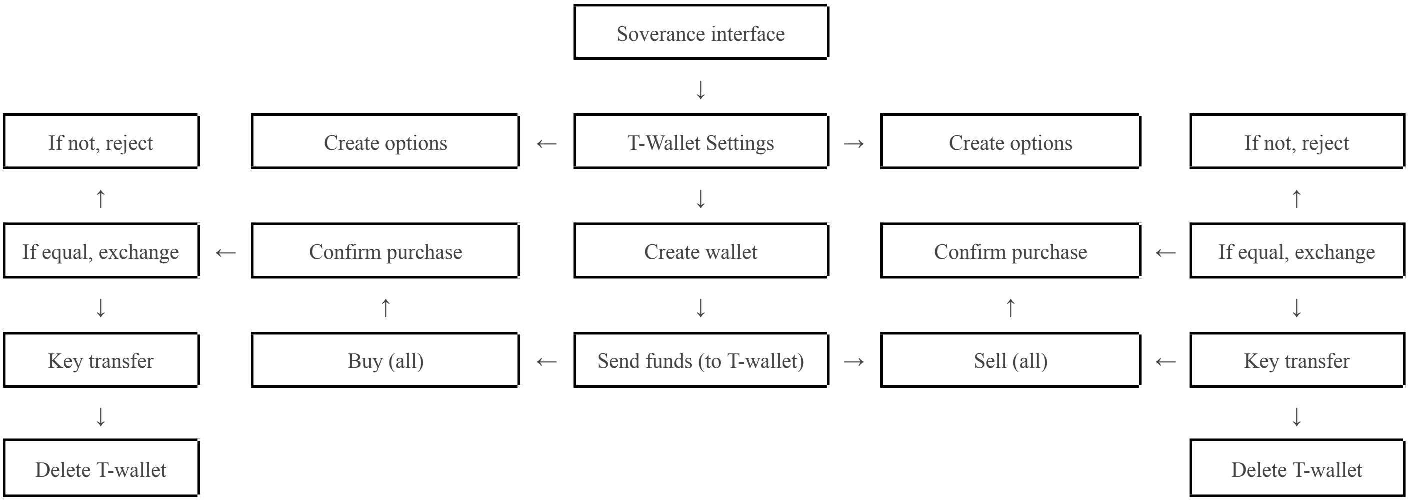
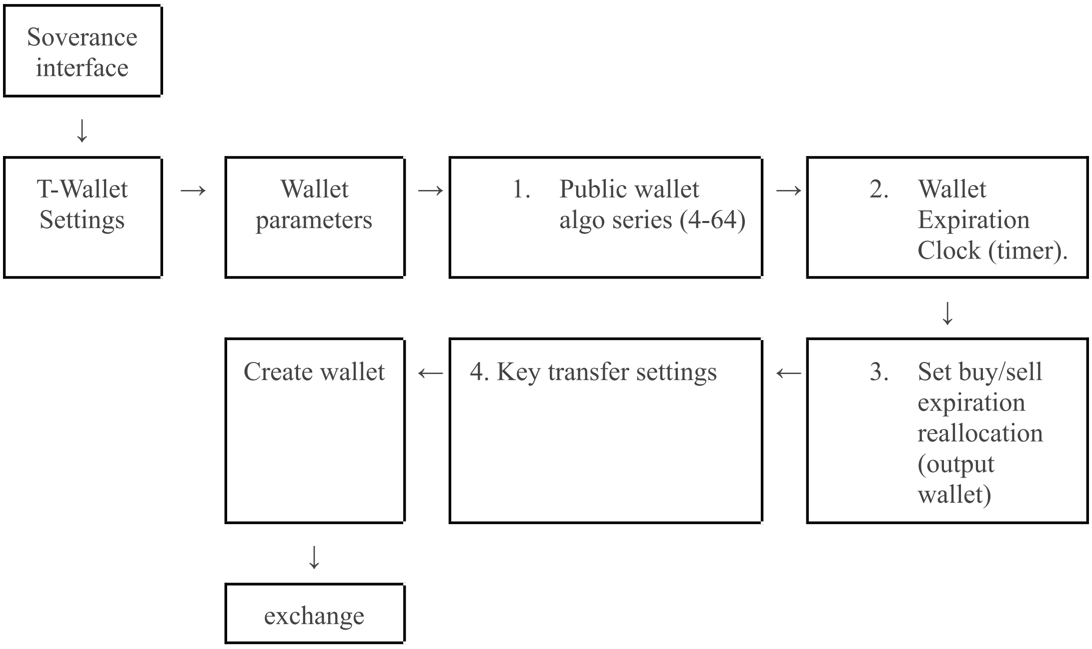
Soverance Marketplace:

Soverance is the privacy wallet solution created to solve the security vulnerabilities currently present as the cold storage standard within the hardware wallet space. With Soverance enabling individuals the ability to hold cryptographic assets, the means to trade these assets anonymously and securely is the next step in providing Sovereignty for all. The Soverance marketplace will be an internal software update that will not create a DEX but rather enable ZK peer-to-peer trade within end-to-end security.

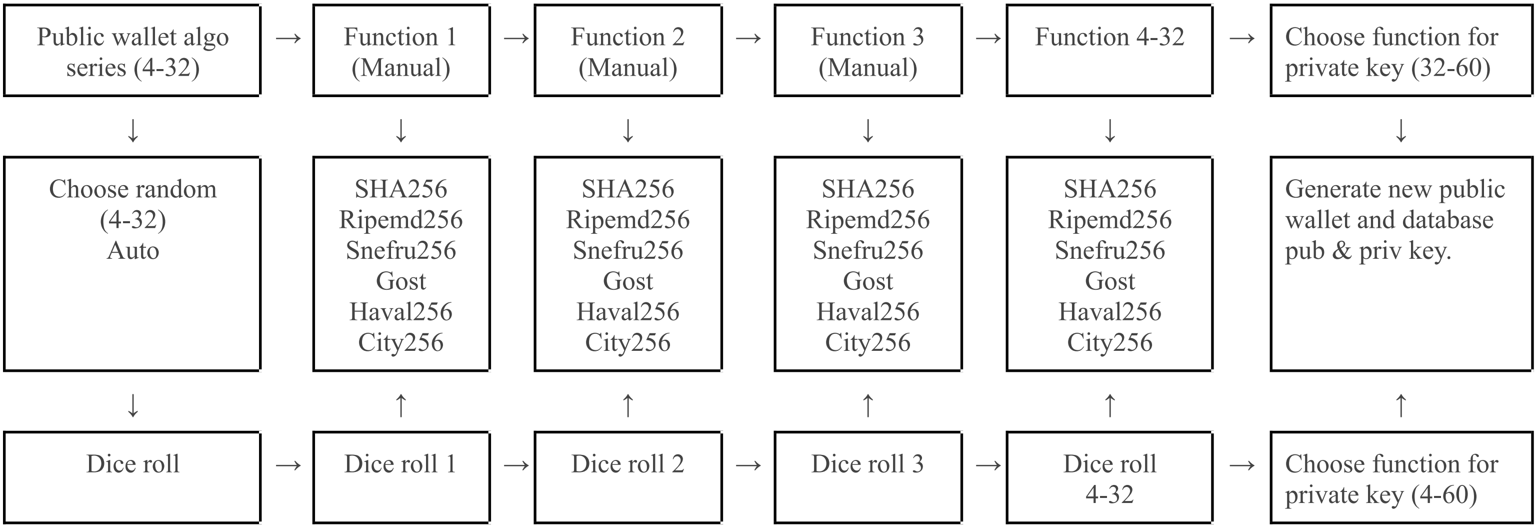
Temporary wallet creation & transfer mechanism:



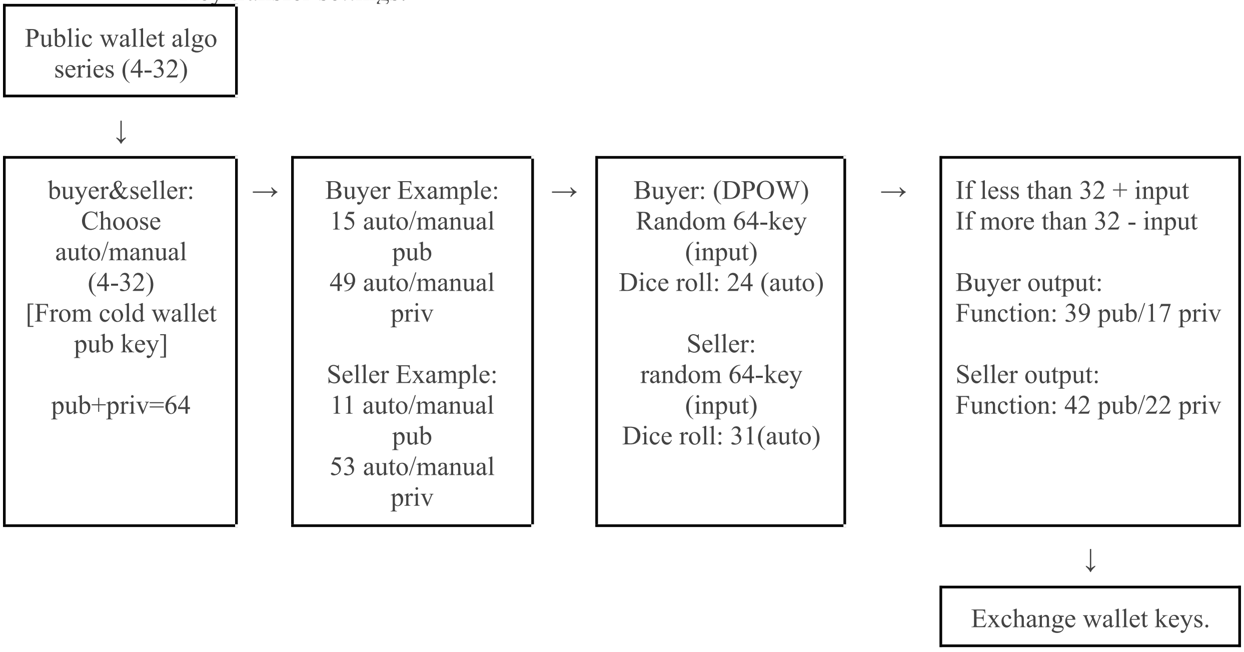
Wallet settings:


Soverance interface
T-Wallet Settings
Wallet parameters
1. Public wallet algo series (4-64)
2. Wallet Expiration Clock (timer).
3. Set buy/sell expiration reallocation (output wallet)
4. Key transfer settings
Create wallet
exchange

Public wallet algo series:


Public wallet algo series (4-32)
Function 1 (Manual)
Function 2 (Manual)
Function 3 (Manual)
Function 4-32
Choose function for private key (32-60)
Choose random (4-32) Auto
SHA256 Ripemd256 Snefru256 Gost Haval256 City256
SHA256 Ripemd256 Snefru256 Gost Haval256 City256
SHA256 Ripemd256 Snefru256 Gost Haval256 City256
SHA256 Ripemd256 Snefru256 Gost Haval256 City256
Generate new public wallet and database pub & priv key.
Dice roll
Dice roll 1
Dice roll 2
Dice roll 3
Dice roll 4-32
Choose function for private key (4-60)

Key transfer settings:


Public wallet algo series (4-32)
buyer&seller: Choose auto/manual (4-32) [From cold wallet pub key]
pub+priv=64
Buyer Example: 15 auto/manual pub
49 auto/manual priv
Seller Example: 11 auto/manual pub
53 auto/manual priv
Buyer: (DPOW) Random 64-key (input)
Dice roll: 24 (auto)
Seller: random 64-key (input)
Dice roll: 31(auto)
If less than 32 + input
If more than 32 - input
Buyer output: Function: 39 pub/17 priv
Seller output: Function: 42 pub/22 priv
Exchange wallet keys.

Definitions:
Wallet 0 - Private, cold storage wallets with infinite cold bins (internal KYC free cold storage).
Wallet 1 - Hot wallet (generic wallet).
T-wallet [Temporary wallet/Tachyon wallet] - timer based wallet made for key exchanging.

SoveranceVPN

To ensure the safety of Soverance user's while connecting to the internet, a decentralised VPN service is possible to retain 100% anonymity for our users. The VPN service is a multi-computer node chain that enables participants the ability to earn yield by hosting synchronisation data and transaction fees within an anonymous proxy server, verifying chain data without IP addresses being public. The system works in tangent with mining data and on-chain data where relay nodes within the VPN are compared to the external chain to provide transactional data

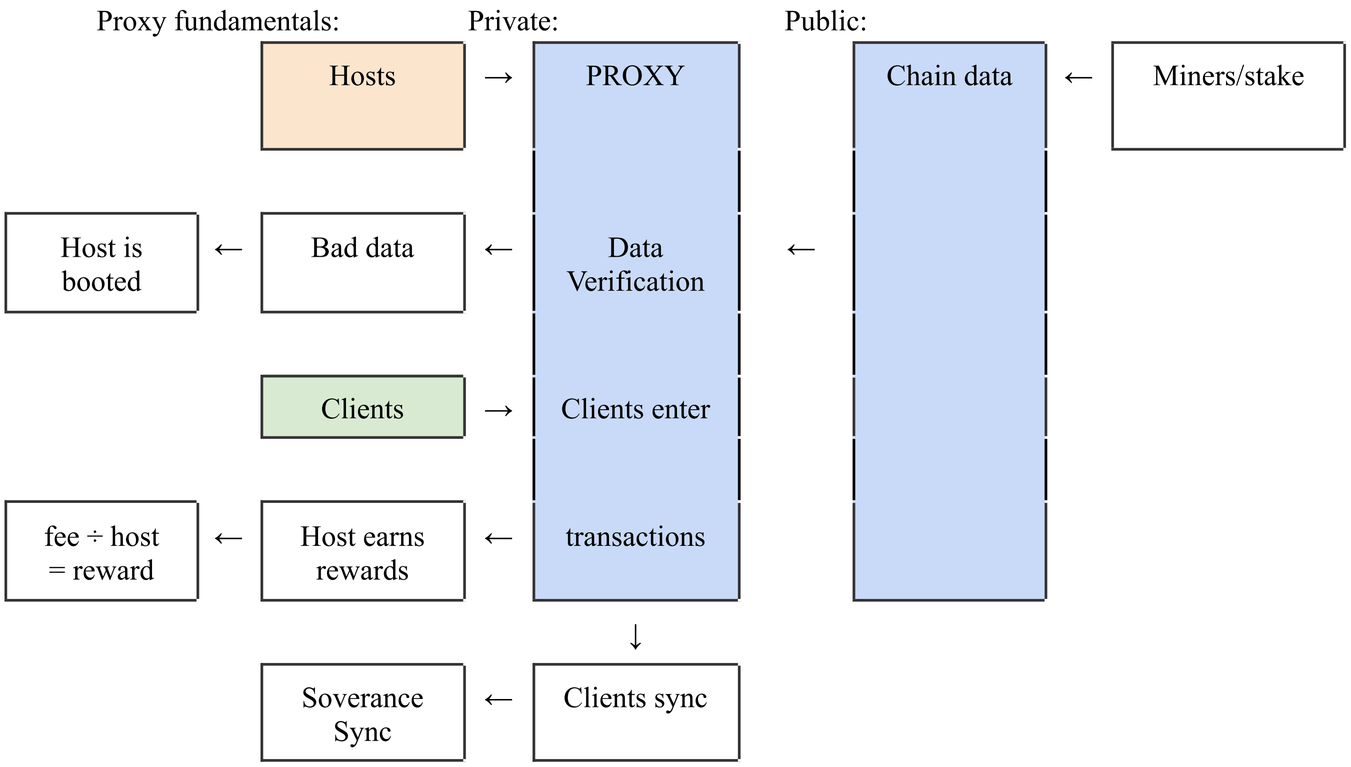
How does it work - Host:

- The VPN is downloaded through the anonymous transaction script via Soverance.org
- The node system is run through the software, forming a VPN.
- The software allows users to be a host or client (VPNH or VPNC).
- The Hosts set up creates or joins a proxy.
    - Proxy : Chain data (synchronisation)
    - Type : Select protocol of listed chains.
- The Hosts launch the VPNH module.
- To become a VPNH and earn rewards, the host will have to be mining the chain and directly transmitting data, or have a method to relay data, in real-time.
- VPNhs with the same task are connected and form a proxy server.
    - The VPNact as a gateways to the proxy and a tunnel from ISP detection.
- The proxy breaks data into bits to be sent through the nodes, get to their endpoints.
    -
- How does it work - Client:
- The client sets an encryption cipher for data transmission through the software portal.
    - The client activates the VPNc and selects the proxy.
- 2 or more nodes attach together to form an “anonymous bitproxy”
- A key exchange is made on connection: Diffie-hellman key exchange.
- The client secures connection to the proxy
- The client searches:
    - Search is sent into bitproxy.
    - The bitproxy is broken into nodes and VPNP hosts retrieves data.
    - Data comes back into bitproxy and is formed together to the correct key
    - The data exits bitproxy and into VPNp.
    - Software decrypts data
    - User sees results.

How it works to ensure true chain data.

- The proxy will be connected to live chain data, if possible.
- data that is not accurate will be thrown out and the malinformed IP addresses will be banned from the server.
- In turn, a decentralized synchronisation system is born.

SoveranceVPN Script Visualised:



Proxy fundamentals: Private: Public:

Hosts → PROXY Chain data ← Miners/stake

Host is booted ← Bad data ← Data Verification ←

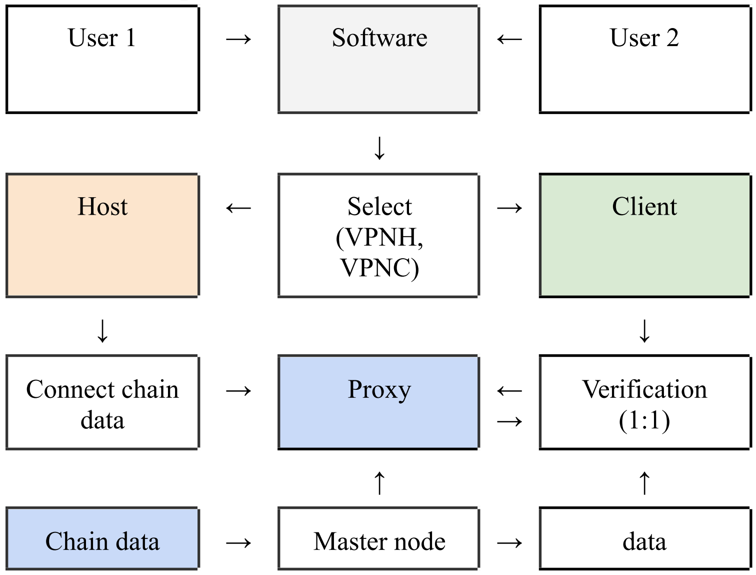
Clients → Clients enter

fee ÷ host = reward ← Host earns rewards ← transactions

↓

Soverance Sync ← Clients sync

Simplified



User 1 → Software ← User 2

↓

Host ← Select (VPNH, VPNC) → Client

↓ ↓

Connect chain data → Proxy ← → Verification (1:1)

↑ ↑

Chain data → Master node → data