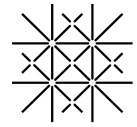


University of Basel

Center for Innovative Finance

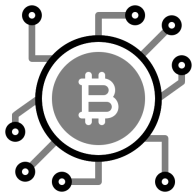
Smart Contracts and Decentralized Finance

Account-based Model

Prof. Dr. Fabian Schär
University of Basel

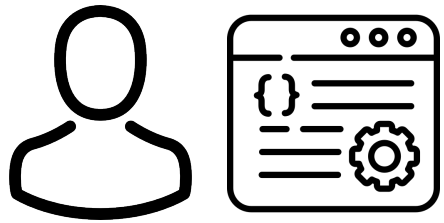
Release Ver.: (Local Release)
Version Hash: 34e09b088b6ea52cabd4443f56991b3f9d4fdc35
Version Date: 2022-09-16 16:04:03 +0200

Programmable Money Paradigms



UTXO Model

- Conditions are on each UTXO.
- Determine how the UTXO can be spend.



Account-based Model

- Conditions are on each account.
- Determine how the account is governed.

Key Takeaway

The ability to implement logic on an account-level is more flexible, more intuitive and easier to handle than conditions on the UTXO-level.

Accounts

Instead of Bitcoin's UTXO model, Ethereum uses actual accounts.

- Each account has a 20 byte address.
- Hexadecimal encoding

Example: `0xD3EEC7A7C7867B9D02248B414F08609A2A6C2AB4`

EIP-55: Capital letter checksum

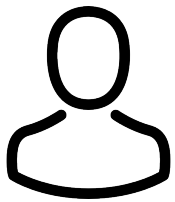
No checksum encoding → Typos were a severe problem.

EIP-55 fixes this: Use keccak hash function $H(x) = h$, where input x corresponds to the address and h to the hash value. For each character (a-f) at position $x[i]$, capitalize $x[i]$, if $h[i] > 7$.

```
Keccak(0xD3EEC7A7C7867B9D02248B414F08609A2A6C2AB4) =
        e1254679f8029ef7c379d60b9a49f4489fb4c872...
```

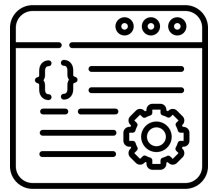
```
0xD3eec7a7C7867B9d02248b414F08609A2A6c2Ab4
```

Different Types of Accounts



Externally Owned Accounts

- Address
- Balance
- Nonce (# of transactions)



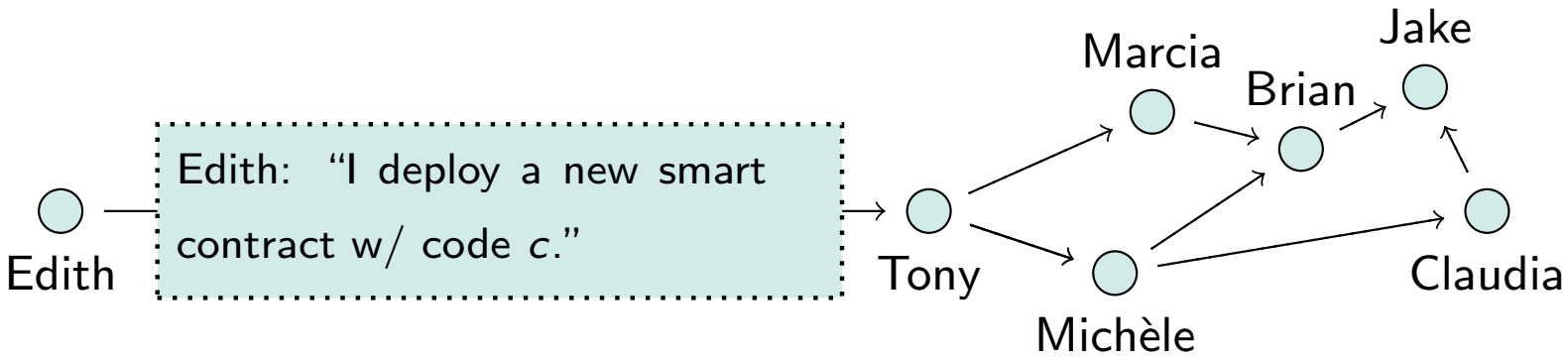
Contract Accounts

- Address
- Balance
- Nonce (# of accounts created)
- Contract code
- Contract storage

Control Over Accounts

Externally Owned Accounts are controlled by a private key. Contract accounts are controlled by the contract code.

How Contract Accounts are created



1. EOA (or CA) issues (internal) transaction with **zero address** as recipient address.
2. The transaction carries the contract code.
3. Transaction confirmation = contract deployment
4. New contract account address = `SHA3.256(address_sender, nonce)`

Account-based Model vs. UTXO-based Model

The account-based model is more intuitive and flexible than the UTXO-based model, but it also has some drawbacks.

Privacy Concerns

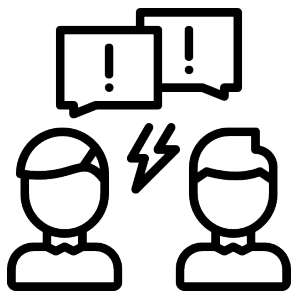
People usually use the same address for many transactions - in contrast to UTXO-based systems where you use a new address for each transaction.

↗ Etherscan

Account-based vs. UTXO-based Model

Transactions getting stuck

If a transaction from an account has not been confirmed yet, subsequent ones (higher nonce) from the same account will not be confirmed.



Unintended Conflicts

If a user uses two or more wallets for the same address and the two are not in sync, two transactions A and B may be conflicting, even if the total balance of the account is larger than the aggregated amount sent with transactions A and B (same Problem with SC).

Creating and Funding A Metamask Account

Exercise 1:

1. Go to `https://metamask.io/` and get the Metamask Browser extension.
2. Set up a new account and connect to Goerli testnet.
3. Backup the mnemonic phrase of your wallet.
4. Choose a Goerli faucet from this list and get some Goerli testnet Ether.

If you are currently managing Mainnet funds with Metamask, make sure **NOT** to use the same mnemonic seed for your testnet transactions. In doing so you will put your ETH and tokens at risk.

Creating and Funding A Metamask Account

Exercise 2:

1. Create a second account with metamask.
2. Transfer 0.1 Goerli ETH from your first account to your second account.
3. Check the transaction details on ↗ Goerli Etherscan.
4. Transfer the balance from your second account, back to your first account.