ETHEREUM

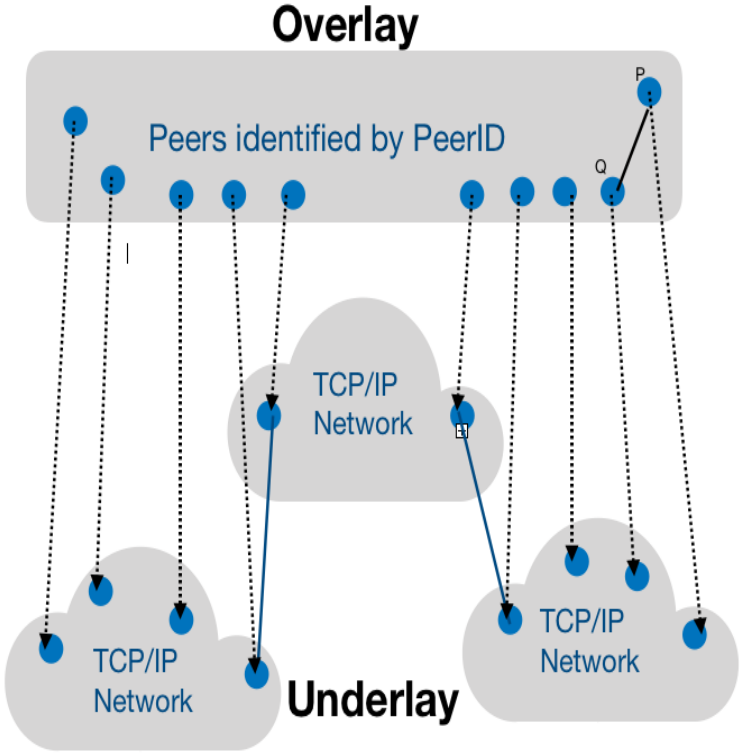
CONF.DR. CRISTIAN KEVORCHIAN

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

ETHEREUM este o platformă open-source globală pentru aplicații decentralizate.

P2P Network

O rețea peer-to-peer (P2P) este o **rețea de suprapunere(overlay network)** - adică este construită ca o abstracție peste Internet-ul public.



Matematic, poate fi considerat un graf orientat G = (V, E), unde V este o familie de noduri iar E este o familie de arce între noduri.

Fiecare nod **p** are un unic identificator **pid**.

Un arc (p, q) din E înseamnă că p realizează o conexiune directă prin care trimite un mesaj către q; adică p poate trimite un mesaj la q prin rețea cu identificarea lui q cu pid ca destinație.

În rețeaua TCP / IP de bază, rareori adresele IP asociate locațiilor fizice din apropiere, realizează conexiuni directe.

Mecanismele de întreținere a retelei P2P sunt utilizate pentru a menține actualizate informațiile de adiacență, asigurând astfel conexiunile între toate nodurile.

Reteaua P2P Ethereum

Software-ul oficial pentru nodul Ethereum, **Geth**, utilizează un protocol de identificare P2P (RLPx Node Discovery Protocol).

RLPx este o stivă de protocoale din categoria protocoalelor de transport bazat pe Kademlia care permite localizarea și stocarea eficientă de conținut într-o rețea P2P pe baza(User Datagram Protocol):

- Node Discovery and Network Formation
- Encrypted transport
- Flow Control
- Peer Reputation
- Security
- localised peer reputation model

Kademlia

Kademlia este o tabelă Hash distribuita pentru decentralizarea rețelelor de calculatoare.

În Ethereum, fiecare nod are asociat un ID, care este hașat cu SHA3 într-o valoare de 256 biți.

Kademlia definește distanța prin metrica XOR, astfel încât distanța dintre două numere de 256 biți este OR-ul lor exclusiv bit.

Fiecare nod are o structură de date formată din 256 bucket-i distincti, în care bucket-ul i stochează informații despre 16 noduri la distanța 2 ^ (i-1) până la 2 ^ i de propriul ID. Pentru a descoperi un nou nod, nodul Ethereum se alege ca țintă x, se parcurg nodurile din bucket-ii lui pentru a identifica 16 noduri cele mai apropiate de ținta x și se solicită fiecăruia să returneze 16 noduri din bucket-ii lor „mai aproape” de ținta x, rezultând cu până la 16x16 noduri recent descoperite. Dintre aceste 16x16 noduri recent descoperite, 16 noduri cele mai apropiate de ținta x sunt solicitate apoi să returneze 16 noduri chiar mai aproape de x. Procesul continuă iterativ până nu se găsesc noi noduri.

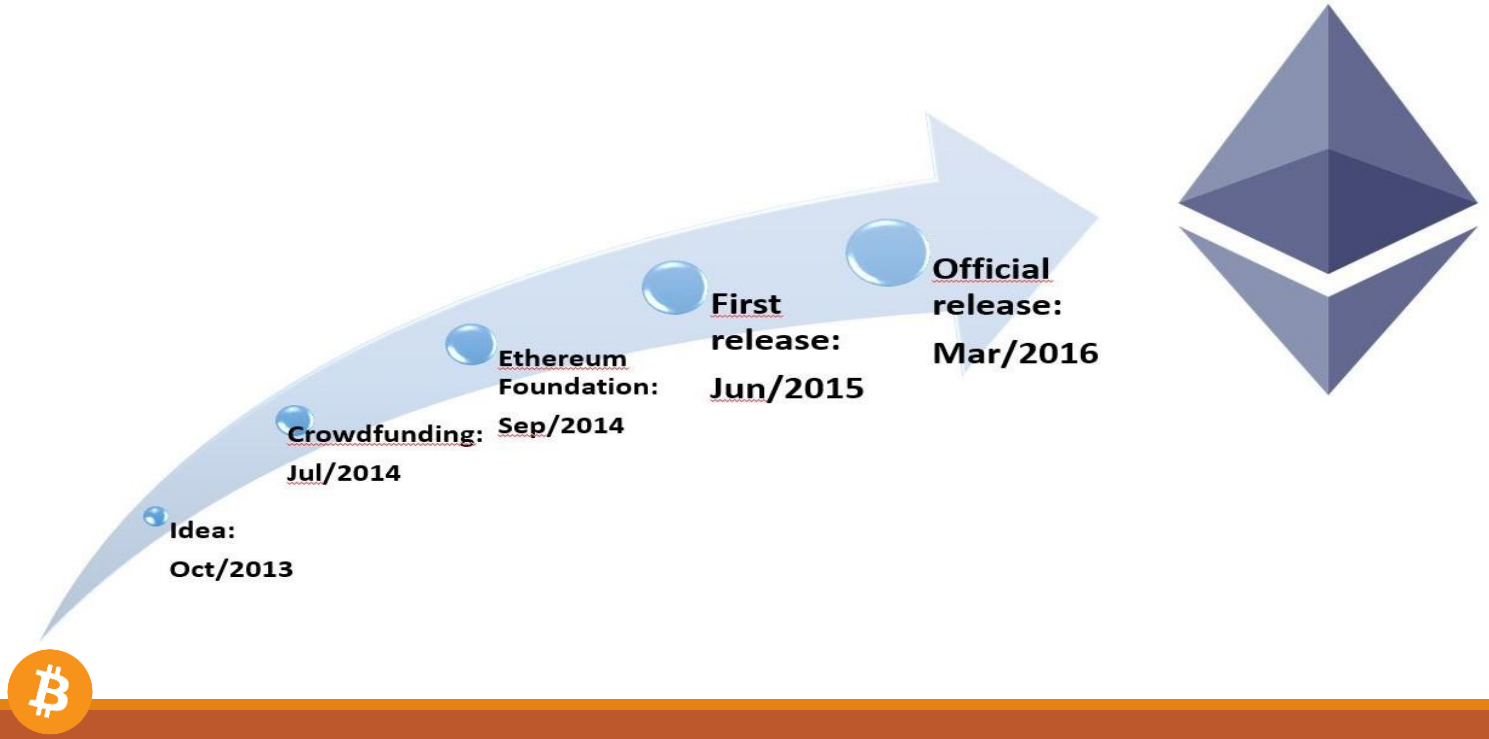
Rețeaua P2P

Fiecare nod P2P rulează următoarea stivă de task-uri:

- Noile transacții sunt plasate prin ”broadcast” la nivelul tuturor nodurilor.
- Prin intermediul fiecărui nod sunt colectate noi transacții într-un block.
- Fiecare nod dezvoltă resurse pentru a valida printr-un algoritm de selecție a blocului(proof-of-work,proof-of-stake). (Dificil de realizat Probabilistic. Cel care va termina mai repede va câștiga)
- Când un nod va fi validat proof-of-work/proof-of-stake, acesta va fi transmis tuturor nodurilor.
- Nodurile acceptă blocul numai dacă toate transacțiile associate acestuia sunt valide (verificarea semnăturii digitale) și nu deja efectuată(verifică toate transacțiile).
- Nodurile confirm acceptarea prin crearea următorului bloc din chain, folosind hash-ul blocului acceptat ca hash anterior.

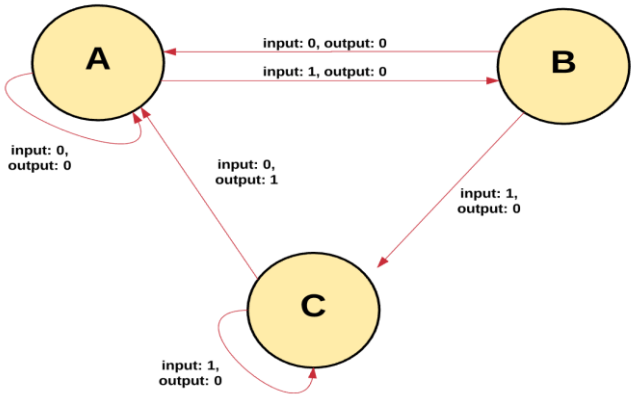
Ethereum este o platformă software descentralizată care permite scrierea și execuția de ”contracte inteligente”(en. Smart Contracts) și aplicații distribuite într-un context computational ”fault tolerant”, securizat și fără posibilitate de a fi controlat de terțe părți. Platforma implementează propria monedă virtuală, Ether. Platforma asociază un limbaj de programare (Turing complet) care rulează peste un blockchain, ajutând arhitecții și dezvoltatorii să dezvolte și să publice aplicații distribuite.

Evoluția ETHEREUM



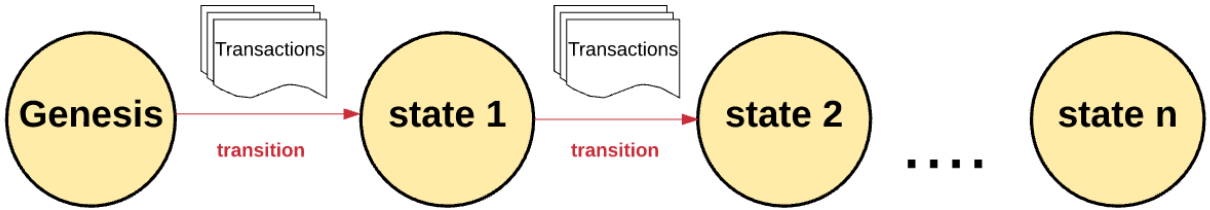
PARADIGMA BLOCKCHAIN ETHEREUM

ETHEREUM blockchain este în esență o mașină-stare bazată-pe-transacții (tcs. O masină-stare este o abstracție bazata pe citirea unor serii de caractere ca input, care vor conduce la trecerea într-o nouă stare odata cu scrierea unor caractere într-o bandă de ieșire).



ESM(ETHEREUM STATE MACHINE)

1. Starea inițială din care pornește ESM se numește ”genesis state”, mașina este în această stare înainte ca orice tranzacție să se producă.
2. Când o tranzacție este executată se realizează tranziția într-o altă stare. Această stare este un ”point in time” asociat stării curente a platformei Ethereum. O stare Ethereum caracterizează milioane de transacții.



Structura Blocului GENESIS

/ **genesis.json**

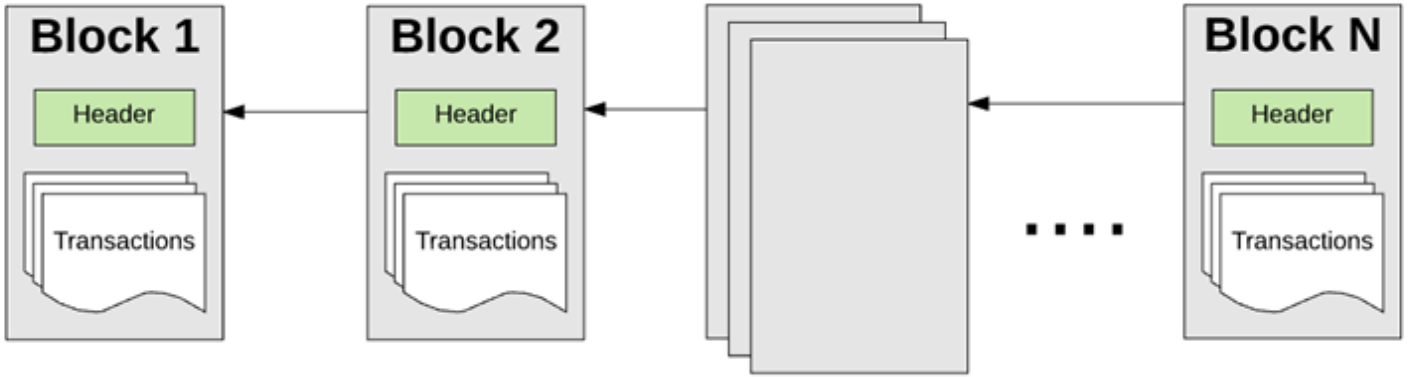
```
{ "alloc": {
    "0xca843569e3427144cead5e4d5999a3d0ccf92b8e": {"balance": "1000000000000000000000000000"},
    "0x0fbdc686b912d7722dc86510934589e0aaf3b55a": {"balance": "1000000000000000000000000000"}                    },
 "config": {
   "chainID": 68,
   "homesteadBlock": 0,
   "eip155Block": 0,
   "eip158Block": 0
 },
 "nonce": "0x0000000000000000",
 "difficulty": "0x0400",
 "mixhash": "0x0000000000000000000000000000000000000000000000000000000000000000",
 "coinbase": "0x0000000000000000000000000000000000000000",
 "timestamp": "0x00",
 "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
 "extraData": "0x43a3dfdb4j343b428c638c19837004b5ed33adb3db69cbdb7a38e1e50b1b82fa",
 "gasLimit": "0xffffffff"
}
```

Blocuri

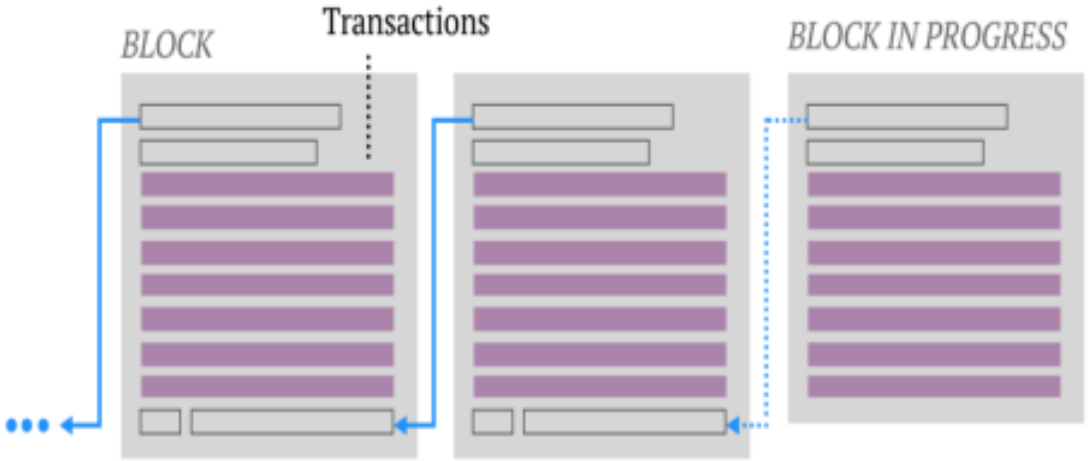
- Tranzacțiile sunt grupate în blocuri, iar fiecare bloc este conectat la cel anterior
- Pentru a trece dintr-o stare în alta este necesar ca transacția să fie validă.
- Pentru a fi clasificată drept ”validă” o transacție trebuie să treacă printr-un proces de validare numit ”**minare**”
- **Minarea se realizează când un grup de noduri(ex. calculatoare) consumă resurse pentru a crea un bloc de transacții valide**
- **Pentru ca un bloc să fie integrat în blockchain este necesar ca fiecare miner să furnizeze o demonstrație matematică drept garanție a validității transacției de tipul:**

dacă demonstrația există atunci blocul trebuie să fie valid

- Procesul se numește ”**proof of work**”


Block 1
Header
Transactions
Block 2
Header
Transactions
Block N
Header
Transactions

Blockchain



Complet distribuită similar BTC

Avantaje:

- **Puternic Securizată**
- **Transparentă**
- **Imutabilă**

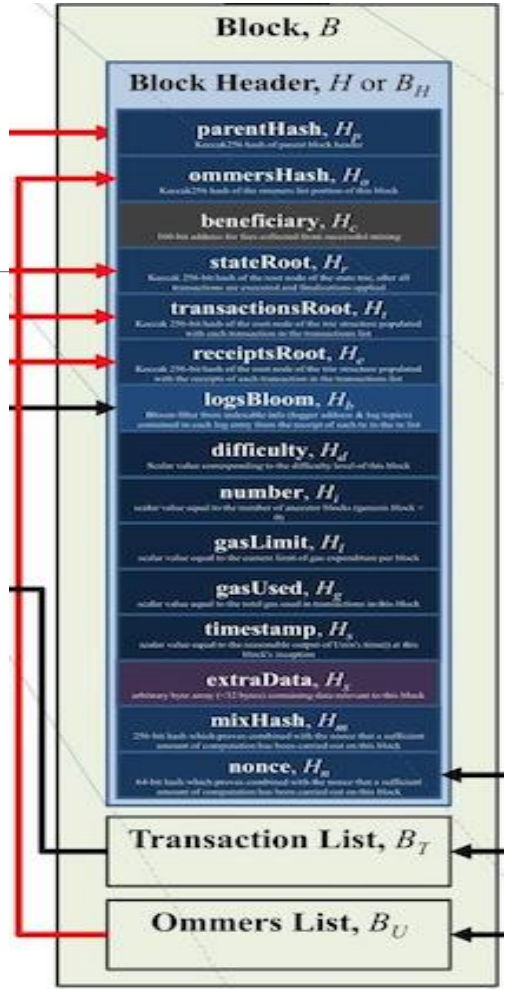
Dezavantaje:

- **Scalarea**
- **Performanțele**

Blockchain Ethereum

Blocurile se compun din trei elemente

- Lista transacțiilor
  - Lista tuturor transacțiilor incluse în bloc
- Header-ul blocului
  - Un grup de 15 elemente
- Lista ”Unchi”(en. Ommer List)
  - Lista ce include toate blocurile ”Unchi”

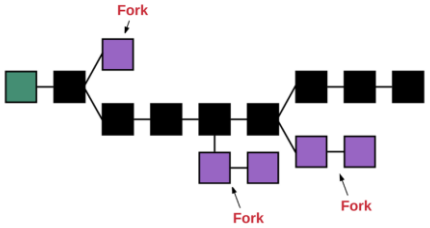


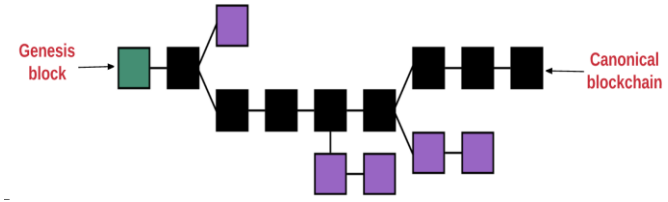
ETHER

- Fiecare ”miner” care validează un nou bloc pentru blockchain este recompensat cu un anumit volum de asset-uri digitale(digital tocken) numit **Ether**
- **Blockchain este un system transacțional de puncte de procesare cu partajarea-stării.**
- **Starea curentă este singura referință globală pe care toată lumea trebuie să o accepte.**
- **Având multiple stări sistemul ar fi haotic pentru că nu am putea decide care stare a fost cea care este asociată transacției corecte.**

Fork

- Generarea de căi multiple în blockchain stă la baza apariției fork-urilor.
- Apariția fork-urilor are efecte disruptive.(ex. 64 milioane ether distribuite prin hacking au condus la reluarea de la o stare anterioara atacului si splitarea Ethereum în două sisteme paralele).
- Pentru a stabili calea validă Ethereum utilizează un protocol numit **GHOST**(**G**reedy **H**eaviest **O**bserved **S**ubtree)
- Protocolul indică faptul că va fi aleasă calea care a oferit cel mai mare volum de calcul.





Dubla cheltuială(Double spending)

”Cheltuiala dublă” reprezintă un atac în care un volum dat de criptomonede este transacționat de multiple ori. Există câteva modalități principale de a efectua o dublă cheltuire:

- Sunt operate două tranzacții conflictuale în succesiune rapidă în rețeaua criptomonedei. Aceasta se numește ”race attac”
- Sunt create două transacții - una care vizează victima și una prin care se crediteaza atactorul. Se face o achiziție de asset-uri de la victimă, se obțin bunurile în baza ei apoi se execută transacția de autocreditare, în intervalul de validare al transacției cu victima.

Concepte de bază

- Criptografic (similar cu Bitcoin)
- Blockchain
  - Conturi-două tipuri(en.Accounts) și Portofele(Wallets)
  - Tranzacții
- Contracte-inteligente(en.Smart Contracts)
  - Solidity
    - Limbaj utilizat pentru dezvoltarea de contracte-inteligente

Fundamente Criptografice

- Funcții Hash
- Criptografie Simetrică
- Criptografie Asimetrică
- Semnături

Hash Functions

- BTC utilizează SHA-256
- Ethereum utilizează Keccak-256
  - Similar to SHA-3
  - Câștigă o competiție de Securitate Cibernetică în 2007
  - Utilizată pentru tot hashing-ul din Ethereum
  - Derivează diferit de block-cipher bazat pe hash-uri sau funcțiile SHA anterioare.

Digital Signatures (Digital Proof)

- Same use-case/cryptographic method (ECDSA) as BTC
- Signer uses private key to generate a signed message
- Signed message can be verified using the signer’s public key
- Hashes are signed in Ethereum, not the data itself

Conturi (en. Accounts)

- Starea-partajată global a Ethereum este marcată la nivelul unor obiecte(conturi) care interacționează între ele printr-un framework de mesagerie.
- Fiecare cont are asociată o stare și o adresă de 160-biti, ca identificatory al contului.
- Sunt înregistrate două categorii de conturi:
  - **Conturi deținute extern-** en. EOA(External Owned Accounts) controlate prin chei private și nu au cod asociat
  - Conturi pe bază de contract – controlate prin cod(contract) și au cod asociat cu interacțiune în blockchain

Contracte deținute extern(en. EOA)

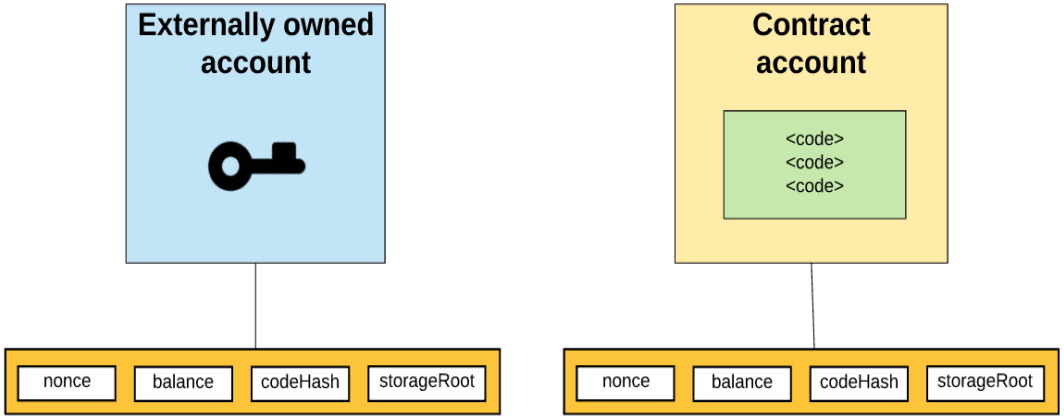
- Un EOA poate trimite mesaje la alt EOA sau la un cont contractual prin crearea si semnarea unei transacții cu cheia private
- Un mesaj dintre doua EOA reprezinta un simplu transfer al unei valori.
- Un mesaj între un EOA și un cont contractual activează un contract(cod) făcând astfel posibilă transferuri de token-uri, realizarea de calculi, creare de noi contracte etc.

Conturi Contractuale (en. Contract Account)

Conturile contractuale nu inițiază noi transacții. Pot genera transacții , ca răspuns la alte transacții

Conturile contractuale pot stoca și executa cod

- Au associate un ”**nonce”** si o ”balanță”
- codeHash – hash-ul asociat codului
- storageRoot conține ”Merkle tree” asociat datelor stocate



Starea Contului

Starea contului consta din patru componente:

- **Nonce** – dacă contul este detinut extern acest numar reprezinta numarul tranzactiilor initiate din acest cont. In cazul conturilor contractuale el desemneaza numarul de contracte.
- **Balanța** Numarul de Wei detinut de aceasta adresa.(Wei = 10^0 Wei,Ada = 10^3 Wei,Babbage = 10^6 Wei, Shannon = 10^9 Wei, **Szabo** = 10^12 Wei, **Finney** = 10^15 Wei, **Ether** = 10^18 Wei).
- **storageRoot** – Hashul nodului radacină asociat Merkle tree.
- **codeHash** - Hashul codului EVM asociat contului

Exemple

Private Key:

0x2dcef1bfb03d6a950f91c573616cdd778d9581690db1cc43141f7cca06fd08ee

- Cheia Privata Ethereum are 66 caractere (with 0x appended).

Adresa:

0xA6fA5e50da698F6E4128994a4c1ED345E98Df50

- Cheia private Ethereum mapează direct adresa.