How the Ethereum blockchain works, for analytics

Shahan Khatchadourian
ConsenSys

March 7, 2017

Outline

1. Ethereum **Protocol Stack**
2. Creating and updating **account** state
3. Creating and updating **contract** state
4. **Peer** activity and **block creation**
5. Data of the blockchain
6. Ethereum Improvement Proposals (EIPs)
7. Interesting social events
8. Challenges

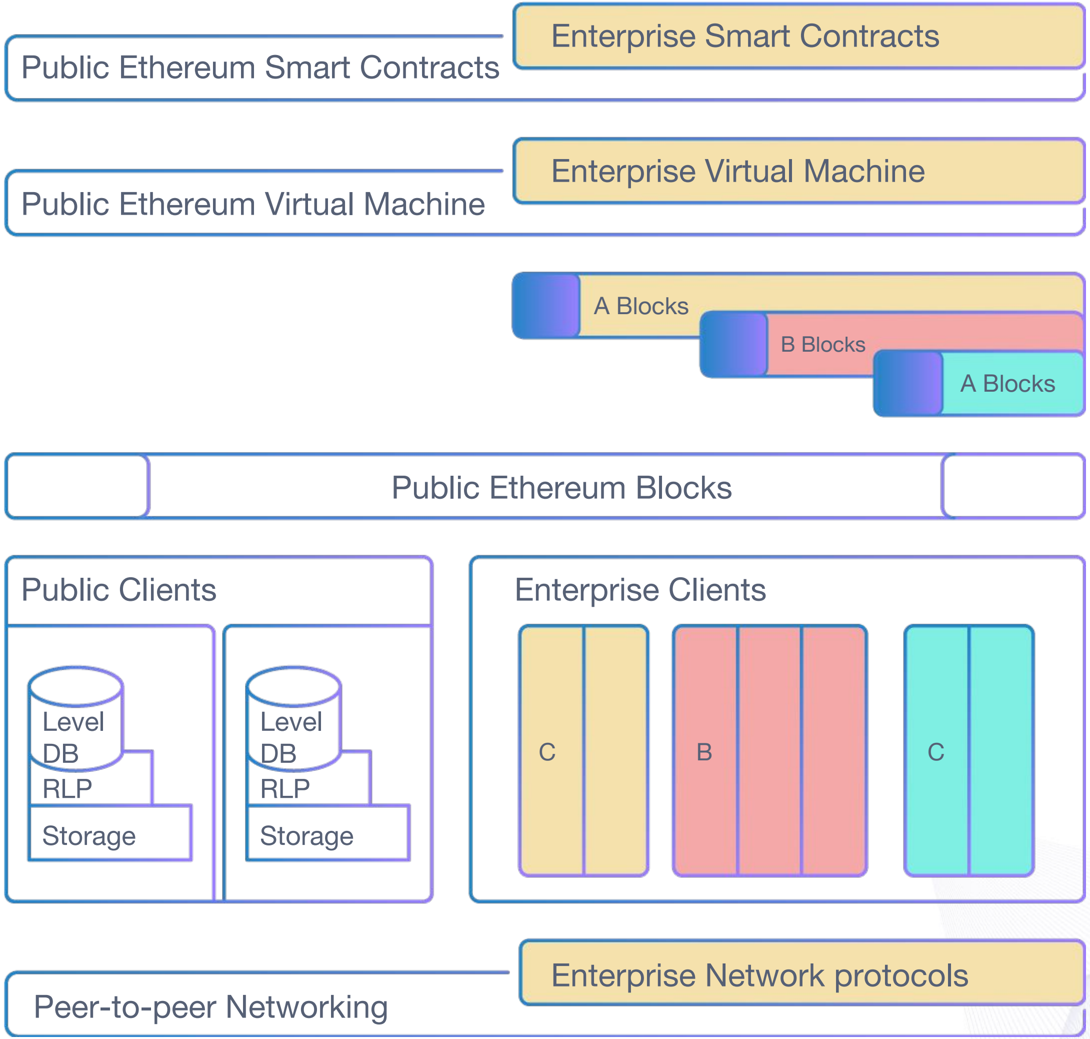
Enterprise Ethereum Protocol Stack

trustless: cryptography

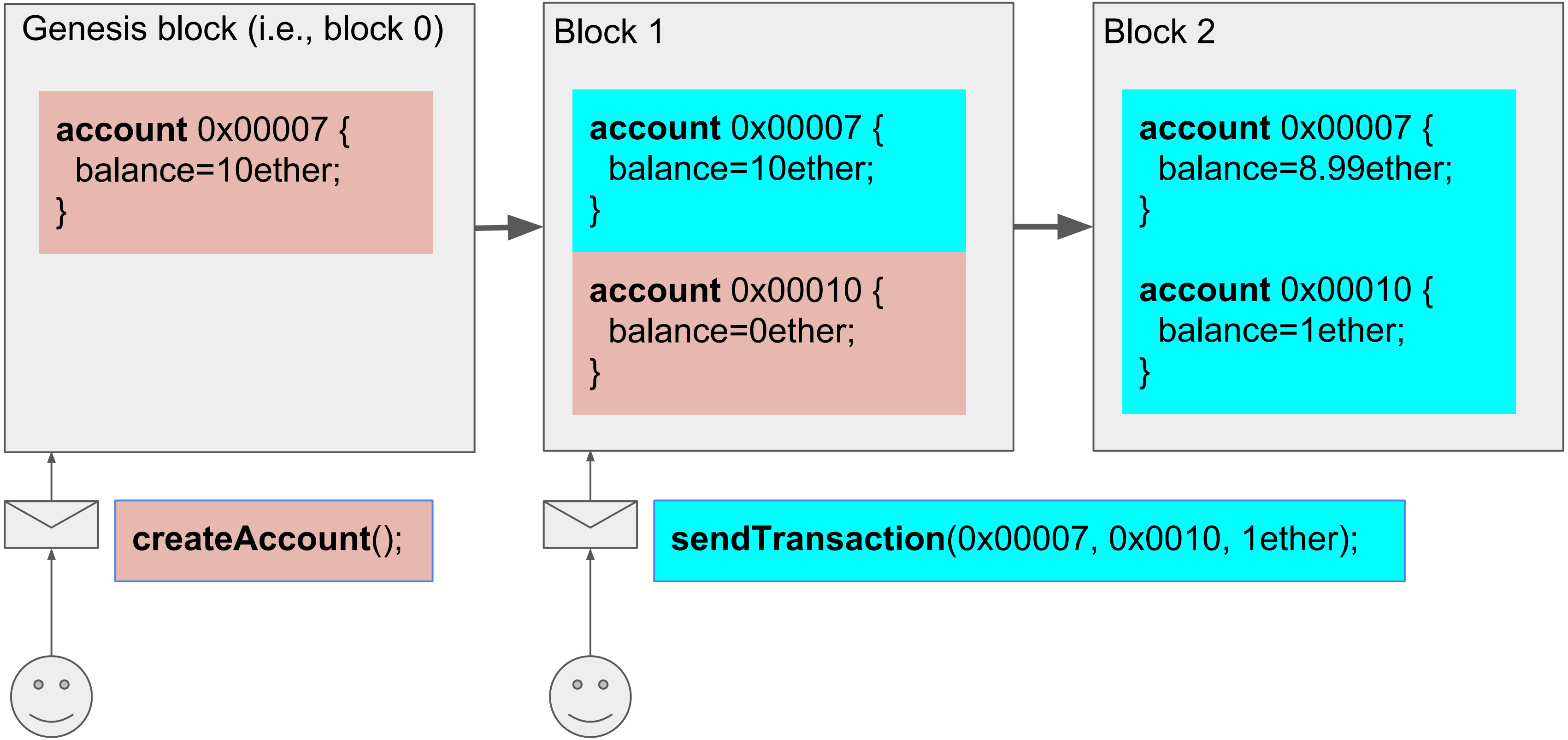
de-centralized: peer-to-peer

flexible: can run code

immutable: state can change but history is kept


Enterprise Smart Contracts
Public Ethereum Smart Contracts
Enterprise Virtual Machine
Public Ethereum Virtual Machine
A Blocks
B Blocks
A Blocks
Public Ethereum Blocks
Public Clients
Level DB
RLP
Storage
Level DB
RLP
Storage
Enterprise Clients
C
B
C
Enterprise Network protocols
Peer-to-peer Networking

Updating **account** state on the Ethereum blockchain



Genesis block (i.e., block 0)

```
account 0x00007 {
  balance=10ether;
}
```

Block 1

```
account 0x00007 {
  balance=10ether;
}
```

```
account 0x00010 {
  balance=0ether;
}
```

Block 2

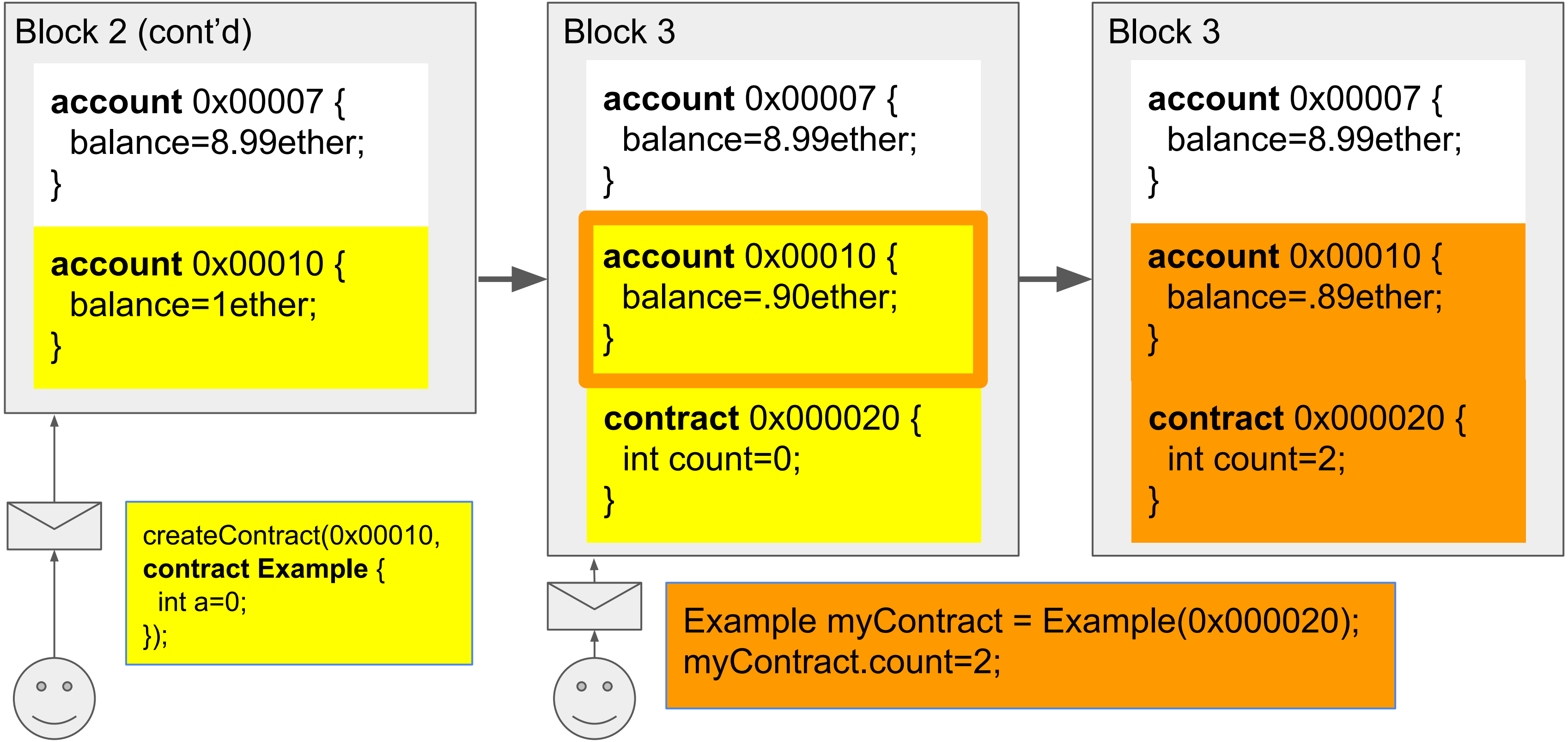
```
account 0x00007 {
  balance=8.99ether;
}
```

```
account 0x00010 {
  balance=1ether;
}
```

createAccount();

sendTransaction(0x00007, 0x0010, 1ether);

Updating **contract** state on the Ethereum blockchain



Block 2 (cont'd)

```
account 0x00007 {
  balance=8.99ether;
}
```

```
account 0x00010 {
  balance=1ether;
}
```

```
createContract(0x00010,
contract Example {
  int a=0;
});
```

Block 3

```
account 0x00007 {
  balance=8.99ether;
}
```

```
account 0x00010 {
  balance=.90ether;
}
```

```
contract 0x000020 {
  int count=0;
}
```

```
Example myContract = Example(0x000020);
myContract.count=2;
```

Block 3

```
account 0x00007 {
  balance=8.99ether;
}
```

```
account 0x00010 {
  balance=.89ether;
}
```

```
contract 0x000020 {
  int count=2;
}
```

Peer activity and block creation

Clients propagate transactions throughout the network.

Miners create blocks of transactions: requires lots of electricity and computational power (GPUs > CPUs; ASICs cannot be used for now).

1. Choose the transactions that go into a block
   - pick in order of decreasing gas payout (gas is credited to the miner)
2. Do some heavy computation for an appropriate value
3. If an appropriate value is found, send it to network with a small verifiable proof
4. The network verifies and accepts the block, miner is rewarded (+gas payout)
   - Adds block to a sequential chain (forks are to be avoided because branches are incompatible)
5. Repeat

Data of the blockchain

- Account balances
- Contract balances
    - Software token balances, e.g., ERC20
- Account and contract transactions
    - Value transactions
    - Gas usage
    - Code execution (hard)
- Block creation rate, hash power, difficulty
- Number of accounts created

Ethereum Improvement Proposals (EIPs)

Proposals for updating the public Ethereum protocol with bug fixes, new features, resolving exploits, and even resolving socially contentious issues. https://github.com/ethereum/EIPs

Proposal to resolve cost of Input/Output operations (they did not use much gas relative to the performance cost), https://github.com/ethereum/EIPs/issues/150

Proposal to **hard fork**: https://github.com/TheDAO/EIPs/blob/master/EIPS/eip-2.mediawiki

A hard form fork indicates a significant change to client software (that everyone on the network runs) and proceeds if there is approval by a majority of miners on the network. Those not in the majority end up forming their own (incompatible) chain.

Interesting ‘social’ events

TheDAO: the most successful crowdfunding in history in which over 28 days the social contract application received $150M from 11,000 contributions,

BUT developers introduced a bug and someone conducted a heist successfully

However; recovery of funds was “successful”. After the heist was discovered, a proposal to ‘steal back’ the money via a hard fork to filter for specific accounts used in the heist.

Challenges

Exact number of machines/miners at the instance a block is mined is unknown

- Perhaps it can be inferred from the hashpower, or the number of reward addresses
- Perhaps rewards address can also be correlated with percentage of rewards earned by mining pools

Contract execution details require understanding of instruction codes:

- no easy way to 'replay' transactions and log the instructions

Conclusion

- Public chain has around 3.5 million blocks, around 20GB (via client)
- Public chain value estimated at > $1bn
- The only public chain that has a high value and has interesting properties such as smart contracts
- Continues to grow, increasing enterprise adoption