**Blockchain Basics:**

**Bitcoin and Blockchain:**

Bitcoin is one of the first protocol to utilize the revolutionary blockchain technology. The Bitcoin Whitepaper, authored by the pseudonymous `Satoshi Nakamoto`, described how Bitcoin could facilitate peer-to-peer transactions within a decentralized network using cryptography. `Bitcoin` as a superior digital store of value, often referred to as digital gold. There is a fixed amount of Bitcoin, similar to the scarcity of gold. You can learn more about this in the [Bitcoin Whitepaper](<https://bitcoin.org/bitcoin.pdf>).

A blockchain is a digital ledger that records transactions across many computers in a secure and decentralized manner. Each block contains a number of transactions, and every new block is linked to the previous one, forming a chain.

**Ethereum and Smart Contracts:**

A few years after Bitcoin’s creation, Vitalik Buterin and others founded `Ethereum`, which builds upon the blockchain infrastructure, but with additional capabilities. With Ethereum, you can create decentralized transactions, organizations, and agreements without a centralized intermediary. This was achieved through the addition of `smart contracts`.

Though the concept of smart contracts was originally conceived in 1994 by \*\*[Nick Szabo]([https://en.wikipedia.org/wiki/Nick\_Szabo)\*\*](https://en.wikipedia.org/wiki/Nick_Szabo)**), Ethereum made it a reality.

**What is Smart Contracts?:**

Smart contracts are a set of instructions executed in a decentralized way without the need for a centralized or third party intermediary.

Smart Contract functionality is the primary difference between blockchains like `Ethereum` and `Bitcoin`. Technically `Bitcoin` does have smart contracts but they’re intentionally `turing incomplete`.

In web3, a smart contract is a self-executing contract with the terms of the agreement directly written into code. They run on blockchains and automatically execute when predetermined conditions are met, without the need for intermediaries. \_Example\_: A smart contract for an escrow service.

**The Oracle Problem:**

Smart contracts face a significant limitation – they cannot interact with or access data from the real world. This is known as the `Oracle Problem`.

Blockchains are deterministic systems, so everything happens within their ecosystem. To make smart contracts more useful and capable of handling real-world data, they need external data and computation.

Oracles serve this purpose. They are devices or services that provide data to blockchains or run external computation. To maintain decentralization, it’s necessary to use a decentralized Oracle network rather than relying on a single source. This combination of on-chain logic with off-chain data leads to `hybrid smart contracts`.

Hybrid smart contracts combine on-chain code (running on a blockchain) with off-chain data and computations provided by oracles. This allows the contracts to interact with data and systems outside their native blockchain. \_Example\_: A smart contract for insurance that uses real-world data (like weather or flight delays) provided by oracles.

Oracles in web3 are intermediaries that provide smart contracts with external data. They act as bridges between blockchains and the outside world, allowing smart contracts to execute based on real-world events and data. \_Example\_: A weather oracle provides data for a smart contract that triggers crop insurance payments based on rainfall data.

**Chainlink:**

\*\*[Chainlink]([https://chain.link/)\*\*](https://chain.link/)**) is a popular decentralized Oracle network that enables smart contracts to access external data and computation. Chainlink is also blockchain agnostic – so it’s going to work with any chain out there.

**Layer 2 Scaling Solutions:**

As blockchains grow, they face scaling issues. Layer 2, or L2, solutions have been developed to address this. L2 solutions involve other blockchains hooking into the main blockchain, essentially allowing it to scale. There are two primary types of L2 solutions:

1. Optimistic Rollups: eg. Optimism, Arbitrum

2. Zero-Knowledge Rollups: eg. ZK Sync, Polygon ZK EVM

Layer 2 solutions in web3 are technologies built on top of a blockchain (Layer 1) to improve its scalability and efficiency. These solutions handle transactions off the main chain, reducing congestion and fees, and then settle the final state on the main chain. \_Example\_: The Lightning Network for Bitcoin.

**Ethereum/EVM (Ethereum Virtual Machine):**

Ethereum is a blockchain platform known for its smart contract functionality. The Ethereum Virtual Machine (EVM) is its computation engine that executes smart contracts. Ethereum allows developers to build decentralized applications and is the basis for many web3 projects. \_Example\_: ERC-20 tokens, a standard for creating fungible tokens on Ethereum.

**Dapp (Decentralized Application):**

A Dapp is an application that runs on a decentralized network, typically a blockchain. It is powered by smart contracts and operates without a central authority. Dapps can serve various purposes, from finance to gaming. Example: Uniswap, a decentralized finance application.

**Web3:**

Web3 is a term used to describe the new paradigm of the internet powered by blockchain and smart contracts. Unlike the previous versions of the web, web3 Is permissionless and relies on decentralized networks rather than centralized servers. This ushers in an era of censorship-resistant and transparent agreements and transactions, often called an ownership economy.