openHPI Course: Blockchain – Revealing the Myth

Ethereum Solution

Prof. Dr. Christoph Meinel

Tatiana Gayvoronskaya

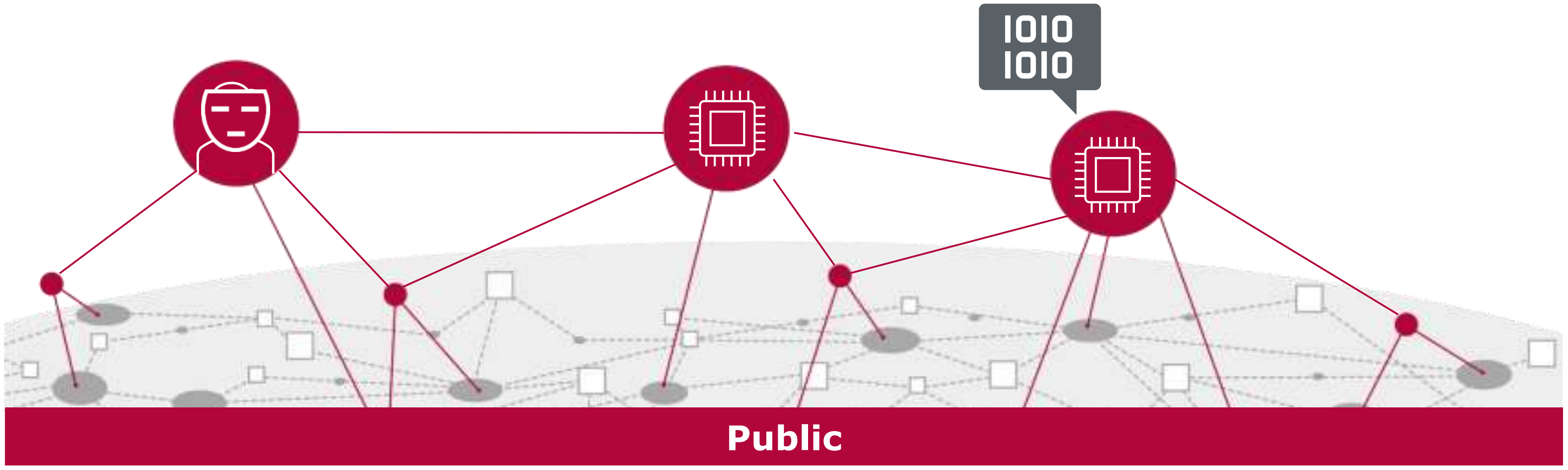
Hasso Plattner Institute
University of Potsdam, Germany

Ethereum
Beginning of a New Blockchain Generation

- 2014 was the **birth year of Ethereum** and thus the beginning of a **new generation** of blockchain-based projects
- This new generation not only expands the possibilities of blockchain technology, but also **redefines its meaning**
- In contrast to Bitcoin technology, Ethereum deals with the **current state of an account**
- This gives the new generation the name – **account-based blockchain solutions**

Ethereum
Two Types of Accounts

- Ethereum system has **two types** of **accounts**:
  - **external** accounts
  - **internal** accounts
- **External accounts** are comparable to a bank account and **belong to the users** of the Ethereum system
- **Internal accounts** are assigned to the **autonomous objects** (so-called smart contracts)



Ethereum
External Account

- **External accounts** have
  - an "account number," more specifically an **address** (comparable to the Bitcoin address) and
  - **information** about the **balance** and **transactions** that have been made via the address
- **Users** of the Ethereum system can "**transfer**" Ethereum coins **(Ether)** by **transactions** to other users or **activate internal accounts**

Ethereum
Internal Account – Autonomous Agents

- **Internal accounts** or “**smart contracts**” are **far more than cryptographic “boxes”** with specific values that can **only be unlocked if certain conditions are met**
- **Smart contracts** can better be described as **“autonomous agents”** that exist within the Ethereum system
- They have **accounts** as users do and account numbers – more specifically **addresses**
- **Autonomous agents** have control over their **own contents**, e.g., over the values they contain, conditions, and the Ether balance that can be used for system-dependent fees

Ethereum
Internal Account – Controlled by Code

- Internal accounts are **controlled** by their **contract code**
- **Smart contracts** always **execute a certain part of their source code** if they are "triggered" by a **special message** from another smart contract or a user through a **transaction**
- The code can implement **any rules and conditions** and thus allow complex applications, so-called decentralized applications or **DApps** for short
- DApps run **without any central "coordinator"** in a specially created environment on computers of all **full nodes**:

  → **Ethereum Virtual Machine (EVM)**
- Network of these "**autonomous agents**" together with user accounts form a **censorship-resistant, decentralized world computer**

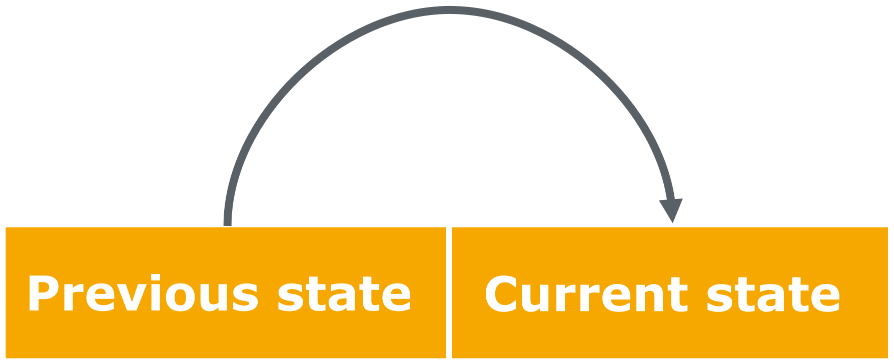
Ethereum
Account Content

- In general, both Ethereum accounts contain four fields:
  - **nonce** – a counter used to make sure each transaction can only be processed once
  - account's current Ether **balance**
  - account's **contract code** (empty by external accounts)
  - account's **storage** (empty by external accounts)
- If the **source code** of a smart contract account is **activated** by a message or a transaction, it can **access the internal storage** to **read and write**, to **send messages** to other smart contracts, or to **create new smart contracts**

Nonce	**Balance**	**Code**	**Storage**

Ethereum
Transaction as a Bridge Between Two States

- **Current state of an account**, like the current state of UTXO in Bitcoin-based systems, is **updated by a transaction**
- In this way, a transaction represents a **bridge** – a valid transition **between two states** – the previous one and the new one
- **Transaction structure** in the Ethereum system is significantly **more complex** compared to the Bitcoin system



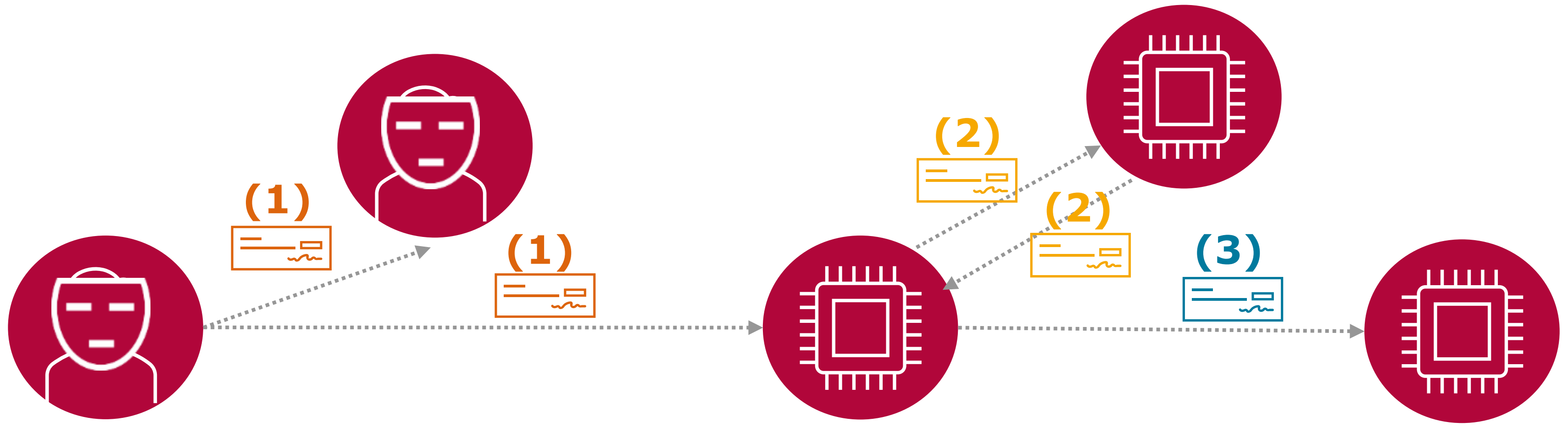
Ethereum
Types of Transactions in Ethereum

In the Ethereum system **two types of transactions** are distinguished:

(1) Transactions that are **"exchanged" between the accounts**

- transactions that are **made by external accounts (1)** and
- so-called messages that are **exchanged between the internal accounts (2)**

(2) Transactions that are **used to create new smart contracts** (contract creation transactions) **(3)**



Ethereum
Transaction Content

An **Ethereum transaction** consists of the following:

- **Nonce** – value that corresponds to the number of transactions carried out by the sender
- **Recipient address** – In case of a contract creation transaction, this field is empty
- **Value** – amount of Ether to be transferred from sender to recipient. In case of a contract creation transaction, this is the amount of Ether for the newly created smart contract account
- **Data** – used **for signing the transaction** and to determine the sender of transaction
- **Smart contract code** – for contract creation transaction
- **Data for a message** – transactions that are exchanged between the smart contracts
- And two specific values – **gasPrice** and **gasLimit**

Ethereum
gasPrice

gasPrice – fee charged in the Ethereum system for **every calculation step in a smart contract**

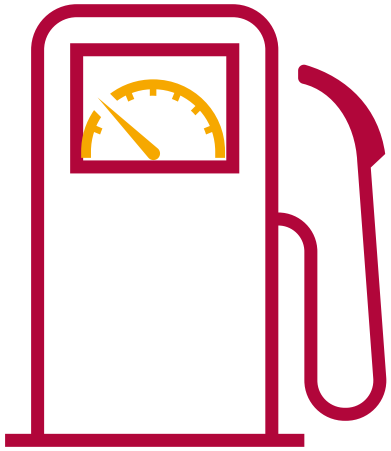
- Imposed for protection against denial-of-service attacks
- Whereby every user, including attackers, should pay for every resource that is used (including calculation, bandwidth, storage)
- Fee is measured in units of **gas** and paid in Ether
- gasPrice is noted in determining the cost to execute the transaction

1 gwei = 0.000000001 ETH

Ethereum
gasLimit

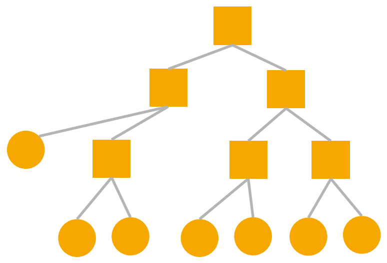
gasLimit – Value corresponding to the **maximum amount of gas** that is to be used in **executing the transaction**

- Paid in advance before any calculation is made; cannot be increased later
- gasLimit is used to avoid accidental or other calculation problems in the code
- For this reason, in every transaction a limit is set on the number of calculation steps that can be carried out in the code



Ethereum
Block

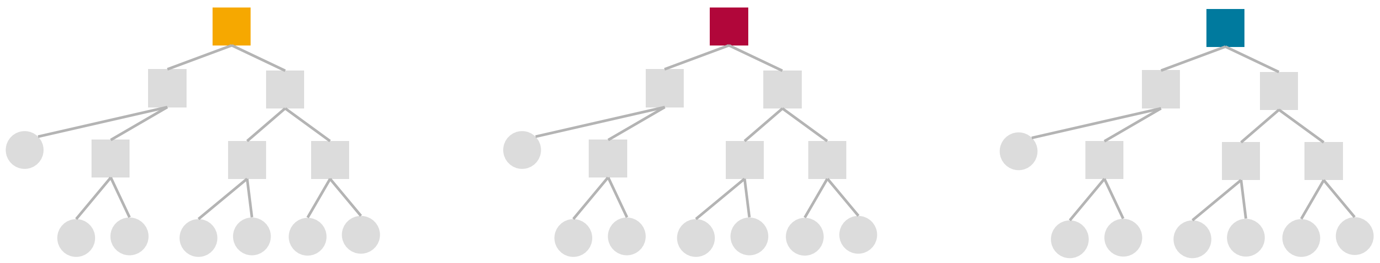
- In comparison to the Bitcoin block header, which consists of **six entries**, there are **fifteen entries** stored in an **Ethereum block header**
- This confirms once again the **higher complexity of the Ethereum** system compared to the Bitcoin system
- Ethereum system uses an advanced technology for a cryptographically authenticated data structure, namely the **Merkle Patricia Tree**
- This structure enables a **fast search** for contents, is **easy to implement**, and needs **little storage space**



Ethereum
Merkle Patricia Tree

Ethereum block header consist of three Merkle Patricia Roots:

- **transactionsRoot** – Merkle Patricia Root of the transaction list
- **stateRoot** – root of the states. There is a global state tree that is updated over time
- **receiptsRoot** – root of the receipts. In the Ethereum system a receipt is created for each transaction that contains the specific information regarding its execution



Summary

- We have once again shown the complexity of a secure system for electronic transactions without relying on trust
- The implementation of blockchain technology is now moved
  - from the area of cryptocurrency or decentralized registry
  - to a programmable, decentralized trust infrastructure

Recommended Literature and References

Recommended literature:

- For a **broad overview** of the **aims and objectives** of Ethereum, we would recommend the following exciting explanation by Vitalik Buterin
- For more detailed information about ethereum **in the context of the big picture of blockchain**, we would recommend our book "Blockchain: Hype or Innovation"

References:

- V. Buterin, Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform, (2013)