**Intro to Oracles – getting real world price data**

**Introduction:**

With the rapid advancement of blockchain technology and the growing adoption of decentralized finance platforms (DeFi), the necessity to support multiple digital currencies has significantly increased. Enabling users to utilize their preferred digital currencies extends market reach and improves the usability of an application.

This lesson will walk you through adding a currency conversion feature to the `fundMe` contract and setting price thresholds with Chainlink Oracle, a decentralized network for external data.

**USD values:**

Currently, our contract will require the transaction value to be greater than one Ethereum (ETH). If we want to give the users the flexibility to spend instead 5 USD, we would need to update our contract. We can begin by specifying the new value with a state variable

`uint256 public minimumUSD = 5

at the top of the contract.

The next step would be changing the condition inside the `fund` function, to check if the `value` sent is equal to or greater than our `minimumUSD`. However, we are facing a roadblock here: the `minimumUSD` value is in USD while the transaction message value is specified in ETH.

**Decentralized Oracles:**

The USD price of assets like Ethereum cannot be derived from blockchain technology alone but it is determined by the financial markets. To obtain a correct price information, a connection between off-chain and on-chain data is necessary. This is facilitated by a decentralized Oracle network.

This blockchain limitation exists because of its deterministic nature, ensuring that all nodes univocally reach a consensus. Attempting to introduce external data into the blockchain, will disrupt this consensus, resulting in what is referred to as a smart contract connectivity issue or the Oracle problem.

For smart contracts to effectively replace traditional agreements, they must have the capability to interact with real-world data.

Relying on a centralized Oracle for data transmission is inadequate as it reintroduces potential failure points. Centralizing data sources can undermine the trust assumptions essential for the blockchain functionality. Therefore, centralized nodes are not enough for external data or computation needs. Chainlink addresses these centralization challenges by offering a decentralized Oracle Network.

Chainlink offers ready-made features that can be added to a smart contract. And we’ll address some of them:

* **Data Feeds**
* **Verifiable Random Number**
* **Automation (previously known as “Keepers”)**
* **Functions**

**Chainlink Data Feeds:**

Chainlink Data Feeds are responsible for powering over $50 billion in the DeFi world. This network of Chainlink nodes aggregates data from various exchanges and data providers, with each node independently verifying the asset price.

They aggregate this data and deliver it to a reference contract, the price feed contract, in a single transaction. Each contract will store the pricing details of a specific cryptocurrency

**Chainlink VRF:**

The Chainlink VRF (Verifiable Random Function) provides a solution for generating provably random numbers4, ensuring true fairness in applications such as NFT randomization, lotteries, and gaming. These numbers are determined off-chain, and they are immune to manipulation.

**Chainlink Automation (previously known as “Keepers”):**

Another great feature is Chainlink’s system of Keepers. These nodes in listen for specific events and, upon being triggered, automatically execute the intended actions within the calling contract.

**Chainlink Function:**

Finally, Chainlink Functions allow API calls to be made within a decentralized environment. This feature is ideal for creating innovative applications and is recommended for advanced users with a thorough understanding of Chainlink ecosystem.

**Conclusion:**

Chainlink Data Feeds will help integrate currency conversion inside of our `FundMe` contract. Chainlink’s decentralized Oracle network not only addresses the ‘Oracle problem’, but provides a suite of additional features for enhancing every dApp capabilities.