NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Ethereum

Construction of the "World Computer"

Dr Sourav SEN GUPTA
Lecturer, SCSE, NTU

ethereum
ETHERBROWSER
PEER-TO-PEER MESSAGING
GENERALIZED BLOCKCHAIN
PROGRAM ANYTHING

Vitalik Buterin explains Ethereum

Ethereum on YouTube | 29 Jul 2014

https://www.youtube.com/watch?v=TDGq4aeevgY

Generalized Blockchain

State Transition System

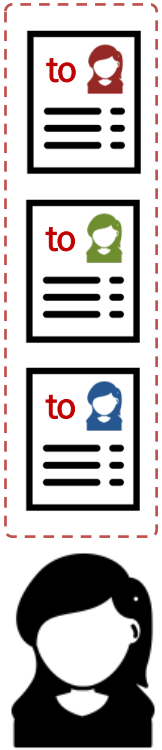
Recall : Bitcoin Transactions

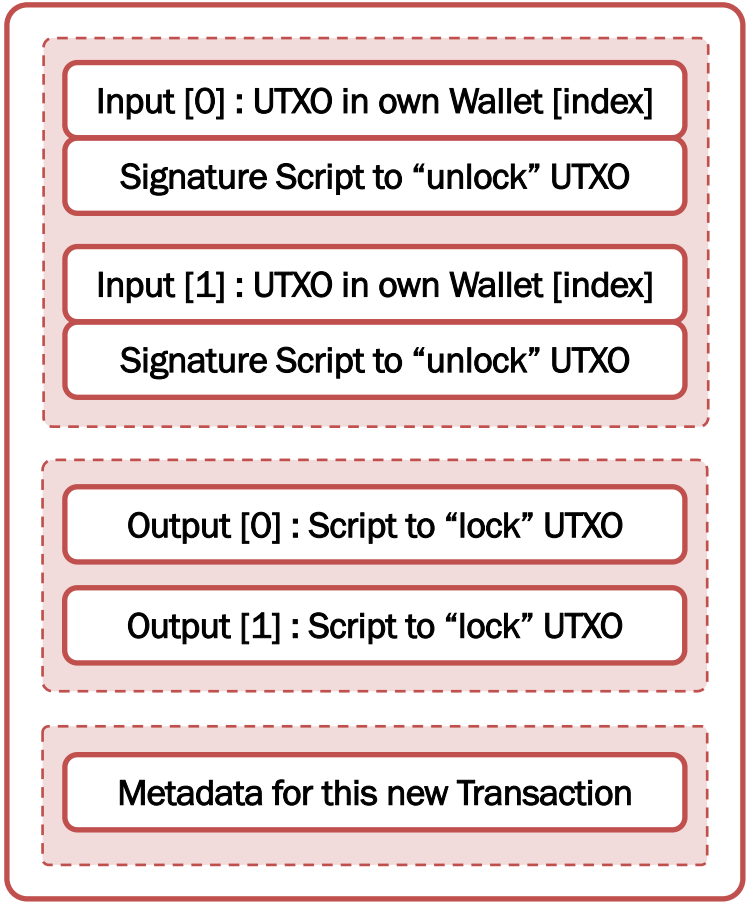
In a **Bitcoin Transaction**

- Input UTXOs are “spent”
- Output UTXOs are “created”
- Total Coins are “preserved”

In case of a **Bitcoin User**

- UTXOs constitute “balance”
- UTXOs owned by “private keys”
- UTXOs spent using “signatures”



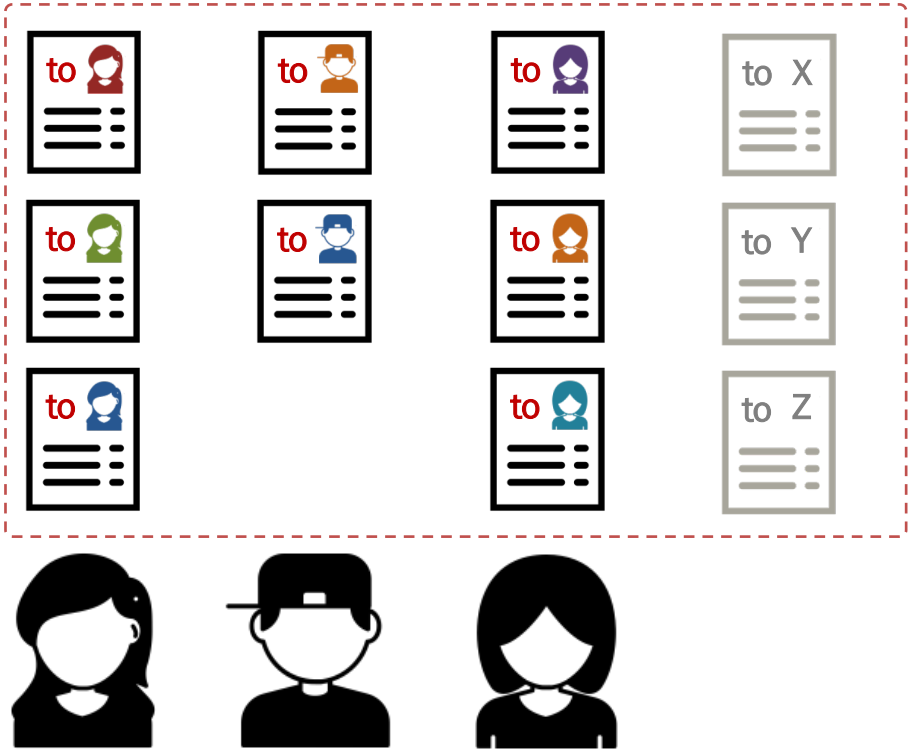


“State” of Bitcoin Blockchain

State of Bitcoin Blockchain consists of the “Record of Ownership” of all **unspent Coins**, that is, the UTXOs.

- UTXOs are **not** Transactions
- UTXOs are **unspent outputs**
- **Mechanism** to record Coins

There is no User Account in Bitcoin.



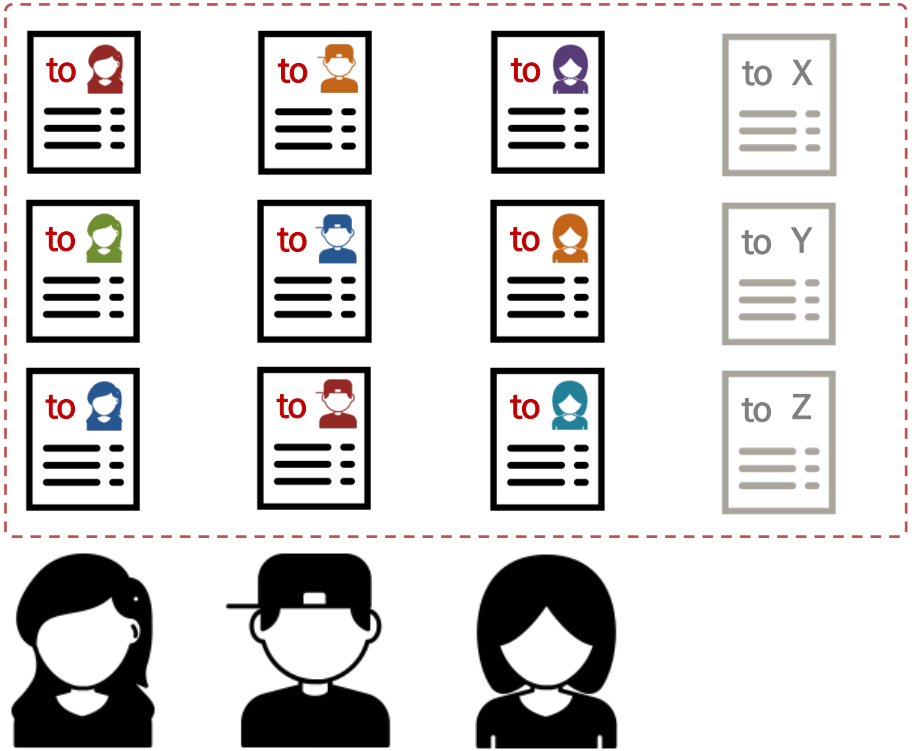
“State Transition” by Transactions

Bitcoin Transaction acts on a State of the Bitcoin Blockchain to alter it.

$\texttt{APPLY}(S_n, \texttt{Tx}) \rightarrow S_{n+1}$

(may also result in execution Error)

Example Transaction



UTXO-based State Transition

```
S_n   = { 2937…5d1c [0],   ce22…62f5 [0],   855f…7396 [1],   97b1…ec0d [1],
          4b93…dfd9 [0],   824e…cd8e [1],   824e…cd8e [2],   824e…cd8e [3]   }

Tx    = { input    : [ 2937…5d1c [0],   824e…cd8e [2] ],
          output   : [ 3e3f…f12c [0],   3e3f…f12c [1] ],
          scripts  : [ signature scripts for the input UTXOs ]  }

S_n+1 = { ce22…62f5 [0],   855f…7396 [1],   97b1…ec0d [1],   4b93…dfd9 [0],
          824e…cd8e [1],   824e…cd8e [3],   3e3f…f12c [0],   3e3f…f12c [1]   }
```

Verifying transaction accuracy and checking double-spending is efficient.

Checking for “balances” in specific Bitcoin “Accounts” is quite inefficient.

UTXO-based State Management [1]

Once a Block is received by any Bitcoin full node,

Check if the previous block referenced by this block exists and is valid.
Check that timestamp of this block is greater than that of previous block.
Check that the "proof of work" on this block is valid with verifiable nonce.

Let S_n be the "global state" of the blockchain after the previous block.

$S = S_n$; for all *Tx* in this Block : S = **APPLY**(S, *Tx*) ; return $S_{n+1} = S$;
if **APPLY** returns execution error in case of any *Tx* : return False and exit.

S_{n+1} will be the "updated global state" of the blockchain after this block.

[1] reading: "Ethereum Whitepaper" (https://ethereum.org/en/whitepaper/)

Account-based State Management

Account for every User of the Chain

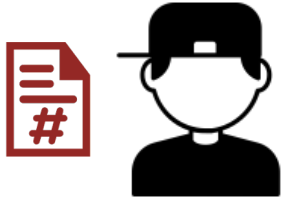
- Address of the Account
- Balance of the Account
- Number of Transactions
- Extra Information (if any)

Checking for “balances” in specific “Accounts” becomes quite efficient.

Verifying transaction accuracy and checking double-spend inefficient.

Address : Identifier of the Account
Balance : Amount of Coins owned
Nonce : Number of Transactions
Data : Extra information (if any)

Address : Identifier of the Account
Balance : Amount of Coins owned
Nonce : Number of Transactions
Data : Extra information (if any)



Address : Identifier of the Account
Balance : Amount of Coins owned
Nonce : Number of Transactions
Data : Extra information (if any)

“State” of Ethereum Blockchain

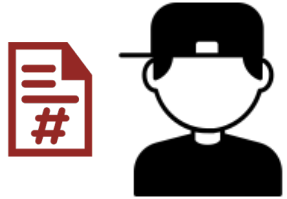
State of Ethereum Blockchain is made up of the “Accounts” of all **Users of the Chain** (and some more)

- Every User has an Account
- Every Account has Address
- Every Account has Balance

Mechanism : Standard Accounting

Address : 0x40aa48…996e02
Balance : 183 Ether (in Wei)
Nonce : 34
Data : None

Address : 0xe7494e…14aC3d
Balance : 24 Ether (in Wei)
Nonce : 8
Data : None



Address : 0xc42bfA…4e89a4
Balance : 0 Ether (in Wei)
Nonce : 0
Data : None

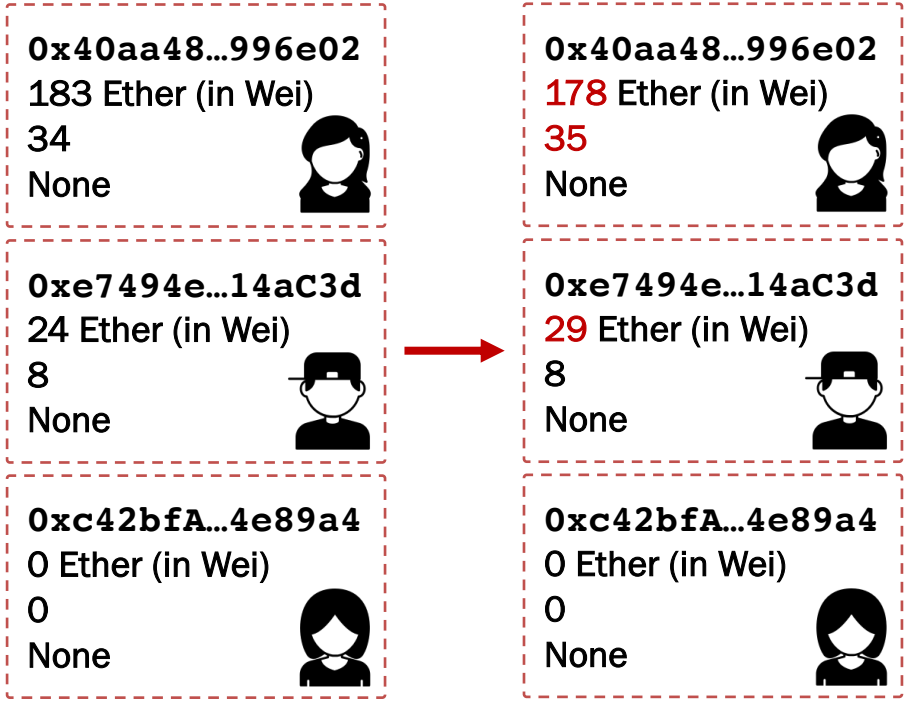
“State Transition” by Transactions

Ethereum Transaction acts on the State of the Ethereum Blockchain.

$APPLY(S_n, Tx) \rightarrow S_{n+1}$

(may also result in execution Error)

Example Transaction



Ethereum Transaction

Transaction Hash:	0xfdfe0bb1f05c441e34cc65118390463d6921cfce65b22a347f7b4160fa414fe3	Transaction Identifier
Status:	Success	Status of Transaction
Block:	10926965 1597 Block Confirmations	Block where Recorded
Timestamp:	5 hrs 50 mins ago (Sep-24-2020 06:34:26 PM +UTC) | Confirmed within 38 secs	Time and Confirmation
From:	0x40aa48213402969228e75752edb902a7e3996e02	Sender's Address
To:	0xe7494e7724d120bec42bfa4e89a47858b214ac3d	Recipient's Address
Value:	10.97188441 Ether ($3,827.32)	Total Ether Transacted
Transaction Fee:	0.001638 Ether ($0.57)	Transaction Fee to Miner

The base currency for Ethereum is **Ether**, used both for Transactions and Fee.

ref : https://etherscan.io/tx/0xfdfe0bb1f05c441e34cc65118390463d6921cfce65b22a347f7b4160fa414fe3

Ethereum Blockchain

Accounts and Transactions

Accounts in Ethereum [1]

State of Ethereum is made up of Accounts, each with a 20-byte address.

- Externally Owned Accounts : Controlled by Private Keys : "Human Users"
- Contract Accounts : Controlled by Contract Code : "Autonomous Agents"

Account State : `{ nonce, balance, storage, code }`

- Externally Owned Accounts primarily operate with `{ nonce, balance }`
- Contract Accounts primarily operate with contract `{ storage, code }`

Ethereum presents a complex ecosystem of Users and Autonomous Agents.

[1] reading: "Ethereum Whitepaper" (https://ethereum.org/en/whitepaper/)

Externally Owned Accounts

State : { nonce, balance,
storage, code }

Action : Send Transactions

- to transfer Ether from balance
- to trigger some Contract Code
- to create some new Contract

Address : 0x40aa48…996e02

Nonce : 34
Balance : 183 Ether (in Wei)
Storage : None
Code : None

Transactions sent by an Externally Owned Account updates **Ethereum State**

- by **changing balance** of self and other EOA (plus transaction fee to miners)
- by **creating a new contract account** or by triggering **execution of a contract**

ref : https://etherscan.io/address/0x40aa48213402969228e75752edb902a7e3996e02

Contract Accounts

State : { nonce, balance, storage, code }

Action : Send Messages

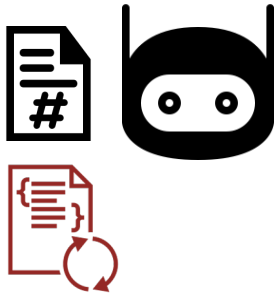
- to transfer Ether to some EOA
- to trigger some Contract Code
- to create some new Contract

Action : Execute Contract Code

- on receiving “calls” from others
- to update storage and balance
- to send Messages as required

Address : 0x2a0c0D…50c208

Nonce : 257
Balance : 38223 Ether (in Wei)
Storage : storageRoot (trie)
Code : codeHash (contract)



```
pragma solidity ^0.4.16;
mapping (address =>
          mapping (address => uint256))
          public tokens;
contract Token { … }
contract Exchange { … }
```

ref : https://etherscan.io/address/0x2a0c0dbecc7e4d658f48e01e3fa353f44050c208

Transaction

Digitally signed instruction constructed and sent by Externally Owned Account.

`nonce`	:	Number of transactions sent by the Sender
`to`	:	160-bit address of the Recipient's Account
`value`	:	Number of Wei to transfer to the Recipient
`v,r,s`	:	Values for Sender's Signature and Identity
`data/init`	:	Byte array for Input Data or Code Creation
`gasPrice`	:	Number of Wei to be paid for the Execution
`gasLimit`	:	Maximum amount of gas for the Execution

Transactions accomplish either Ether Transfer, Message or Contract Creation.

Types of Transactions

Main “payload” for an Ethereum Transaction : `{ value, data/init }`

Transactions can be of two types : Message Calls and Contract Creations

- `{value, data = Null}` : Ether Transfer/Payment to the Recipient
- `{value = Null, data}` : Invocation of Contract Code at Recipient
- `{value, data}` : Ether Transfer/Payment as well as Code Invocation
- `{value = Null, data = Null}` : Valid but useless (waste of gas)
- `{value, init = Code}` : Contract Creation if sent to address `0x00`

`value` and `data/init` are interpreted in accordance with the `to` address

Message Call : Transfer/Payment

```
{ value = X Wei, data = Null }
```

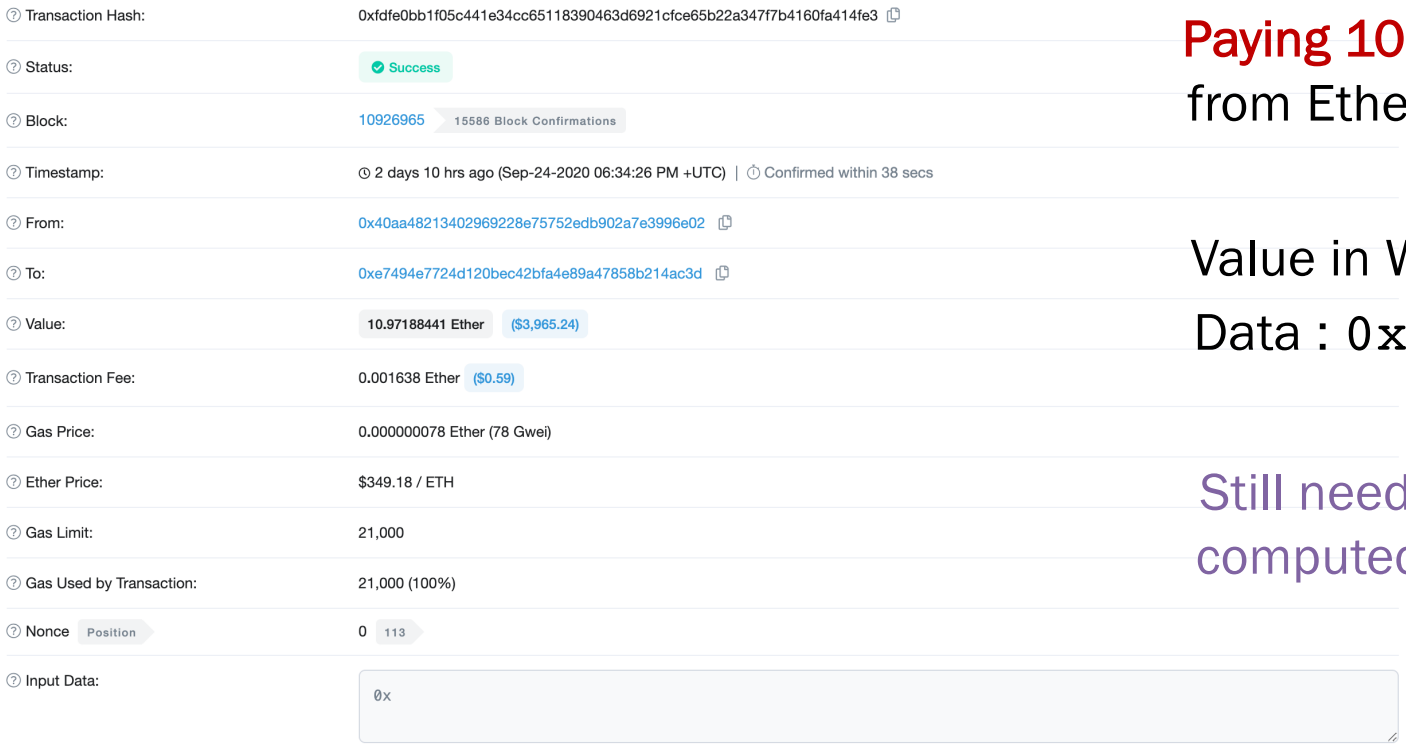
If recipient (`to` address) is an EOA

- Record **state change** : Add X Wei to Recipient balance, and deduct from Sender
- EOA not found : Update local state with Recipient address and initialize to X Wei

If recipient (`to` address) is a Contract

- Call **fallback function** in Recipient contract and execute if the code is “payable”
- No fallback function : Add X Wei to Contract balance, and deduct from Sender

Example : Transfer/Payment



Transaction Hash:	0xfdfe0bb1f05c441e34cc65118390463d6921cfce65b22a347f7b4160fa414fe3
Status:	Success
Block:	10926965 15586 Block Confirmations
Timestamp:	2 days 10 hrs ago (Sep-24-2020 06:34:26 PM +UTC) | Confirmed within 38 secs
From:	0x40aa48213402969228e75752edb902a7e3996e02
To:	0xe7494e7724d120bec42bfa4e89a47858b214ac3d
Value:	10.97188441 Ether ($3,965.24)
Transaction Fee:	0.001638 Ether ($0.59)
Gas Price:	0.000000078 Ether (78 Gwei)
Ether Price:	$349.18 / ETH
Gas Limit:	21,000
Gas Used by Transaction:	21,000 (100%)
Nonce Position	0 113
Input Data:	0x

Paying 10.97188441 Ether
from Ethereum EOA to EOA

Value in Wei (10^{18} Wei = 1 Ether)

Data : `0x` (denotes pure Payment)

Still needs “execution fee” computed in terms of Gas.

ref : https://etherscan.io/tx/0xfdfe0bb1f05c441e34cc65118390463d6921cfce65b22a347f7b4160fa414fe3

Message Call : Code Invocation

`{ value = Null, data = Contract ABI# }`

If recipient (`to` address) is a Contract

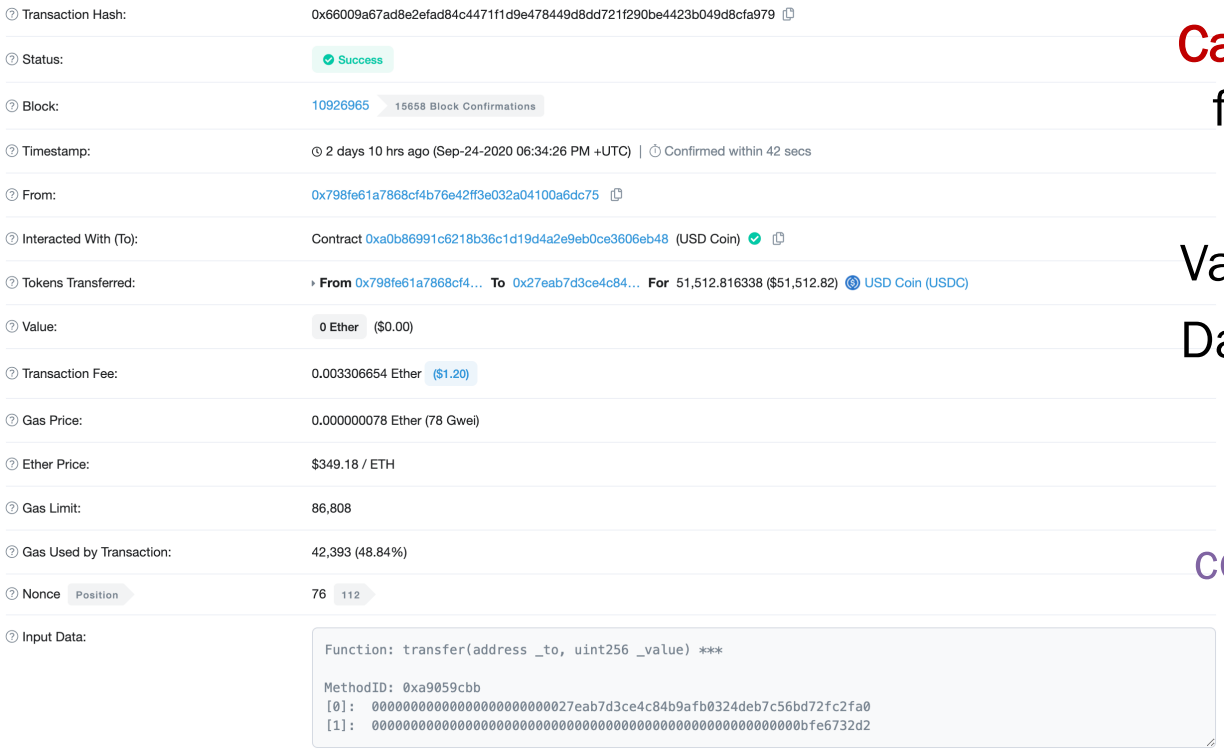
- Call the method specified by **function selector** with specified **input arguments**
- Contract ABI Specs : https://solidity.readthedocs.io/en/develop/abi-spec.html

If recipient (`to` address) is an EOA

- Interpretation of data is up to the Wallet used in the EOA (most wallets ignore).
- Completely ignored by Ethereum protocol and is not subject to consensus rules.

: Contract Application Binary Interface (ABI) specified for invoking Ethereum Contracts or methods therein.

Example : Code Invocation



Transaction Hash:	0x66009a67ad8e2efad84c4471f1d9e478449d8dd721f290be4423b049d8cfa979
Status:	Success
Block:	10926965 15658 Block Confirmations
Timestamp:	2 days 10 hrs ago (Sep-24-2020 06:34:26 PM +UTC) | Confirmed within 42 secs
From:	0x798fe61a7868cf4b76e42ff3e032a04100a6dc75
Interacted With (To):	Contract 0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48 (USD Coin)
Tokens Transferred:	From 0x798fe61a7868cf4… To 0x27eab7d3ce4c84… For 51,512.816338 ($51,512.82) USD Coin (USDC)
Value:	0 Ether ($0.00)
Transaction Fee:	0.003306654 Ether ($1.20)
Gas Price:	0.000000078 Ether (78 Gwei)
Ether Price:	$349.18 / ETH
Gas Limit:	86,808
Gas Used by Transaction:	42,393 (48.84%)
Nonce Position	76 112
Input Data:	Function: transfer(address _to, uint256 _value) *** MethodID: 0xa9059cbb [0]: 000000000000000000000000027eab7d3ce4c84b9afb0324deb7c56bd72fc2fa0 [1]: 0000000000000000000000000000000000000000000000000000000000bfe6732d2

Calling function transfer(…)
from Contract “USD Coin”

Value : 0 (denotes code invocation)

Data : Method, Parameters

Must pay “execution fee”
computed in terms of Gas.

ref : https://etherscan.io/tx/0x66009a67ad8e2efad84c4471f1d9e478449d8dd721f290be4423b049d8cfa979

Message Call : Code Invocation + Payment

```
{ value = X Wei, data = Contract ABI# }
```

If recipient (`to` address) is a Contract

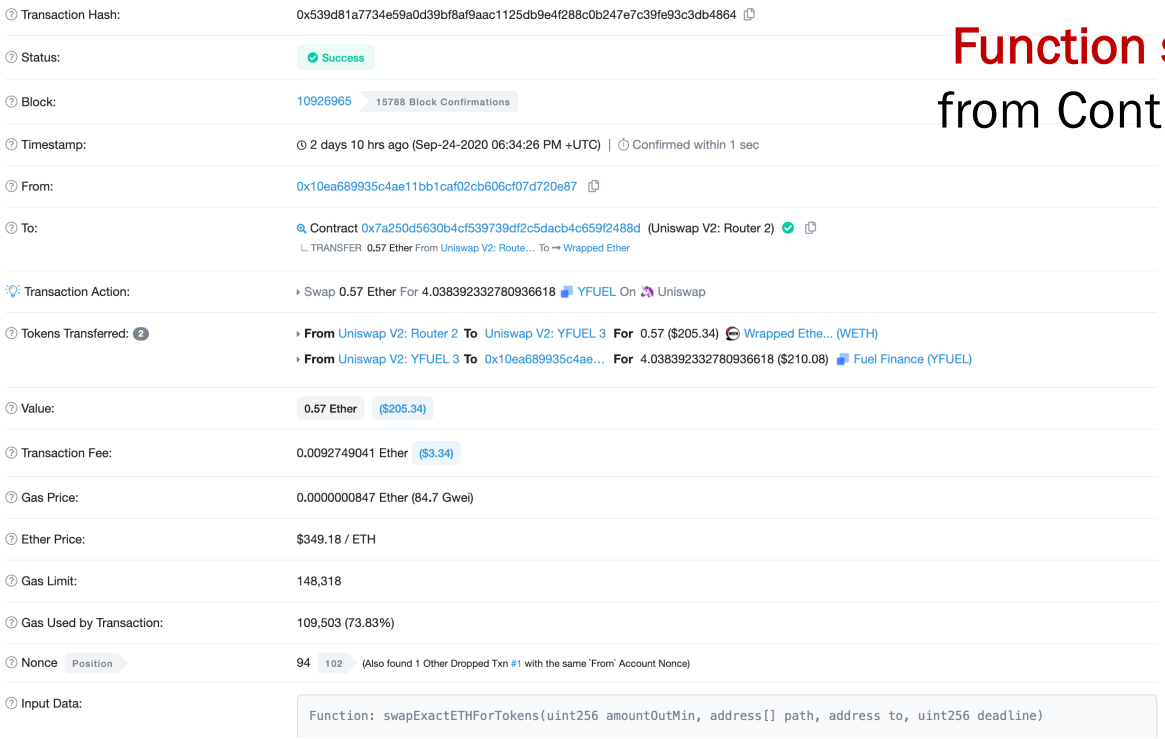
- Call the method specified by **function selector** with specified **input arguments**
- Utilizes the input value as per the function or adds it to the Contract's balance

If recipient (`to` address) is an EOA

- Record **state change** : Add X Wei to Recipient balance, and deduct from Sender
- Interpretation of data is up to the Wallet used in the EOA (most wallets ignore).

: Contract Application Binary Interface (ABI) specified for invoking Ethereum Contracts or methods therein.

Example : Code Invocation + Payment



Transaction Hash:	0x539d81a7734e59a0d39bf8af9aac1125db9e4f288c0b247e7c39fe93c3db4864
Status:	Success
Block:	10926965 15788 Block Confirmations
Timestamp:	2 days 10 hrs ago (Sep-24-2020 06:34:26 PM +UTC) | Confirmed within 1 sec
From:	0x10ea689935c4ae11bb1caf02cb606cf07d720e87
To:	Contract 0x7a250d5630b4cf539739df2c5dacb4c659f2488d (Uniswap V2: Router 2) TRANSFER 0.57 Ether From Uniswap V2: Route… To Wrapped Ether
Transaction Action:	Swap 0.57 Ether For 4.038392332780936618 YFUEL On Uniswap
Tokens Transferred: 2	From Uniswap V2: Router 2 To Uniswap V2: YFUEL 3 For 0.57 ($205.34) Wrapped Ethe... (WETH) From Uniswap V2: YFUEL 3 To 0x10ea689935c4ae… For 4.038392332780936618 ($210.08) Fuel Finance (YFUEL)
Value:	0.57 Ether ($205.34)
Transaction Fee:	0.0092749041 Ether ($3.34)
Gas Price:	0.0000000847 Ether (84.7 Gwei)
Ether Price:	$349.18 / ETH
Gas Limit:	148,318
Gas Used by Transaction:	109,503 (73.83%)
Nonce Position	94 102 (Also found 1 Other Dropped Txn #1 with the same `From` Account Nonce)
Input Data:	Function: swapExactETHForTokens(uint256 amountOutMin, address[] path, address to, uint256 deadline)

Function swapExactETHForTokens(…)
from Contract “Uniswap V2: Router 2”

Value in Wei (10^{18} Wei = 1 Ether)

Data : Method, Parameters

Must pay “execution fee” computed in terms of Gas.

ref : https://etherscan.io/tx/0x539d81a7734e59a0d39bf8af9aac1125db9e4f288c0b247e7c39fe93c3db4864

Special Transaction : Contract Creation

`{ value = X Wei, init = Compiled Bytecode }`

If recipient (`to` address) is `0x0`

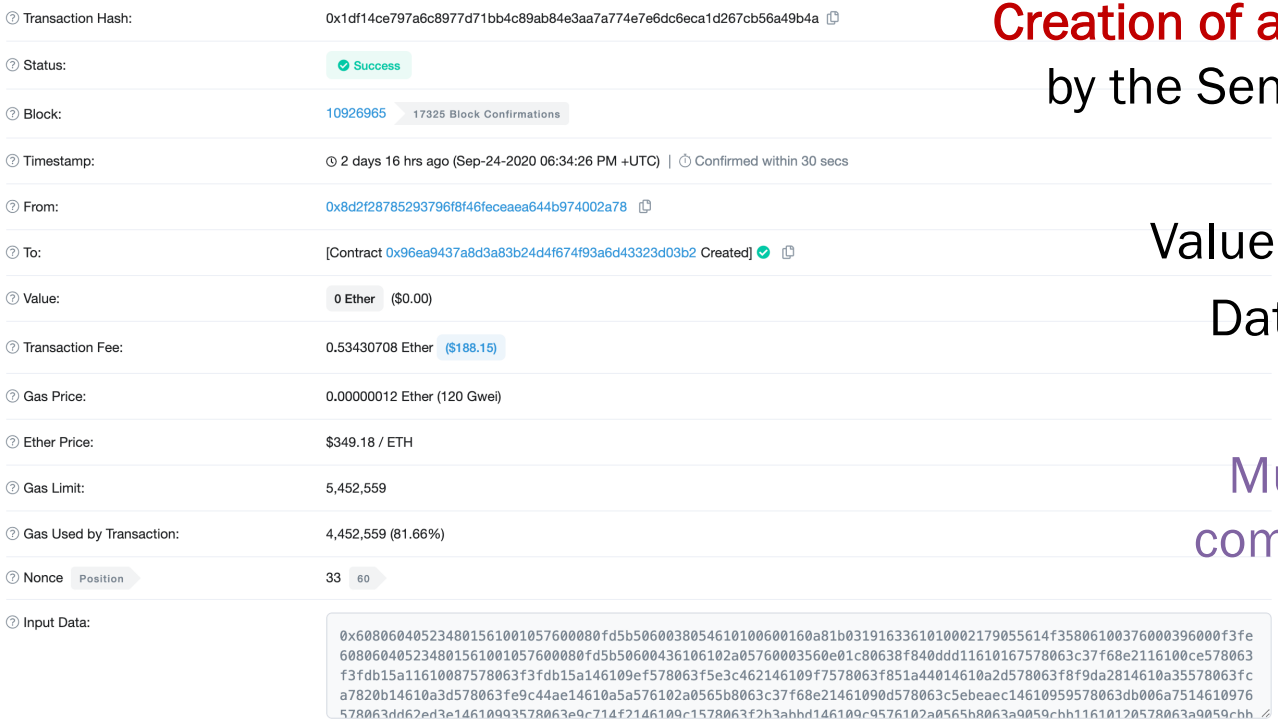
- Create **new contract account** with the **compiled bytecode** sent as `init/data`
- If `value` is non-zero, the **initial balance** for the contract account is set at X Wei

If there is no `init/data` sent

- `value` will be **lost forever**, as Ether sent to `0x0` can not be recovered or spent.
- Such cases of dumping Ether should rather use the designated **burn address**.#

: Designated burn address for Ethereum : https://etherscan.io/address/0x000000000000000000000000000000000000dead

Example : Contract Creation



Transaction Hash:	0x1df14ce797a6c8977d71bb4c89ab84e3aa7a774e7e6dc6eca1d267cb56a49b4a
Status:	Success
Block:	10926965 17325 Block Confirmations
Timestamp:	2 days 16 hrs ago (Sep-24-2020 06:34:26 PM +UTC) | Confirmed within 30 secs
From:	0x8d2f28785293796f8f46feceaea644b974002a78
To:	[Contract 0x96ea9437a8d3a83b24d4f674f93a6d43323d03b2 Created]
Value:	0 Ether ($0.00)
Transaction Fee:	0.53430708 Ether ($188.15)
Gas Price:	0.00000012 Ether (120 Gwei)
Ether Price:	$349.18 / ETH
Gas Limit:	5,452,559
Gas Used by Transaction:	4,452,559 (81.66%)
Nonce Position	33 60
Input Data:	0x6080604052348015610010576000080fd5b5060038054610100600160a81b0319163361010002179055614f3580610037600039600f3fe6080604052348015610010576000080fd5b506004361061102a05760003560e01c80638f840ddd11610167578063c37f68e2116100ce578063f3fdb15a11610087578063f3fdb15a146109ef578063f5e3c462146109f7578063851a44014610a2d578063f8f9da2814610a35578063fca7820b14610a3d578063fe9c44ae14610a5a576102a0565b8063c37f68e21461090d578063c5ebeaec14610959578063db006a7514610976578063dd62ed3e14610993578063e9c714f2146109c1578063f2b3abbd146109d9578063957610a0565b8063a9059cbb1461120578063a9059cbb

Creation of a new Contract Account by the Sender (no initial balance)

Value : 0 (provides no initial balance)

Data : Compiled Bytecode

Must pay “execution fee” computed in terms of Gas.

ref : https://etherscan.io/tx/0x1df14ce797a6c8977d71bb4c89ab84e3aa7a774e7e6dc6eca1d267cb56a49b4a

Messages from Contracts

Contract Code can also send **Messages** to EOA or other Contract accounts.

Contract messages are triggered by some Parent Transaction sent by EOAs.

Internal Transactions/Messages

- create : Create a new Contract Account by sending a Message to address `0x0`
- call : Transfer/Pay a specified amount of Ether to a specified EOA or Contract
- call : Invoke specified function within own/another specified Contract account

Internal transactions are not serialized; they are always triggered externally.

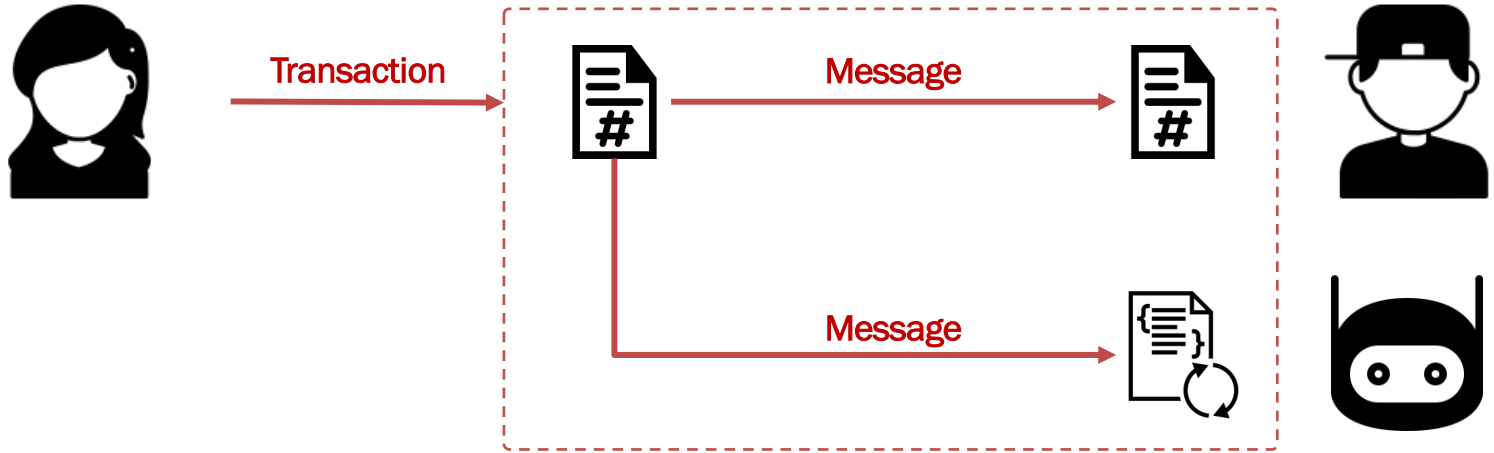
Example : Internal Transactions

Block	Age	Parent Txn Hash	Type	From	To	Value
10926965	2 days 17 hrs ago	0x7bc436e35a70da19…	create	0xa4d82f73949cb2ed1…	0x7d08ea46fbca6eb44…	0 Ether
		0x0a7187b03769f394d…	create	0xa4d82f73949cb2ed1…	0x0f1873de7d94fb4c4…	0 Ether
Contract Creation		0x23166446f1068aa37…	create	0xa4d82f73949cb2ed1…	0x9cecbf554c0495ca3…	0 Ether
Function Invocation		0x539d81a7734e59a0…	call	Uniswap V2: Router 2	Wrapped Ether	0.57 Ether
Ether Transfer		0xd970c0b39660621a…	call	0x056cbc3d1926b50b…	0x05635c9c354e98cc5…	1.87228948514411 Ether
		0xe003a910dcf99d095…	call	0x73282a63f0e3d7e96…	0x98945bc69a554f8b1…	0.005 Ether
		0xe003a910dcf99d095…	call	0x73282a63f0e3d7e96…	0xf31749a95dc62ce3f9…	1.995 Ether
		0x1062832447fc44faf6…	call	0xec0cec1f0ec5dff3d6…	0x2a549b4af9ec39b03…	0.86752841 Ether
		0xc7e1dc80e37f42eac…	call	Wrapped Ether	0x860bd2dba9cd475a…	26.917453132580439 Ether
		0xc7e1dc80e37f42eac…	call	0x860bd2dba9cd475a…	Wrapped Ether	26.502119100081978 Ether
		0xa7492296c7f598021…	call	Kraken 5	Kraken 4	0.04678539365 Ether

ref : https://etherscan.io/txsInternal?block=10926965

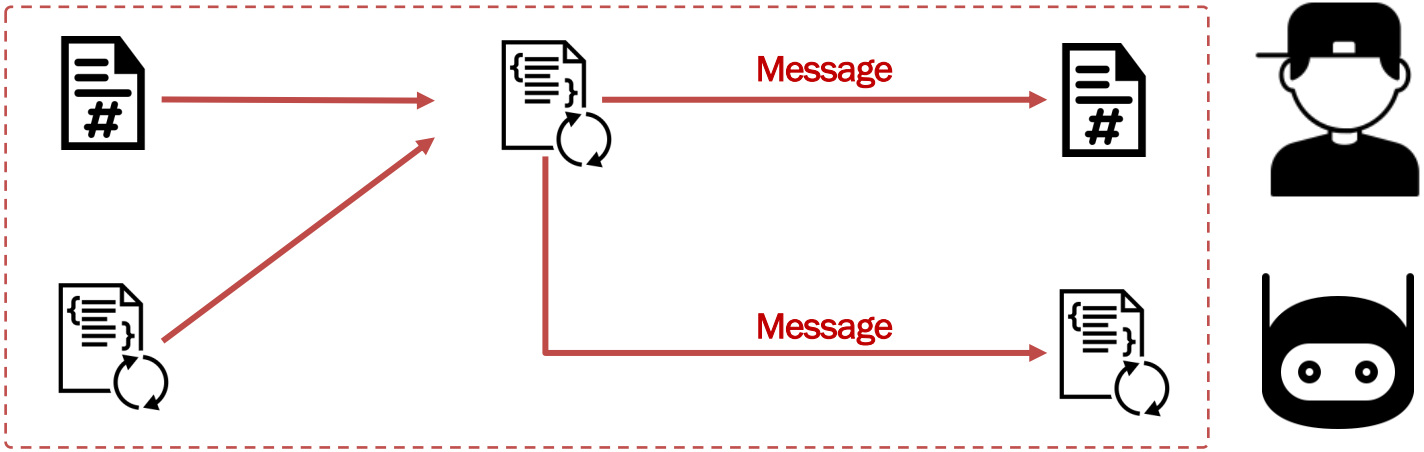
Interaction Patterns (EOA)

Accounts communicate via Messages, triggered by Transactions from EOAs.



Interaction Patterns (CA)

Contracts initiate Messages, as a chain-reaction to Transactions from EOAs.



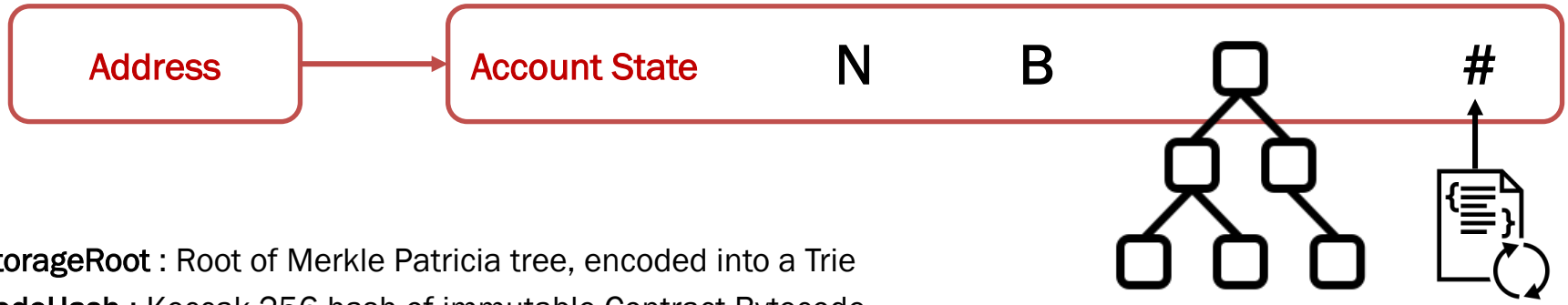
Ethereum Blockchain

State-Transition Machine

Account State in Ethereum

Account State is a generic representation of Ethereum accounts (EOA or CA).

Address : State = `{ nonce, balance, storageRoot, codeHash }`



storageRoot : Root of Merkle Patricia tree, encoded into a Trie

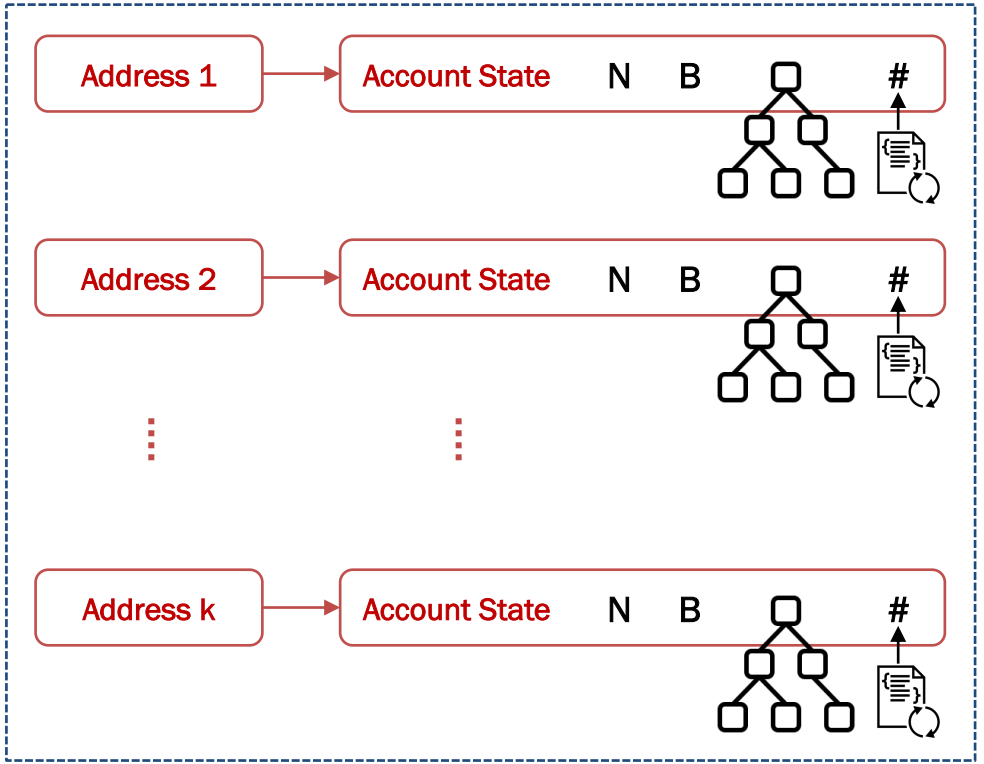
codeHash : Keccak-256 hash of immutable Contract Bytecode

Ethereum “World State”

The **world state** of Ethereum is a **map** between Addresses and corresponding Account States.

- Addresses are 160-bit identifiers
- States **not stored** on Blockchain
- Mapping maintained in a trie DB
- Root Hash identifies “world state”

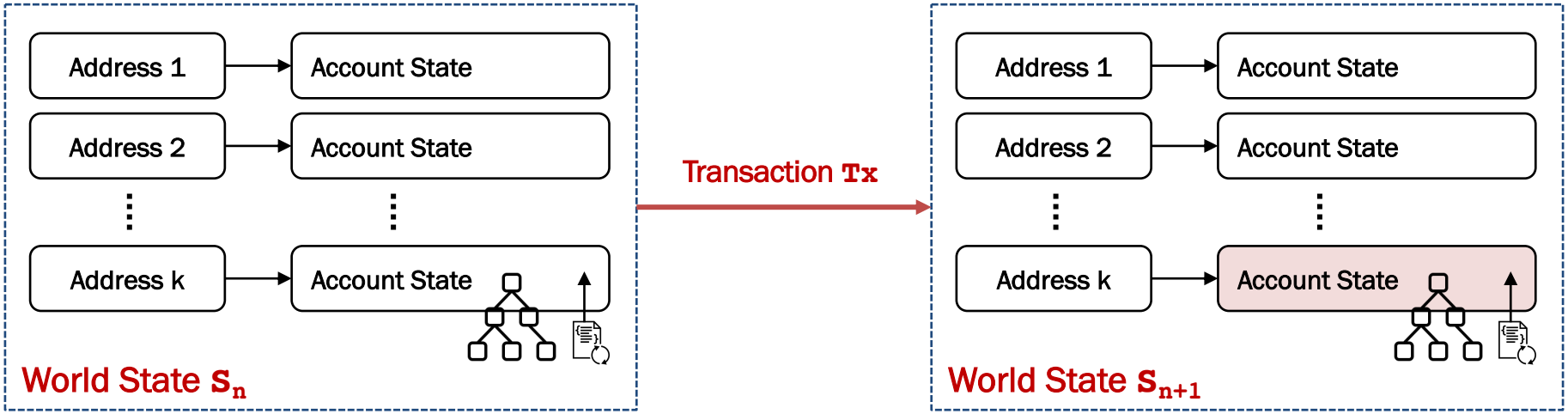
State : Collection of (Code + Data)



State Transition

Transactions act as **atomic digitally-signed instructions** on the **world state.**

Transactions and resultant messages update one or more Account States.



State-Transition Chain

Ethereum state machine was initiated with a "**genesis state**" in the beginning.

At each **epoch**, previous state updates to the next state through **Transactions**.