

Ethereum



for noobs

How to process a new block on Ethereum fullnode

Hội anh em blockchain "thiện lành"

Profile



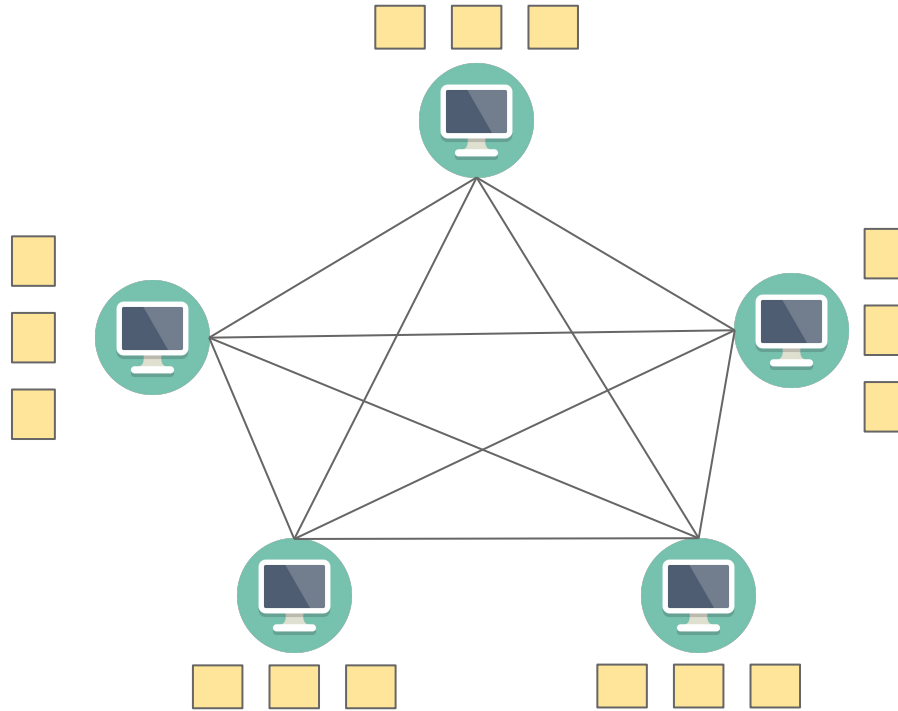
Cong Le

- Education
 - Graduated from Bach Khoa University with major Computer Science
 - Studying Master program at Bach Khoa University
- Working Experiences
 - +4 years in Software Development
 - +3 years in Blockchain Technology
 - Senior Dev in Infinito Wallet project

Contents

- Process a new block
- Switch to a new chain
- Query data via RPCs

Process a new block



A list of fullnodes on Ethereum network



Feature Tip: Ethereum charts & statistics at your fingertip. [Learn More!](#)



Ethereum Blockchain Explorer

Quick links: [ERC-20 Tokens](#) [ERC-721 Tokens](#)

All Filters ▾

Search by Address / Txn Hash / Block / Token / Ens

Search



ETHER PRICE

\$266.08 @ 0.02908 BTC (-1.67%)



LATEST BLOCK

7986471 (13.2s)

TRANSACTIONS

476.82 M (10.6 TPS)

ETHEREUM TRANSACTION HISTORY IN 14 DAYS



MARKET CAP

\$28.351 Billion



DIFFICULTY

2,024.50 TH

HASH RATE

163,187.95 GH/s

Latest Blocks

Bk	7986471 40 secs ago	Miner Spark Pool 91 txns in 10 secs	2.05807 Eth
Bk	7986470 50 secs ago	Miner Ethermine 136 txns in 19 secs	2.08879 Eth
Bk	7986469 1 mins 10 secs ago	Miner Ethermine 213 txns in 15 secs	2.16179 Eth

Transactions

Tx	0x29586b2aa5... 40 secs ago	From 0xe4d05f54eff833ab... To 0xb1298e180afd7e3...	0.00052 Eth
Tx	0xbd804e8aa5... 40 secs ago	From 0xd9a7f6484804bc6... To 0xb1298e180afd7e3...	0.00052 Eth
Tx	0x3ea0707a2b... 40 secs ago	From 0x2fadd4fc6c442f78... To 0x3161a3a2469469...	0 Eth

Block #7986472

Overview

Comments

Block Height:	7986472 < >
Timestamp:	43 secs ago (Jun-19-2019 04:04:19 AM +UTC)
Transactions:	140 transactions and 2 contract internal transactions in this block
Mined by:	0x5a0b54d5dc17e0aadc383d2db43b0a0d3e029c4c (Spark Pool) in 6 secs
Block Reward:	2.060072629887170892 Ether (2 + 0.060072629887170892)
Uncles Reward:	0
Difficulty:	2,015,925,566,716,380
Total Difficulty:	10,663,194,973,875,573,686,795
Size:	19,891 bytes
Gas Used:	7,995,494 (99.94%)
Gas Limit:	8,000,029

Transactions

For Block [7986471](#)

A total of 91 transactions found
















First

<

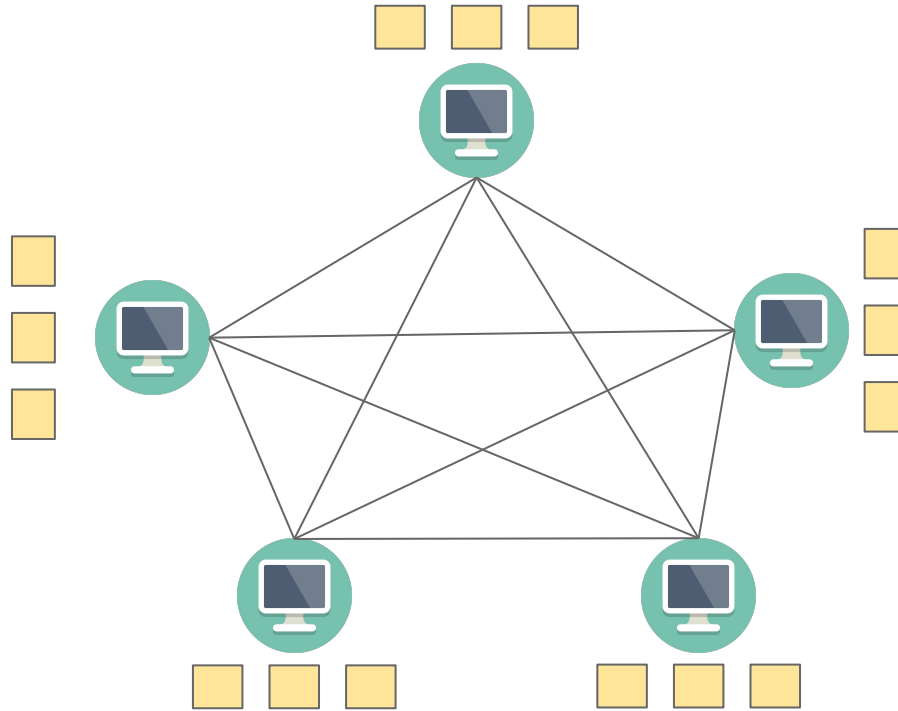
Page 1 of 2

>

Last

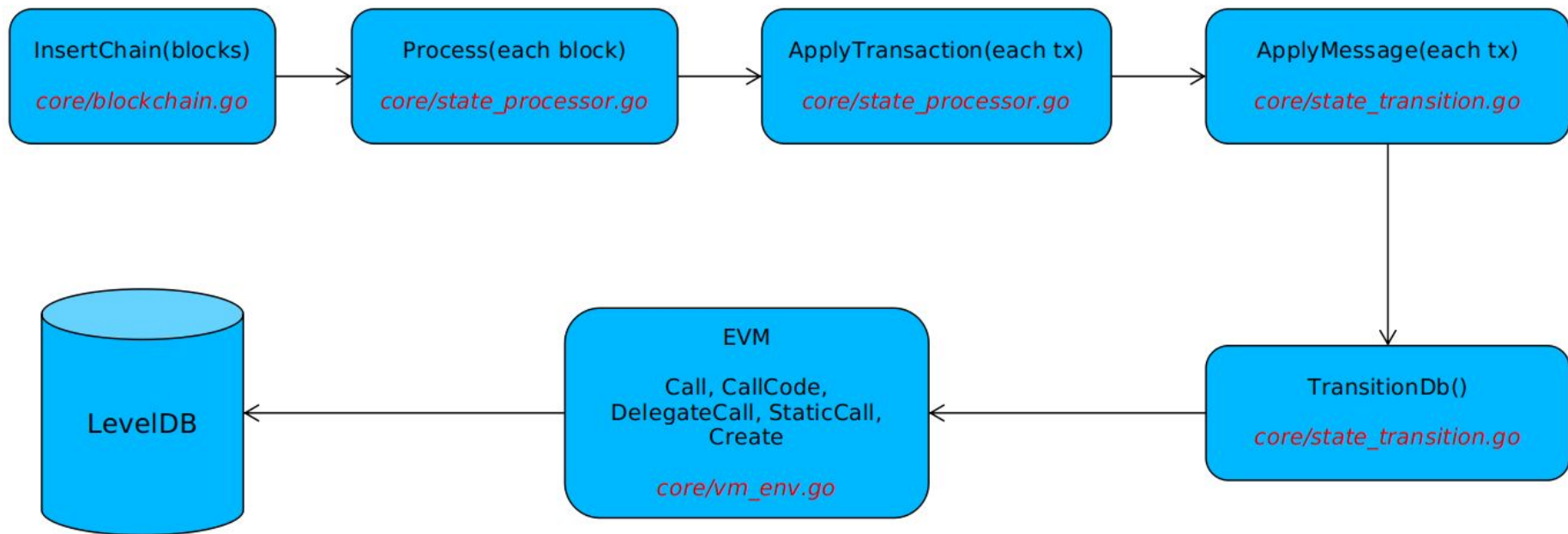
Txn Hash	Block	Age	From	To	Value	[Txn Fee]
0x29586b2aa585db...	7986471	1 min ago	0xe4d05f54eff833ab...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084
0xbd804e8aa53735...	7986471	1 min ago	0xd9a7f6484804bc6...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084
0x3ea0707a2b8436...	7986471	1 min ago	0x2fadd4fc6c442f78...	  0x3161a3a2469469...	0 Ether	0.00014692
0xf531c9ccf068092...	7986471	1 min ago	0x2fadd4fc6c442f78...	  0x3161a3a2469469...	0 Ether	0.00014718
0x6ee606a8ad5e35f...	7986471	1 min ago	0x864af554242ff204...	 0xc50489a925e9c8...	0 Ether	0.00013726
0x6dda6c254434de...	7986471	1 min ago	0xa3bd36054f577e2...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084
0x417843f5838b67f...	7986471	1 min ago	0x6a9a314aac0325...	  0x498d99de4268ce...	0 Ether	0.00063562
0x3ca62410aa8777...	7986471	1 min ago	0x89c8b42bbb181d...	  0x498d99de4268ce...	0 Ether	0.00069562
0x21d977667b34d3...	7986471	1 min ago	0xdaf07ec839e878b...	  0x498d99de4268ce...	0 Ether	0.00063562
0xf0e140f342e5b9fd...	7986471	1 min ago	0xdcecc323bb78f97...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084

Process a new block

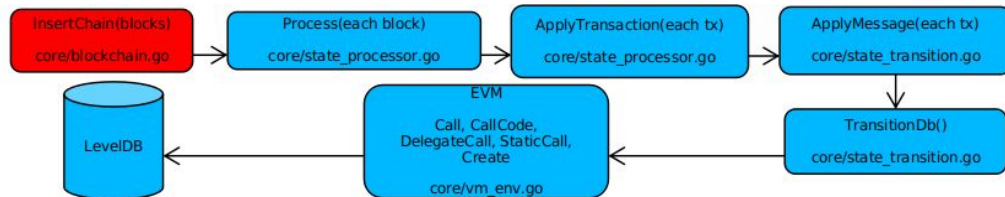


A list of fullnodes on Ethereum network

Process a new block



Process a new block

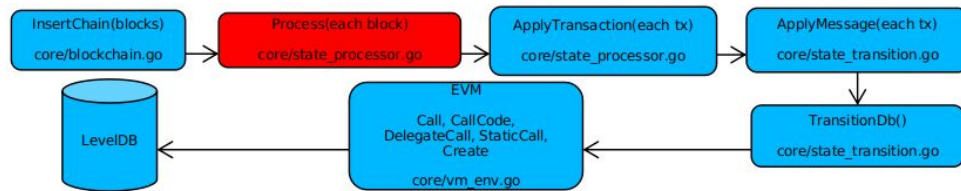


```
func (bc *BlockChain) insertChain(chain types.Blocks, verifySeals bool) (int, []interface{}, []*types.Log, error) {
    // Something here
    ...
    // No validation errors for the first block (or chain prefix skipped)
    for ; block != nil && err == nil; block, err = it.next() {
        // Something here
        ...
        receipts, logs, usedGas, err := bc.processor.Process(block, state, bc.vmConfig)
        t1 := time.Now()
        if err != nil {
            bc.reportBlock(block, receipts, err)
            return it.index, events, coalescedLogs, err
        }
        // Validate the state using the default validator
        if err := bc.Validator().ValidateState(block, parent, state, receipts, usedGas); err != nil {
            bc.reportBlock(block, receipts, err)
            return it.index, events, coalescedLogs, err
        }
        t2 := time.Now()
        proctime := time.Since(start)

        // Write the block to the chain and get the status.
        status, err := bc.WriteBlockWithState(block, receipts, state)
        t3 := time.Now()
        if err != nil {
            return it.index, events, coalescedLogs, err
        }
    }
}
```

core/blockchain.go

Process a new block

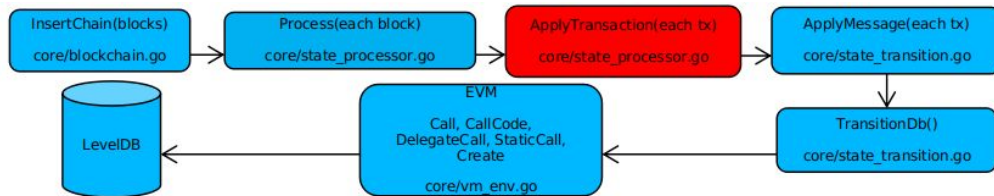


```
func (p *StateProcessor) Process(block *types.Block, statedb *state.StateDB, cfg vm.Config) (...) {
    // Something here
    ...
    // Iterate over and process the individual transactions
    for i, tx := range block.Transactions() {
        statedb.Prepare(tx.Hash(), block.Hash(), i)
        receipt, _, err := ApplyTransaction(p.config, p.bc, nil, gp, statedb, header, tx, usedGas, cfg)
        if err != nil {
            return nil, nil, 0, err
        }
        receipts = append(receipts, receipt)
        allLogs = append(allLogs, receipt.Logs...)
    }
    // Finalize the block, applying any consensus engine specific extras (e.g. block rewards)
    p.engine.Finalize(p.bc, header, statedb, block.Transactions(), block.Uncles(), receipts)

    return receipts, allLogs, *usedGas, nil
}
```

core/state_processor.go

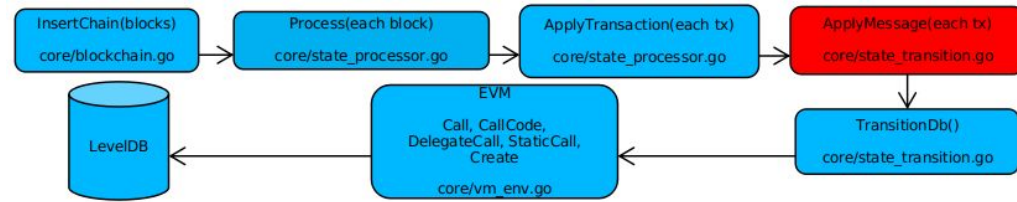
Process a new block



```
func ApplyTransaction(config *params.ChainConfig, bc ChainContext, author *common.Address, ...) (...) {  
    // Something here  
    ...  
    // Create a new context to be used in the EVM environment  
    context := NewEVMContext(msg, header, bc, author)  
    // Create a new environment which holds all relevant information  
    // about the transaction and calling mechanisms.  
    vmenv := vm.NewEVM(context, statedb, config, cfg)  
    // Apply the transaction to the current state (included in the env)  
    _, gas, failed, err := ApplyMessage(vmenv, msg, gp)  
    if err != nil {  
        return nil, 0, err  
    }  
}
```

core/state_processor.go

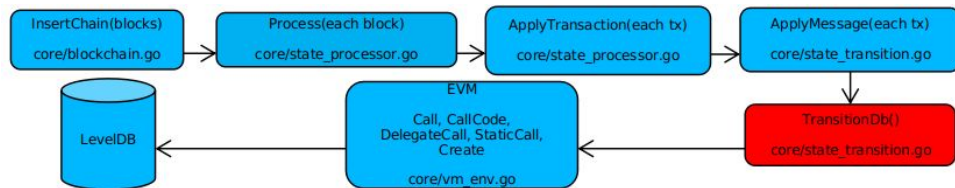
Process a new block



```
func ApplyMessage(evm *vm.EVM, msg Message, gp *GasPool) ([]byte, uint64, bool, error) {  
    return NewStateTransition(evm, msg, gp).TransitionDb()  
}
```

core/state_transition.go

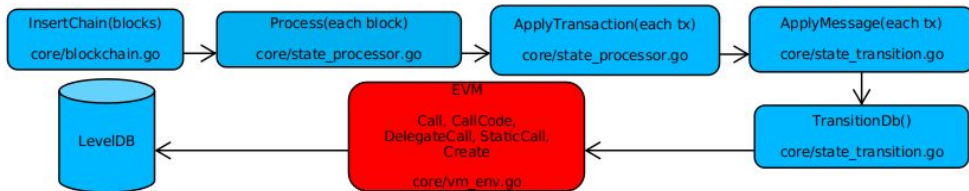
Process a new block



```
func (st *StateTransition) TransitionDb() (ret []byte, usedGas uint64, failed bool, err error) {  
    // Something here  
    ...  
    if contractCreation {  
        ret, _, st.gas, vmerr = evm.Create(sender, st.data, st.gas, st.value)  
    } else {  
        // Increment the nonce for the next transaction  
        st.state.SetNonce(msg.From(), st.state.GetNonce(sender.Address())+1)  
        ret, st.gas, vmerr = evm.Call(sender, st.to(), st.data, st.gas, st.value)  
    }  
    if vmerr != nil {  
        log.Debug("VM returned with error", "err", vmerr)  
        // The only possible consensus-error would be if there wasn't  
        // sufficient balance to make the transfer happen. The first  
        // balance transfer may never fail.  
        if vmerr == vm.ErrInsufficientBalance {  
            return nil, 0, false, vmerr  
        }  
    }  
    st.refundGas()  
    st.state.AddBalance(st.evm.Coinbase, new(big.Int).Mul(new(big.Int).SetUint64(st.gasUsed()), st.gasPrice))  
    return ret, st.gasUsed(), vmerr != nil, err  
}
```

core/state_transition.go

Process a new block



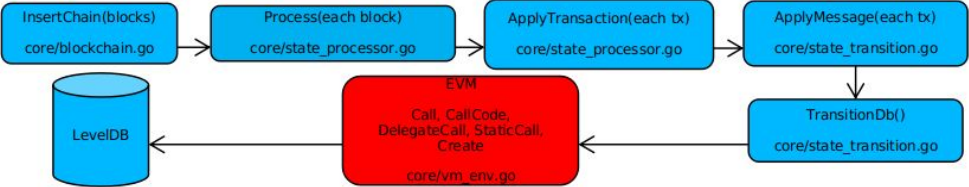
```
// Create creates a new contract using code as deployment code.
func (evm *EVM) Create(caller ContractRef, code []byte, gas uint64, value *big.Int) (contractAddr common.Address, ...) {
    contractAddr = crypto.CreateAddress(caller.Address(), evm.StateDB.GetNonce(caller.Address()))
    return evm.create(caller, &codeAndHash{code: code}, gas, value, contractAddr)
}
```

```
// create creates a new contract using code as deployment code.
func (evm *EVM) create(caller ContractRef, codeAndHash *codeAndHash, gas uint64, value *big.Int, address common.Address) (...) {
    // Something here
    ...

    // Ensure there's no existing contract already at the designated address
    contractHash := evm.StateDB.GetCodeHash(address)
    if evm.StateDB.GetNonce(address) != 0 || (contractHash != (common.Hash{}) && contractHash != emptyCodeHash) {
        return nil, common.Address{}, 0, ErrContractAddressCollision
    }
    // Create a new account on the state
    snapshot := evm.StateDB.Snapshot()
    evm.StateDB.CreateAccount(address)
    if evm.ChainConfig().IsEIP158(evm.BlockNumber) {
        evm.StateDB.SetNonce(address, 1)
    }
    evm.Transfer(evm.StateDB, caller.Address(), address, value)
}
```

core/vm/evm.go

Process a new block



0xae0b8a30b10f341... 7756097 38 days 20 hrs ago 0xffb8db803d38d794... IN Contract Creation 0 Ether 0.00250167

Overview

State Changes

Now

Comments

Transaction Hash:

0xae0b8a30b10f341699ac3509f402a73edf0d1fd578bcbcb788000fe652ad6888b

Status:

Success

Block:

7756097

248438 Block Confirmations

Timestamp:

38 days 20 hrs ago (May-14-2019 03:38:04 AM +UTC)

From:

0xffb8db803d38d794b5785ee0ac09f83b429d11b5

To:

[Contract 0x409b512e1cf94500877c5353b2a0c13b2d24914f Created]

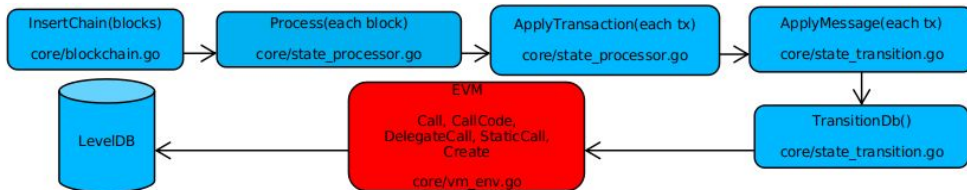
Value:

0 Ether (\$0.00)

Transaction Fee:

0.00250167874446 Ether (\$0.74)

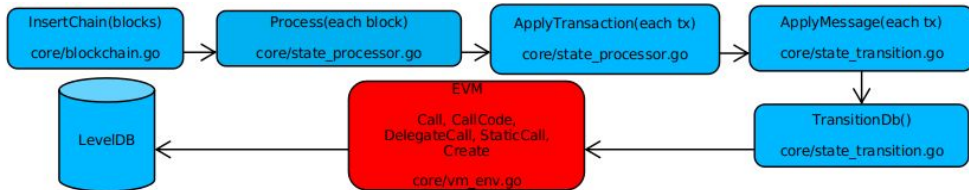
Process a new block



```
func (evm *EVM) Call(caller ContractRef, addr common.Address, input []byte, gas uint64, value *big.Int) (...) {  
    // Something here  
    ...  
    evm.Transfer(evm.StateDB, caller.Address(), to.Address(), value)  
    // Initialise a new contract and set the code that is to be used by the EVM.  
    // The contract is a scoped environment for this execution context only.  
    contract := NewContract(caller, to, value, gas)  
  
    // Something here  
    ...  
    ret, err = run(evm, contract, input, false)  
  
    // When an error was returned by the EVM or when setting the creation code  
    // above we revert to the snapshot and consume any gas remaining. Additionally  
    // when we're in homestead this also counts for code storage gas errors.  
    if err != nil {  
        evm.StateDB.RevertToSnapshot(snapshot)  
        if err != errExecutionReverted {  
            contract.UseGas(contract.Gas)  
        }  
    }  
    return ret, contract.Gas, err  
}
```

core/vm/evm.go

Process a new block



0xe5d2cc5068988a... 8004527 29 secs ago 0xe56d37ea4d0d5e... → 0x5d1d54ced0377df... 0.00560173 Ether 0.000021

Overview State Changes New Comments

Transaction Hash:

0xe5d2cc5068988a48b2f41f1c615f35c5b1e32c829664bbd4d85e60752dedc433

Status:

Success

Block:

8004527 27 Block Confirmations

Timestamp:

5 mins ago (Jun-22-2019 12:03:05 AM +UTC)

From:

0xe56d37ea4d0d5e18449eac405577c93120b9d06b

To:

0x5d1d54ced0377dfdf696ec7bff72f4c5009c0b59

Value:

0.00560173 Ether (\$1.65)

Transaction Fee:

0.000021 Ether (\$0.0062)

Switch to a new chain

```
// WriteBlockWithState writes the block and all associated state to the database.
func (bc *Blockchain) WriteBlockWithState(block *types.Block, receipts []*types.Receipt, state *state.StateDB) (...) {
    // Something here
    ...

    currentBlock := bc.CurrentBlock()
    localTd := bc.GetTd(currentBlock.Hash(), currentBlock.NumberU64())
    externTd := new(big.Int).Add(block.Difficulty(), ptd)

    // Something here
    ...

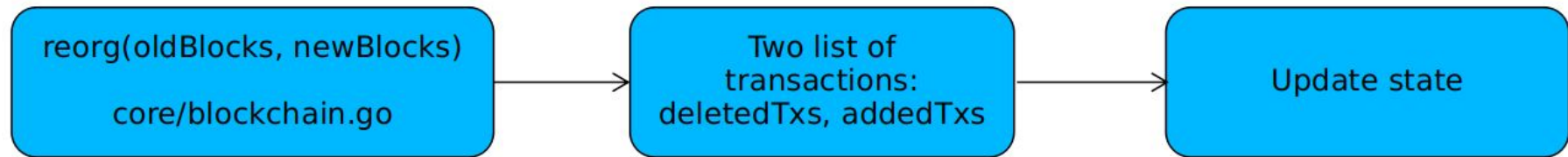
    // If the total difficulty is higher than our known, add it to the canonical chain
    // Second clause in the if statement reduces the vulnerability to selfish mining.
    // Please refer to http://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf
    reorg := externTd.Cmp(localTd) > 0
    currentBlock = bc.CurrentBlock()

    // Something here
    ...

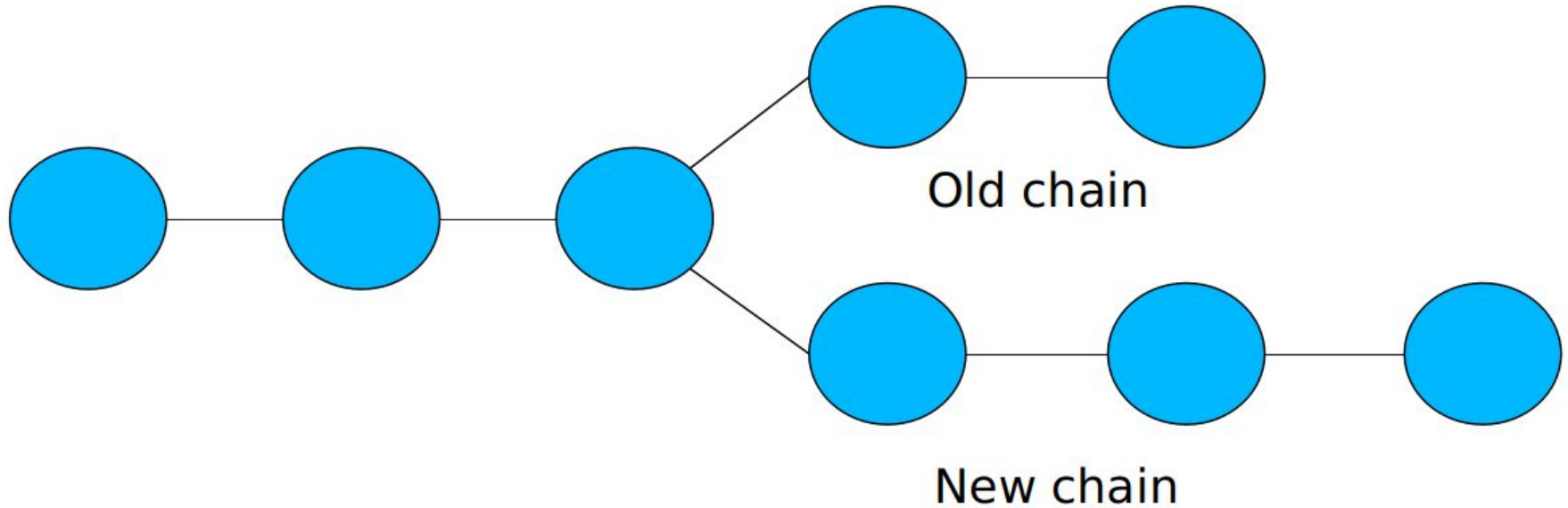
    if reorg {
        // Reorganise the chain if the parent is not the head block
        if block.ParentHash() != currentBlock.Hash() {
            if err := bc.reorg(currentBlock, block); err != nil {
                return NonStatTy, err
            }
        }
    }
}
```

core/blockchain.go

Switch to a new chain



Switch to a new chain



Block #7986472

Overview

Comments

Block Height:	7986472 < >
Timestamp:	43 secs ago (Jun-19-2019 04:04:19 AM +UTC)
Transactions:	140 transactions and 2 contract internal transactions in this block
Mined by:	0x5a0b54d5dc17e0aadc383d2db43b0a0d3e029c4c (Spark Pool) in 6 secs
Block Reward:	2.060072629887170892 Ether (2 + 0.060072629887170892)
Uncles Reward:	0
Difficulty:	2,015,925,566,716,380
Total Difficulty:	10,663,194,973,875,573,686,795
Size:	19,891 bytes
Gas Used:	7,995,494 (99.94%)
Gas Limit:	8,000,029

Transactions

For Block [7986471](#)

A total of 91 transactions found
















First

<

Page 1 of 2

>

Last

Txn Hash	Block	Age	From	To	Value	[Txn Fee]
0x29586b2aa585db...	7986471	1 min ago	0xe4d05f54eff833ab...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084
0xbd804e8aa53735...	7986471	1 min ago	0xd9a7f6484804bc6...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084
0x3ea0707a2b8436...	7986471	1 min ago	0x2fadd4fc6c442f78...	  0x3161a3a2469469...	0 Ether	0.00014692
0xf531c9ccf068092...	7986471	1 min ago	0x2fadd4fc6c442f78...	  0x3161a3a2469469...	0 Ether	0.00014718
0x6ee606a8ad5e35f...	7986471	1 min ago	0x864af554242ff204...	 0xc50489a925e9c8...	0 Ether	0.00013726
0x6dda6c254434de...	7986471	1 min ago	0xa3bd36054f577e2...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084
0x417843f5838b67f...	7986471	1 min ago	0x6a9a314aac0325...	  0x498d99de4268ce...	0 Ether	0.00063562
0x3ca62410aa8777...	7986471	1 min ago	0x89c8b42bbb181d...	  0x498d99de4268ce...	0 Ether	0.00069562
0x21d977667b34d3...	7986471	1 min ago	0xdaf07ec839e878b...	  0x498d99de4268ce...	0 Ether	0.00063562
0xf0e140f342e5b9fd...	7986471	1 min ago	0xdcecc323bb78f97...	 0xb1298e180afd7e3...	0.000521464 Ether	0.000084

Query data via RPCs

eth_blockNumber

Returns the number of most recent block.

Parameters

none

Returns

QUANTITY - integer of the current block number the client is on.

Example

```
// Request
curl -X POST --data '{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":1}'

// Result
{
  "id":83,
  "jsonrpc": "2.0",
  "result": "0xc94" // 1207
}
```


Query data via RPCs

eth_getBlockByHash

Returns information about a block by hash.

Parameters

1. `DATA` , 32 Bytes - Hash of a block.
2. `Boolean` - If `true` it returns the full transaction objects, if `false` only the hashes of the transactions.

Example Parameters

```
params: [  
  '0xe670ec64341771606e55d6b4ca35a1a6b75ee3d5145a99d05921026d1527331',  
  true  
]
```

Query data via RPCs

Example

```
// Request
curl -X POST --data '{"jsonrpc":"2.0","method":"eth_getBlockByHash","params":["0xe670

// Result
{
  "id":1,
  "jsonrpc":"2.0",
  "result": {
    "number": "0x1b4", // 436
    "hash": "0xe670ec64341771606e55d6b4ca35a1a6b75ee3d5145a99d05921026d1527331",
    "parentHash": "0x9646252be9520f6e71339a8df9c55e4d7619deeb018d2a3f2d21fc165dde5eb5",
    "nonce": "0xe04d296d2460cfb8472af2c5fd05b5a214109c25688d3704aed5484f9a7792f2",
    "sha3Uncles": "0x1dcc4de8dec75d7aab85b567b6ccdd41ad312451b948a7413f0a142fd40d49347",
    "logsBloom": "0xe670ec64341771606e55d6b4ca35a1a6b75ee3d5145a99d05921026d1527331",
    "transactionsRoot": "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cad001622fb5e3",
    "stateRoot": "0xd5855eb08b3387c0af375e9cdb6acfc05eb8f519e419b874b6ff2ffda7ed1dfff",
    "miner": "0x4e65fda2159562a496f9f3522f89122a3088497a",
    "difficulty": "0x027f07", // 163591
    "totalDifficulty": "0x027f07", // 163591
    "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "size": "0x027f07", // 163591
    "gasLimit": "0x9f759", // 653145
    "gasUsed": "0x9f759", // 653145
    "timestamp": "0x54e34e8e" // 1424182926
    "transactions": [{...},{...}]
    "uncles": ["0x1606e5...", "0xd5145a9..."]
  }
}
```

Query data via RPCs

eth_getBlockByNumber

Returns information about a block by block number.

Parameters

1. `QUANTITY|TAG` - integer of a block number, or the string `"earliest"`, `"latest"` or `"pending"`, as in the [default block parameter](#).
2. `Boolean` - If `true` it returns the full transaction objects, if `false` only the hashes of the transactions.

Example Parameters

```
params: [  
  '0x1b4', // 436  
  true  
]
```

Returns

See [eth_getBlockByHash](#)

Query data via RPCs

`eth_getTransactionByHash`

Returns the information about a transaction requested by transaction hash.

Parameters

1. `DATA` , 32 Bytes - hash of a transaction

Example Parameters

```
params: [  
  "0x88df016429689c079f3b2f6ad39fa052532c56795b733da78a91ebe6a713944b"  
]
```

Query data via RPCs

Example

```
// Request
curl -X POST --data '{"jsonrpc":"2.0","method":"eth_getTransactionByHash","params":["

// Result
{
  "jsonrpc":"2.0",
  "id":1,
  "result":{
    "blockHash":"0x1d59ff54b1eb26b013ce3cb5fc9dab3705b415a67127a003c3e61eb445bb8df2",
    "blockNumber":"0x5daf3b", // 6139707
    "from":"0xa7d9ddbe1f17865597fbd27ec712455208b6b76d",
    "gas":"0xc350", // 50000
    "gasPrice":"0x4a817c800", // 200000000000
    "hash":"0x88df016429689c079f3b2f6ad39fa052532c56795b733da78a91ebe6a713944b",
    "input":"0x68656c6c6f21",
    "nonce":"0x15", // 21
    "to":"0xf02c1c8e6114b1dbe8937a39260b5b0a374432bb",
    "transactionIndex":"0x41", // 65
    "value":"0xf3dbb76162000", // 42900000000000000
    "v":"0x25", // 37
    "r":"0x1b5e176d927f8e9ab405058b2d2457392da3e20f328b16ddabcebc33eaac5fea",
    "s":"0x4ba69724e8f69de52f0125ad8b3c5c2cef33019bac3249e2c0a2192766d1721c"
  }
}
```

Query data via RPCs

eth_getBalance

Returns the balance of the account of given address.

Parameters

1. `DATA` , 20 Bytes - address to check for balance.
2. `QUANTITY|TAG` - integer block number, or the string `"latest"` , `"earliest"` or `"pending"` , see the [default block parameter](#)

Example Parameters

```
params: [  
  '0xc94770007dda54cf92009BFF0dE90c06F603a09f',  
  'latest'  
]
```

Returns

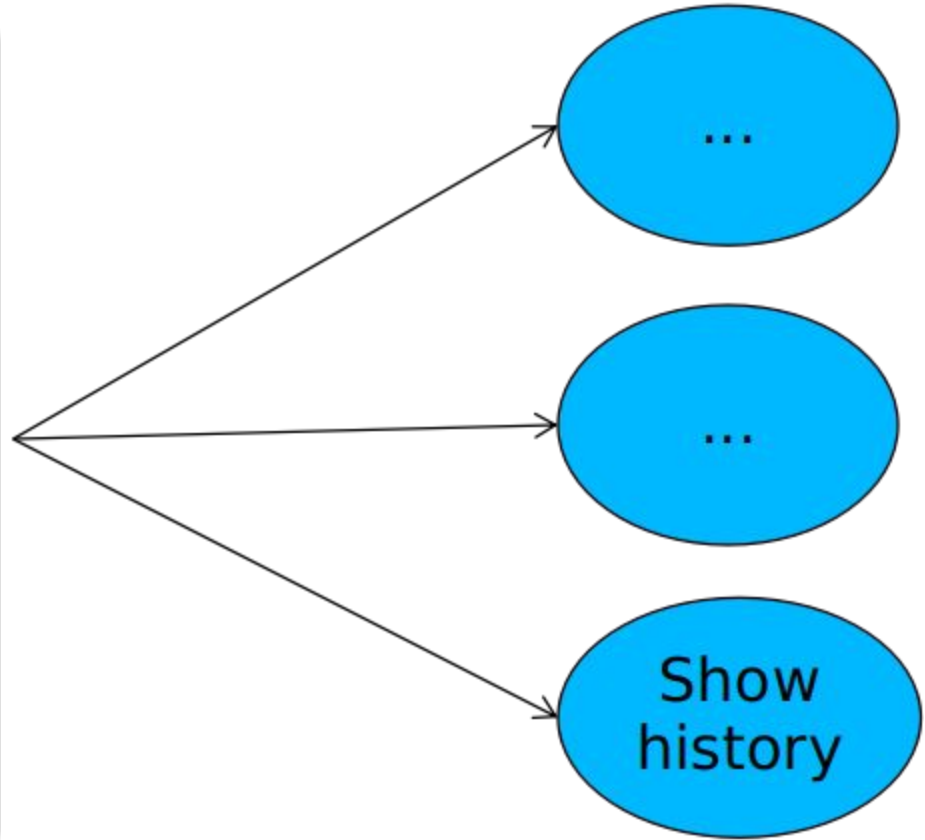
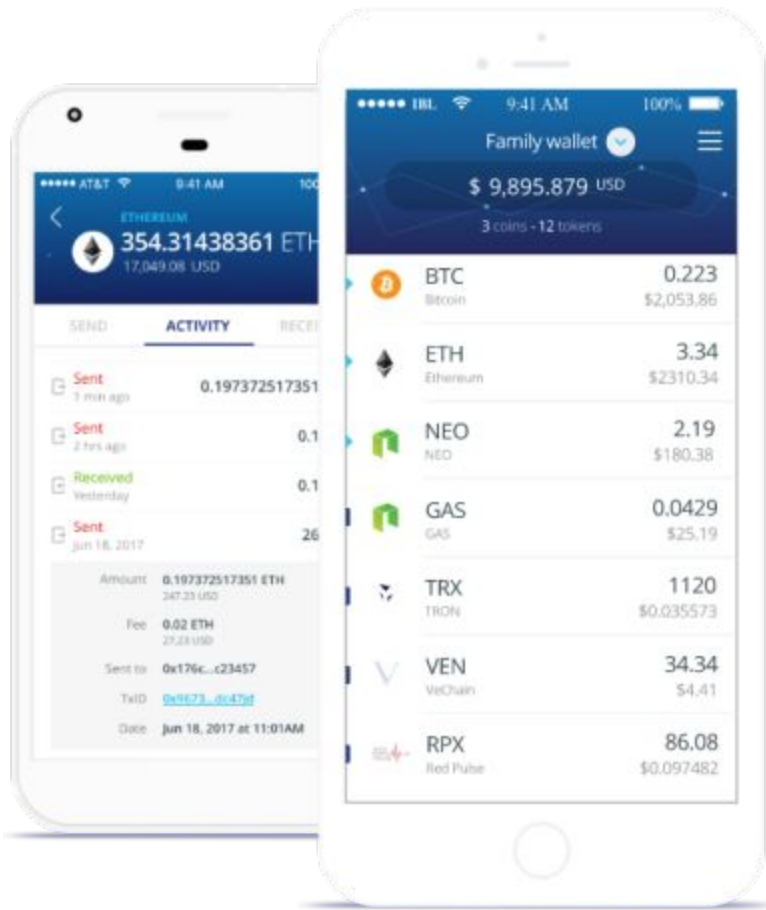
`QUANTITY` - integer of the current balance in wei.

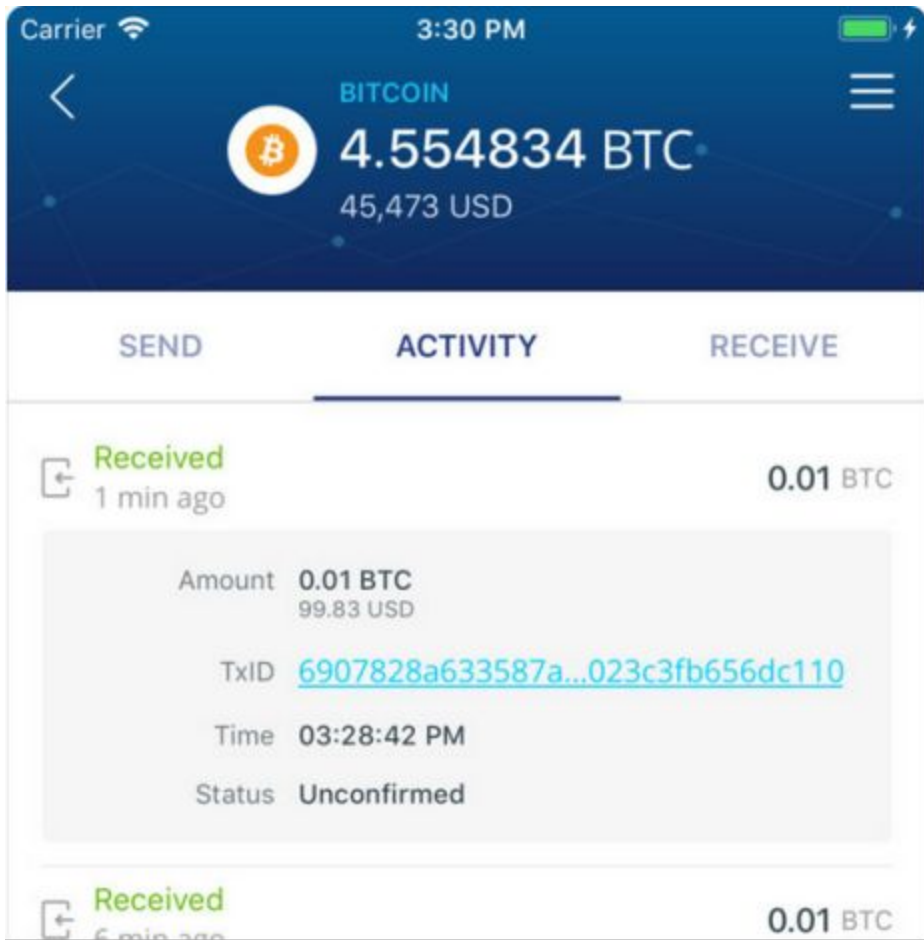
Query data via RPCs

Example

```
// Request
curl -X POST --data '{"jsonrpc":"2.0","method":"eth_getBalance","params":["0xc9477000

// Result
{
  "id":1,
  "jsonrpc": "2.0",
  "result": "0x0234c8a3397aab58" // 158972490234375000
}
```





How to get history ?





User of Infinito wallet

Has transaction
in Block 10
+ 1 ETH

...

Has transaction
in Block
1.000.000
- 0.5 ETH

...

Has transaction
in Block
5.000.000
+ 10 ETH



Hey, I wanna check my activities, please help ?!



Searching all data ...



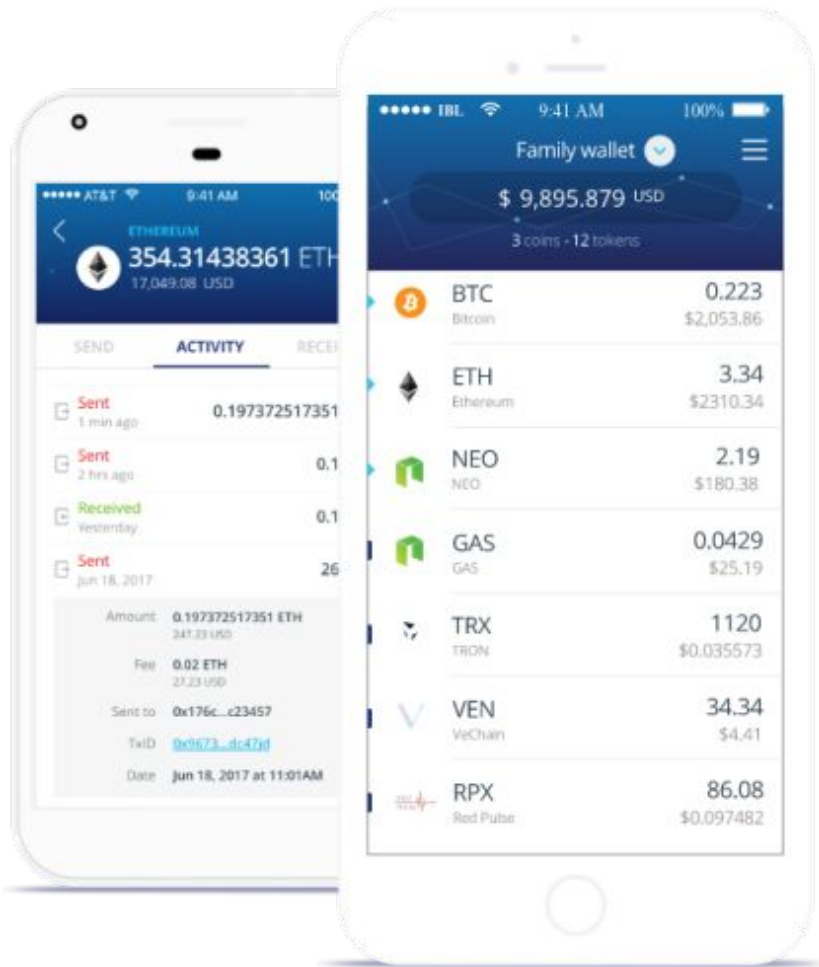
Dev



I'll try my best, just wait for 1 day, is it OK ?

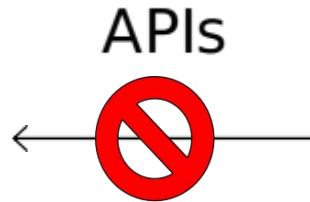
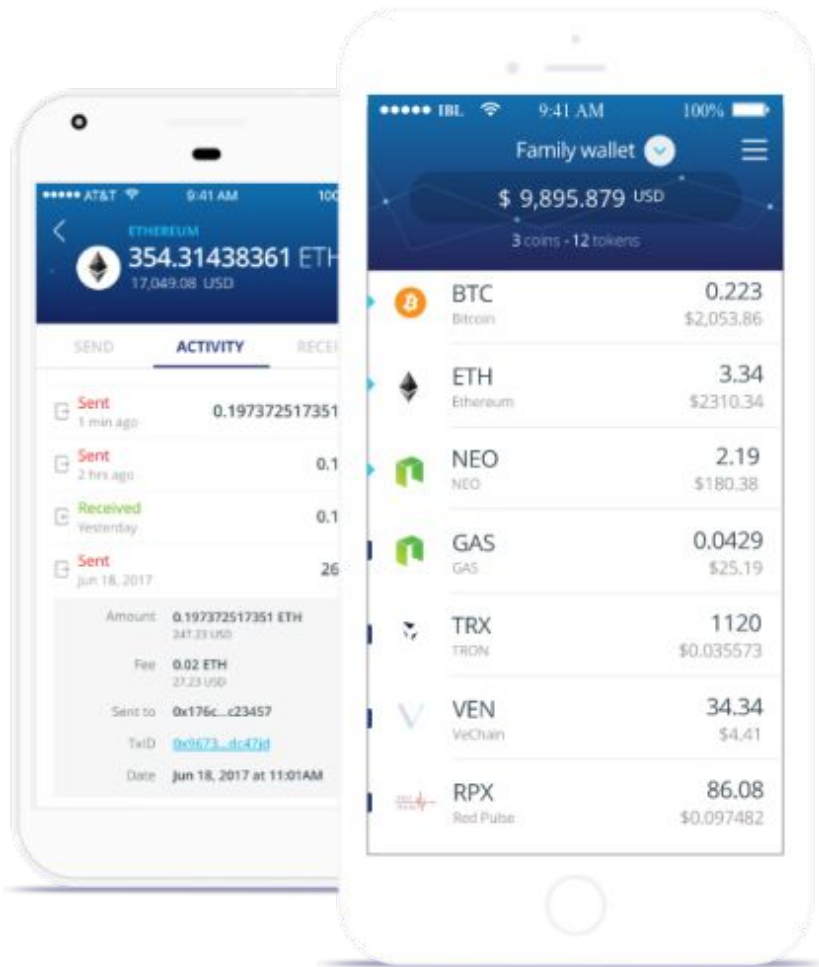


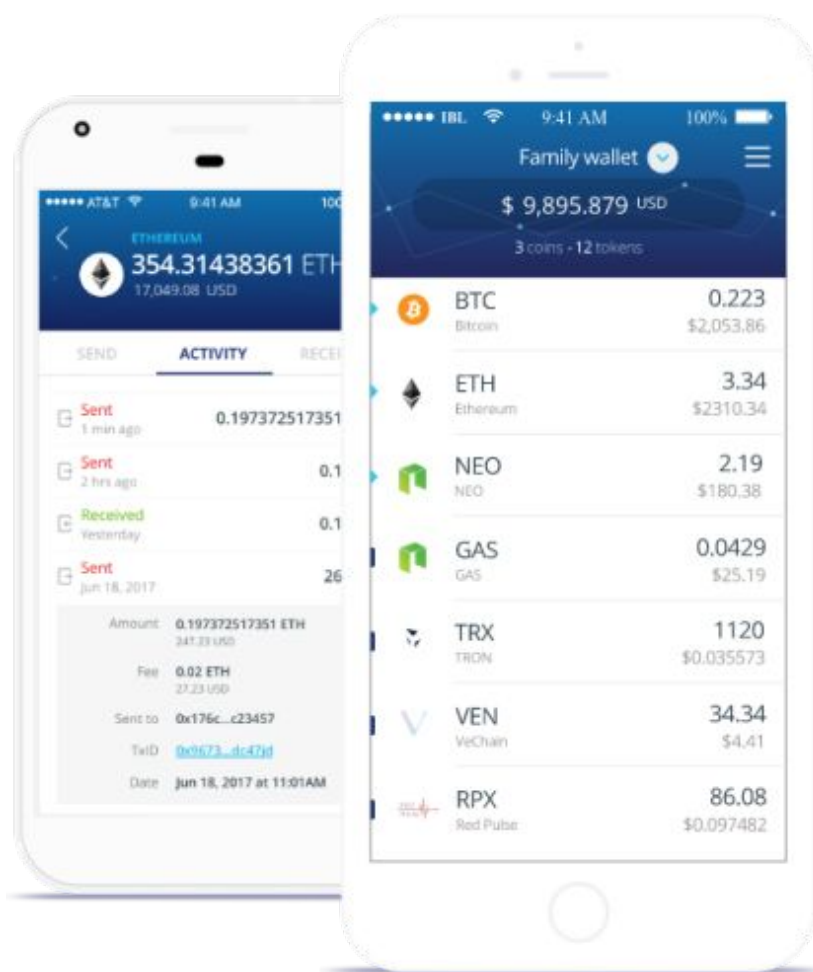
OMG ! Are you kidding me ?!

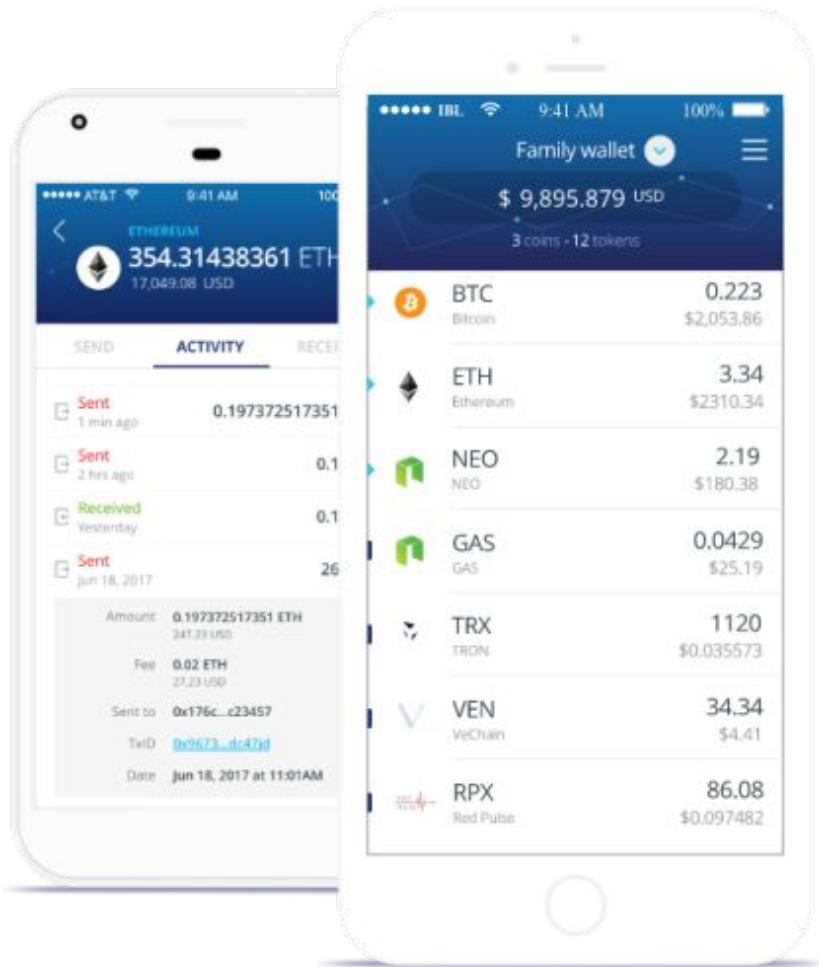



APIs







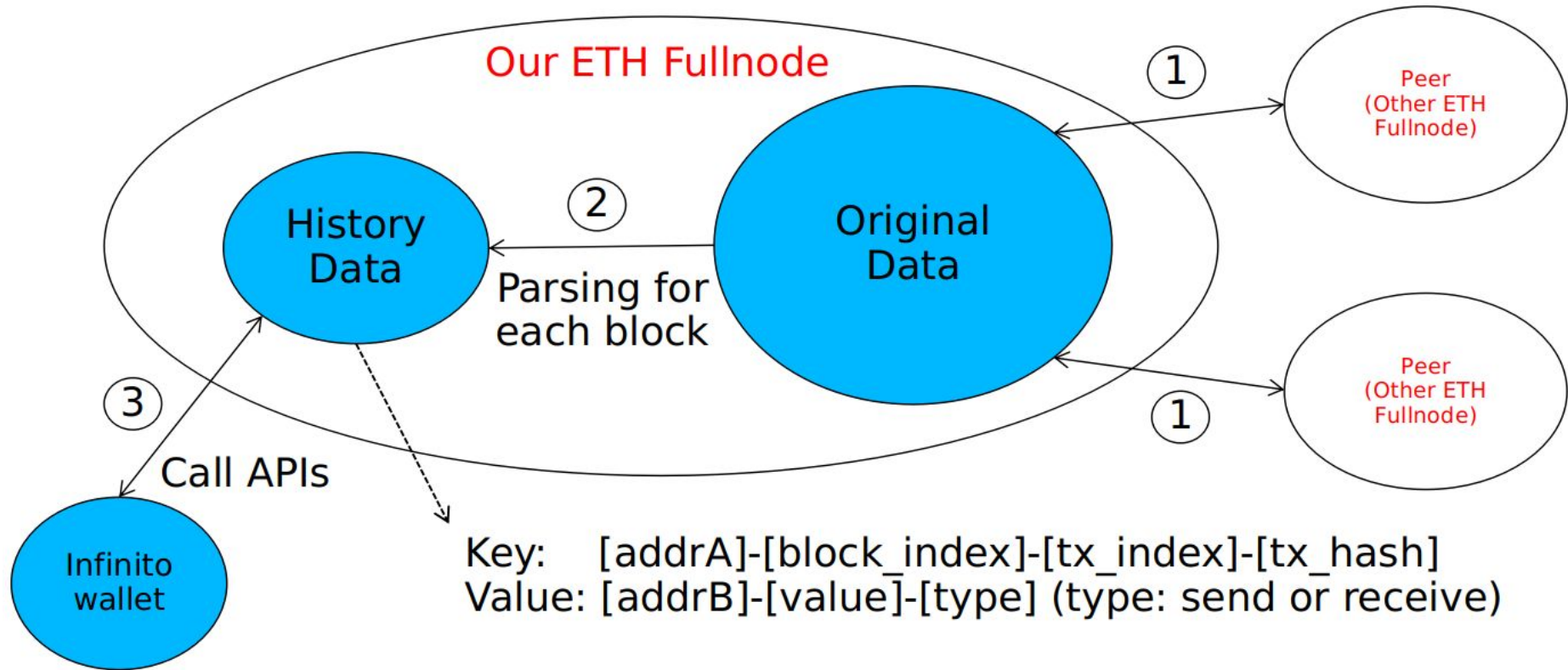


APIs with 

←



IBL's Services



Assume that we have a new block like this:

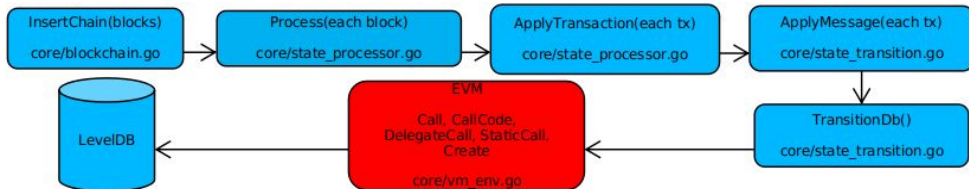
```
"Block": {  
  "Transactions": {  
    "tx": A send 10 ETH to B  
  }  
}
```

We need to save this activity:

(**key**: A - block_index - tx_index - tx_hash, **value**: B - 10 - 0)

(**key**: B - block_index - tx_index - tx_hash, **value**: A - 10 - 1)

Process a new block



```
func (evm *EVM) Call(caller ContractRef, addr common.Address, input []byte, gas uint64, value *big.Int) (...) {  
    // Something here  
    ...  
    evm.Transfer(evm.StateDB, caller.Address(), to.Address(), value)  
    // Initialise a new contract and set the code that is to be used by the EVM.  
    // The contract is a scoped environment for this execution context only.  
    contract := NewContract(caller, to, value, gas)  
  
    // Something here  
    ...  
    ret, err = run(evm, contract, input, false)  
  
    // When an error was returned by the EVM or when setting the creation code  
    // above we revert to the snapshot and consume any gas remaining. Additionally  
    // when we're in homestead this also counts for code storage gas errors.  
    if err != nil {  
        evm.StateDB.RevertToSnapshot(snapshot)  
        if err != errExecutionReverted {  
            contract.UseGas(contract.Gas)  
        }  
    }  
    return ret, contract.Gas, err  
}
```

core/vm/evm.go

Thanks for listening!