Problem in a network is that each machine has its own state

Blockchain is a mixture of network and a database

If one thing changes in one place it changes everywhere

It’s a book of transactions (blocks) and its append only

That is why the blockchain is deterministic

The current value of a wallet is starting 0 plus all transactions that are in/out. Therefore we need to be fully synched to work with them.

Ethereum adds a virtual machine that can execute code. In each transaction I can add some data which can be a code that perform logic (smart contracts) as well as input/output to functions that change the state. A smart contract main role is to update the state of the blockchain. This logic is known as smart contracts and that is what we code. Therefore programs are immutable, they never change.

Note that "contracts" in Ethereum should not be seen as something that should be "fulfilled" or "complied with"; rather, they are more like "autonomous agents" that live inside of the Ethereum execution environment, always executing a specific piece of code when "poked" by a message or transaction, and having direct control over their own ether balance and their own key/value store to keep track of persistent variables.

**Ethereum Accounts**

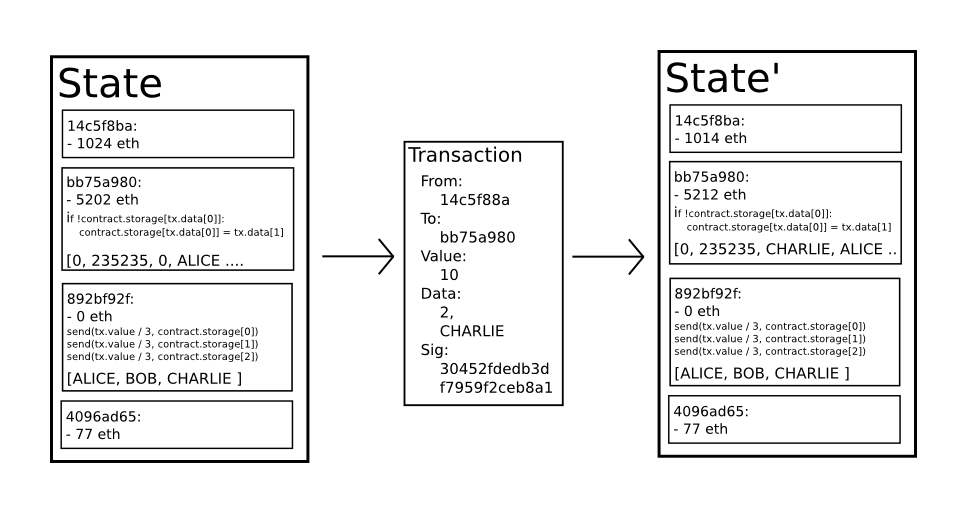
In Ethereum, the state is made up of objects called "accounts", with each account having a 20-byte address and state transitions being direct transfers of value and information between accounts. An Ethereum account contains four fields:

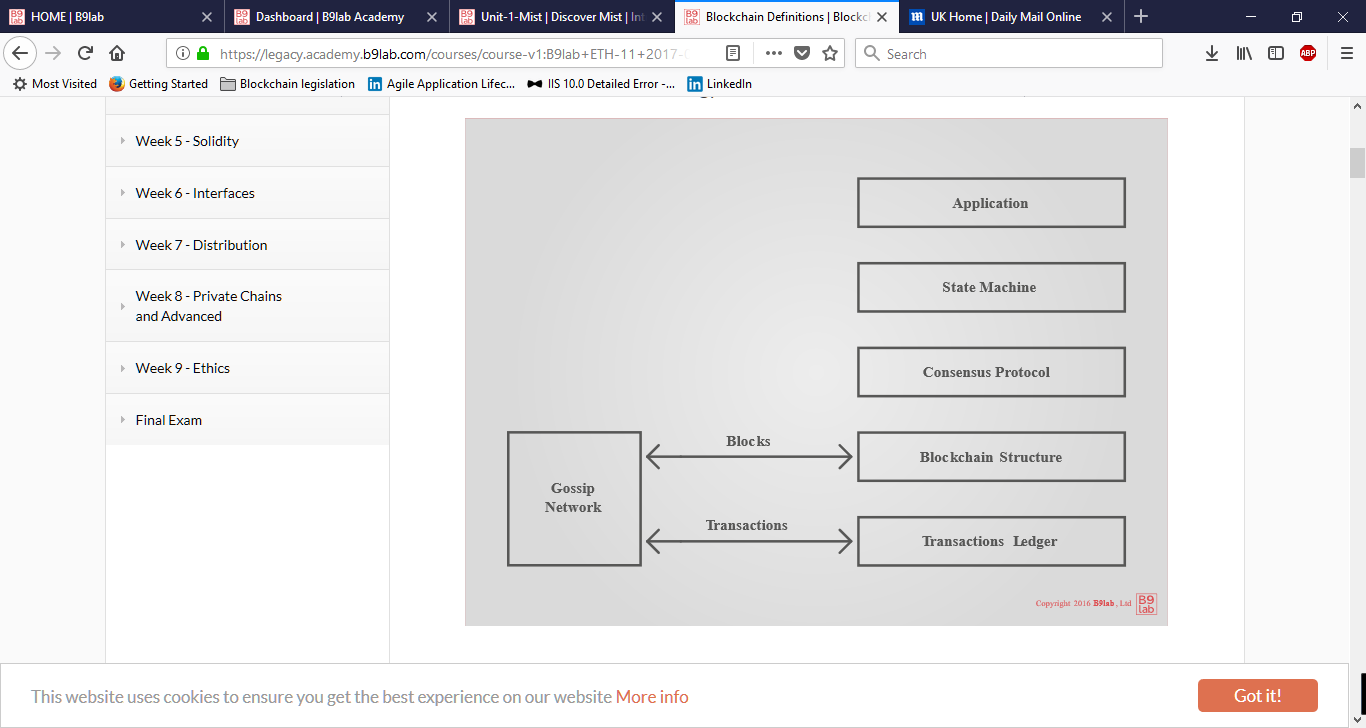
• The nonce, a counter used to make sure each transaction can only be processed once

• The account's current ether balance

• The account's contract code, if present

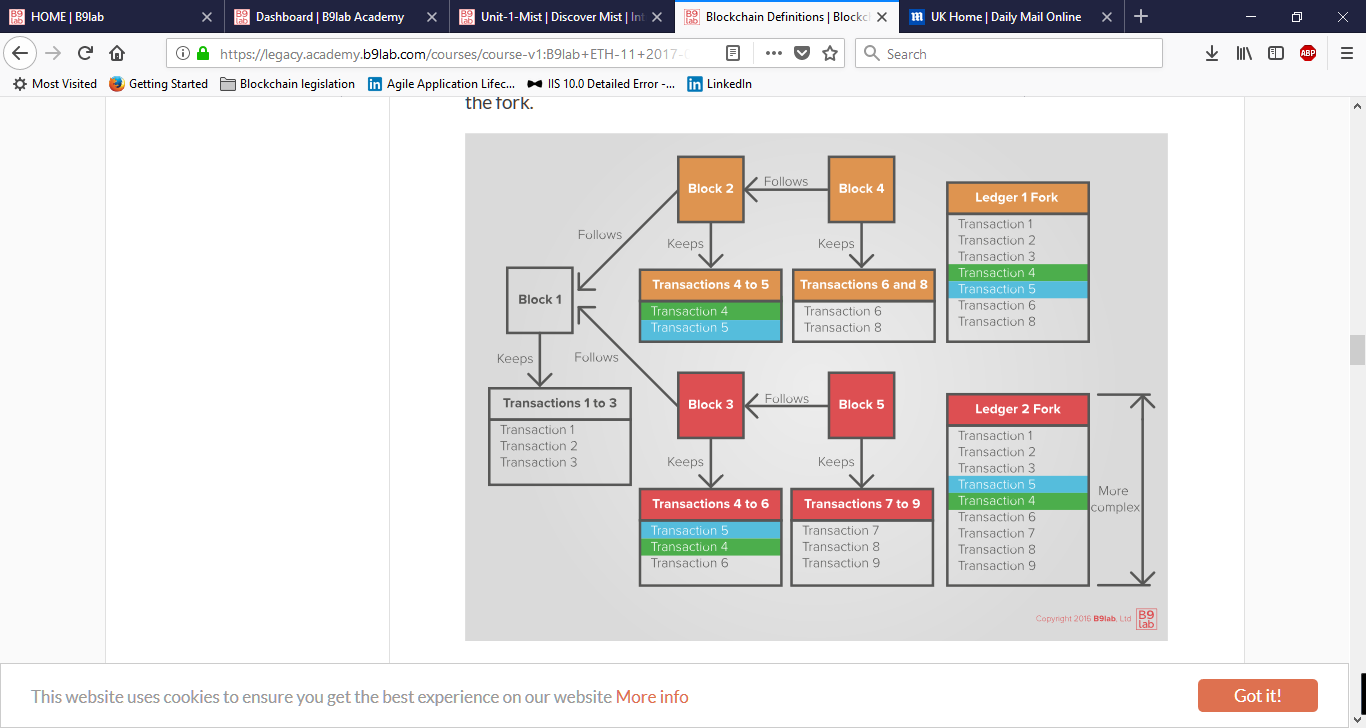
• The account's storage (empty by default)



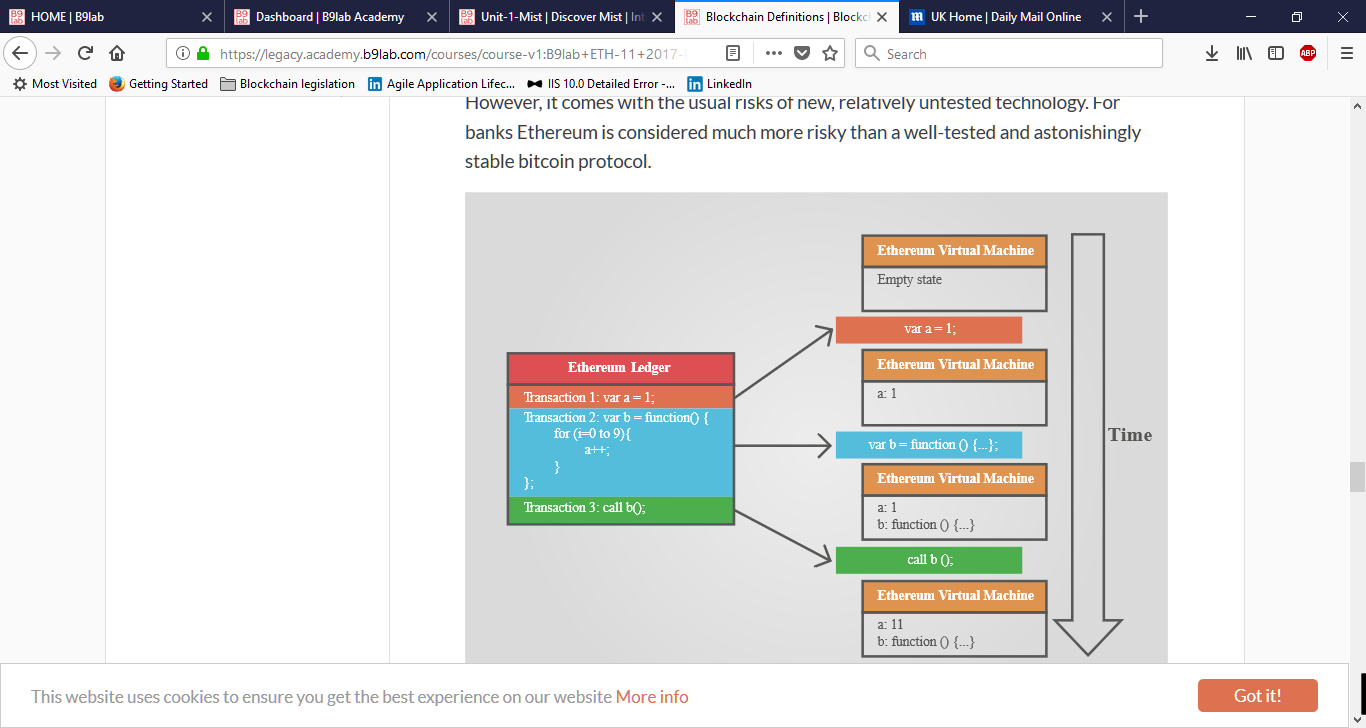


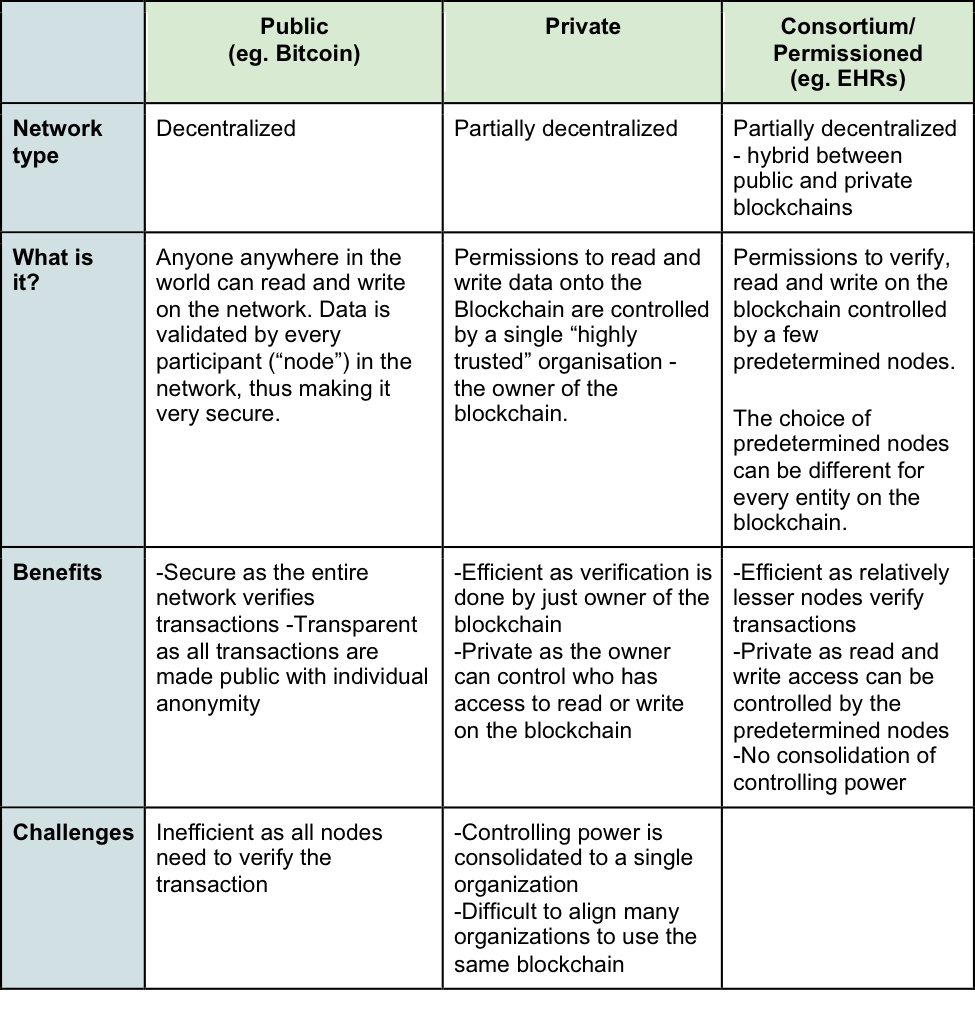
A transaction is an atomic event, an event whose sub-event parts make no sense in isolation.

**Forking**



Ledger 1 Fork is known as uncle





<https://github.com/ethereum/wiki/wiki/White-Paper>

<http://gavwood.com/Paper.pdf>