Smart Contracts

John R Williams, MIT

The FinTech Revolution

jrw@mit.edu

Ethereum Wallet

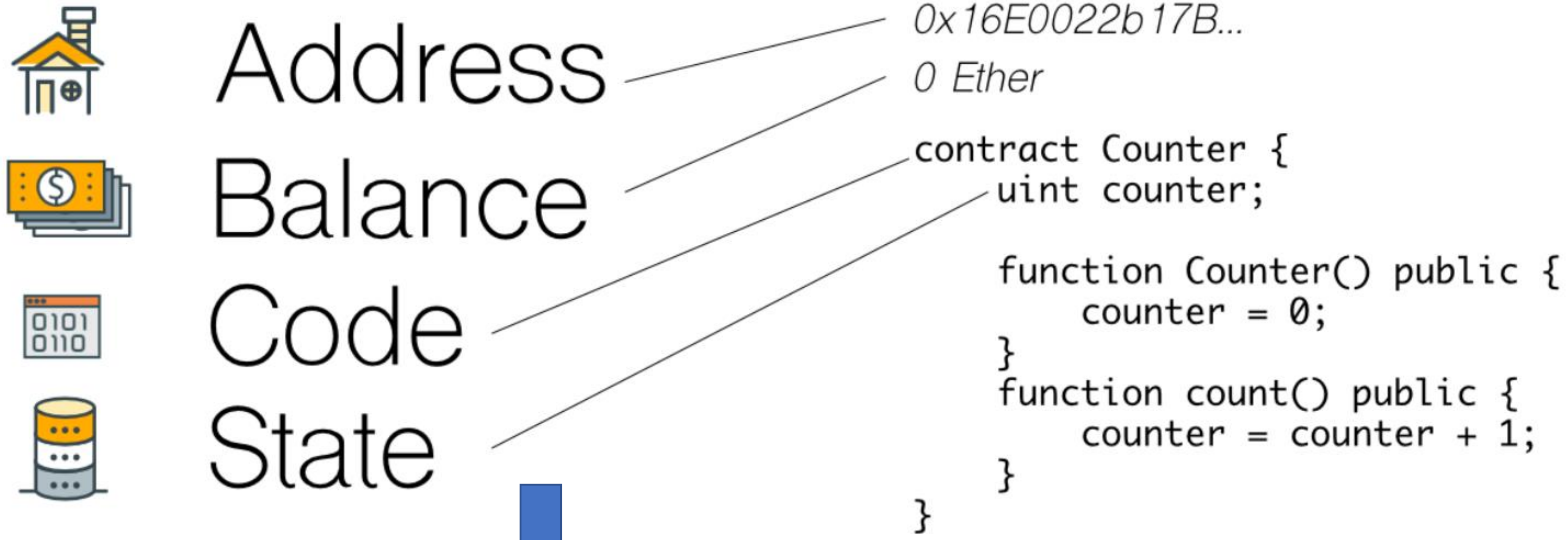
Address — *0x16E0022b17B...*

Balance — *0 Ether*

State

Logs – changes are logged and events raised

Ethereum Smart Contract Structure



```
contract Counter {
         uint counter;

         function Counter() public {
               counter = 0;
         }
         function count() public {
               counter = counter + 1;
         }
}
```

events in the log can be monitored

It would be nice if a smart contract could do this

Two parties agree to the terms of a contract, and it is written as code into the blockchain.

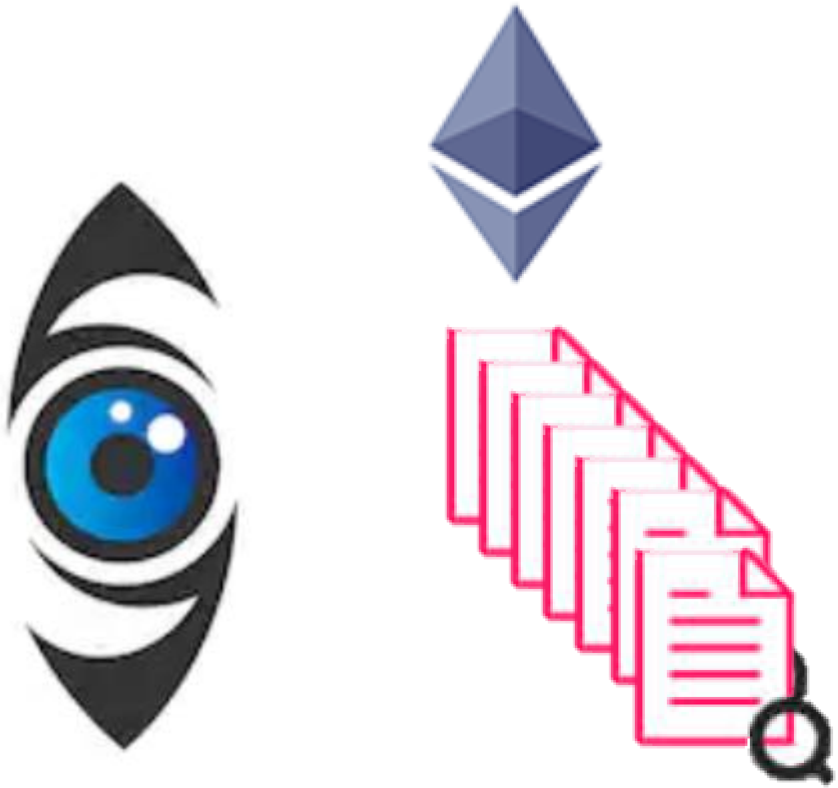
When a triggering event — like an expiration date — occurs, the contract executes itself according to the coded terms.

Smart contract can’t watch the world for events

Oh I see a tornado is coming I should sell everything

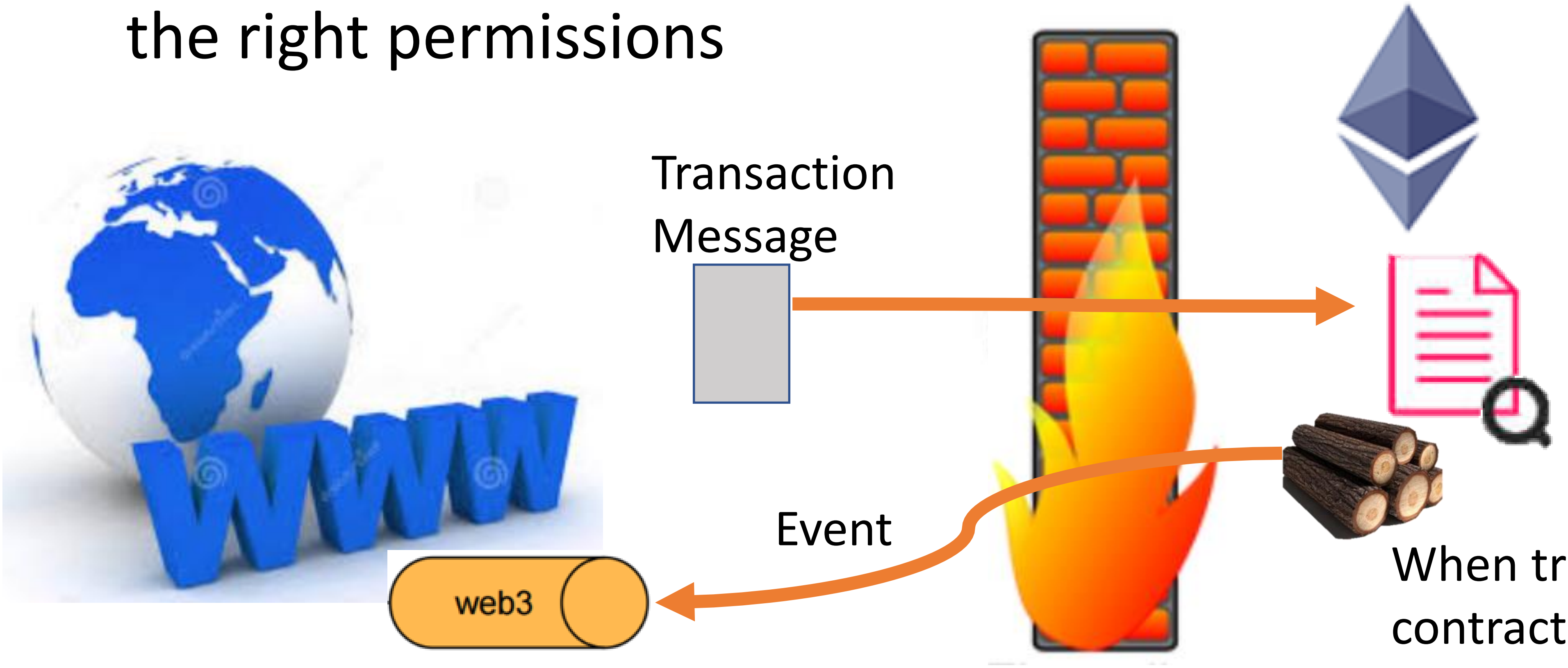
Every Ethereum node as a version of the smart contract

We can’t synchronize if each version of the contract has its own independence to act?



Workflow of updating a contract

Transaction can update state of all copies of contract if it has the right permissions



Client App

When transaction or contract is put into block log is wirtten

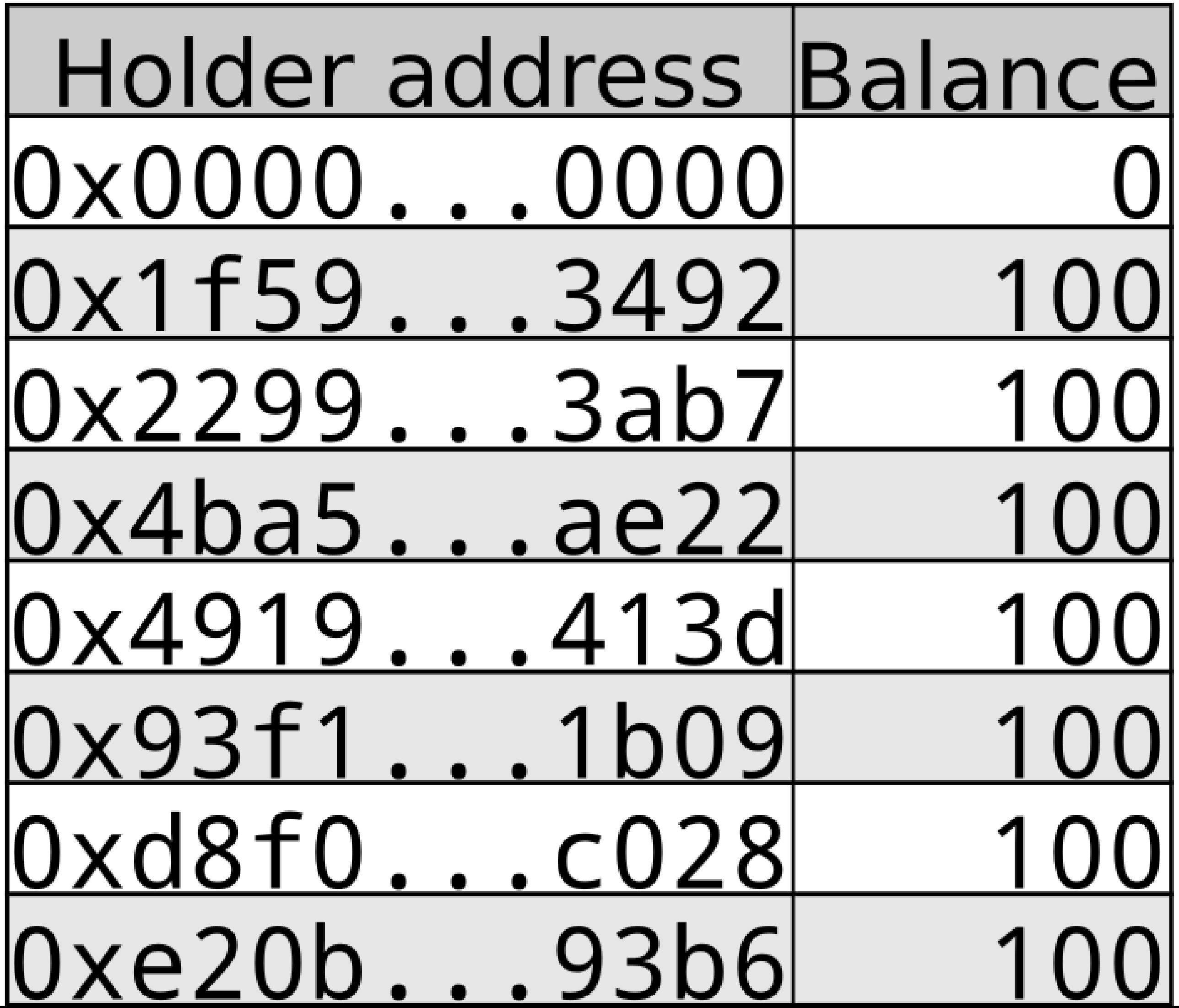
Ethereum smart contracts cannot

- Watch for events in the outside world
- Query the internet
- Query web servers
- Run code continuously to monitor the blockchain
- Watch for events on the blockchain

Ethereum contract limitations

- Ethereum virtual machine (EVM) constraints
- High costs (the contract runs on every Ethereum node)
- Gas limits (every transaction on a contract uses gas)
- No confidentiality/privacy on code visibility
- No ability to scale across “servers”
- Code is immutable and is locked in the blockchain (bad code cannot be modified)
- Should contain a “kill” function to disable rogue code. (DAO story)
- Boot up of node requires every smart contract ever written is run again

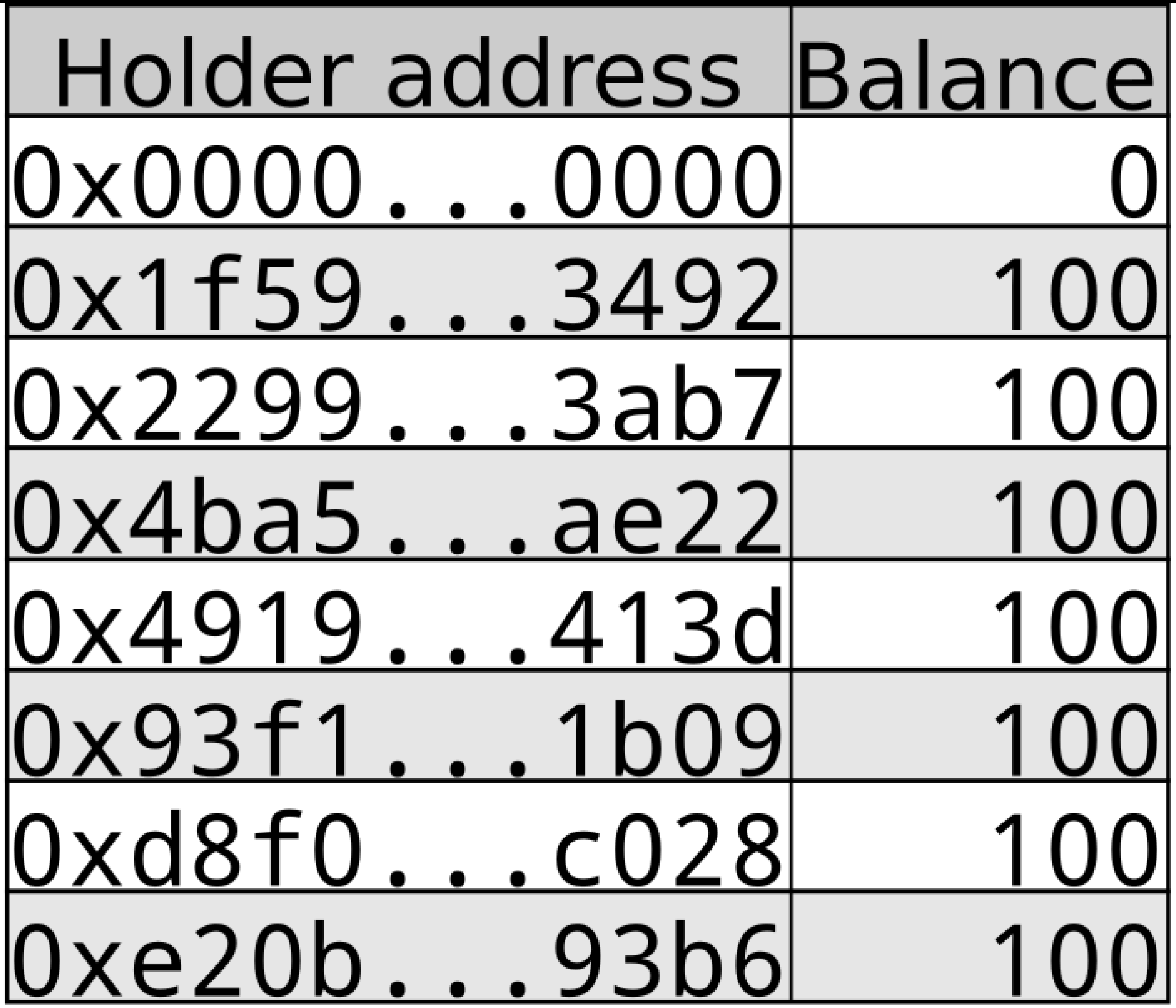
ERC20 standar tokens – Smart Contract



Holder address	Balance
0x0000...0000	0
0x1f59...3492	100
0x2299...3ab7	100
0x4ba5...ae22	100
0x4919...413d	100
0x93f1...1b09	100
0xd8f0...c028	100
0xe20b...93b6	100

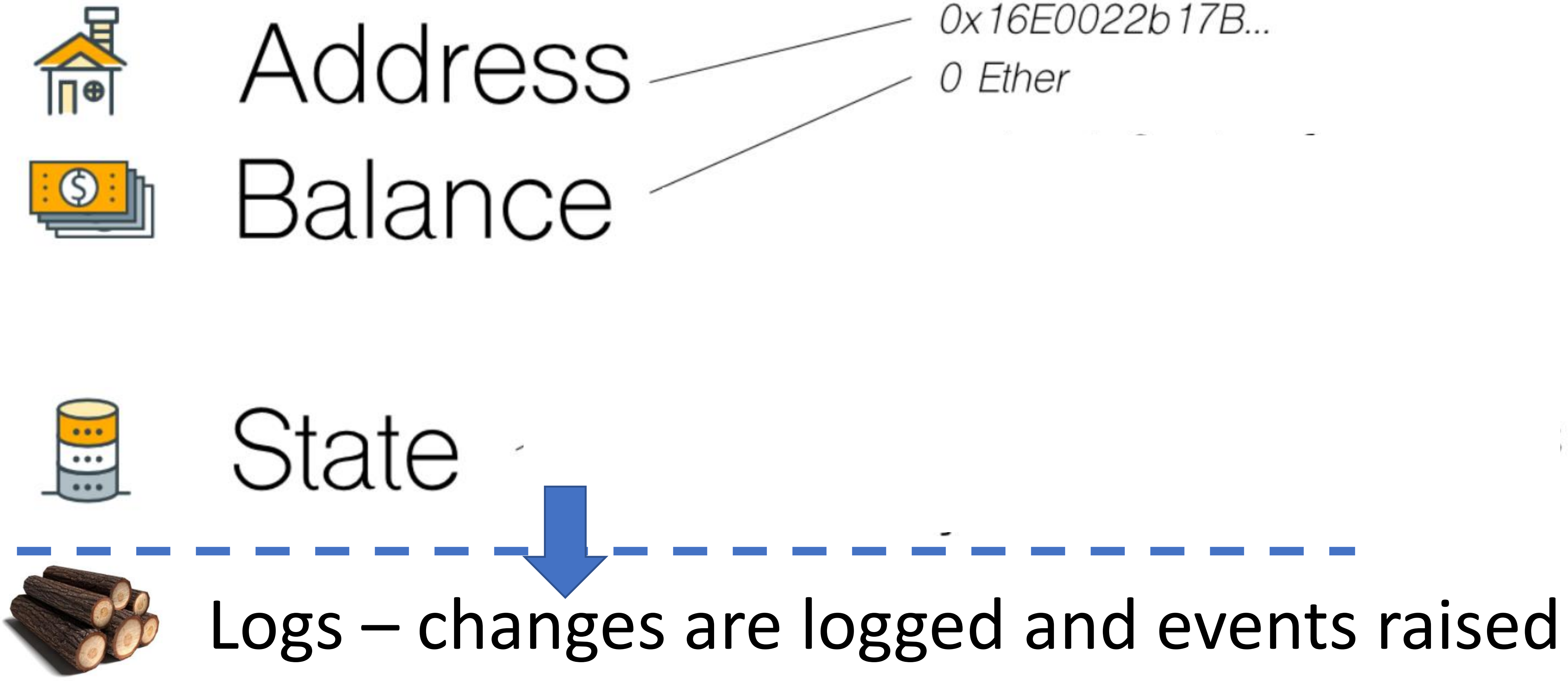
ICO Tokens

The initial contract for token creation has a “wallet” holding every exchange of that token.



Holder address	Balance
0x0000...0000	0
0x1f59...3492	100
0x2299...3ab7	100
0x4ba5...ae22	100
0x4919...413d	100
0x93f1...1b09	100
0xd8f0...c028	100
0xe20b...93b6	100

External Ethereum Account



Address

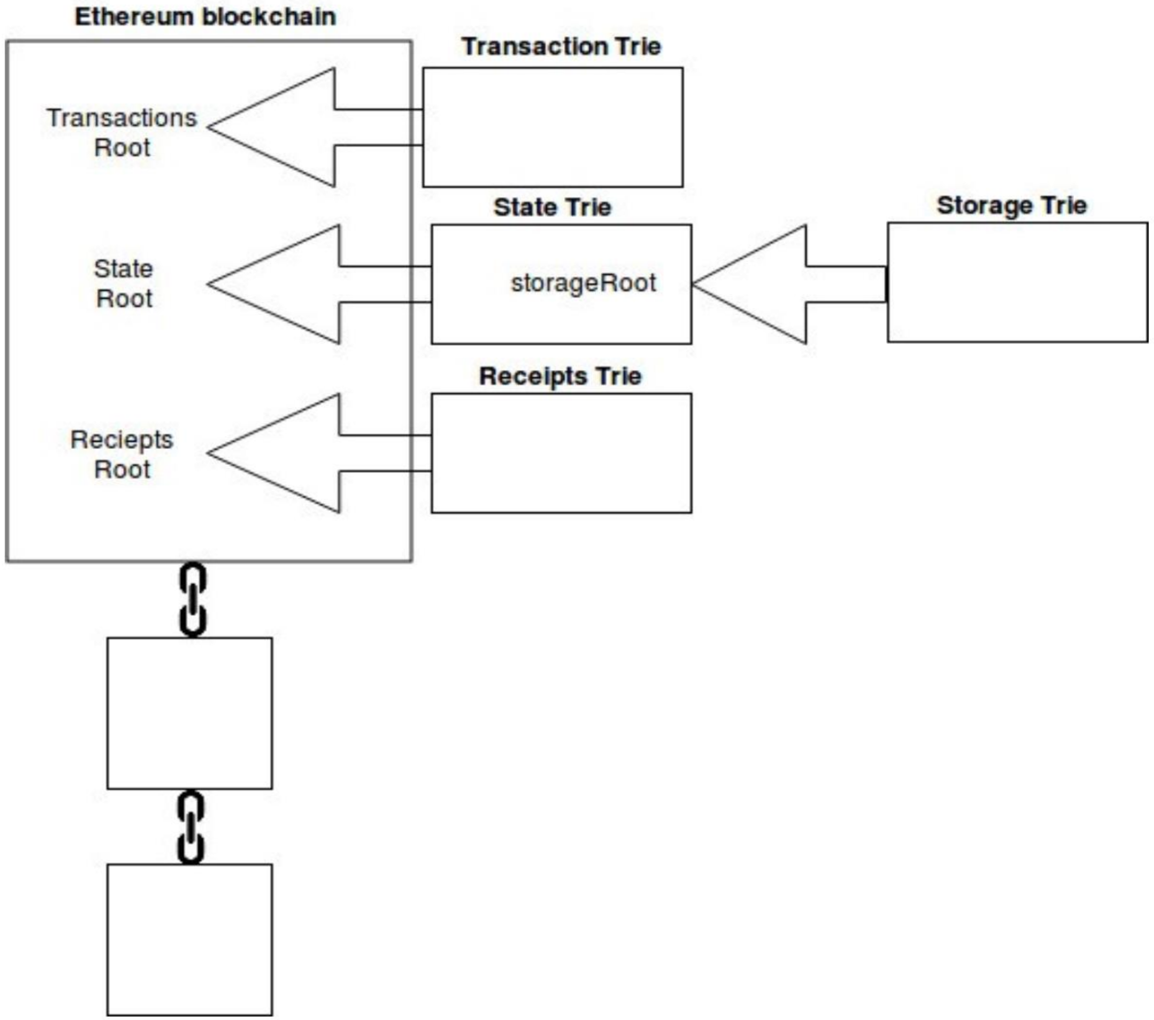
Balance

0x16E0022b17B...

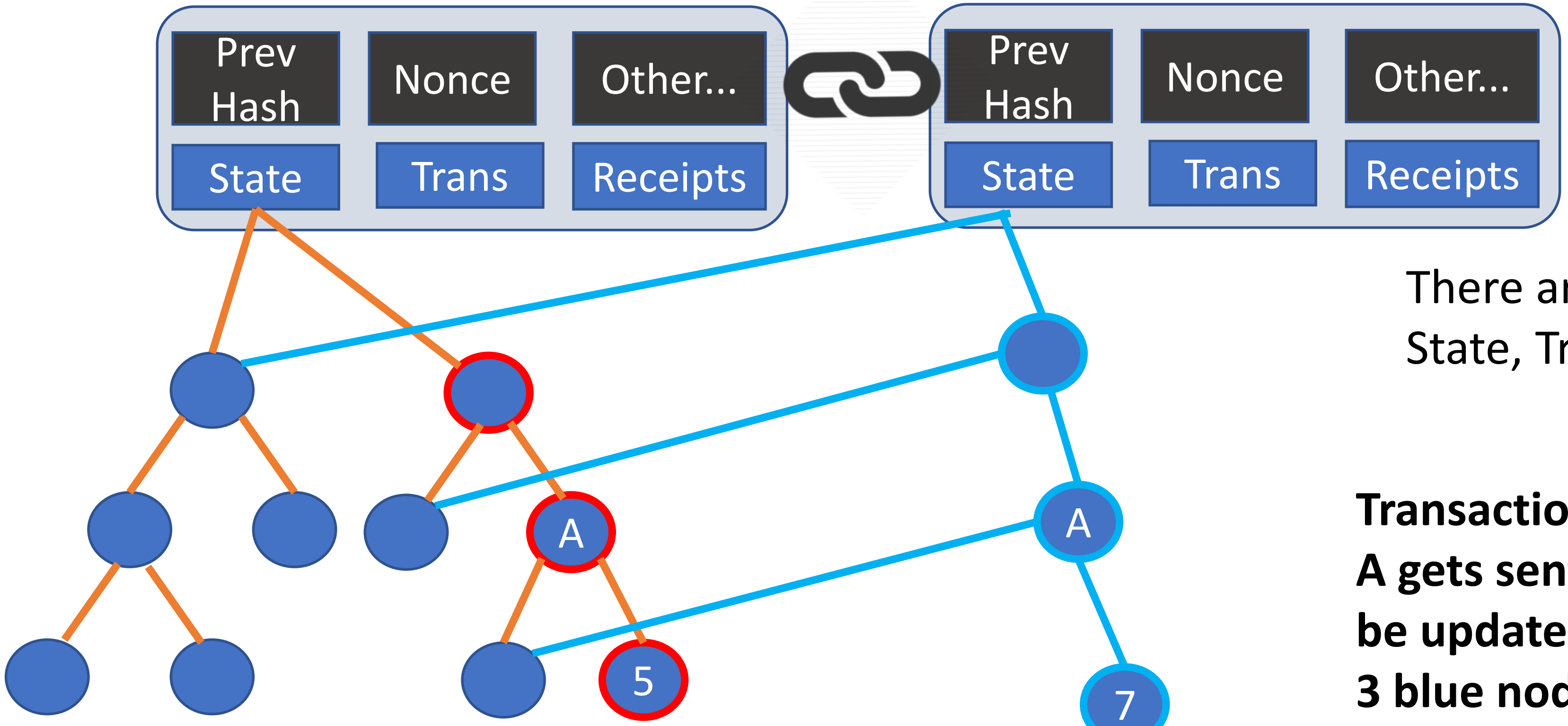
0 Ether

State

Logs – changes are logged and events raised


Ethereum blockchain
Transaction Trie
Transactions Root
State Trie
Storage Trie
State Root
storageRoot
Receipts Trie
Reciepts Root

Example Patricia Tries for Accounts in Ethereum



There are separate tries for State, Transactions and Receipts

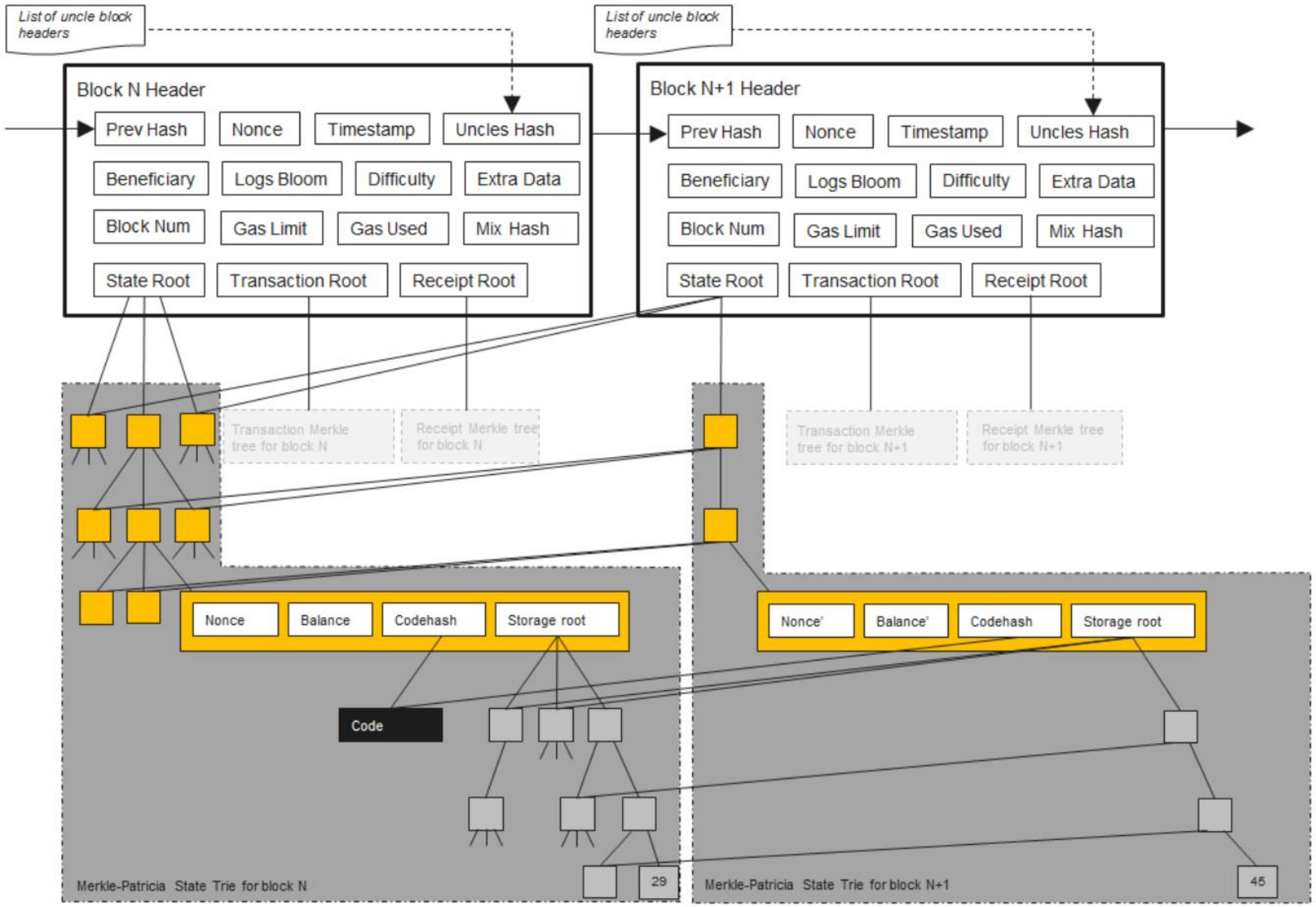
Transaction Updates State

A gets sent 2 ether so State must be updated.

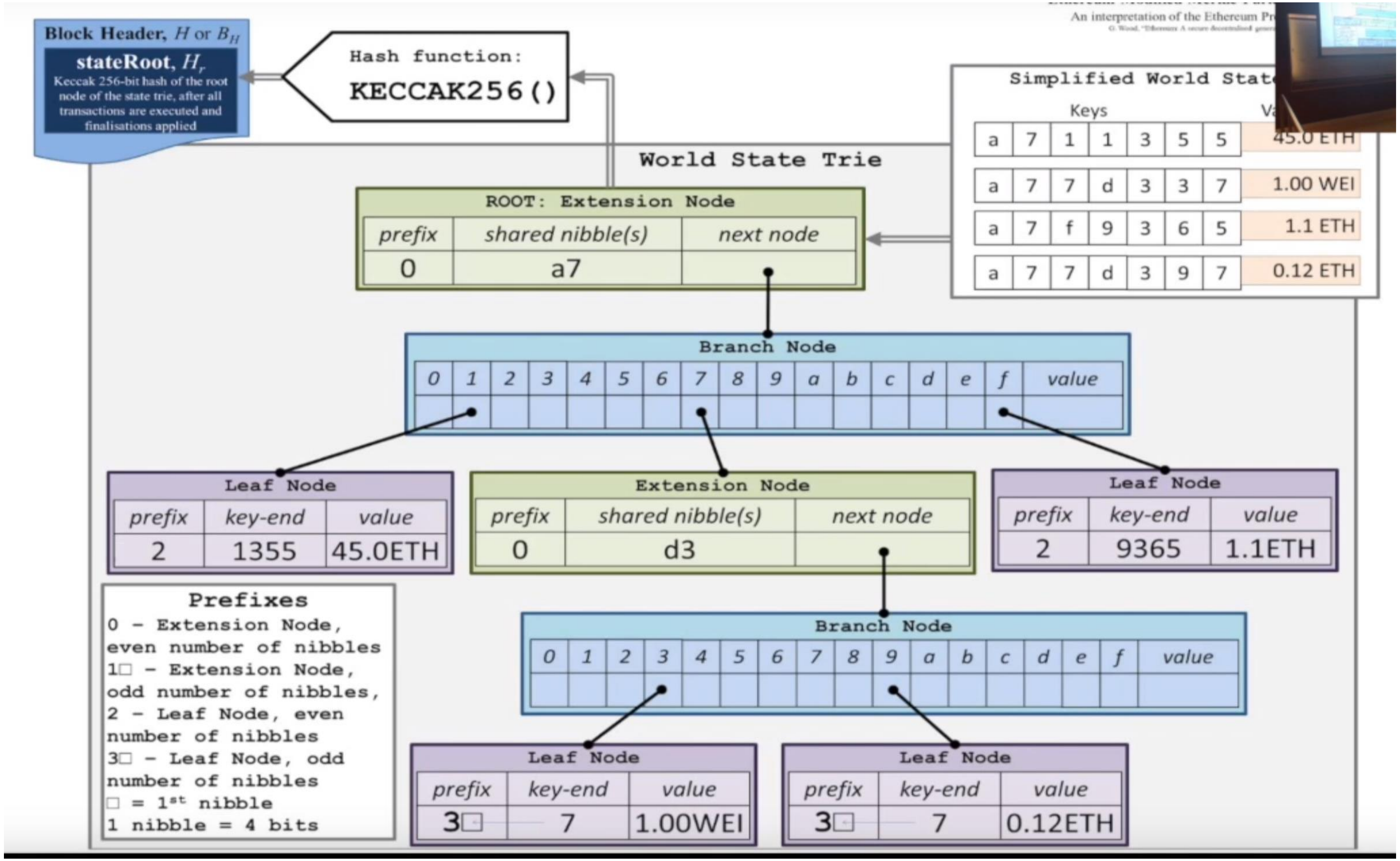
3 blue nodes must be added.

Red nodes can be deleted.

Data Structure for Accounts in Ethereum



Data Structure for Accounts in Ethereum



An interpretation of the Ethereum Pr

G. Wood, "Ethereum: A secure decentralised gener

Block Header, H or B_H

stateRoot, H_r
Keccak 256-bit hash of the root node of the state trie, after all transactions are executed and finalisations applied

Hash function:
KECCAK256()

Simplified World State

Keys							Va
a	7	1	1	3	5	5	45.0 ETH
a	7	7	d	3	3	7	1.00 WEI
a	7	f	9	3	6	5	1.1 ETH
a	7	7	d	3	9	7	0.12 ETH

World State Trie

ROOT: Extension Node

prefix	shared nibble(s)	next node
0	a7	●

Branch Node

0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	value
	●						●								●	

Leaf Node

prefix	key-end	value
2	1355	45.0ETH

Extension Node

prefix	shared nibble(s)	next node
0	d3	●

Leaf Node

prefix	key-end	value
2	9365	1.1ETH

Branch Node

0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	value
			●						●							

Leaf Node

prefix	key-end	value
3□	7	1.00WEI

Leaf Node

prefix	key-end	value
3□	7	0.12ETH

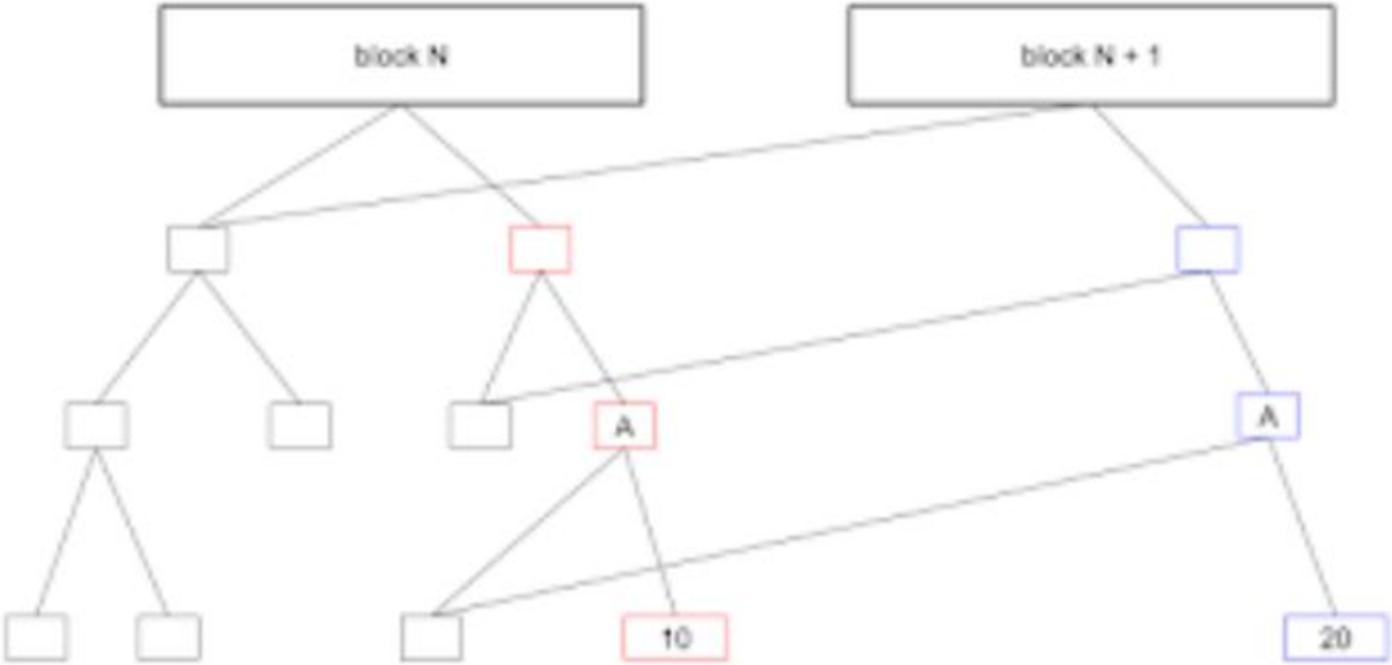
Prefixes

0 – Extension Node, even number of nibbles
1□ – Extension Node, odd number of nibbles,
2 – Leaf Node, even number of nibbles
3□ – Leaf Node, odd number of nibbles
□ = 1st nibble
1 nibble = 4 bits

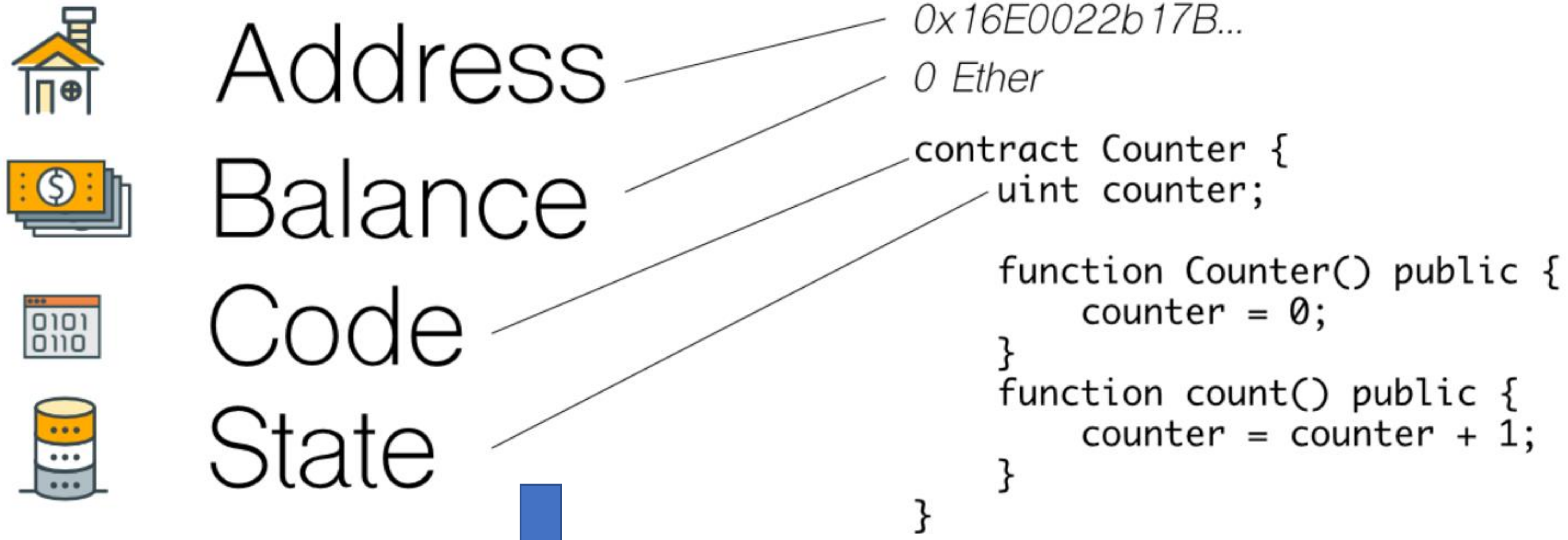
State Pruning https://medium.com/codechain/ethereums-state-trie-pruning-45ea73ed2c78

A gets sent 10
So 3 blue nodes must be added.
Red nodes could be deleted if we
are absolutely sure N+1 is added

Red nodes are deleted after 127
state changes !!!



Ethereum Smart Contract Structure



```
contract Counter {
    uint counter;

    function Counter() public {
        counter = 0;
    }
    function count() public {
        counter = counter + 1;
    }
}
```

events in the log can be monitored

Patricia Tries Exercises and Description

https://medium.com/cybermiles/diving-into-ethereums-world-state-c893102030ed

Ethereum Design Decisions

https://ethereumbuilders.gitbooks.io/guide/content/en/design_rationale.html