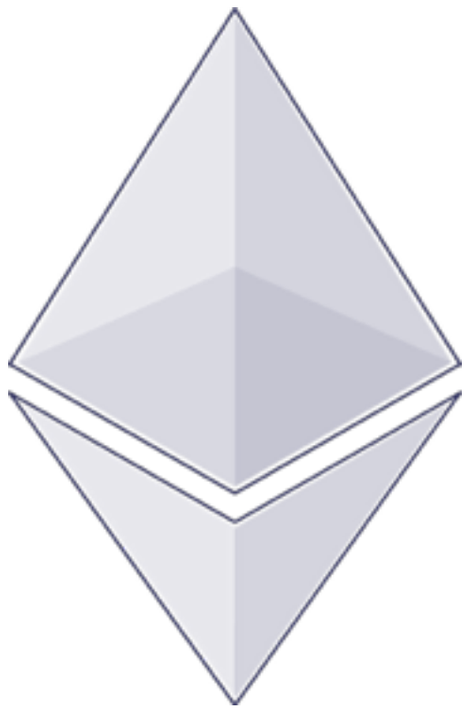


ethereum

Oliver Batista

O que é?

- Plataforma open-source para construir e distribuir aplicações descentralizadas
- Sem intermediários, o usuário está no controle de suas informações pessoais e financeira o tempo todo
- 100% peer-to-peer com "criptografia de fábrica", a prova de censura e inspeção de pacotes
- Tecnologia de Consenso Decentralizado, onde os nós são recompensados por proteger a rede

Por quê?

- Modelos Centralizados
- Confiança de dados pessoais e financeiros a terceiros
- Perda de privacidade (monetização)
- Ponto único de ataque e falhas

Histórico

- Protocolos e-cash (1980/90) - Chaumian blinding garantindo privacidade
- Wei Dai b-money (1998) - Consenso descentralizado e criação de valor pela solução de problemas computacionais
- Hal Finney (2005) - b-money + Hashcash, Reusable proof of work

Bitcoin

- Satoshi Nakamoto (2009) - gerência de propriedade por chaves públicas criptográficas e algoritmo de consenso para manter registro de posse (proof of work)

Ledger

From	To	Amt
Bill	Alice	15
Jon	Ann	3
Bob	Ryan	30

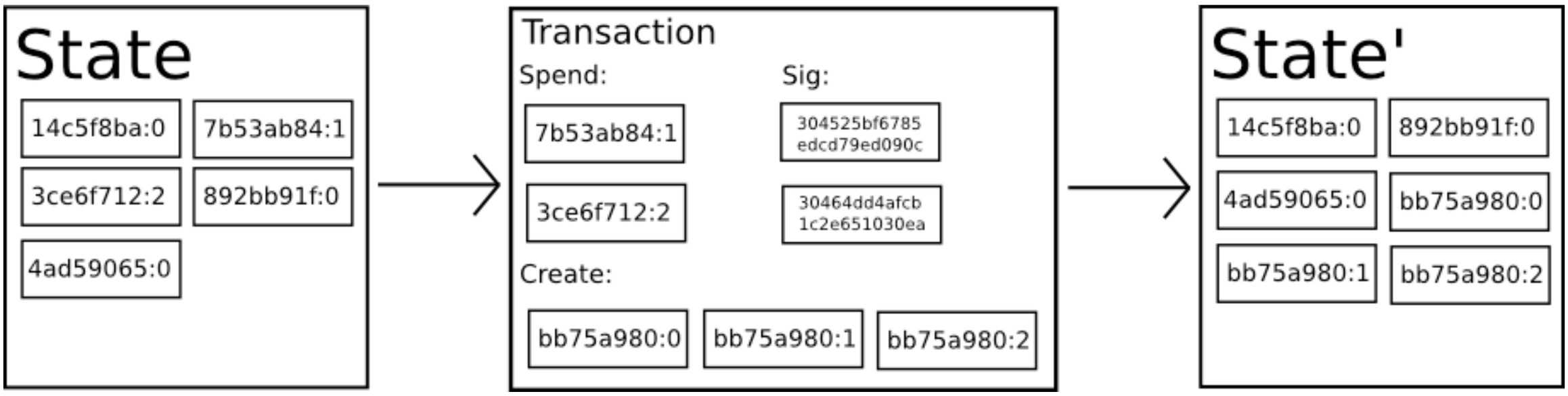
Unverified

From	To	Amt
Alice	Bob	10

Sistema de Transição de Estados

- "Livro Contábil" do Bitcoin como um sistema de transição de estados
- Estado é o status de propriedade de todas moedas
- *APPLY(S,TX) -> S' or ERROR*
- *APPLY*({ Alice: $50, Bob: $50 },"envie $20 de Alice para Bob") = { Alice: $30, Bob: $70 }

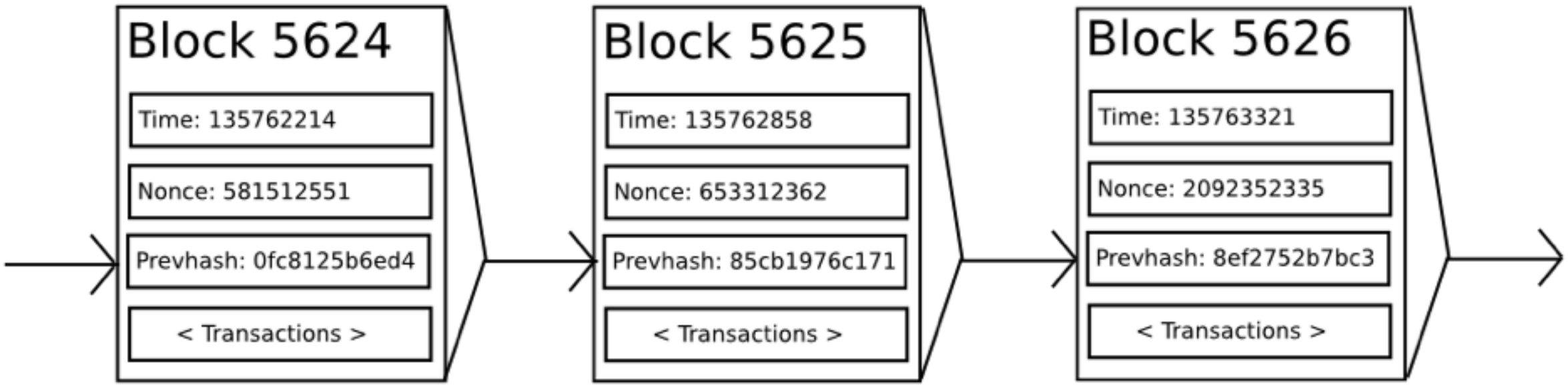
Sistema de Transição de Estados



Blockchain

- Distribuindo Consenso
- Cada nodo produz continuamente pacotes de transações chamados blocos
- Bloco a cada 10 minutos (Bitcoin)
- Cada bloco contendo referência (hash) do bloco anterior, timestamp, nonce e lista de todas transações

Blockchain


Block 5624
Time: 135762214
Nonce: 581512551
Prevhash: 0fc8125b6ed4
< Transactions >
Block 5625
Time: 135762858
Nonce: 653312362
Prevhash: 85cb1976c171
< Transactions >
Block 5626
Time: 135763321
Nonce: 2092352335
Prevhash: 8ef2752b7bc3
< Transactions >

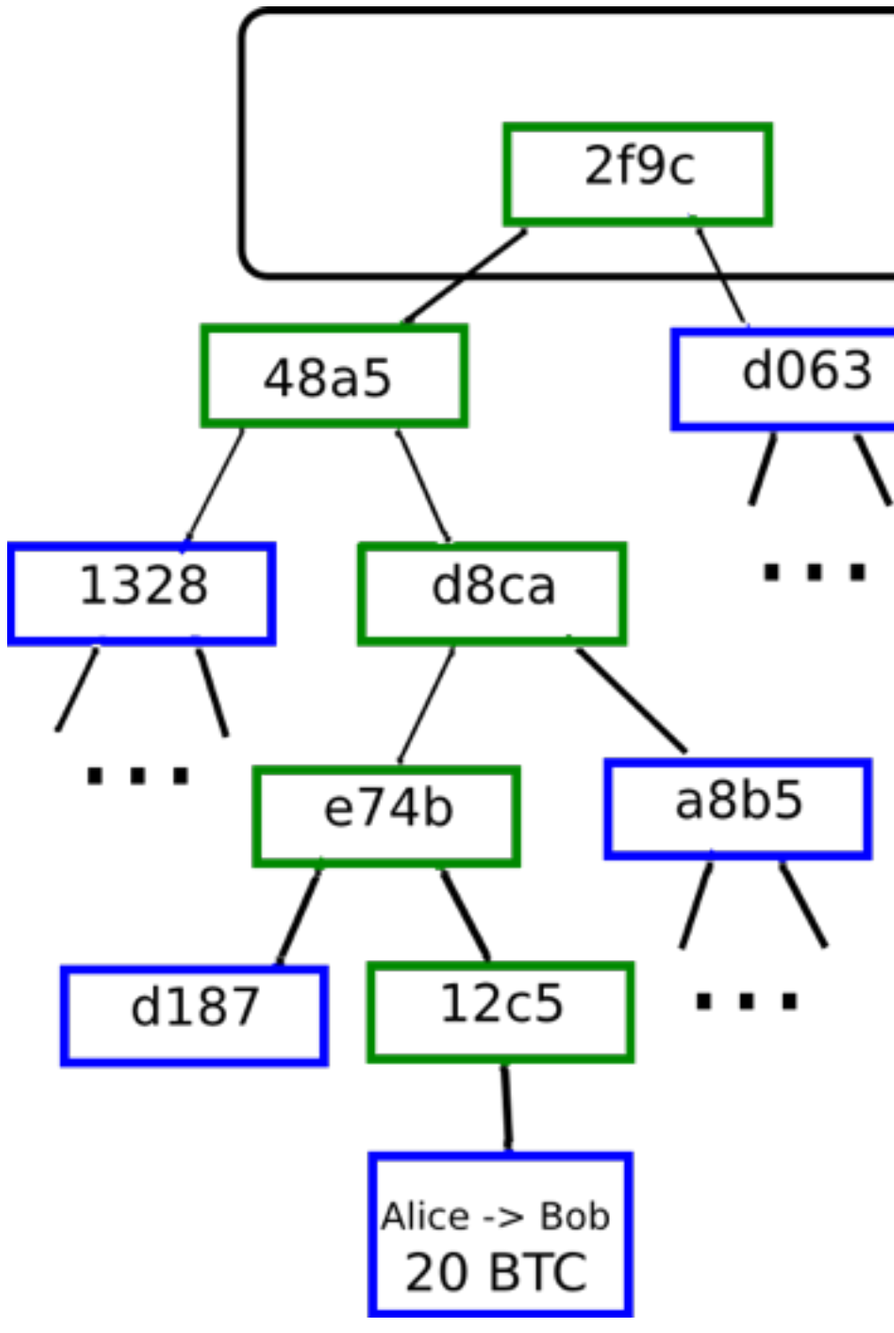
Blockchain

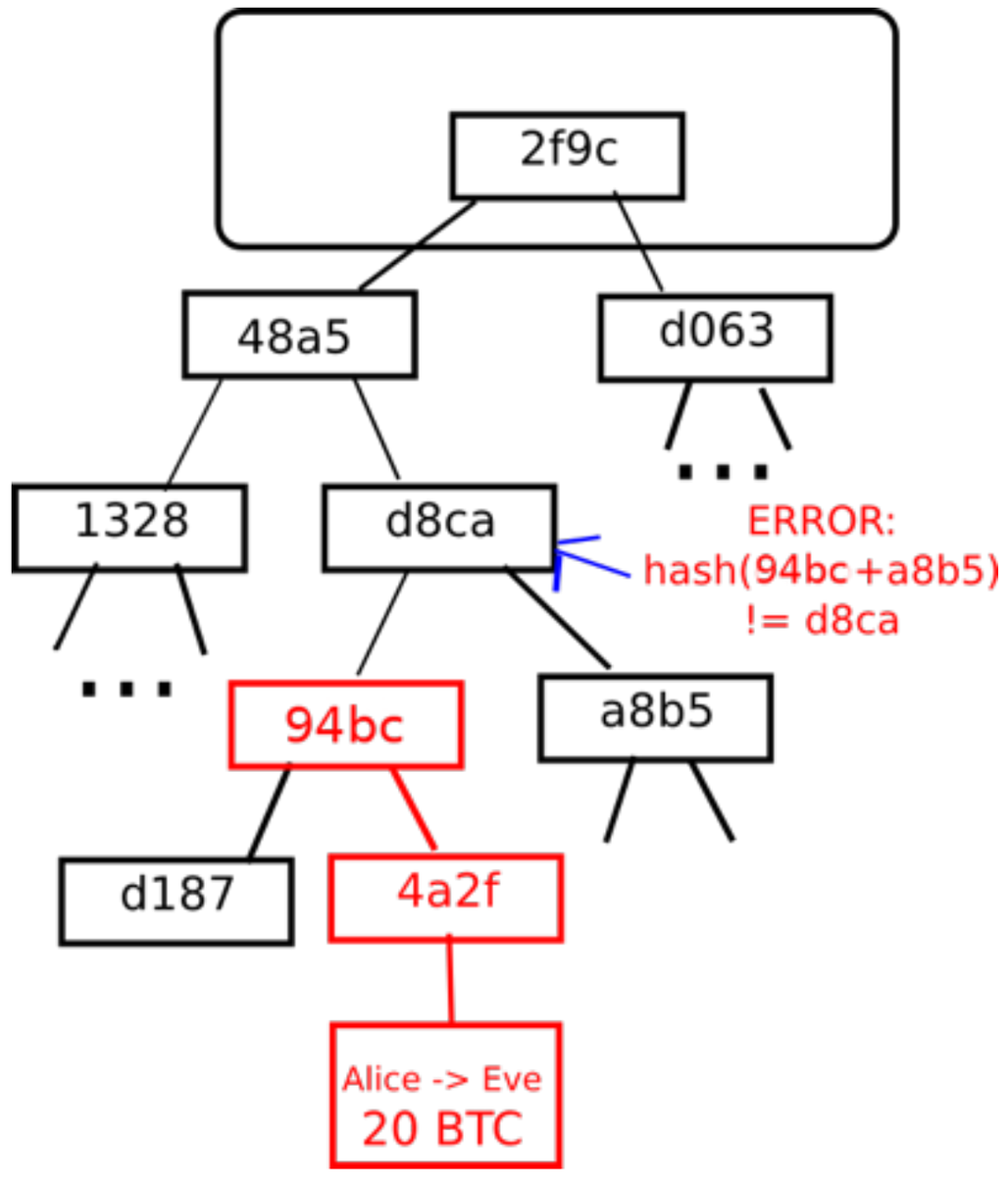
- O estado atual é uma abstração, sequências de transações são registradas no lugar
- Criação de Blocos válidos depende de “proof of work”, trabalho recompensado em Bitcoin
- Hash duplo (SHA256) de cada bloco deve ser menor que a “dificuldade" atual da rede
- Devido a imprevisibilidade do SHA256, criação se dá por tentativa e erro

Árvores de Merkle

- Estrutura de dados multinível para registrar transações
- Tipo de árvore binária, dados ficam em nós folha, nós acima compostos pelo hash de dois nós filhos
- Cabeçalho do bloco contém a hash raiz dessa árvore que registra todas transações

Árvores de Merkle





Aplicações Alternativas

- Namecoin - registro de nomes
- Colored Coins - criação de moeda própria, "pintando bitcoins"
- Meta Coins - Protocolo em cima do Bitcoin, com função de transição de estado própria

Scripting

- Não é Turing-Completo
- Desconhece seu valor total
- Não possui um “estado”
- Desconhece a blockchain

Ethereum

- Protocolo alternativo para construção de aplicações descentralizadas
- Blockchain com uma ling. de programação Turing-Completa
- Qualquer um possa escrever contratos inteligentes e apps com suas próprias regras de propriedade e função de transição de estado
- Namecoin em duas linhas de código

Contas

- No lugar de um simples registro de valor, uma estrutura de dados que contenha código, possua memória e "saiba onde está."
- Contas Externas vs Contas Contrato

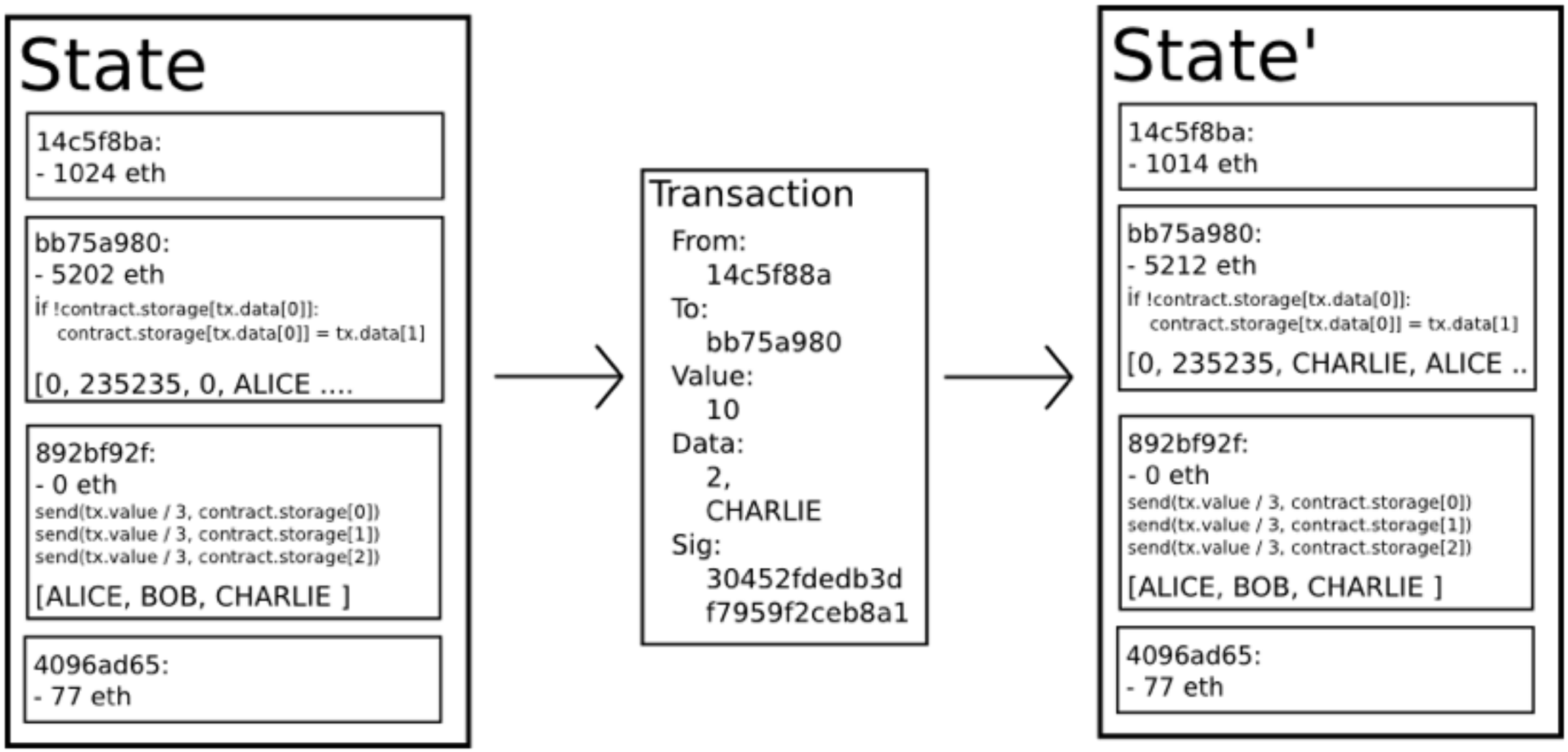
Transações

- Pacotes de dados assinados com mensagem a ser enviado a partir de conta externa
- Contém destinatário, remetente, ether, campo de dados, STARTGAS, GASPRICE
- Prevenção a negação de serviço, loop infinito e desperdício computacional

Mensagens

- Comunicação entre contratos
- Contém remetente, destinatário, ether, campo de dados, STARTGAS
- Um contrato em execução pode repassar funções para outros contratos, sempre "pagando" por isso

Transição de Estados




```
if !self.storage[calldataload(0)]:
    self.storage[calldataload(0)] = calldataload(32)
```

Execução de Código

- Ethereum Virtual Machine Code, linguagem bytecode
- Implementações em Python, Lisp e Go
- Código de contrato é parte da função de transição de estado que é parte do algoritmo de validação de bloco

Blockchain e Mineração

- Blocos contém lista de transações, o estado mais recente, número e dificuldade
- Não guarda o estado completo e sim as diferenças em relação ao bloco anterior
- Dessa forma não é necessário fazer download de todo histórico da blockchain economizando espaço dessa forma

Smart Contracts

- Proposto por Nick Szabo em 1994
- Programas de computador capazes de executar termos de um contrato.
- Exemplo: Vending Machine
- Organizações Autônoma Descentralizadas

Aplicações

- Derivativos Financeiros
- Mercados Preditivos (Augur)
- Sistemas de Identidade e Reputação
- Armazenamento de Arquivos
- Data feed descentralizado
- Computação nas nuvens (Golem)

Aplicações

- Testamentos
- Registro propriedade intelectual
- Sistemas de votação
- Crowdfunding
- Seguros