chain that is longer than the chain containing the original transaction. If she manages to do that, the original transaction is considered invalid because the miners will always work on the longest chain. To protect against such an attack, the recipient of a transaction should wait for at least 6 confirmations until she accepts the payment. The longer she waits, the more unlikey it is that a 51 percent attack succeeds [13] [27].
In the second case the attacker can ignore the transactions to a specific Bitcoin address by not including them in her blocks. When another miner includes transactions for the address, she can fork the blockchain and invalidate these blocks as long as she has the majority of the mining power in the network. [13]
A majority attack is also possible with less than 51 percent of the hashing power, but the probability that an attack succeeds is reduced. Although it is not possible for a single miner to control even 1 percent of the network, centralized mining pools offer a high risk for for-profit attacks by a mining pool operator [13].

3.1.5 Limitations of Bitcoin

In addition to simple monetary transactions, cryptocurrency networks can in theory be used to store arbitrary data and perform arbitrary state transitions and thus enable more complex and powerful applications such as Smart Contracts or Distributed Anonymous Organizations (DAOs) [27].
There have been several attempts to retrofit the simple scripting language of Bitcoin to implement various smart contracts. However, since this scripting system has a low expressiveness and is not touring complete, many tasks are computationally very expensive to perform or not even possible. It is, therefore, preferable to use a general purpose smart contract scripting language. The Ethereum Virtual Machine (EVM), which is used in this application, is the first system that incorporates such capabilities [31].

3.2 Ethereum

Ethereum is a distributed computing platform that uses a blockchain to store not only the state of user accounts, but also program code and its associated state. It allows for the distributed execution of arbitrary code and provides a suitable platform for the development of smart contracts discussed further in section 3.3. Ethereum was specifically designed to allow anyone to write smart contracts and decentralized applications. It supports several scripting languages that can be compiled to byte code that is executed on the Ethereum Virtual Machine (EVM). [27][37].
As of May 2017, Ethereum is the second largest cryptocurrency with a market capitalization of over 8 billion US-Dollars [30].

The following subsections discuss the most important aspects of Ethereum.

3.2.1 Accounts

An account is an object that stores the state of a users account balance or of a contract. The first account is called an externally owned account (EOA), because it belongs to an external entity and is controlled by a private key. It can send messages by creating and signing transactions with its private key [27][37].
The second account is called a contract account and its state is controlled by the contract code. When a contract account receives a message, its code is executed. It can also send messages to other contracts or create new contracts [27][37].
The EVM does not differentiate between the 2 account types. Every account has a key-value store called a storage and a balance in wei that can be changed by sending transactions that include ether [37].

3.2.2 Transactions and messages

Transactions A transaction is a data package that is signed by an externally owned account. It contains the sender of the transaction, the recipient of the transaction, the amount of ether sent, an optional data field, a gas limit and a gas price field [27].

Messages A message is like a transaction that is sent from one contract to another contract. It behaves similar to an ordinary function call in other programming languages. Every time the running code of a contract account executes the “call” opcode, a message is generated and sent to the recipient contract or externally owned account. Both, transactions and calls, can be used to create new contracts, to invoke functions of a contract or to transfer ether to a contract or to an externally owned account [27] [37].

Ether Ether is the currency token used by Ethereum to pay for transaction fees. Developers have to provide ether when they deploy contract code to the blockchain and users have to spend or burn ether when they invoke transactions on a contract. Ether can be exchanged against other fiat currencies via various cryptocurrency exchanges [27][37].
Ether can be divided further into smaller units, the smallest among them is called wei. One ether is equal to 10^{18} wei [27][37].

Gas Gas is used to pay transaction fees in Ethereum. Gas is not a currency, but an internal pricing unit that decouples the market price of the ether from the cost of transactions. 1 unit of gas represents the price for the most simple operation executed on the EVM. The price for one unit of gas, the gas price, can be dynamically adjusted to adapt to fluctuating values of the ether currency. The gas price can be defined by the originator of a transaction and miners decide whether they want to include the transactions in their blocks or not. [37].
The gas limit or startgas value of a transaction determines how many computational steps a transaction is allowed to execute. The more lines of code a contract executes and the more memory and storage it uses, the higher the gas limit of the initial transaction needs

to be [37].
Transaction fees and the gas limit help to prevent the execution of faulty code, like infinite loops and they help to save computational resources on the network. The gas limit also disincentivises potential denial-of-service attacks on the network [27][37].

3.2.3 Blockchain and mining

The Ethereum blockchain is very similar to Bitcoin. The most important difference is the fact, that a block not only contains a list of transactions but also the whole state of the network. The state is stored in a data structure called "Patricia Tree" [27].
The Patricia Tree is a modified Merkle Tree that is optimized for the insertion and deletion of nodes. It stores the state of all contract and externally owned accounts. Every block stores a reference to the root of the tree and updates only the parts that changed because of the effects of the transactions in that block. This allows new nodes to only download the Patricia Tree instead of all blocks to retrieve the state of all accounts and therefore saves a considerable amount of disk space. It is estimated that if this concept would be applied to Bitcoin, it would require a node to store between 5 and 20 times less data [27].
Ethereum also uses a different Proof-of-work algorithm, called Ethash, that produces a block every 12 seconds in average compared to 10 minutes in Bitcoin. This has the advantage, that transactions can be processed faster and a recipient of a transaction does not have to wait long until she can consider a transaction to be safe. It also increases the interactivity of applications that interact with contracts on the blockchain. Further, Ethash is memory hard and therefore ASIC resistant [37].
A negative effect of the fast block time is that the stale rate, the rate at which blocks that are not part of the main chain are produced, is increased. This is a security risk that can lead to centralization to mining pools since many miners will not be rewarded for their effort in mining new blocks. While in Bitcoin, such blocks are considered "orphan" and are no longer used, the GHOST protocol of Ethereum also allows for the inclusion of such "uncle" blocks and rewards the miners of them [27].
In contrast to Bitcoin, the mining reward for a block is static and exactly 5.0 ether. Successful miners also collect all gas that is used in the transactions of a block. Miners of "uncle" blocks receive 7/8 of the static block reward [37].
The total amount of ether issued in a year is statically bounded to 1/3 of the pre-sale, which is approximately 18 million ether. It is estimated that approximately 1% of the total monetary base is lost every year due to the death of key owners, lost of private keys or transactions to empty addresses. Therefore the supply of ether grows at a disinflationary rate until the rate of annual loss and destruction of ether will balance the rate of issuance and the currency no longer grows [27].

3.2.4 Light clients

A protocol for fully or partially light clients in the Ethereum network is still under development As of April 2017. However, fully and partially light clients will play an important role in the future.

A partially light client is a client that verifies every block but stores almost nothing on its hard disk. Instead it uses DHT GET requests to access blocks from other nodes and validates them locally. The goal is to use almost only partially light clients in the future that store nothing but the last few thousand blocks [35].

A fully light client does not process most transactions. It can access transactions and states by recursively downloading nodes from the Merkle Tree of a block or from the Patricia Tree of the network. For example, if a light client wants to know the state of an account, it requests the root of the Patricia Tree from a source and then downloads nodes recursively until it arrives at the desired value. If it wants to check for a specific transaction, it can request the block number and index for it and recursively search for the transaction in the Merkle Tree of that block. Fully light clients will also be able to collectively validate blocks and watch for specific events that took place on the blockchain. The latter functionality is useful to observe the state of a contract on the blockchain. All operations except the event watching have a complexity of $\mathcal{O}(\log n)$, which makes them highly efficient and suitable for smart phones and other hardware constraint devices [35].

3.3 Ethereum Smart contracts

A smart contract is a special account that stores executable code together with its associated data and an account balance on the blockchain. Smart contracts have an address (a public key) and are created by transactions. Transactions are also used to interact with a contract on the blockchain by sending money to its account balance or by executing code. To execute the code of a contract, a function call containing the functions name and its parameters is binary encoded and sent to the contract in the data field of the transaction [31] [37].

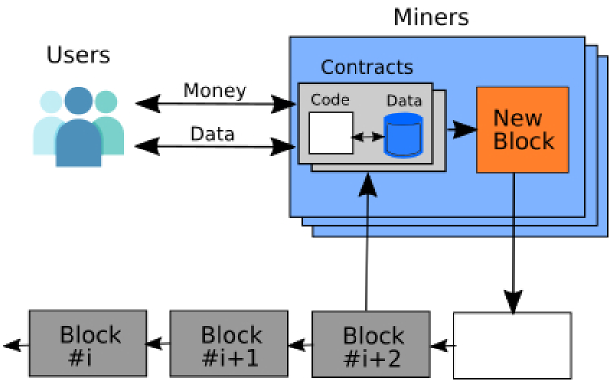

Figure 3.5: Execution of a smart contract on the blockchain.

Figure 3.5 illustrates the interaction of externally owned accounts (users) with a contract. Every time a contract receives a message from another contract or a transaction from a user, it can receive ether or execute a function that is specified in the data field. In the same way, the contract can send money from its balance to other accounts or execute functions on other contracts through broadcasting of messages [31].

Execution of the code takes place on all mining nodes in the network concurrently which reach consensus over the new state of the contract using a proof-of-work algorithm. The persistent variables of a contract are stored on the storage, a key-value store associated with the contract that is persisted on the blockchain. Access to the storage is very expensive (20000 units of gas per 256 bit word) because it has to be stored on every full node in the network. Intermediate results of computations are stored in the memory, a non-persistent byte-array. The state as well as the code of a contract are public and the code of a contract cannot change retrospectively [27] [37].

3.3.1 Solidity

Ethereum supports different script languages for writing smart contract code. The most popular among them is Solidity, a contract oriented high-level language with a static type system that has a syntax very similar to javascript [40].

The main construct in Solidity is the contract. Similar to a class, a contract can contain fields, functions, function modifiers, struct types and enum types and it is also inheritable.[40].

The following subsections briefly describe the most important concepts of a contract in