents a decentralized compute infrastructure that respects private user data while allowing apps to run both decentralized and on par with cloud-hosted apps. The TEA Project offers a compute layer that sits adjacent to blockchain while offering the same level of decentralization and trust. We're a fully decentralized compute layer without the limitations of smart contracts.

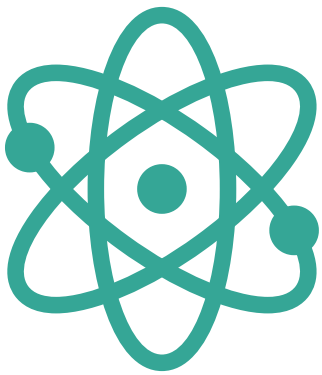
The TEA Project creates a decentralized compute network capable of running full speed apps by combining three roots of trust:

1. Blockchain (Ethereum) is used to store the remote attestation results of TEA's compute nodes and to enforce economic incentives for node runners (miners).



2. TEA uses the onboard TPM chips of the compute nodes to run compute tasks privately. A natural byproduct of this architecture is that user data remains private and isn't leaked outside the enclave.

3.Atomic clocks form the basis of TEA's Proof of Time consensus which reliably orders transactions from the decentralized nodes. This forms the basis of the state machine that maintains the present status of all the apps running on TEA's compute layer



THE SOLUTION

teaproject.org

2-Proof of Trust:

A central TEA principle is that if the execution environment can be trusted, then the execution result can be trusted as well. The consensus mechanism of other protocols are generally concerned with the execution result and not the integrity of the compute environments. TEA's compute layer consensus verifies the integrity of its compute nodes using the trust data from the onboard trusted platform modules (TPM), with the resulting status written to the Ethereum blockchain.

The TEA Project has solved the TPS issues inherent in ledger-based blockchains by moving to a new form of consensus that's based on atomic clocks. We don't need to solve complex math problems because we're not requiring nodes to calculate a consensus. The TEA Project is using atomic clocks and GPS satellites in combination with hardware security modules to reach a non-traditional consensus.

It could be said that these two new consensus mechanisms, Proof of Trust and Proof of Time, are the TEA Project's main innovations that unlock a full-speed decentralized web for apps that require data security and censorship-resistance. By combining blockchain with GPS and trusted hardware, the TEA Project can meet the needs of large-scale, high frequency apps while ensuring permission-less decentralization and data privacy protection.

HOW TEA PROJECT SOLVES A BIG WEB2 PROBLEM

teaproject.org

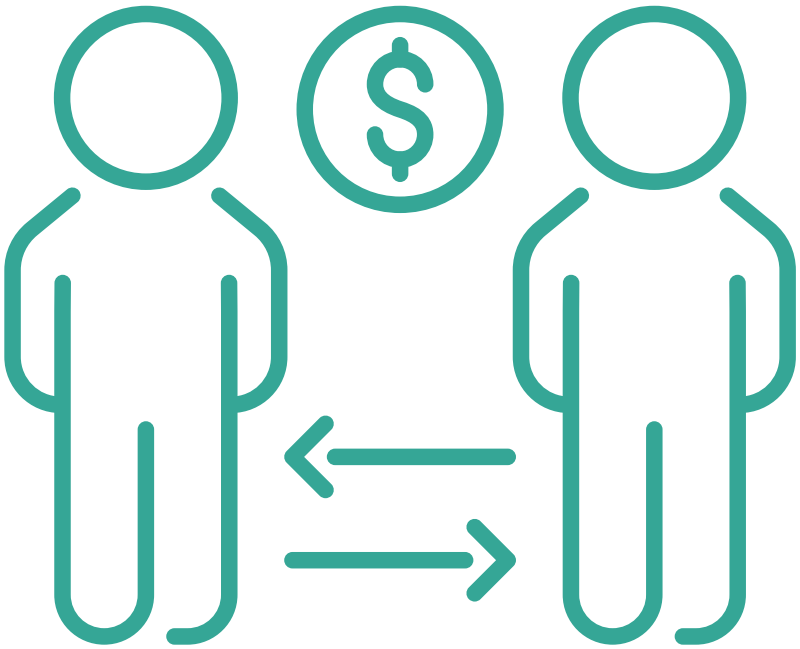
The web2 world of cloud-hosted apps has created an emerging need for apps that truly protect **private user data**. End-users want to use apps without giving companies access to their private data without their consent. The TEA Project is able to meet this emerging privacy demand as data running inside of TApps remains within each node's enclave. The trustability of each compute node is vouchsafed by the platform itself. The technologies built into the TEA platform protects the data as well as the integrity of the result. Under the watchful eye of hardware security modules, the decentralized app’s computation is based on the expected code and the expected data. No one can alter the input code, the input data, or the output result. If any of the nodes have been tampered with or hacked, the TPM will report it. We verify the result by verifying the execution environment: if the execution environment is trustable, then we can trust the result of an app running in that trusted environment. The result of the computation is correct given the computing environment is verified through proof of trust data that is shared in the network and verified via remote attestation.

HOW TEA PROJECT SOLVES A BIG WEB2 PROBLEM

teaproject.org

One of the major benefits of using apps deployed on the TEA Project is that computing only happens within the protected enclaves of the decentralized TEA Project nodes. This means that the node hosting the app code and private user data has no access while the app executes with the user-provided data. After the app execution is completed, the contents of the enclave is destroyed. The end-user gets the result from the app and there are no traces of their private data left anywhere on our network. There's no chance of data breaches during the entire process.

Giving users back the privacy of their own data also implies they can monetize it as they wish. In web2, users have little to no say in how companies monetize their private data. In TEA's version of web3, each user can choose to keep their data private or offer it to companies who pay for the privilege of using it. Private data has always been valuable but we need something beyond the existing centralized web2 architecture for users to be fairly compensated for it.



DEVELOPER EXPERIENCE

The push towards full-speed web3 dApps lacks one missing piece: a development framework for building rich web3 dApps. A big reason that web3 has not yet become a reality yet is this lack of a framework. The TEA Project wants to make it easier for developers to make the jump to web3 by using our familiar development framework as their entry point. Only 1.52% of developers commonly use Solidity, a specialized language for Ethereum smart contracts according to a Stack Overflow dev survey.

The TEA Project will have an SDK available to help shorten development time on our platform. Creating dApps for the TEA Project will always be straightforward for developers as our native binary format is WebAssembly (WASM). WASM, the native language of TApps, is fast, secure, and easy to output to from a variety of different programming languages. This flexibility makes it easier for developers to take the leap into web3 by using the TEA Project as their entry point.

The TEA framework makes building such dApps straightforward as it provides an SDK, wide programming language compatibility through WebAssembly, and code hosting through TEA's decentralized compute nodes. The TEA Project platform also handles billing and allows developers to deploy their code as microservices. These microservices are like libraries that other developers can call as part of their larger programs. The difference here is that the microservice authors get paid everytime their code is called from another developer's code.

DEVELOPER EXPERIENCE

teaproject.org

As an infrastructure provider for web3, the TEA Project isn't itself a developer of web3 dApps. We'll make available reference TApps along with tutorials to help others build on TEA, but the work of actually producing viable web3 dApps will be up to the developers themselves. It's an open question which apps will make the successful transition from web2 to web3. It's not the TEA Project's role to pick the winners here. There will be a natural organic movement of certain classes of apps away from centralized hosting that require better data privacy and decentralization. The TEA Project's job is to ensure that our platform is the easiest to build on and lives up to the promise of full-speed decentralization when developers come looking for the ideal platform to deploy their dApps on.

There are many benefits for developers to use the TEA Project platform as a transition point to web3. One of the primary 3-tier architecture that has a front-end, an execution tier, and a database available for app developers.

Benefits for developers

Front-end (e.g. written in html, css, and javascript) downloaded from IPFS and running in the end-user's browser.

The developer's back-end code runs in any hosting node owned by one of the individual miners.

A database tier that allows developers to change the current state of their app

TEA PROJECT TECHNICAL DETAILS - COMPUTE LAYER

teaproject.org

The TEA Project's Compute Layer

TEA stands for Trusted Execution and Attestation. TEA Project uses trusted hardware, blockchain, and time as its three roots of trust to create a secure and decentralized cloud computing platform. Combining these technologies allows the TEA Project to scale without compromising security or decentralization.

In brief, we can say that the layer-1 blockchain we use, Ethereum, enforces the trust status of each of the hosting nodes. Any hosting nodes deemed not trustable via remote attestation will not be allowed to run on our compute layer and will have its monetary deposit slashed by the layer-1. The consensus algorithm running on our compute layer uses remote attestation to verify the execution environment where the code is run. The TEA Project's philosophy is that if the execution environment, code, and input data are trusted, then the execution result can also be trusted.

The TEA Project aims to build a platform that allows code and data to run inside TEA nodes with complete trust. The technologies built into the TEA platform protects the data and makes sure:

- The computation is based on the expected code and expected data. No one can alter the input code, the input data, or the output result.
- Anyone outside the TEA module's hardware has no access to what's inside the module.
- After the task is done, anyone can verify with Proof of Trust data that their code and data was processed in an untampered enclave.

Developers themselves don't need any special knowledge about blockchain or consensus algorithms to get the decentralization and security benefits inherent in the TEA Project's compute layer. They only need to set WebAssembly as their code compilation target, which allows them to use their preferred programming languages. TApps follow the same three-tier architecture prevalent in cloud computing (front-end, back-end, and database), but TEA Project TApps run fully decentralized.

The Technical Implementation of the TEA Project's Compute Layer

The goal of the TEA Project is to create a decentralized compute platform where each hosting node is a secure execution environment. Once a node is certified trustable (after successfully passing remote attestation), it can run app code in its HSM-protected hardware enclave that interacts with decrypted customer data. The system is designed such that the hosting nodes don't have access to the customer's data or app code once it enters the hosting node's protected enclave.
There are many technologies that help form the TEA Project's secure, decentralized infrastructure for running TApps.

The hardware security modules provide the protected enclaves for app execution while TEA's data storage uses IPFS/libp2p to form a peer-to-peer network. Because IPFS is publicly accessible, storing data on IPFS means everyone can access it as long as the content identifier (CID) is known. In order to protect the data, everything TEA stores in IPFS will be encrypted. The encryption key will never be stored in IPFS or any persistent media and only resides in the memory of the TEA modules.

Only a minimized operating system, the TEA Runtime, is running inside the TEA hosting nodes. TEA's runtime environment is WebAssembly (Wasm) that is specially configured to run inside a secure enclave. Wasm is the only code allowed to run in the TEA runtime, as Wasm bytecode is considered one of the most secure forms of executables. The security built into the WebAssembly language itself also helps to secure the network. Using Wasm is also conducive for developing applications in the TEA ecosystem as most modern programming languages can be compiled into the WebAssembly code format. Developers can stick with their preferred programming languages to write their apps which speeds up development on the TEA platform.

TEA PROJECT TECHNICAL DETAILS - BLOCKCHAIN

teaproject.org

A Summary of How TEA Project Uses Blockchain

TEA uses its layer-1 blockchain to provide:

- Economic incentives and penalties that form the basis of its token economy.
- Trust information storage written to the CML NFTs's metadata (ERC-721).

The trust information stored within each CML NFT's metadata determines if the corresponding compute layer node can run or not.

- Each compute layer node must be in good standing based on the information reported by the node's HSM under the watch of remote attestation. If a node's CML NFT has indicated it failed remote attestation, its deposit will be slashed and the node will be taken offline.
- The layer-1 blockchain keeps the trust status on each of the compute layer nodes to ensure that they are all trustable.
- Because all all compute nodes that pass remote attestation are trustable, we no longer have to worry about Byzantine faults on our compute layer. The second root of trust for our compute layer, time, ensures that transactions are always properly ordered in our state machine using our non-traditional consensus.

Economics-wise, using VRF to randomly select which node will lead the execution of a client's computation job ensures that it would be very costly to attack the TEA network. We use token incentives to encourage many participants to run a node themselves and others to buy CML tokens which entitle the investors to a share of the node’s mining profits. If a node is found to be tampered as revealed by remote attestation, the CML’s tokens are slashed and all token buyers will lose their investment. This is an example of how TEA uses economic incentives to reinforce well-

Caveats About TEA's Use of Blockchain

The TEA Project's layer-1 is currently Ethereum but TEA's compute layer can be run on top of any smart contract-based blockchain. TEA was designed with portability in mind and can change the underlying layer-1 it runs on as long as there's a smart contract deployed there to communicate with.

TEA's considers blockchain a root of trust in that it uses token incentives governed by the blockchain to incentivize good behavior among nodes. Given that each compute layer node must at least have a deposit (in TEA tokens) as well as have purchased a CML NFT for a mining license, it stands to reason that bad behavior such as tampering with the hardware will result in economic loss for the node runner. Blockchain is thus used by the TEA Project indirectly and isn't essential to the platform's functioning like TEA's other two roots of trust, secure hardware and time.

The TEA Project has two main tokens: TEA (ERC-20) and the CML NFT (ERC-721). These entities exist primarily in our compute layer and are mirrored to our main layer-1 chain, the Ethereum blockchain. When an end-user withdraws TEA from our compute layer to Ethereum, we burn the amount that was withdrawn. Similarly, an end-user's deposit into the compute layer will burn the same amount of tokens on Ethereum. Optionally, we may at some point turn off the bridge to layer-1. This bridge is technically breakable if there are any legal concerns depending on the needs of enterprise. At that point, fiat can be used to fund the compute layer operations and any TEA or CML that the fiat buys stays only within the compute layer.

Remote Attestation

The remote attestation process is used in conjunction with the hardware security modules (HSM) onboard the mining computers to ensure each node is trustable.

- Multiple nodes are randomly called to conduct remote attestation on any particular node under inspection.
- The results of these remote attestations are stored in the compute layer's database.
- Successful remote attestation on a candidate node ensures that the machine's HSM hasn't been tampered with. Therefore, the secure enclave overseen by the HSM is trustable and will protect the confidentiality of both the code and data processed within the enclave.

It's through this remote attestation process that the TEA Project has been able to create a completely trustable peer-to-peer network of compute layer nodes that are protected and monitored by hardware TPM chips. Every TEA node, through the remote attestation process, can verify the integrity of any other TEA node.

This is the fundamental premise of the TEA Project's concept of trust. If a node shows full integrity under remote attestations performed by multiple peer nodes, then its enclave is secure, and the resulting calculations (i.e. app output) can be trusted.

How TEA Conducts Remote Attestation

In the TEA network, each trusted computing node has a Root of Trust (RoT), a key pair generated by the HSM manufacturer. The private key for this RoT is held inside the hardware and not known to anyone, including the manufacturer. Each node's public key is recorded in a database on TEA's compute layer.

TEA Project currently runs on AWS Nitro for its compute nodes. Unlike hardware TPM chips that generate Proof of Trust data, the remote attestation process will involve interacting with the KMS of each Nitro instance. The process of remote attestation of AWS Nitro nodes proceeds as follows:

1. Periodically, the TEA Project will select a random node to run remote attestation on. It will request:

- The node's KMS to generate its attestation document.
- The node to generate its hardware fingerprint and fingerprints for all the software modules running in the node. These will need to be signed by the Nitro node's private key before being returned to the requestor.

2. The attestor will compare the KMS-generated attestation document to the hardware fingerprints claimed by the software (TEA runtime) running in the node. Each attesting node:

- Verifies the signature of the attestation documents.
- Verifies that the two attestation documents have no discrepancies for the hardware fingerprints.

3. TEA's compute layer uses a Raft consensus protocol where a leader node randomly selects remote attestation nodes to challenge the randomly selected node being tested. If a two-thirds (2/3) majority of attestor nodes (including the leader) receive a positive result from verifying the target node's requested attestation material, the tested node will have passed the consensus verification and be marked as trustable to use. Note that trustable nodes are the only ones selectable to perform the remote attestation process. Therefore, as long as the network can keep a minimum ratio of 2/3 trustable nodes, we can trust the consensus result. To maintain this minimum threshold, the TEA Project team will initially bootstrap the network with clean nodes and control the birth rate of new nodes by controlling the issuance of Camellia NFTs needed to run nodes on the TEA network.
4. Finally, the result of the attestation process (pass / fail) is recorded on the layer-1 blockchain, to the CML NFT's metadata (ERC-721).

In the event that a CML is marked as failing remote attestation, it will no longer be allowed to run on TEA's network and its bonding curve tokens will be slashed to 0 and forfeited by all of its owners.

TEA PROJECT TECHNICAL DETAILS - TAAS

teaproject.org

TEA's Biggest Innovation is Trust as a Service (TAAS)

In the TEA Project, computing nodes become trusted through a remote attestation process. The concept of trusted computing has so far been stuck in the limiting paradigm of one-computer = one-metal-box. This old thinking limits its expansion and possible uses cases in a distributed world. In particular, it needs a solution like the TEA Project to make a decentralized network of hosting nodes trustable without sacrificing app execution speed. In the TEA Project, trust is already taken care of by the remote attestation process. That allows our compute layer nodes to all trust each other by default and no longer need to watch for Byzantine faults.

The TEA nodes' hardware security modules allow the compute nodes to offer a hardware-protected enclave where code and data can run securely. Anyone outside the TEA module's hardware has no knowledge about what happens inside the module. Neither the TEA node's owner nor the server connected to the module will know, nor can anyone predict which node is running any particular task. This gives the TEA network security as no information can be breached during this secure workflow.

The TEA Project's goal is to build a platform that allows code and data to run inside trusted TEA modules without needing to trust anything else. The technologies built into the TEA platform protects the data and the integrity of the result. Developers can build their apps on TEA Project and benefit from TEA's Trust as a Service to run secure apps.
The decentralized app's computation is based on the expected code and the expected data. Once a node is proven to be trustable, no one can alter the input code, the input data, or the output result. The result of the computation is correct given the computing environment is verified through proof of trust data that is shared in the network and verified via remote attestation. The TEA Project's decentralized apps can thus run on par with cloud computers and do so with a an even higher level of trust.

TEA PROJECT TECHNICAL DETAILS - TAAS

teaproject.org

TEA's Biggest Innovation is Trust as a Service (TAAS)

In the TEA Project, computing nodes become trusted through a remote attestation process. The concept of trusted computing has so far been stuck in the limiting paradigm of one-computer = one-metal-box. This old thinking limits its expansion and possible uses cases in a distributed world. In particular, it needs a solution like the TEA Project to make a decentralized network of hosting nodes trustable without sacrificing app execution speed. In the TEA Project, trust is already taken care of by the remote attestation process. That allows our compute layer nodes to all trust each other by default and no longer need to watch for Byzantine faults.

The TEA nodes' hardware security modules allow the compute nodes to offer a hardware-protected enclave where code and data can run securely. Anyone outside the TEA module's hardware has no knowledge about what happens inside the module. Neither the TEA node's owner nor the server connected to the module will know, nor can anyone predict which node is running any particular task. This gives the TEA network security as no information can be breached during this secure workflow.

The TEA Project's goal is to build a platform that allows code and data to run inside trusted TEA modules without needing to trust anything else. The technologies built into the TEA platform protects the data and the integrity of the result. Developers can build their apps on TEA Project and benefit from TEA's Trust as a Service to run secure apps.
The decentralized app's computation is based on the expected code and the expected data. Once a node is proven to be trustable, no one can alter the input code, the input data, or the output result. The result of the computation is correct given the computing environment is verified through proof of trust data that is shared in the network and verified via remote attestation. The TEA Project's decentralized apps can thus run on par with cloud computers and do so with a an even higher level of trust.

Transaction Consensus Using Time

The TEA Project uses two different types of consensus on its compute layer:

- The consensus used in remote attestation as governed by the hardware security modules of the compute nodes.
- The consensus used to order transactions as governed by the GPS modules of the compute nodes.

The TEA Project uses time as the basis of its consensus to order transactions as reported by the (trusted) GPS modules of the compute nodes. The TEA Project's state machine nodes are in charge of synchronizing the current state between multiple nodes so that the transaction sequence is consistent across all the state machine nodes. We do this by relying on the accurate time provided by the atomic clock of GPS satellites.

1. Every transaction has a timestamp attachment.
2. The timestamp originates from an atomic clock or a GPS and is considered 100% trustable.
3. State machine nodes receive transactions and orders them by their timestamps.
4. Because the sequence is the same across all state machine nodes, we've reached a consensus on the order of transactions.

TEA Project's Algorithm for Txn Consensus

The design of the TEA Project's compute layer consensus ensures that even if transactions are missing (e.g. the transaction is delayed by network congestion), the ordering of transactions will still be correct. Before we describe the txn consensus algorithm, let's go over some useful terminology.

A compute node will send its transactions to multiple state maintainer nodes as part of the consensus mechanism. These duplicate recipients are referred to as replicas.

There's a buffer period after which reported transactions will be dropped. To ensure that most replicas can be synchronized to a consistent state, the TEA Project's state machine requires a short waiting queue due to network latency.

Since time is stable in our universe, it follows that each replica can achieve strong consistency using time as a root of trust for consensus. The buffer time is adjustable by the TEA Project and is currently set at 3 seconds. For example, if a node's current clock is at T, then any reports of txns with timestamps earlier than T - 3 seconds will be dropped.
The foundation of the compute layer's consensus on txn ordering is the trusted timestamp. These are trustable by virtue of the HSM that watch over the GPS modules on the compute node machines.
A conveyor belt is used to order txns within each replica. Transactions can be added but never removed from each replicas conveyor belt.

The TEA Project is able to achieve a reliable ordering of transactions through the following algorithm:

- The compute nodes send transaction hashes and receipts to multiple state machine nodes, referred to as replicas. Because it takes time to package up a transaction and create its hash, the original transactions don't have timestamps of when they were sent out. The followup receipts have timestamps that show when the original txn hash was sent out.
- Replicas match the hashes of transactions with their followup receipts. Replicas can re-order transactions as long as they remain within the buffer period relative to the now time of the replica's clock.
- Replicas are always broadcasting their latest txns and a decentralized, peer-to-peer sync function between replicas keeps the transactions ordered. The sync function can only add a missing transaction hash to a replica when it's missing on their conveyor compared to other replicas.

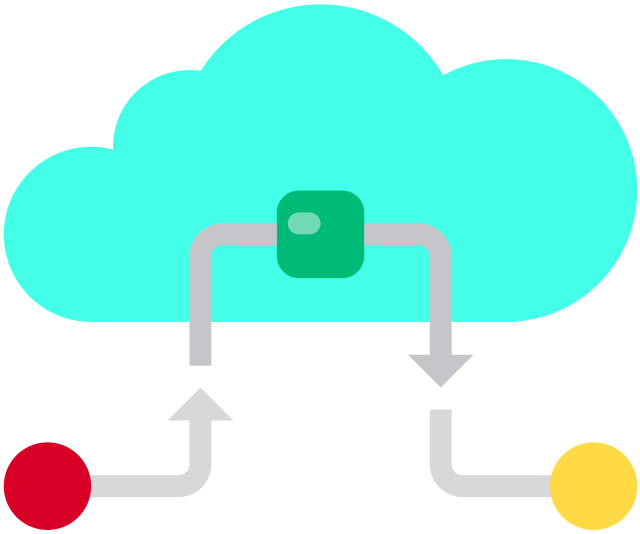
The consensus algorithm proceeds as new transactions land on each replica's conveyor belt until eventually everyone ends up with the same state. There might be a question of whether the transactions could ever be out of order across the replicas. But the reported time is recognized under the supervision of trusted hardware and used as the basis for the final ranking of all replicas. There might be a chance that a transaction is reported late (e.g. because of network congestion), but not of there being actual differences in the trusted timestamps across replicas.

State Maintainer Node RAM Storage

The TEA Project's state machine only needs to store these state change in RAM and not the transaction itself. Since the on-board hardware security modules of the mining nodes allows attestation to be run on them, they can be guaranteed to be trustable and don't need to sync up a historic ledger of all previous transactions. Once all nodes are trustable, any node can get the latest state from a nearby node's RAM. Syncing up to the latest state through reading a nearby node's RAM is a quick process, much quicker than having to reconstruct the entire ledger to get the latest block.

In the **TEA Project**, transactions are processed with the resulting state change stored in the RAM memory residing in the enclave of the state machine nodes.
Every state maintainer node will have the same copy of the state in their memory, as well as the SQL database instance.

There's also a persistent backup kept to IPFS but in an encrypted format. The SQL database instance refers to the GlueSQL database that uses the on-board GPS modules of the state maintainer nodes to update the current state continuously. IPFS is only used to write backups of the current database state. The GlueSQL database can be used by developers the same as they use SQL databases in the cloud computing world.



The TEA Project also offers an eventually consistent CRDT database built on OrbitDB that can be used ad hoc by TApps. This type of DB allows for short-term inconsistencies in the business logic of TApps. The TEA Project uses OrbitDB databases built on top of IPFS that run through our compute layer nodes for these transactions.

CRDT stands for conflict-free replication data type, which allows conflict-free mergers between different replications. There are a couple of benefits for TApp developers to use CRDT databases:

- CRDT is fast as it has no time delay and doesn't need to wait for others. Allowances are made for new transaction reports which are added non-destructively. This is in contrast to the TEA Project's strong consistency state machine which must wait a minimum amount of time for confirmation.
- CRDT storage cost is cheap, relying on IPFS for decentralized hard drive storage instead of the more expensive RAM storage.

References and Further Reading

- An interactive TPM simulator: https://google.github.io/tpm-js/
- The official website of the Trusted Computing Group who's responsible for maintaining the open standards of TPM chips: https://trustedcomputinggroup.org/
- Information on Google Spanner which also uses time as a root of trust for ordering transactions: https://en.wikipedia.org/wiki/Spanner_(database)
- A research article detailing how GPS can be used with TPM to create trustable time synchronization among nodes in a network: https://www.mdpi.com/2076-3417/11/18/8288/htm

DEVELOPER EXPERIENCE

teaproject.org

Code runs as WASM binaries, which allows developers to use their preferred programming language. When a client requests app execution, encrypted data (either local or hosted on IPFS) and encrypted binary code (hosted on IPFS) are decrypted within the TPM-protected enclaves of TEA's decentralized compute nodes. No trust is needed between developers, the hosting nodes, or the end user. The TEA Project ensures the developers and hosting nodes are paid through a transaction and the user gets the desired result from the app.

It's important to us that developers can build on the TEA Project without having to be experts in blockchain / decentralized design or having to learn a new language. In the TEA Project, developers only have to upload their code to the developer portal. Developers have no responsibility for renting a server and doing sysadmin tasks as they have to in web2. It's up to the end-user to choose which hosting node to use.

Everything in the TEA Project is designed to look from the app's point of view like it's running on a normal cloud computing architecture, with the magical benefit that it's running fully decentralized. The TEA Project design handles these details so developers and decentralized app users don't have to think about them.

DEVELOPERS FUNDING

teaproject.org

Developers can optionally use TEA Project's built-in bonding curve to generate investment in their TApp. This optional funding mechanism allows TApps to leverage expected future revenue into early development funding. The TAppStore is where the entire TEA ecosystem meets: developers to publish their TApps, curators to publicize and invest in new trending TApps, and consumers to spend their TEA tokens on useful TApps they want to use.

TEA PROJECT ECOSYSTEM

The TEA Project ecosystem's main demographic groups are miners, developers, and consumers who each play a part in maintaining a healthy decentralized computing platform. A strong mining community must be developed to provide the infrastructure before developers are onboarded. Similarly, developers must develop compelling apps before consumers can be enticed to enter the ecosystem. Each demographic will be encouraged to enter the TEA Project through a variety of methods.

TEA PROJECT ECOSYSTEM

teaproject.org

1. MINERS :

The TEA Project aims to build a healthy ecosystem from the ground up through miners who provide the hosting nodes.

- Miners "plant" CML into their mining hardware equipped with a TPM and either an atomic clock or GPS module to "harvest" TEA tokens.
- Mining machines host web3 applications and are rewarded in TEA tokens based on the app's consumed computing resources.

2. DEVELOPERS :

Miners are necessary for the actual computing infrastructure just as developers are necessary for building usable apps on top of that infrastructure. This segment of the rollout will include tech education & outreach on how to build on the TEA ecosystem.

- Hackathon events and grant program released.
- TEA SDK available helping developers build with the TEA dev framework.
- TAppStore launched showcasing rich dApps running on the TEA platform.
- In addition to outreach to traditional web2 devs, TEA will also partner with web3 development frameworks to introduce TEA Project to devs who've already onboarded onto web3.

3. CONSUMERS:

Consumers form the demand that rewards both the miners who provide the hosting nodes and the developers who create the TApps. We hope to see a positive feedback loop: as more consumers enter the ecosystem, developers can focus on making TApps that meet consumer demand. Additionally, consumers can invest in each TApp through its bonding curve and promote the TApp on social media. Becoming a curator for a burgeoning TApp helps the consumer invest in the TApps that they use. The TApp tokens they hold will increase in price along the bonding curve as more buyers push the supply higher. And as consumer interest in TApps increase, they'll be incentivized to exchange ETH for TEA which in turn will support the TEA token price and grow the TEA economy.

COMPETITION

The emerging web3 strategies for solving data privacy issues have been fraught with compromises.

- Multiple projects have adopted Intel's SGX chip specification to provide a trusted execution environment (TEE) within the CPU itself. However, Intel is a central authority which introduces centralization, the hazards of which were realized when Intel decided to discontinue their SGX chip line for consumer CPUs. Additionally, projects reliant on SGX are limited to running everything in the CPU. But the CPU doesn't provide enough resources for applications such as AI which also need TPU and GPU access.

- Other projects have chosen to leave everything encrypted within their platform to preserve data privacy. This type of design requires datacenter-level hardware to decrypt these data streams in a timely manner, which makes running a node an institutional endeavor outside the reach of ordinary miners.

- The limitations of smart contract-based web3 platforms is often counteracted by introducing centralized hosting to do the heavy lifting for the dApp. For a gaming dApp, this could mean that while game items are tracked on the blockchain the actual gameplay happens on centralized servers. This of course turns supposed dApps into hybrid dApps with all of the data security issues that are typical with centralized hosting.

Smart contracts run slowly because of the consensus required to confirm transactions on the blockchain. Because Ethereum's mining nodes are not trustable (e.g. any of them could present an alternate ledger it hopes to pass off as the truth), a BFT consensus algorithm is necessary to deal with scenarios where not all nodes are telling the truth. The TEA Project is able to sidestep this quandary of decentralized nodes by performing remote attestation on all of its compute layer nodes. And we no longer have to worry about Byzantine faults because all of TEA’s compute nodes are trustable as determined by remote attestation.

COMPETITION

teaproject.org

The TEA Project's non-traditional consensus uses time to ensure that transactions are properly ordered in our state machine. Proof of time gives TEA a consensus on the sequence of events between all decentralized nodes, which allows them to share a global state machine. The TPM onboard the compute nodes ensures that the GPS units haven't been hacked or altered in anyway which ensures that the timestamps coming from each node are trustable. Because of our non-traditional consensus, our compute layer doesn't have any "TPS" (txns per second) limitations as it's not burdened with the slow BFT consensus that operates on blockchain.

It should be noted that the primary function of the smart contract is to keep the trust status of TEA's compute nodes. The smart contract also provides a list of IP addresses from which end-users can load the TAppStore. And although TEA Project is currently deployed to Ethereum, it can be deployed as a smart contract to multiple layer-1 blockchains

The TEA Token Economy

A healthy token economy is essential to incentivize behavior that helps support the TEA ecosystem. To this end, we've designed three main types of tokens to help the TEA Project achieve its goals:

1. The TEA utility token.
2. The CML NFT that comes with mining privileges.
3. Bonding curve tokens

The TEA Utility Token

Our first token, TEA, is a utility token with a supply of 100 million tokens pre-allocated in the genesis block. TEA is also given as a reward to miners for running public services (e.g. remote attestation) on the TEA network. Note that there's no inflation: the 100 million TEA tokens is a hard cap with mining rewards generated through a taxation process on state maintainer nodes.

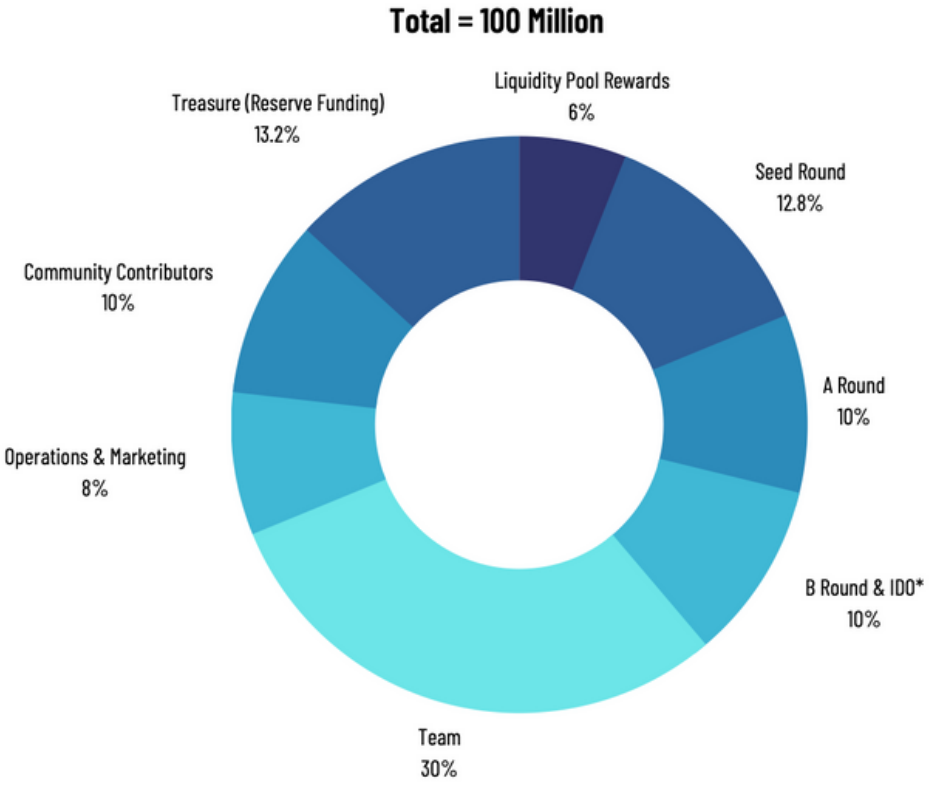
TEA has a variety of uses on the platform:

- TEA pays for gas when performing any transaction on the network which is paid out to miners.
- TEA is paid by consumers when using a TApp.
- TEA can be used to purchase tokens issued on a bonding curve representing different entities in the TEA ecosystem (explained below).

text

TOKEN DISTRIBUTION & USE OF FUNDS

teaproject.org



Vesting Schedules	Immediate Unlock	Vesting
Seed, A, B Rounds	10.00%	5% per month for 18 months
Team and Community	0.00%	2 month lockup 5% per month for 20 months

*Seed round includes post-seed round

**Any investment rounds and other allocations not completed will go to the treasury

***The treasury includes the early mining reward fund which is used to pay the miners remote attestation rewards during the early stages after the mainnet launch

**** If B Round / IDO are skipped, this amount will go to a subsidy pool to incentivize miners in the early stages after mainnet launches

ROADMAP / MILESTONES

teaproject.org

2021 Q1-Q2

Web3 Foundation Open Grant
Seed round secured including investment from Hashkey

2021 Q3-Q4

Testnet starts with dApps running on network
Public mining in preview mode

2022 Q1-Q2

Post-seed round secured
TEA Party dApp released

2022 Q3-Q4

Post-seed round secured
TEA Party dApp released
Layer-1 EVM smart contract compatibility

2023 Q1

Migration to AWS Nitro for all nodes

2023 Q2

Mainnet starts (planned)

2023 Q3

Developer outreach (planned)

OUR PARTNERS

teaproject.org

HASHKEY Capital

DRAPER DRAGON

DragonRoark

JDI Ventures

BITRISE CAPITAL

ORACLES INVESTMENT GROUP

OUR PARTNERS

teaproject.org

WATERDRIP

WATERDRIP CAPITAL

Rebase D.Ventures

YOUBI CAPITAL

SubO capital