

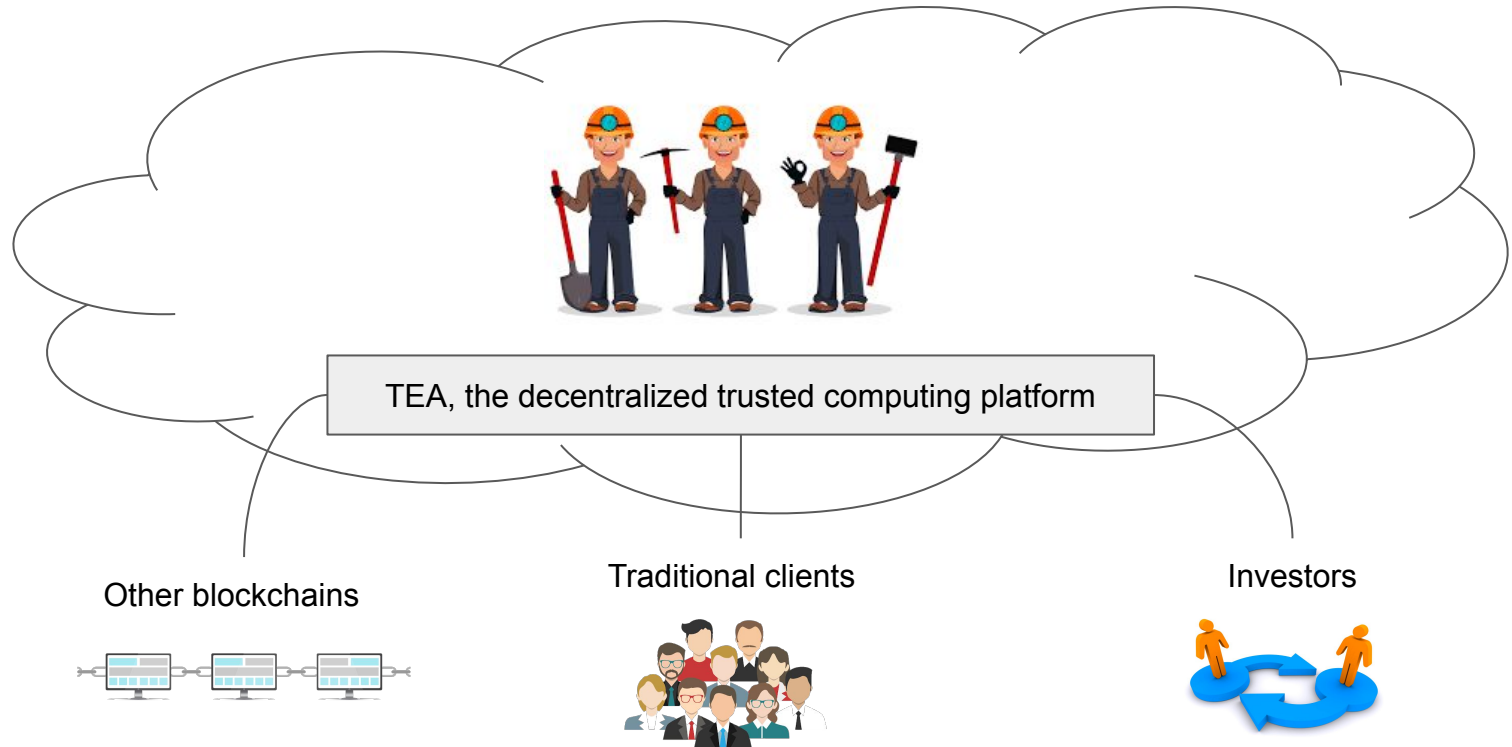
TEA: Trusted Execution & Attestation

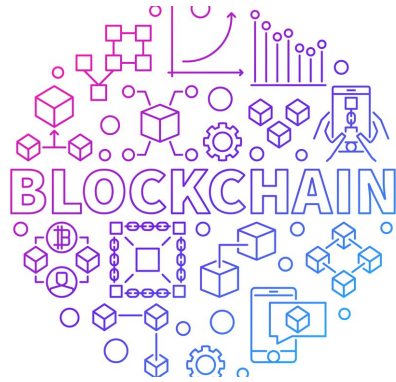
Elevating Decentralized
Trusted Computing to a **T**



www.teaproject.org

Just like Uber and Airbnb, we provide Trust-as-a-Service





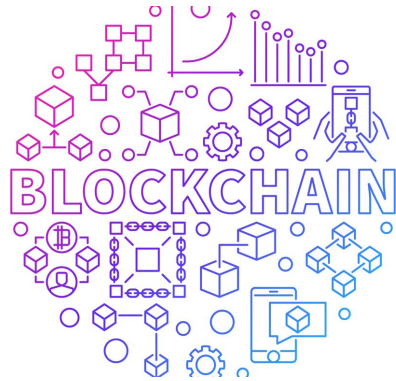
Decentralized but

- Hardly run rich applications
- Need special protocol for privacy
- Pool performance

Run rich application with high performance but

- Centralization
- Privacy breach
- Censorship





Decentralized but

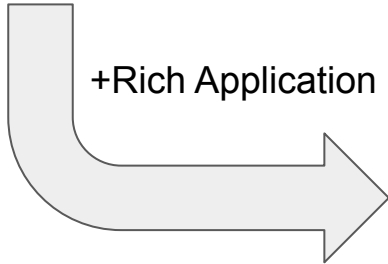
- Hardly run rich applications
- Need special protocol for privacy
- Poor performance

Run rich application with high performance but

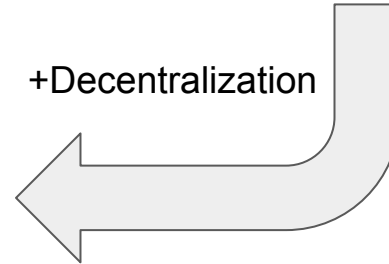
- Centralization
- Privacy
- Censorship



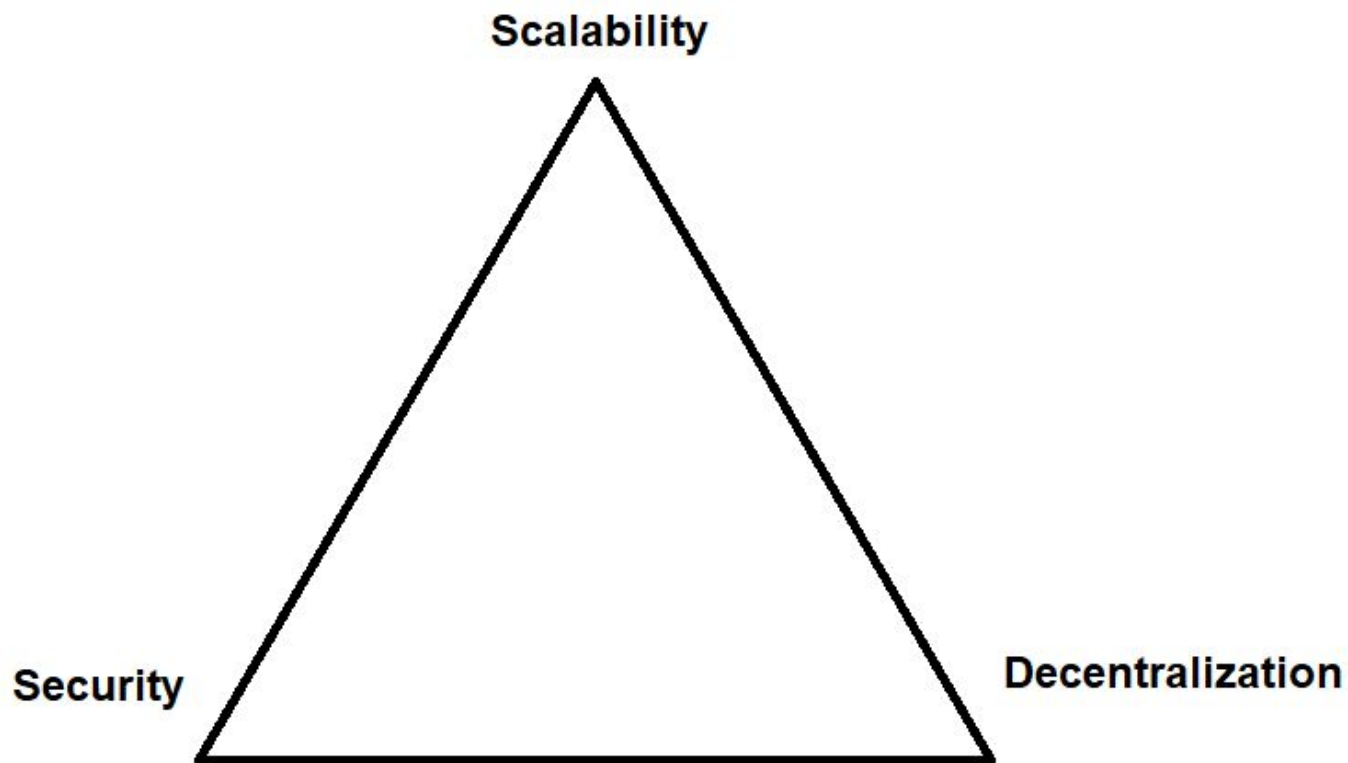
+Rich Application

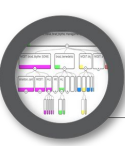


+Decentralization



The Blockchain Trilemma



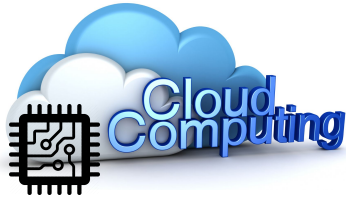


Overview

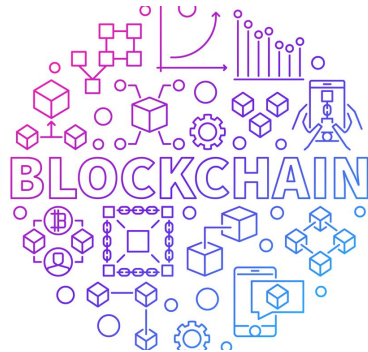


10,000ft

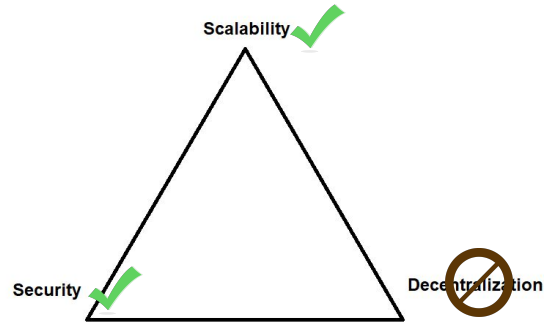
2



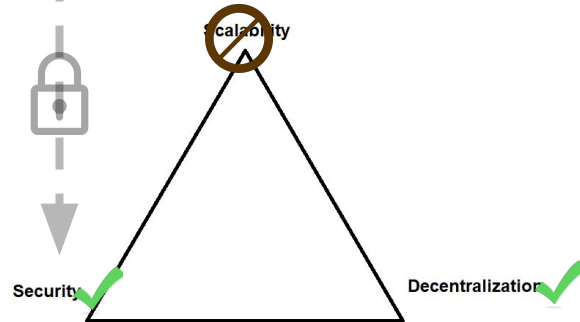
1



The Blockchain Trilemma



The Blockchain Trilemma





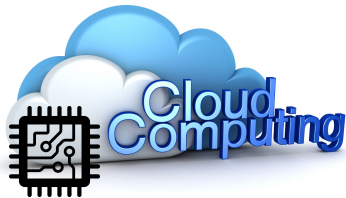
Overview



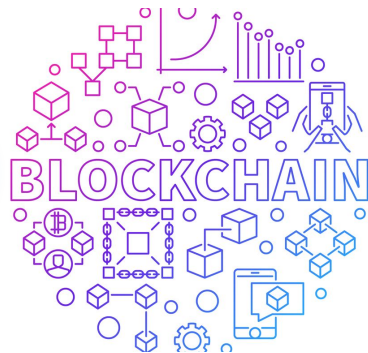
10,000ft

- Rich Applications run on Decentralized Nodes
- Secure Logic and Data inside Hardware Protected Enclave
- Security Chip (Root of Trust - RoT) Generates Proof of Trust (PoT)
- Verify other nodes' PoT and send verification result to Layer 1

2

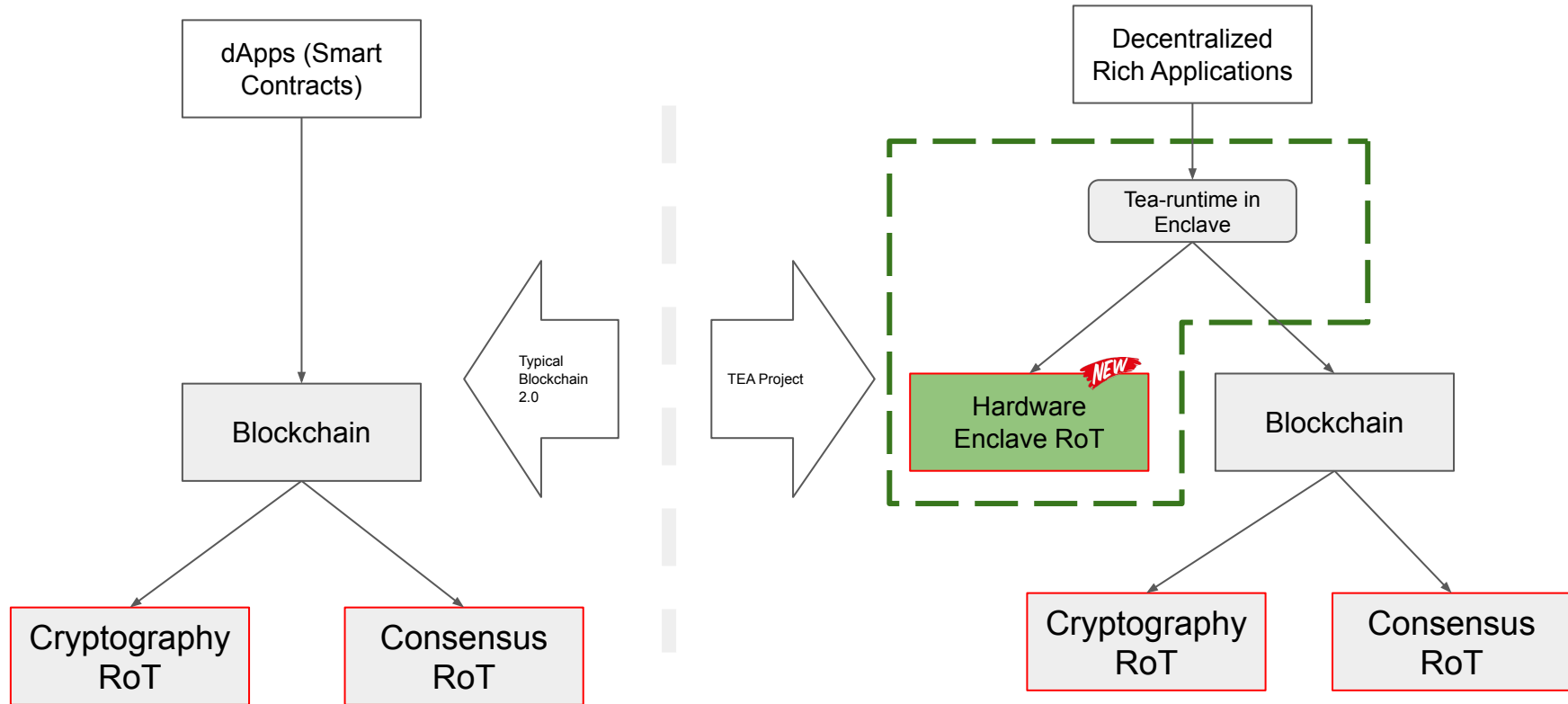


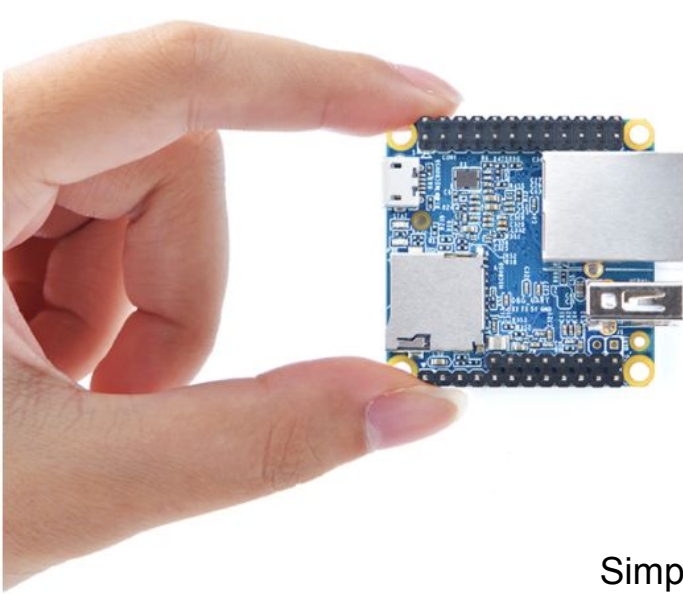
1



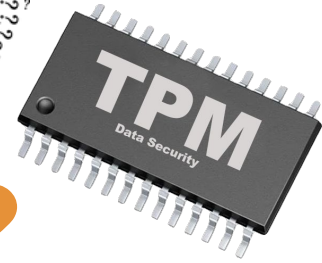
- Immutable Data Storage
- Consensus on the verification result from Layer 2
- Managing Remote Attestation
- Token Economy
- Verify block

Dependencies of RoT





WHY Hardware?



Simply because you cannot trust software!

Other cryptography options?
SMPC, FHE, ZKP

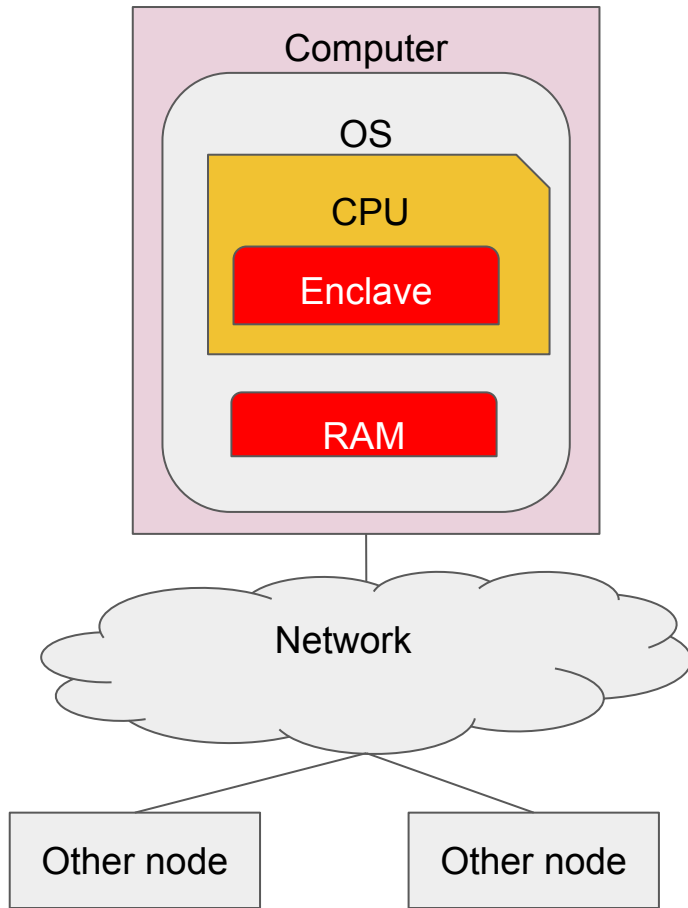
Hardware options?
CPU-based TEE, TCG-based TPM



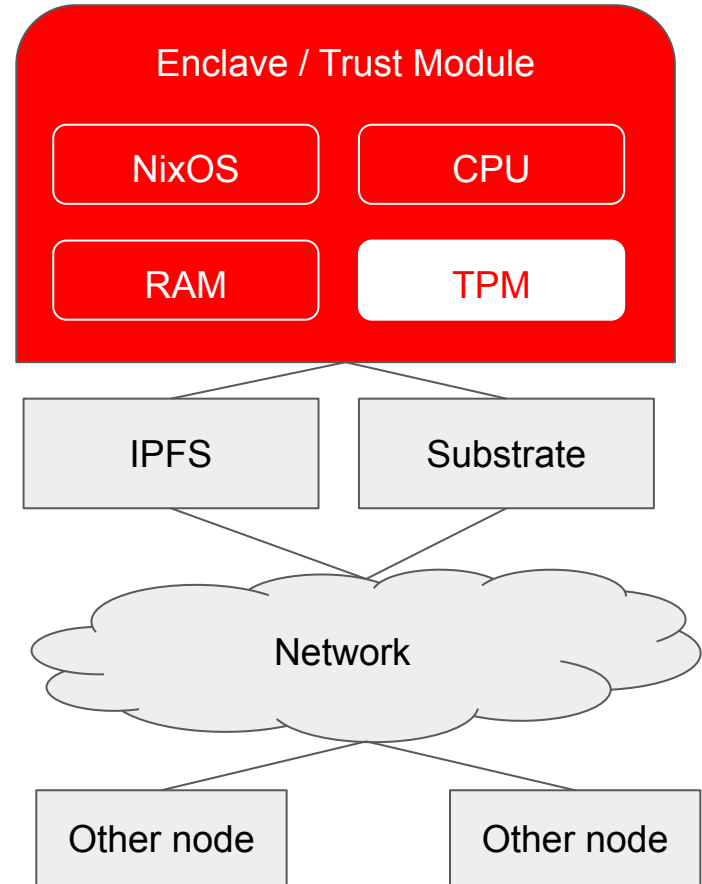
Software stack




TEE



Amazon Nitro / TEA Box



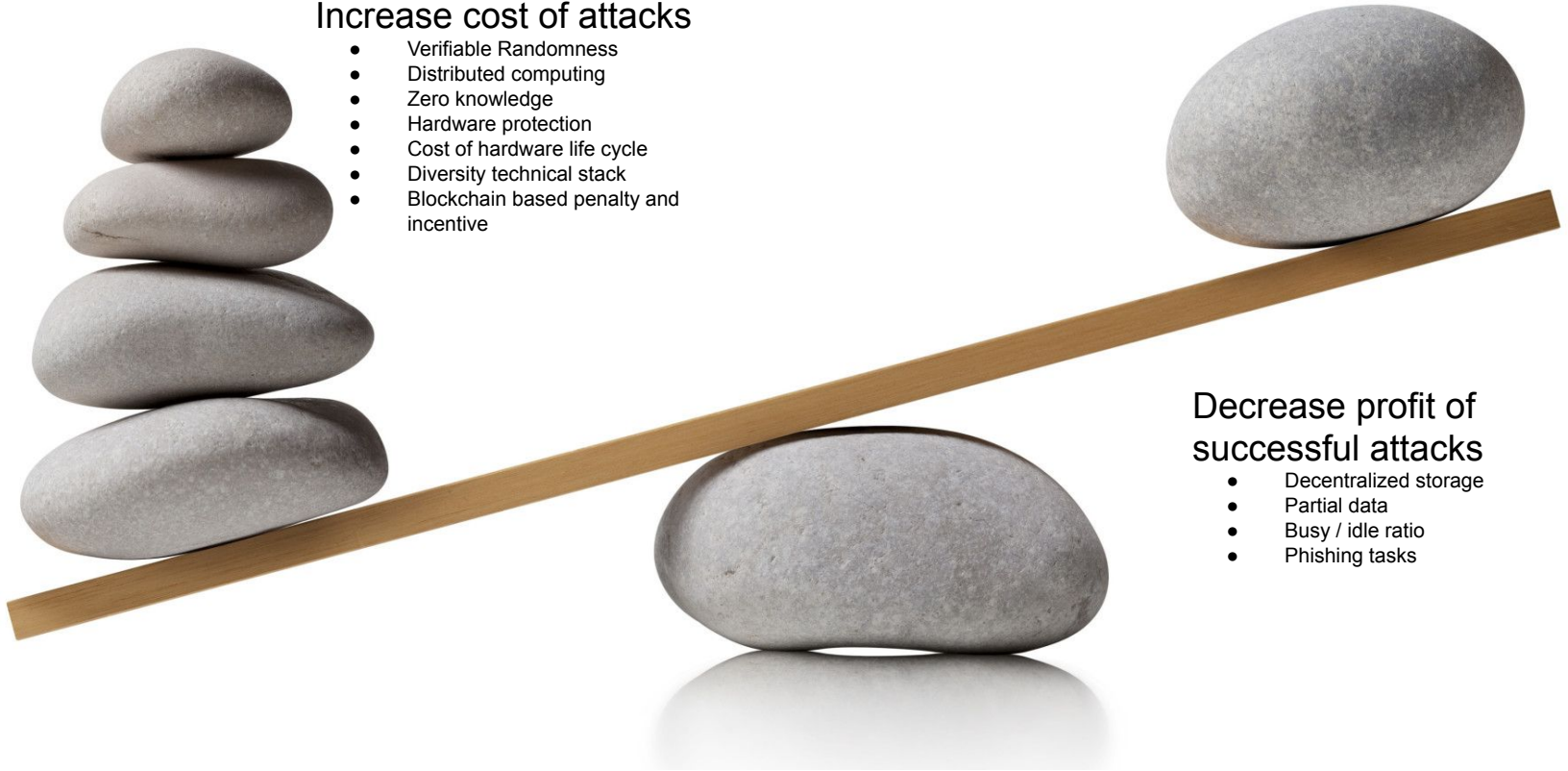
	TEA Support	Technology	RoT Verification	Cloud IaaS 4 Rent?
Google Cloud / MS Azure Confidential Computing	On Roadmap	CPU Based (AMD/Intel)	Centralized Cloud	Y
TEE SGX/SEV/TrustZone	On Roadmap	CPU Based	Centralized by CPU manufacturer	N
Amazon Nitro 	In Development	TPM Based(?)	Centralized Cloud	Y
Trusted Computing (TPM)	Software Simulator Completed	TPM Based	Decentralized	N

Increase cost of attacks

- Verifiable Randomness
- Distributed computing
- Zero knowledge
- Hardware protection
- Cost of hardware life cycle
- Diversity technical stack
- Blockchain based penalty and incentive

Decrease profit of successful attacks

- Decentralized storage
- Partial data
- Busy / idle ratio
- Phishing tasks



Three logical chains rooted from three RoTs

Delegation Chain

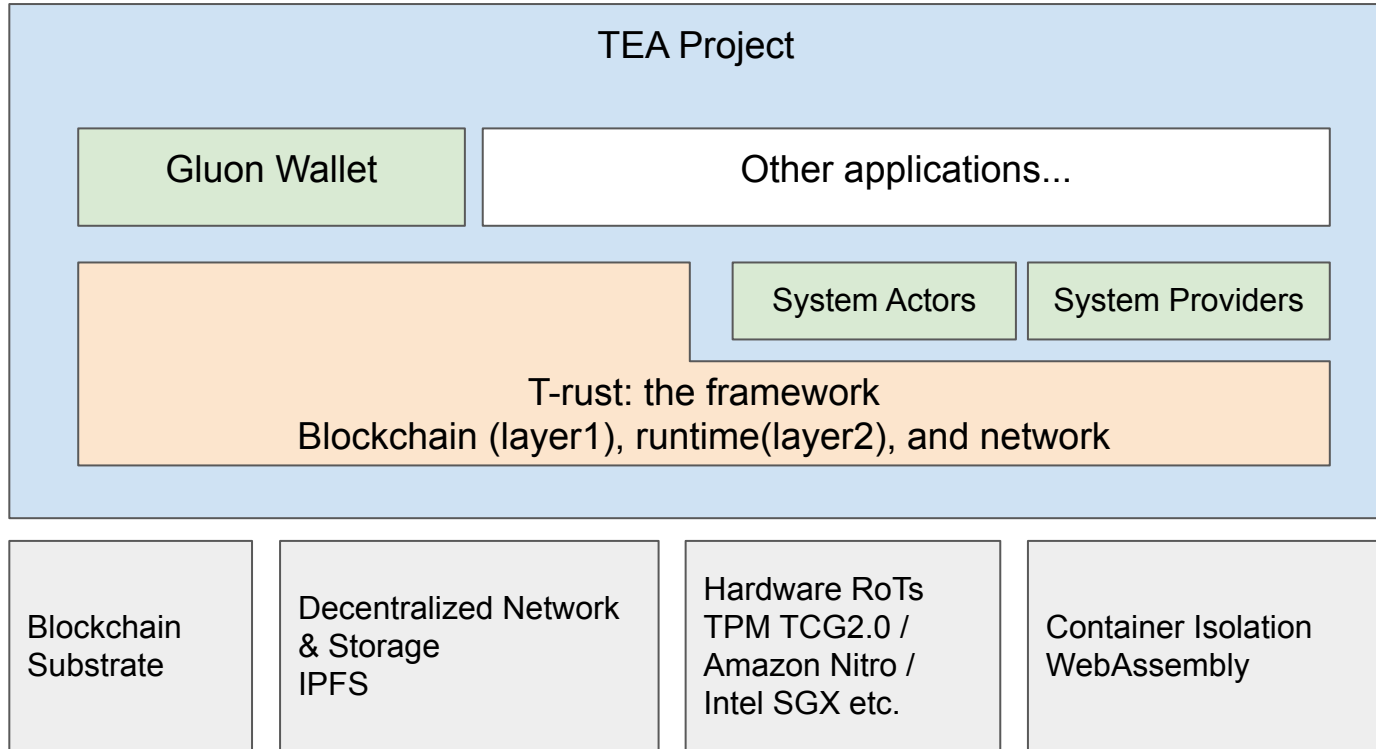
Based on the cryptography RoT, this protocol dispatch the trusted computing tasks to a random and unpredictable trusted execution node while revealing zero knowledge to the public until task completed. Everything is traceable by the public later on.

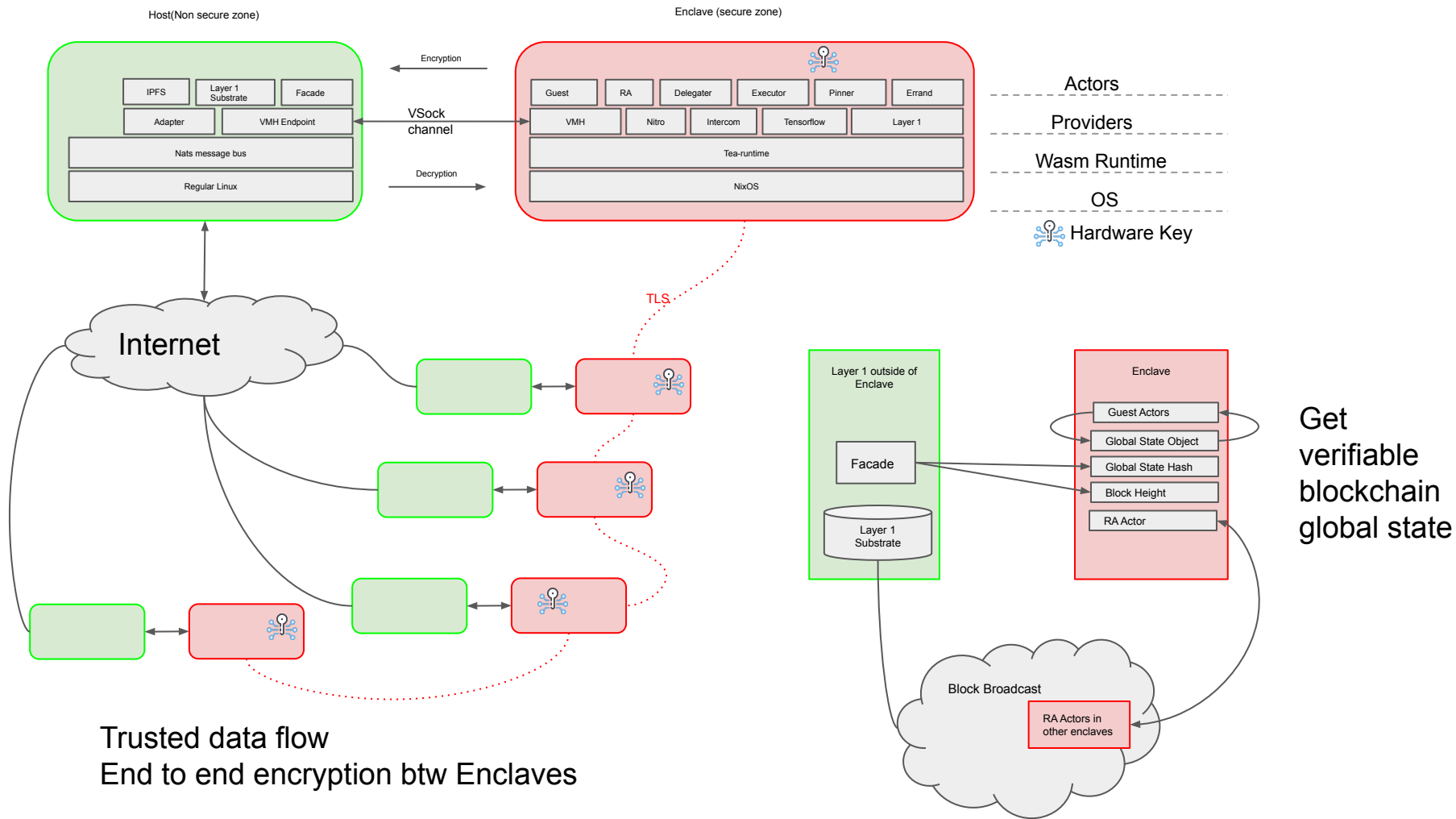
Trust Chain

Based on the hardware RoT(eg. TPM, or Amazon Nitro), All layers of technical stack inside an enclave verify its upper layers integrity before loading, all the way up to the top layer: Wasm actors.

Block Chain

Base on the consensus RoT, the Tea-pellet verifies the Proof of Trust (PoT) from VRF selected Remote Attestation (RA) nodes to determine an enclave reliability. Blockchain is also the storage of essential trust data such as certifications and public keys.





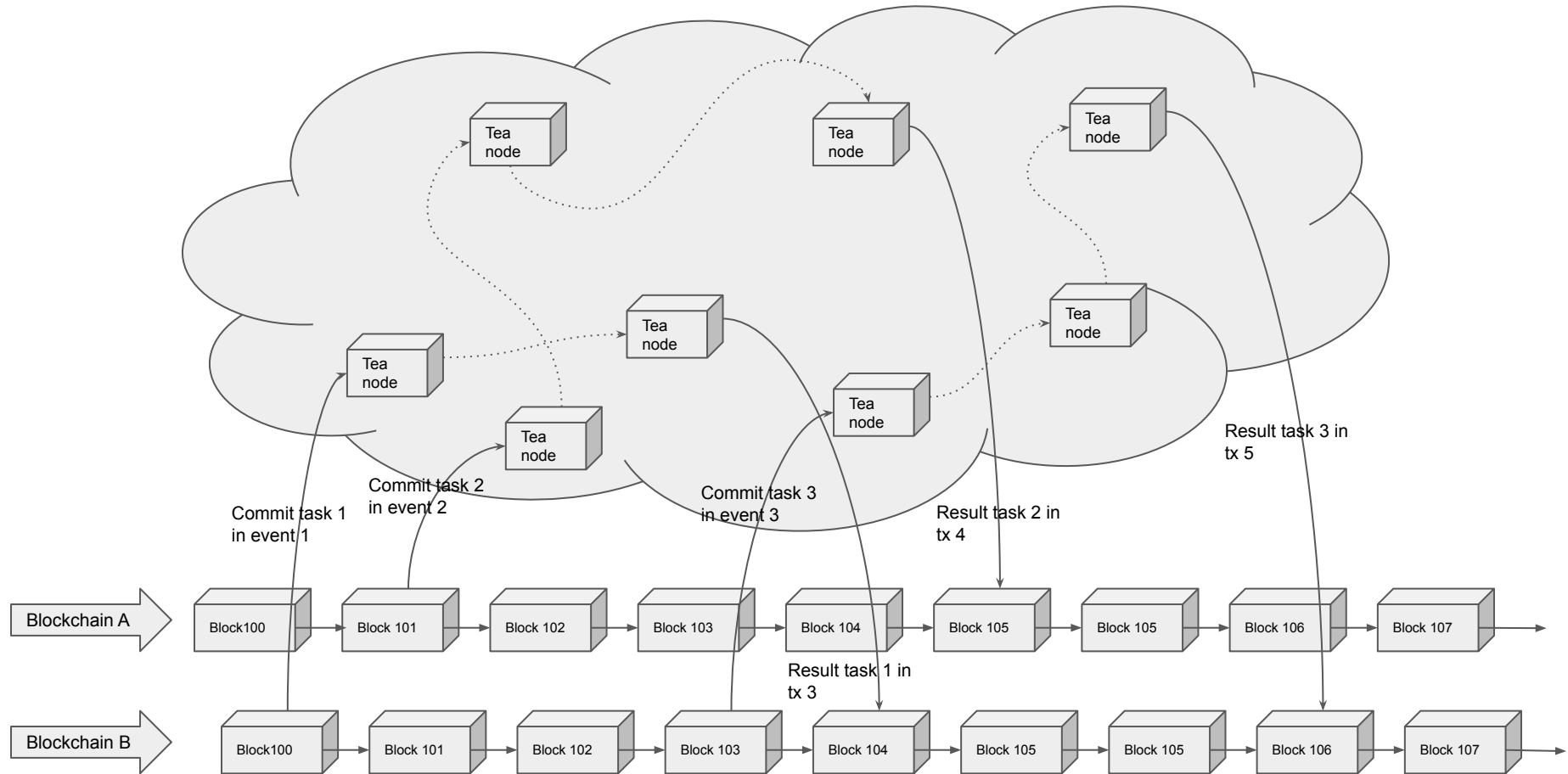
TEA does not aim to speed up smart contracts, but to enable a new kind of Rich dApps

Backend decentralized trusted computing services to smart contracts

Add functions services to IPFS (Interplanetary Functions Services)

Enhanced trusted computing services to traditional cloud computing

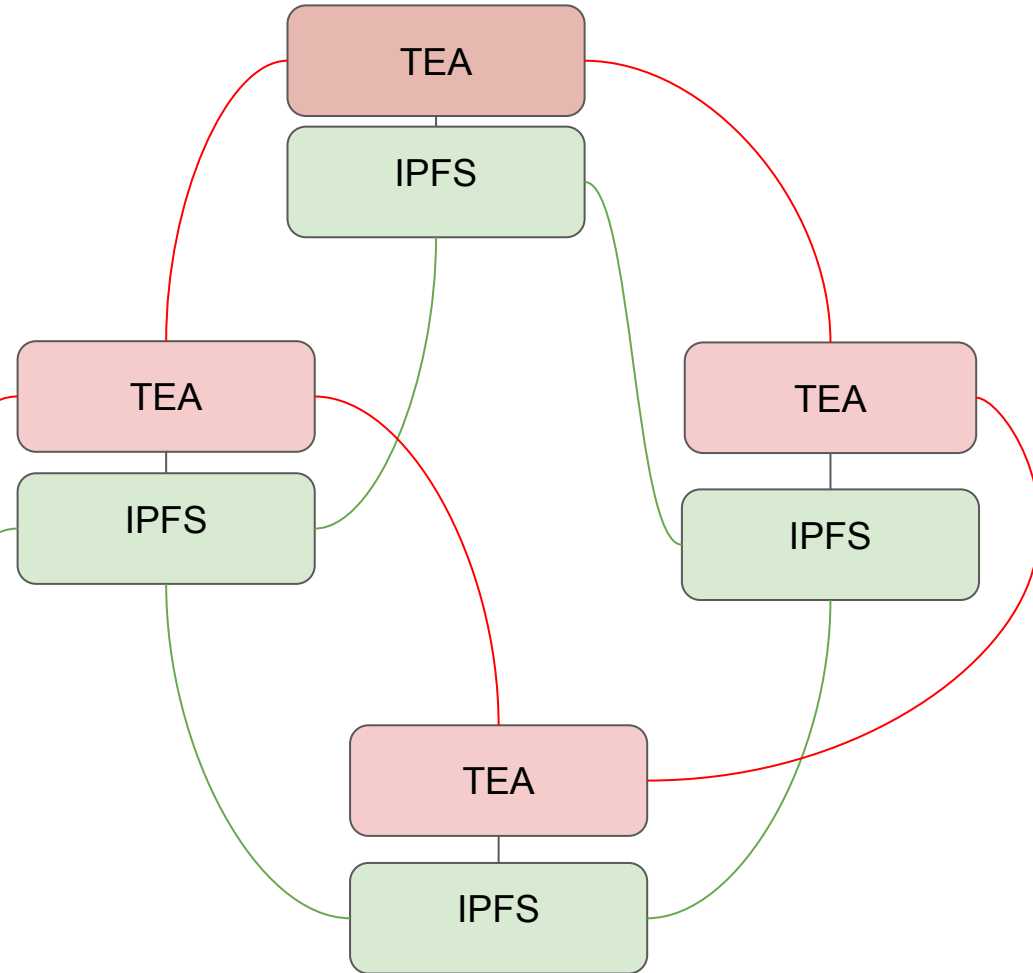
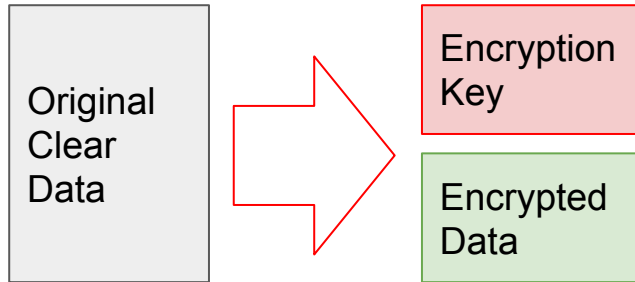
Edge computing

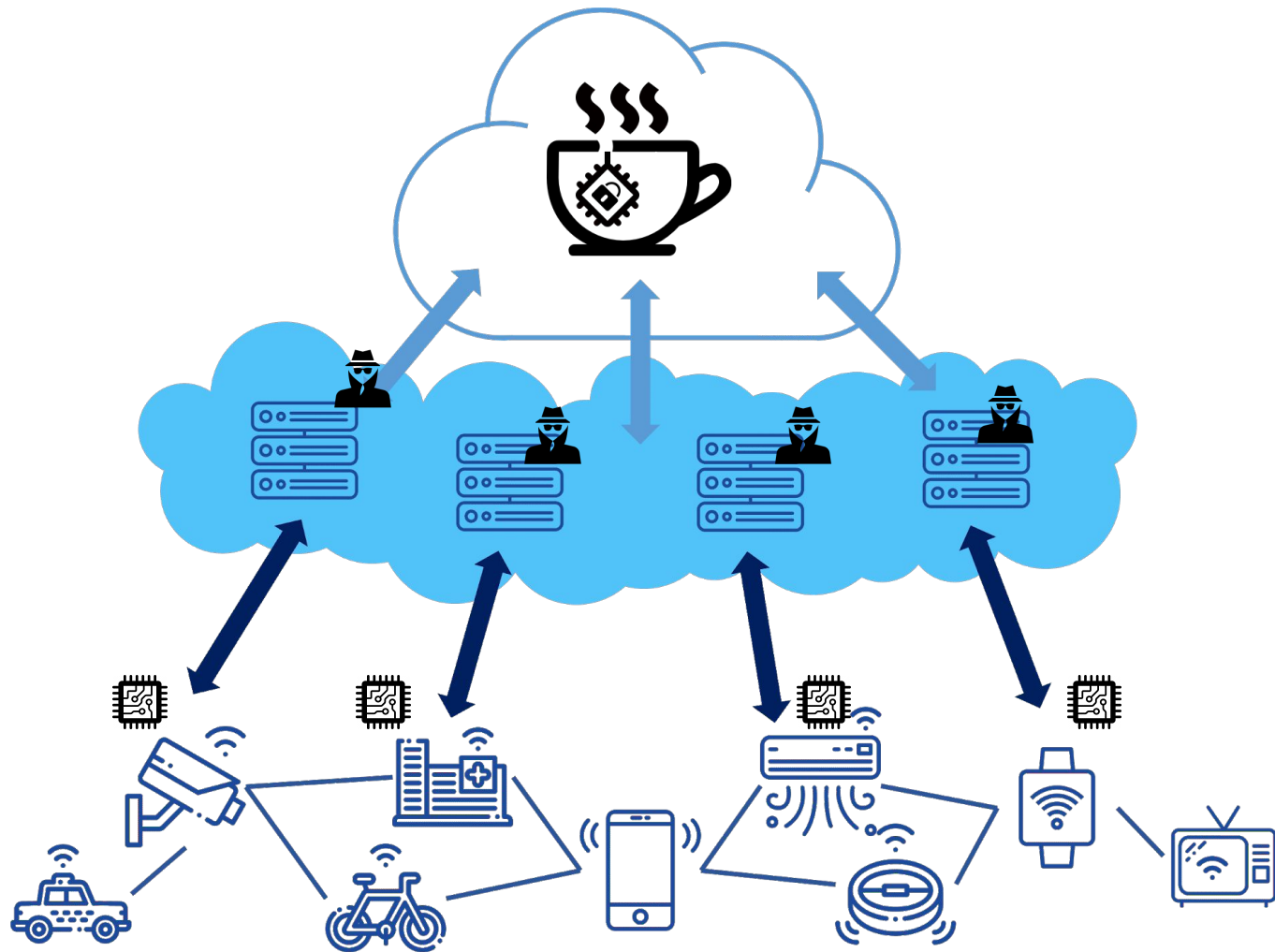


Encrypting data and store securely in TEA network is called Deployment Process

- Encryption Key transfer and store-in-memory between Trusted TEA's Secure Zone only (**Red route**)
- Encrypted data can be store publicly anywhere between IPFS nodes (**Green route**)

Client Secure Zone







Alice: I took a picture of a wild animal. I do not know what it is, but I know it has value to others. I hope to get paid from the usage of my picture but technically have my copyright protected



Bob: I wrote a Tensorflow image recognition model and compile to webassembly as an application. It can determine what is on a picture. I hope to get paid by the usage of my application, but no one can steal my code.

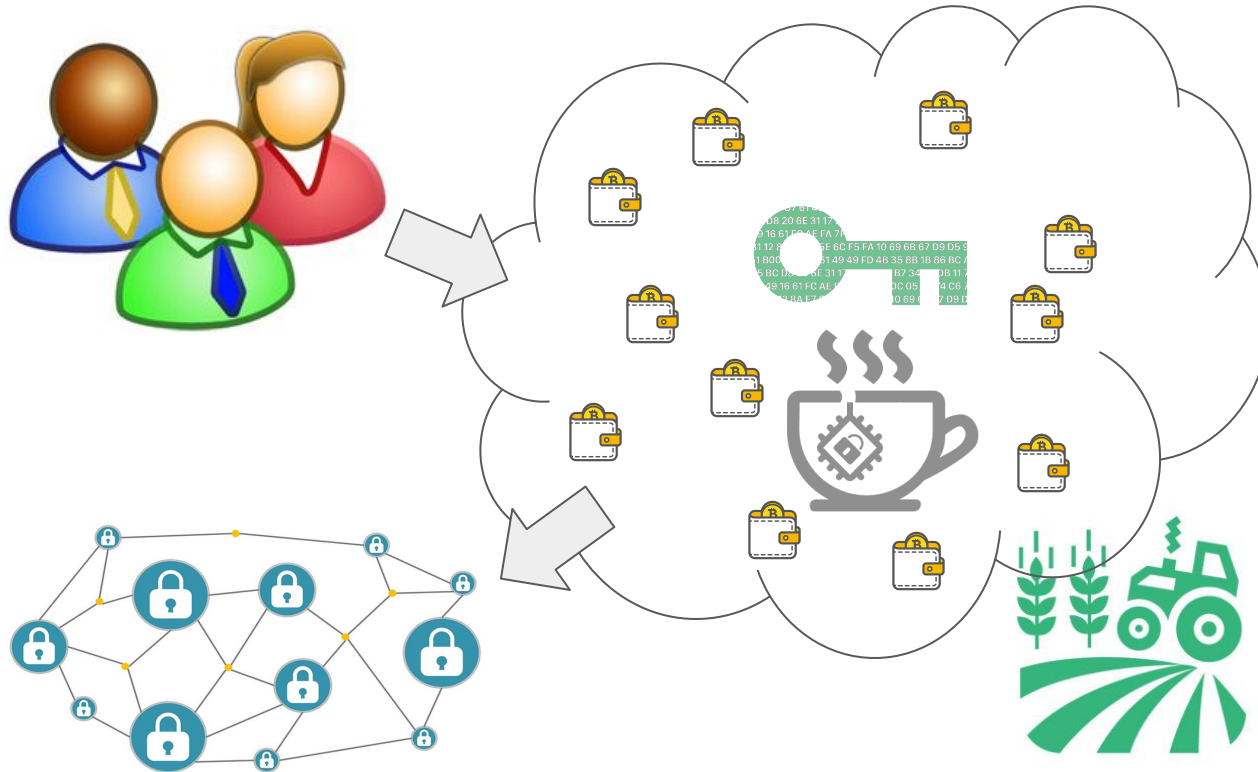


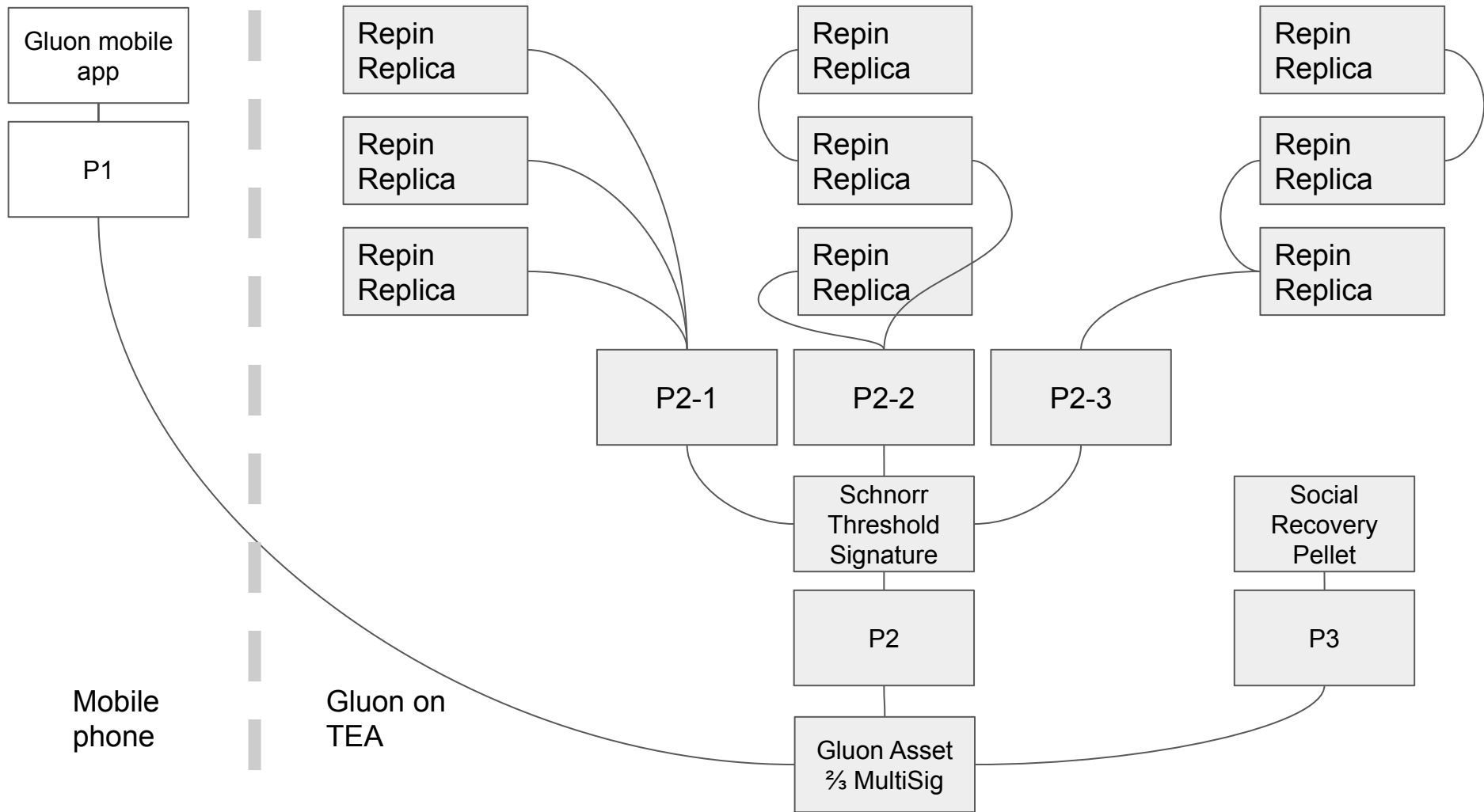
Charlie: I am an IPFS miner. I recently add a TEA module to my mining machine. I am now a IPFS + TEA miner so that I can min both Filecoin and TEA token. How could my clients trust me and send me sensitive information to compute?



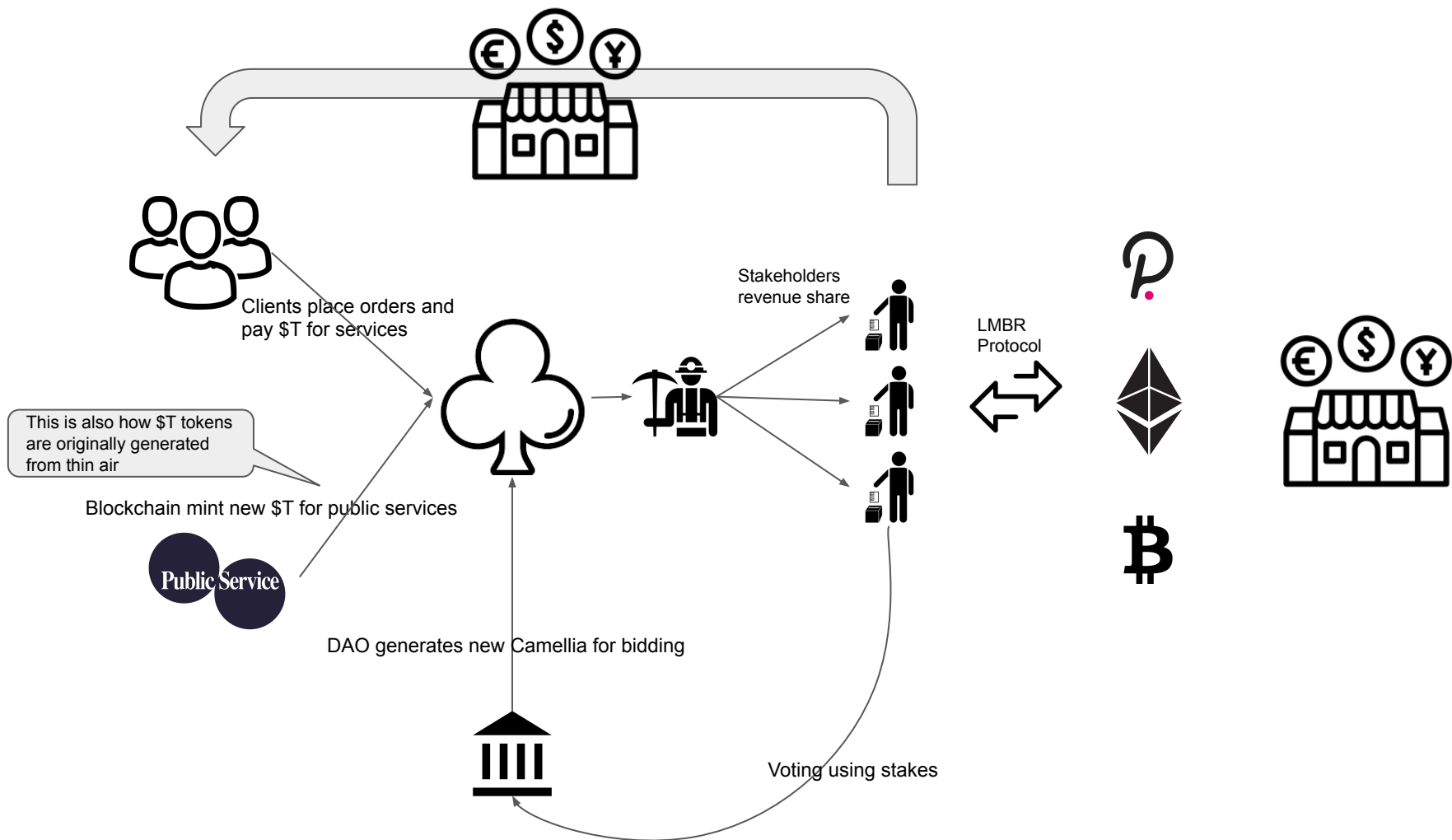
Dave: I am a naturalist and scientist, I have research fund on looking for wild animal behavior by analysing pictures. I cannot analyze every pictures manually, I can pay to run the AI algorithm on wild animal pictures from other paid photographers.

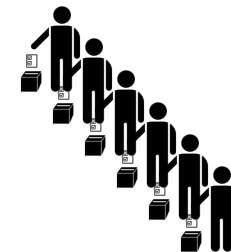
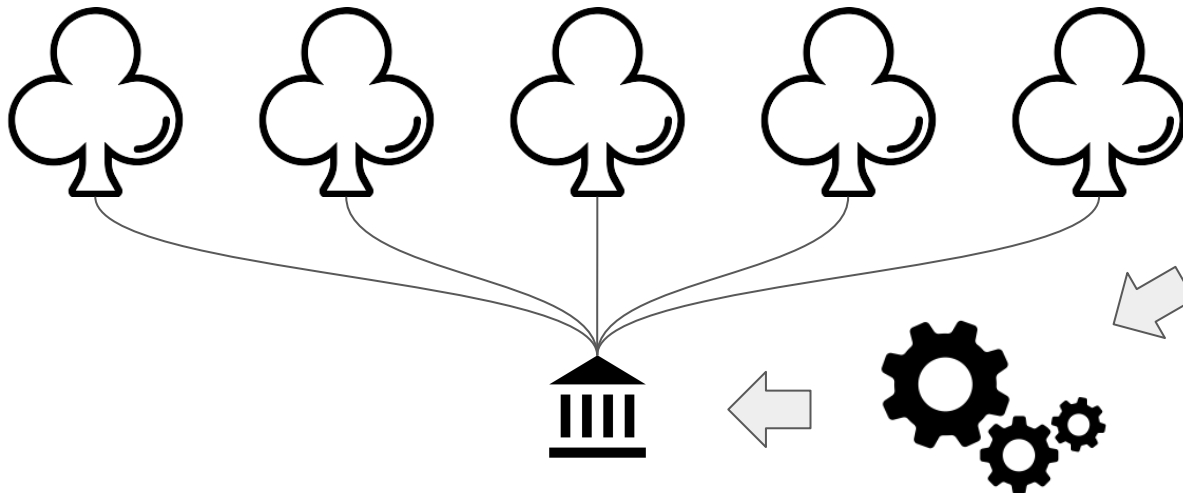
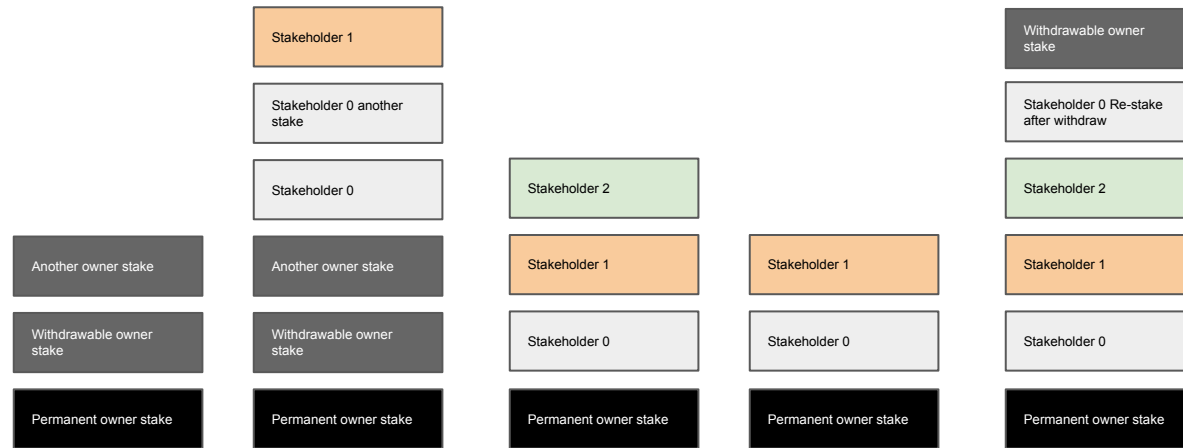
Gluton is NOT a hardware wallet, it is a blockchain powered TaaS(Trust as a Service) so that you do not need to own one





Clients	Miners	Investors	DAO
Buy TEA from market and pay for trusted computing tasks	Earn TEA by performing computing tasks	Earn TEA by staking TEA to miners as a Stakeholder	Burn TEA to control TEA supply
	Own Camellias (active TEA Nodes)	Do not own Camellia	Generate and sell new Camellia seeds based on birth_rate
	Can vote using stakes	Can vote using stakes	
	Earn TEA from public services		Mint new TEA to pay for public services





Camellia's probability to win in the task competition

- Positive correlation to stake weight but no linear
- Positive correlation to age of Camellia (Credit history factor)
- Negative correlation to age of Camellia (Performance factor)
- Capability checker

For stakeholder:

$\text{Revenue_share} = \text{this_slot_weight} \times \text{total_revenue} / \text{total_slots_weight}$

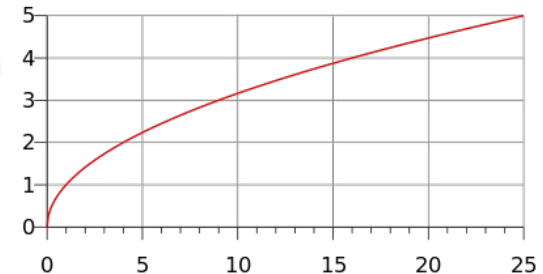
$\text{this_slot_weight} = (\text{this_slot_index} + 1). \text{sqrt}() - \text{this_slot_index}. \text{sqrt}()$

The algorithm encourages stakeholders to evenly distribute their stakes. Esp. to new Camellia which has lower index slots opening

Camellia life circle make it cannot keep best performance forever



Capital flows to where ROI can be maximized

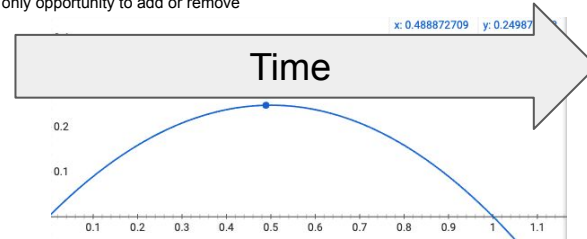
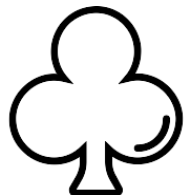
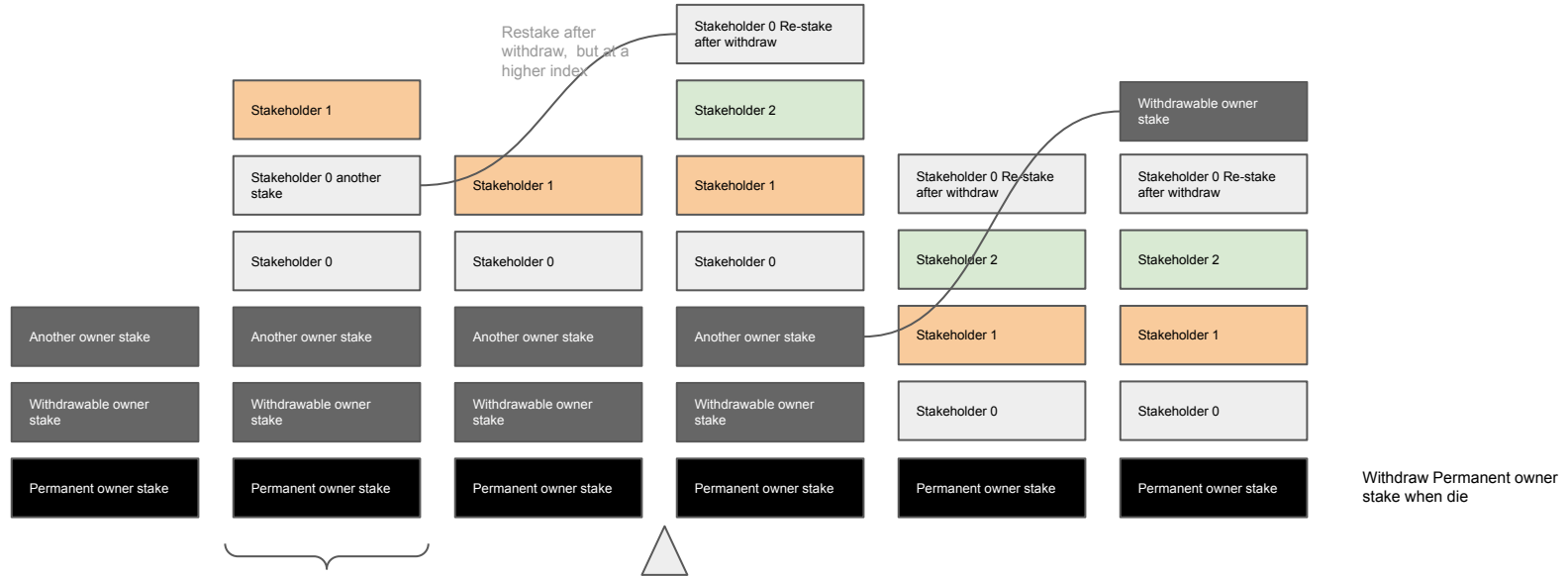


TEA Staking and Camellia Life Cycle

Slot_index
Weight reduced

$$\text{Revenue_share} = \frac{\text{this_slot_weight} \times \text{total_revenue}}{\text{total_slots_weight}}$$

$$\text{this_slot_weight} = (\text{this_slot_index} + 1) \cdot \sqrt{\text{this_slot_index}}$$



Token Economy: TEA

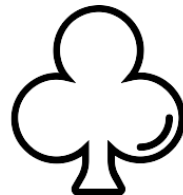
- TEA. Ticker TEA or T\$ internally
- Utility token. Stable coin pegging to computing cost not fiat.
- Used as Gas
- Unlimited supply
- No genesis block supply. Every TEA needs to be mined
- Born from public service rewards burnt by DAO when recycle
- Stake to Camellia for revenue sharing
- Stake to vote
- Stake to be used as weight of probability of task competition



$$\begin{aligned} & \text{computing_time} * \text{cpu_rate} \\ & + \\ & \text{net_traffic} * \text{traffic_rate} \\ & + \\ & \text{storage_time} * \text{storage_size} * \text{storage_rate} \\ & + \\ & \text{tip} \end{aligned}$$

Task Price in \$T

Token Economy: Camellia



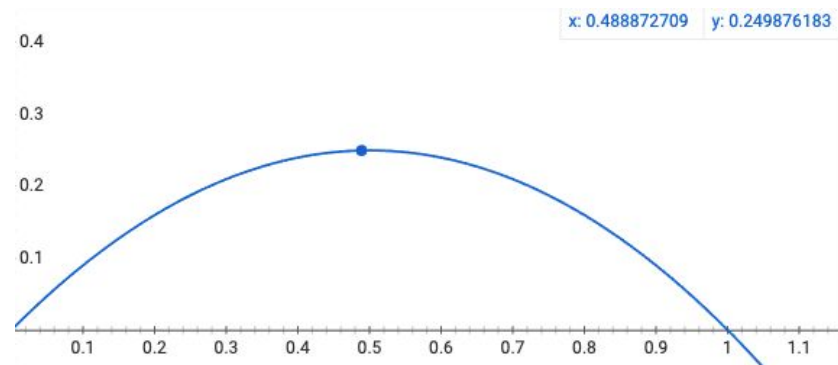
- An internal NFT represent a qualification of mining
- TEA Node can be activated when a Camellia associate with it
- Camellias are owned by miners
- Miners buy new Camellia seeds through a Birth Control Bidding
- Camellia has its life cycle. DAO generate it with birth control. DAO burns it when die.
- Camellia has technical stake used in diversity control
- Presale investors can buy Camellia for pre-mining (forerunner)
- Investors can stake to a Camellia for revenue sharing
- Miners can sell their Camellia, or let it die by withdrawing Permanent Owner Stack.

```
Camellia {  
  tech_stack; // enum of tech stack, used for diversity control  
  rip_block_height; // Camellia die at this block height  
  birth_block_height; // calculate the age. May be used to calculate credit_score  
}
```

$$\text{age_ratio} = (\text{current_block_height} - \text{birth_block_height}) / (\text{rip_block_height} - \text{birth_block_height})$$
$$\text{current_performance} = \text{age_ratio} \times (1 - \text{age_ratio}) \times \text{full_performance}$$

Best strategy of staking to a Camellia

- Invest early to occupy a lower index slot
- Long term hold and wait to its peak performance
- Withdraw when close to the end_of_life



DAO burns TEA and Camellia

- TEA from Camellia seeds auction
- TEA from penalty of bad actors
- Camellia from RIP
- Camellia from penalty of bad actors
- Fees from staking unstaking

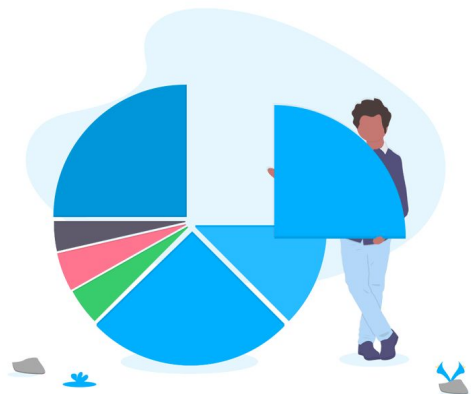
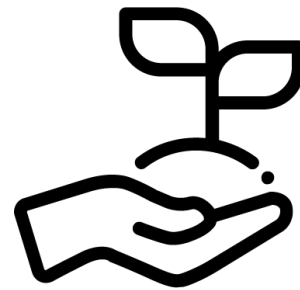
Control the supply of TEA and Camellia, maintain the price.

DAO voting

- Stake as weight. Free TEA cannot vote.
- Birth control rate
- Admission for new tech stack
- Version upgrade
- Penalty rules

Camellia Seeds Bidding: the Birth Control

- Prevent Sybil-Attack
- Enforce diversity by control of mining power between tech stacks
- Create scarcity



TEA Node tech stack

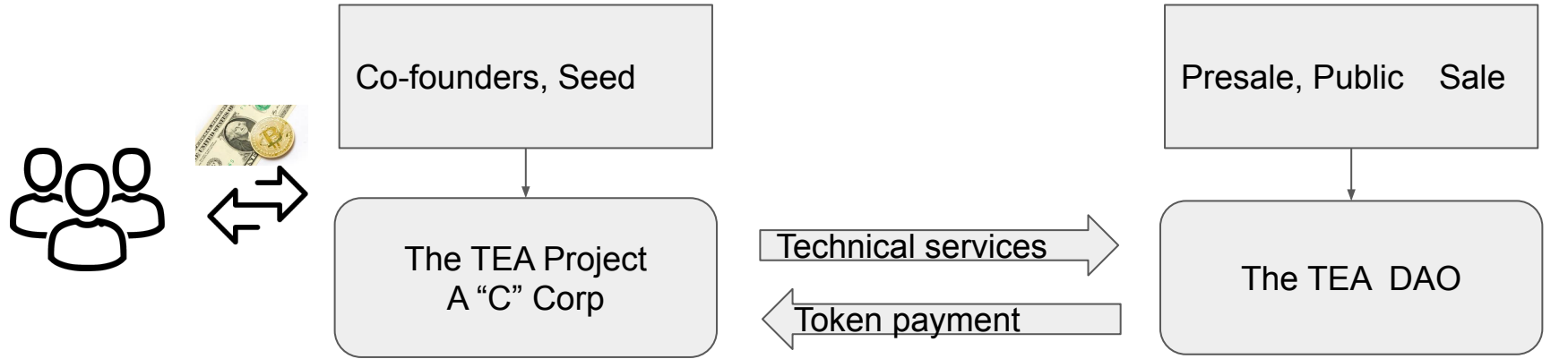


Seeds sale: Co founders and Angel investors. Corp shareholders

Presale: Sale Camellia seeds for pre-ming. Early investors are forerunners

Public sale: Sale Camellia seeds for Public mining.

IDO: Listed on DeX. TEA trading starts



Dev Team Corp revenue comes from:

- Early stage seeds sale
- Mining
- Develop core commercial dApps

Early investors revenue comes from:

- Low cost early pre-mining share
- Early stage staking
- Sale of TEA or Camellia

Miners revenue comes from:

- Client's computing tasks
- Public services tasks
- Resale Camellia

dApps Dev revenue comes from:

- dApps

Some ideas, some questions...

Do not use tea-layer1, use ink smart contract on Polkadot instead?

Make TEA a para-chain of Polkadot. XCMP? TEA becomes a Backend-as-a-Service?

Use DOT as gas directly instead of TEA token?