**How Ethereum Works**

Now that we've covered what ethereum is, let's dive deeper into how the platform functions under the hood.

Consider the online notebook application described in "[What is Ethereum?](https://www.coindesk.com/information/what-is-ethereum/)"

Using ethereum, the app doesn't require one entity to store and control its data. To accomplish this, ethereum borrows heavily from bitcoin's protocol and its blockchain design, but tweaks it to support applications beyond money.

Ethereum aims to abstract away bitcoin's design, however, so that developers can create applications or agreements that have additional steps, new rules of ownership, alternative transaction formats or different ways to transfer state.

The goal of ethereum's 'Turing-complete' programming language is to allow developers to write more programs in which blockchain transactions could govern and automate specific outcomes.

This flexibility is perhaps ethereum's primary innovation, as explained in the guide "[How Ethereum Smart Contracts Work](https://www.coindesk.com/information/how-do-ethereum-smart-contracts-work/)".

**The ethereum blockchain**

The structure of the ethereum blockchain is very similar to bitcoin's, in that it is a shared record of the entire transaction history. Every node on the network stores a copy of this history.

The big difference with ethereum is that its nodes store the most recent state of each smart contract, in addition to all of the ether transactions. (This is much more complicated than described, but the text below should help you get your feet wet.)

For each ethereum application, the network needs to keep track of the 'state', or the current information of all of these applications, including each user's balance, all the smart contract code and where it's all stored.

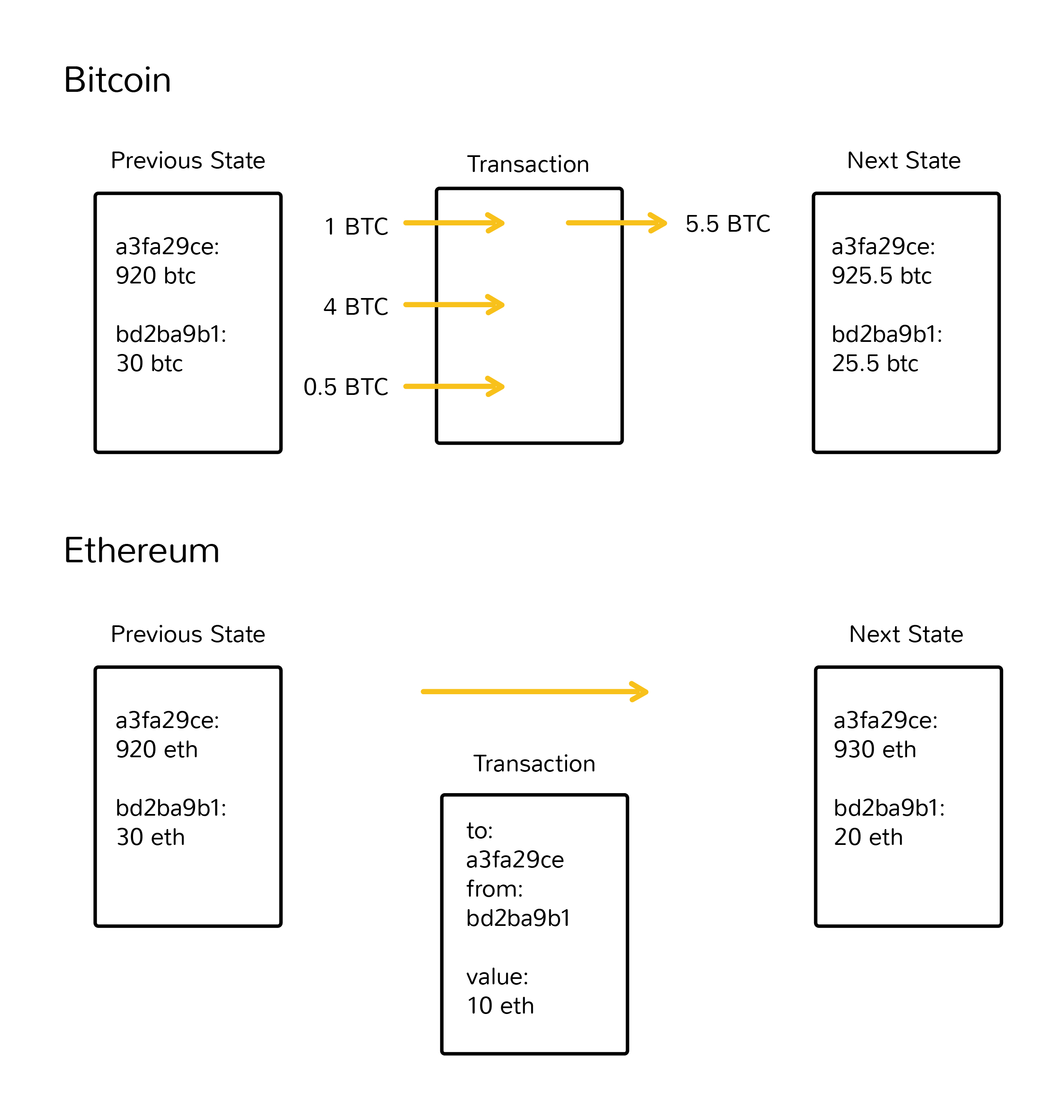
Bitcoin uses unspent transaction outputs to track who has how much bitcoin.

While it sounds more complex, the idea is fairly simple. Every time a bitcoin transaction is made, the network 'breaks' the total amount as if it was paper money, issuing back bitcoins in a way that makes the data behave similarly to physical coins or change.

To make future transactions, the bitcoin network must add up all your pieces of change, which are classed as either 'spent' or 'unspent'.

Ethereum, on the other hand, uses accounts.

Like bank account funds, ether tokens appear in a wallet, and can be ported (so to speak) to another account. Funds are always somewhere, yet don’t have what you might call a continued relationship.

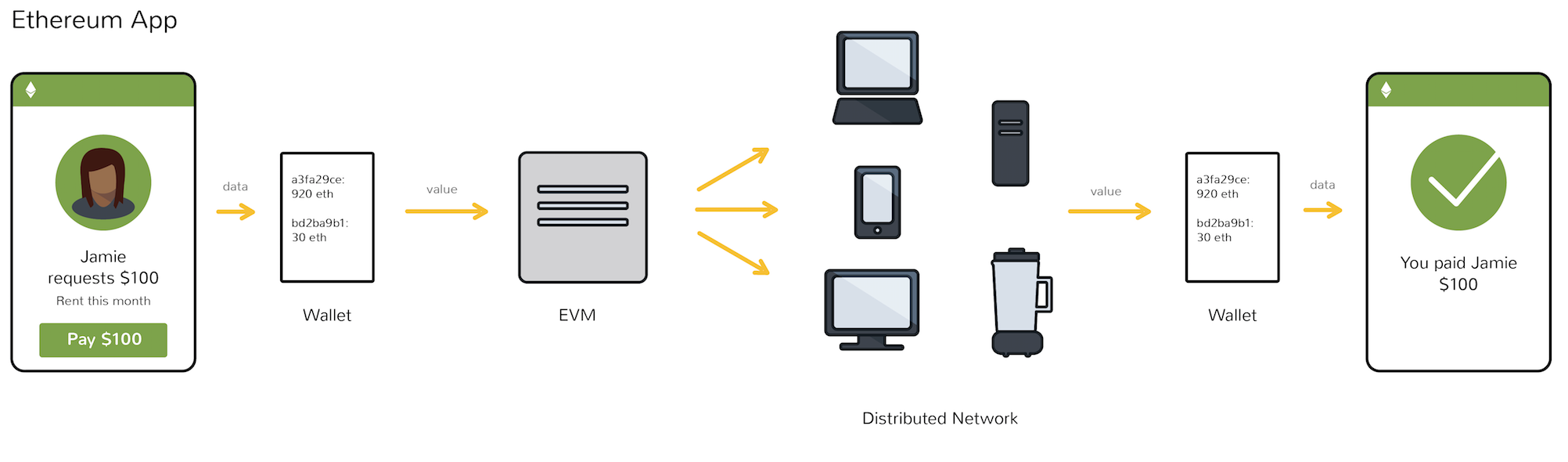


**What is the ethereum virtual machine?**

With ethereum, every time a program is used, a network of thousands of computers processes it.

Contracts written in a smart contract-specific programming languages are compiled into 'bytecode', which a feature called the 'ethereum virtual machine' (EVM) can read and execute.

All the nodes execute this contract using their EVMs.



Remember that every node in the network holds a copy of the transaction and smart contract history of the network, in addition to keeping track of the current 'state'. Every time a user performs some action, all of the nodes on the network need to come to agreement that this change took place.

The goal here is for the network of miners and nodes to take responsibility for transferring the shift from state to state, rather than some authority such as PayPal or a bank. Bitcoin miners validate the shift of ownership of bitcoins from one person to another. The EVM executes a contract with whatever rules the developer initially programmed.

Actual computation on the EVM is achieved through a stack-based bytecode language (the ones and zeroes that a machine can read), but developers can write smart contracts in high-level languages such as Solidity and Serpent that are easier for humans to read and write.

As explained in our guide "[How Ethereum Mining Works](https://www.coindesk.com/information/how-ethereum-mining-works/)", miners are the ones that are preventing bad behavior – like ensuring that no one is spending their money more than once and rejecting smart contracts that haven’t been paid for.

There are a few thousand [ethereum nodes](https://www.ethernodes.org/network/1" \t "_blank) out there, and every node is compiling and executing the same code.

But, you might be thinking, isn't that much more expensive than a normal computation? Yes, it is. That's why the network might only be used only for particular use cases.

The official ethereum [dev tutorial](https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial) concedes this inefficiency, stating:

*"Roughly, a good heuristic to use is that you will not be able to do anything on the EVM that you cannot do on a smartphone from 1999."*