3 Ethereum

In 2013, then 19-year-old Vitalik Buterin published a white paper containing the specification of Ethereum, a new and improved blockchain. Ethereum contains a fully-fledged Turing-complete programming language. "Contracts" are programmed to encode arbitrary state transition functions on the blockchain. Vitalik dropped out of the University of Waterloo after gaining a Thiel Fellowship of 100K USD. Soon after, he raised 18 million USD to support the development of Ethereum. By 2015, Ethereum had launched and it has been growing ever since. At the time of writing the market capitalization of its native cryptocurrency has crossed 15 billion USD.

3.1 ETHER

The native cryptocurrency of Ethereum is called Ether (symbol ETH or Ξ). It is similar to Bitcoin in that it can be transferred to other people or kept as a store of value. At the time of writing 1 ETH is worth about USD$136. Even so, its purpose is different: if Bitcoin is digital gold, ether is more like digital oil.

Ether is designed to fuel the network and incentivize people to share their disk space and computing time with the blockchain. When a user makes a transfer or wants to do a computation, she pays the miners Ether to compensate them for their resource consumption. In a similar vein, when developers build software that uses shared disk space, it will demand Ether to store data.

As an operation consumes resources, it uses up units of **gas**. Similar to Bitcoin's transaction fee, we must specify at which price we value the gas that we will use (the **gas price**). This is usually expressed in small denominations of Ether called Gwei:

Unit	Short	Wei	Ether
Wei	(wei)	1 wei	10^{-18} ETH
KWei	(babbage)	10^{3} wei	10^{-15} ETH
Mwei	(lovelace)	10^{6} wei	10^{-12} ETH
Gwei	(shannon)	10^{9} wei	10^{-9} ETH
Twei	(szabo)	10^{12} wei	10^{-6} ETH
Pwei	(finny)	10^{15} wei	10^{-3} ETH
Ether	(buterin)	10^{18} wei	1 ETH

Setting a high price will ensure that your transaction will be moved to the front of the queue. Since we are executing variable computer programs on the blockchain, it is often not trivial to guess how many operations/units of gas a program will cost. A user must, however, set an upper limit (the **gas limit**) of the amount of total gas she is willing to pay for an operation to succeed. This avoids unpleasant surprises where a program gets stuck and uses enormous amounts of gas, costing the user a fortune. The maximum amount of money ever spent on a transaction is:

$$\text{Fee} = \text{GasPrice} \times \text{GasLimit}$$

.

3.2 ETHEREUM AS A GLOBAL COMPUTER

As with Bitcoin, the Ethereum blockchain consists of nodes and mining computers connected together that make up the blockchain. When a user creates a transaction to interact with the blockchain, that transaction must contain ether and can optionally also include data/code to be

run. That code could be a fully-fledged piece of software or, more often, a "call" (command to execute) to another piece of code that was placed on the blockchain earlier by someone else.

ETHEREUM VIRTUAL MACHINE

When a user calls a piece of code, that code is run by the computers on the blockchain and the resulting state transition (e.g., a transfer of assets) is recorded on the blockchain. The Ethereum blockchain is, in essence, therefore one large clunky virtual machine. It is inefficient in that the computations are repeated by various devices in parallel so you would not want to use it to run games. Instead, you want to use it for computations that require trustworthiness, traceability, transparency, and uncensorableness. This makes Ethereum software ideal for anything related to the management of assets, whether those assets are digital currencies, art, licenses, or anything else that can be represented digitally.

SMART CONTRACTS

What it is

An "off-chain" contract is a written or spoken agreement that is intended to be enforceable by law. It requires a trusted entity such as a lawyer or a judge. More often than not, it requires good faith in the person you are interacting with as going to court is a lengthy and costly process in most countries. Let us look at the definition of smart contracts (as envisioned by Nick Szabo in "The Idea of Smart Contracts" long before Ethereum was created):

> A smart contract is a **computerized transaction protocol that executes the terms of a contract**. The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), **minimize exceptions** both malicious and accidental, and **minimize the need for trusted intermediaries**. Related economic goals include lowering fraud.

In short, a smart contract is code that facilitates, verifies, or enforces the negotiation or execution of a digital contract. It executes automatically when certain conditions or actors trigger it. When routines in the smart contract are executed it can interact with other contracts, store ether, or send ether. Smart contracts are programmed using specialized programming languages such as Solidity and Vyper. Similar to regular software, they must be compiled to EVM bytecode (machine-readable code). This bytecode (the program) can be placed on the distributed ledger. From that point onward, people or other smart contracts can interact with the program by sending transactions to it.

Example

In the following example, we are going to build a name registry. That is, we are going to associate names with Ethereum addresses so that we can transfer Eth or other assets to that name instead of a long and complicated hexadecimal Ethereum wallet address. This is similar to how we do not write IP addresses (e.g., 2a00:1450:400e:80e::200e) when we visit websites. Instead, we visit their domain names (e.g., google.com). A simple example of a smart contract implementing this idea is shown in Algorithm 3.1, it is written in Solidity[1]:

Don't worry if you don't understand exactly what this does yet - our goal here is not to become solidity programmers (yet!). The main use-case is shown in the procedure in Figure 2.

First, the smart contract as shown before is created (uploaded) using a special transaction to the Ethereum blockchain by the developer. Notice how the third line declares that this is indeed a smart contract, followed by its specification. Upon deployment, the smart contract gets

[1]Note that this code is not secure and you shouldn't use this for production purposes!

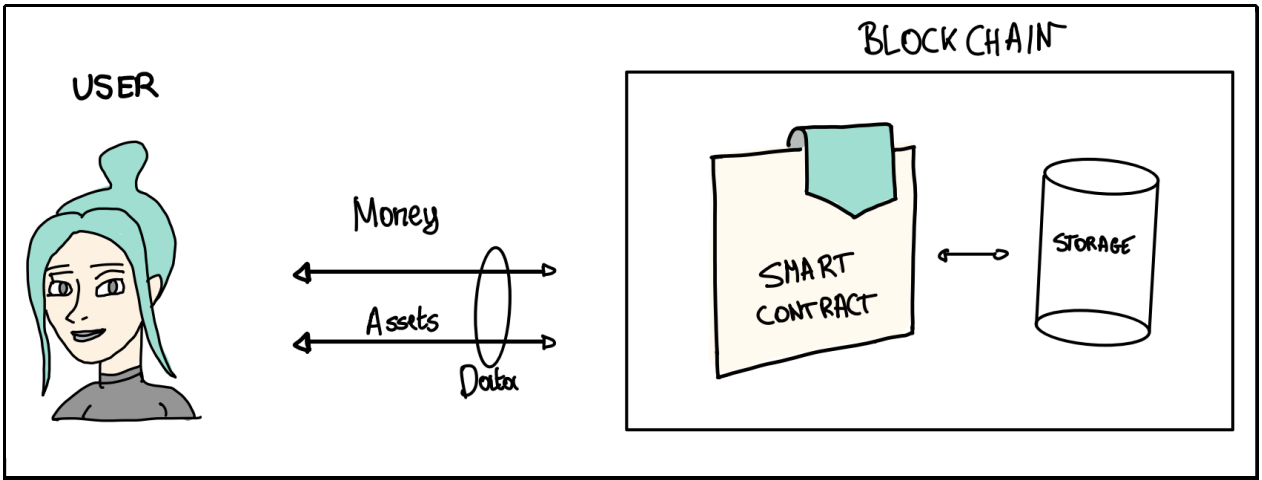

Figure 1: Users interact with the blockchain which contains data about virtual assets such as cryptocurrencies and code (smart contracts) which can act upon the assets.

Algorithm 3.1 A registry smart contract written in Solidity.

```
pragma solidity ^0.5.0;

contract MyRegistry {
  // storage on the Ethereum blockchain
  mapping( string => address ) public registry;

  function registerDomain(string memory domain, address target) public {
    //Can only reserve new unregistered domain names
    require(registry[domain] == address(0));
    // Update the registry
    registry[domain] = target;
  }
}
```

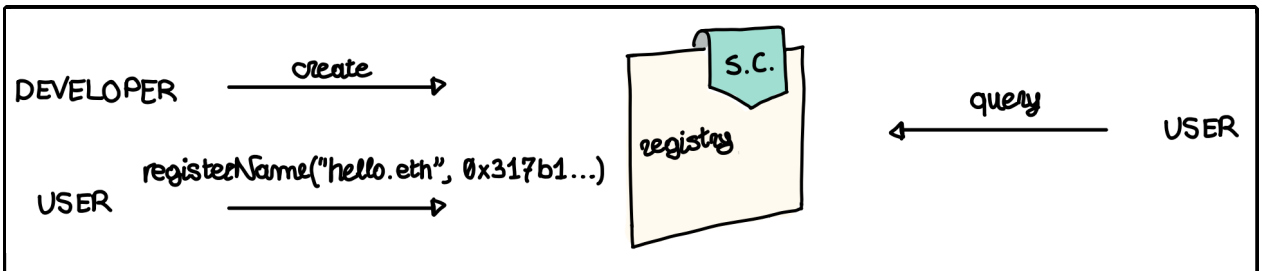

Figure 2: A registry smart contract and its usage.

its own unique Ethereum address which can be used to reference it. Reading the next lines of the contract, we see that the contract contains a "registry" which represents a list of strings (e.g., "myname.eth") and maps those to addresses (e.g., 0x1439..). This registry will be stored in the blockchain itself and can be written to through the registerDomain() function by any user. Upon doing so, the code then proceeds to update the registry to include the registered address. From that point onward, any user can inspect the registry and find the entry that was added.

3.3 WEB3

Smart-contract technology is arguably one of the most important computing innovations since the advent of the internet. We call the infusion of smart contract technology on top of the internet the "Web3" because it enables new applications thereby heralding a third generation of the world-wide-web. Let us take a step back first to see why this is so significant to warrant its own designated era.

The world-wide-web is enabled by the internet which itself emerged out of a combination of technologies - over 40 years in the making by academics and military researchers. The vast majority of protocols that underly the internet are open and freely usable by anyone with a technical background (e.g., UDP, TCP/IP). This open-source mindset is partly what enabled Tim Berners Lee to build upon these base technologies to design what would eventually become the first world-wide-web (web 1.0, the HTTP, and HTML protocols) while working for CERN in Austria around 1990. A kindred mind, Tim too choose the open-source paradigm and this, in turn, ensured an even-level playing field for anyone willing to experiment with the www (i.e., no one company controlled it). Arguably, this helped contributed to it becoming the most successful and quickly adopted technology to date in humankind.

During the mid-2000s, Web 2.0 emerged. Now, users are no longer just consumers of information, but also start producing information (e.g., by uploading blog posts or videos). During Web 2.0 large for-profit technology companies emerged which amassed incredible power through the creation of internet-enabled platforms. These platforms are essentially websites or apps which connect users with each other or with service providers (e.g., Google, Facebook, Amazon, ...). While such centralization of services yields benefits due to economies of scale and specialization, it also runs counter to the original open philosophy of the internet. Unless these platforms are able to sell you products directly, they are mostly monetizing their services by selling your data to a.o. advertizers. Within such surveillance capitalism, there are high incentives for monopolistic structures to emerge and remain closed (not share data, or protocols). To protect their status, these monopolies will buy out any threats to their status at high valuations. This is a classical pattern in business and leads to a stifling of innovation as is evident by the slowdown in innovation on the web as well. Content producers who (necessarily) decide to conform and work with existing platforms are at risk of the changing rules set out by the incumbents. For instance, it is not unheard of for Youtube content creators to suddenly get banned thereby losing their entire business. Similarly, Facebook, Apple, or Google changing rates on their app stores could very well mean the death of companies who develop and sell through them.

The defining property of Web3 is that intermediary platforms are (mostly) bypassed in favor of decentralization. It is much more a design philosophy rather than one particular technology or application itself but it is more easily understood by giving an example. The clearest win for society is the Creator Economy that Web3 unlocks. Instead of needing, say, Youtube with its ads, content creators (artists, developers, djs, musicians, writers, ...) can connect directly with their supporters. The digital content that they create can be programmed (using smart

contracts like NFTs) such that they have full control over the terms that they set without being beholden to corporate contracts. It also unlocks innovative incentivization schemes as is shown in the next example.

> **Resale Royalty** A popular blockchain-enabled commission structure allows the artists to gain from resales as opposed to just the initial sales. This can be very impactful for young creators who necessarily command low initial sales prices. In 2018, then 21-year-old Robbie Barrat sold his digital artwork for a mere $176. In 2021, it was resold for 100 Ethereum earning him a whopping $11,031 commission in the process[a]. Enforcing such a structure in the physical world is a difficult problem.
>
> [a] Transaction 0xe78acb5f60978fc4903d2f75fc9ba58d67e76a0730201cfb755aeda91826ef0e

For users, buying an audio file from an artist directly not only ensures that they are being supported the most, moreover it means that users are allowed to do with the music as they please (transfer, resell, showcase in a real or virtual home, ...). Contrast this with something like iTunes where you can only use your purchased songs within Apple's proprietary ecosystem, being barred from reselling, transferring, or lending them out. In line with an open philosophy, Web3's decentralization benefits the creators, and the consumers at the expense of intermediaries who arguably provide less value to the table. When you port these ideas to other industries like real estate, gaming, etc you start to get an idea of how impactful the concept of Web3 really can be.

	WWW	**Web 2.0**	**Web3**
Economy	Information Economy	Platform Economy	Token Economy
Characteristics	Read-Only	Read/Write	Read/Write/Execute
Monetization	Offline	Ads, Selling Products	Token Model
Payments	Offline	Credit Cards, Paypal	Ethereum, Blockchain, ..

Table 1: Key characteristics of different versions of the world-wide-web.

3.4 REFERENCES