SINERGIJA UNIVERSITY
FACULTY OF COMPUTING AND INFORMATICS

USING REAL-WORLD DATA IN BLOCKCHAIN SYSTEMS

- Final Thesis -

Mentor
Dalibor Radovanović

Student
Nikola Pavlov

Bijeljina, 2024.

SINERGIJA UNIVERSITY

FACULTY OF COMPUTING AND INFORMATICS
Bijeljina , Raje Baničića

Candidate: Nikola Pavlov
Index number: 2021230012
Study department: Computing & Informatics

Thesis: Using real-world data in blockchain systems

Basic subjects:

1. **Description of the problems that blockchain systems encounter when providing "off-chain" data**
2. **Implementation of the Oracle system and how to prevent a centralized point of failure**
3. **Challenges in synchronizing "off-chain" data with a real-time blockchain system**

MENTOR:

Bijeljina, 25.10.2024..

____
Dr Dalibor Radovanović, prof.

DEAN:

____
Doc. dr Saša Adamović

USING REAL-WORLD DATA IN BLOCKCHAIN SYSTEMS

ABSTRACT:

The basic principles behind blockchain protocols, as well as their applications, are described in this paper. Different implementations of the decentralized oracle networks, as well as the importance of real-world data for the functioning and improvement of the blockchain systems, are shown through the description of the process of obtaining the real-world data. Different ways of applying such systems are shown, as well as trade-offs when choosing different solutions and the risks that come with those solutions. Determinism issues and challenges affecting data synchronization are explained.

KEYWORDS:

Blockchain, oracle, decentralization, smart contracts, data, information, protocol, token, consensus, cryptography, cryptocurrency, ethereum, program, application

KORIŠĆENJE PODATAKA IZ STVARNOG SVETA U BLOKČEJN SISTEMIMA

SAŽETAK:

U diplomskom radu su opisani osnovni principi blokčejn protokola, kao i njihova primena. Kroz opis procesa dobavljanja podataka iz stvarnog sveta, prikazane su različite implementacije decentralizovanih oracle mreža kao i važnost realnih podataka za funkcionisanje i unapređivanje blokčejn sistema. Prikazani su različiti načini primene ovakvih sistema, kao i kompromisi pri odabiru različitih rešenja, kao i rizici koji dolaze sa tim rešenjima. Objašnjeni su problemi i izazovi determinizma koji utiču na sinhronizaciju podataka.

KLJUČNE REČI:

Blokčejn, oracle, decentralizacija, pametni ugovori, podaci, informacije, protokol, token, konsenzus, kriptografija, kriptovaluta, ethereum, program, aplikacija

TABLE OF CONTENTS

1. INTRODUCTION

1.1. THE HISTORY OF DECENTRALIZED FINANCIAL SYSTEMS

October 31, 2008, the date when the "Bitcoin Whitepaper" document was first published, by a person under the pseudonym "Satoshi Nakamoto", is most often taken as the official date of the creation of blockchain technology. This document describes the technology of a new, decentralized digital money - the cryptocurrency called "Bitcoin".

Although this date can be taken as the official moment of the birth of blockchain technology, decentralized systems were already widely in existence.

Even in 1983, the American cryptographer David Chaum conceived the idea of cryptographic electronic money called "ecash". Later, in 1995, he implemented this idea through "Digicash" - an early version of cryptographic electronic transactions.

In 1998, cryptography and computer science experts Wei Dai and Nick Szabo described their versions of digital money, "b-money" and "bit gold".

The Bitcoin protocol was only created in January 2009, and has brought great innovation due to the decentralization, security measures, and financial incentives it creates for those who maintain the protocol.

1.2. THE PHILOSOPHY BEHIND THE BLOCKCHAIN SYSTEMS

Blockchain systems are by the very nature of the protocol design very closed, which means that within the system itself, we can only see the data recorded in that system, i.e. the list of transactions, and we have no data from the outside world. This kind of mechanism was created on purpose, to maximally reduce the number of potential security threats, but when implemented, it increases the difficulty of using the blockchain system itself by ordinary users.

In the example of cryptocurrencies, if we take the Bitcoin currency and look at it outside of the Bitcoin protocol itself, this currency on different markets will always have somewhat different exchange values in classic currencies, i.e. dollars or euros. The market itself dictates the price, and there is no official price for one Bitcoin.

If we look at the value of one Bitcoin inside the protocol, it will always be equal to one Bitcoin, because it is not designed to be used outside of the protocol itself.

Every market and platform for selling or buying Bitcoin currency is only an abstraction by a third party, centralized or decentralized, to sell or buy Bitcoin for some other currency, while within the system one Bitcoin is always equal to one Bitcoin.

This sounds like it can be taken for granted, but the complexity emerges if we look at other cryptocurrencies, which in addition to being able to transact and transfer from one address to another, also can have smart contracts at the protocol level.

1.3. SMART CONTRACTS

Smart contracts were first conceptualized in 1994 by computer scientist Nick Szabo, who defined them as computerized transaction protocols that can independently execute the terms of a contract.

The first protocol-level smart contracts in the blockchain system appear together with the Ethereum network created in 2013 by developer Vitalik Buterin.

Ethereum is the first blockchain protocol that, in addition to transactions between two addresses, introduced smart contracts that enable the execution of application code at the blockchain level.

This is made possible by the creation of the "Ethereum Virtual Machine" environment (referred to as "EVM") - a decentralized virtual environment that executes code across all Ethereum nodes.

Nodes, using EVM, execute smart contract code using "gas", a unit of measurement that indicates the power and complexity of computing operations required to execute the computer code of the smart contract itself.

This means that the more complex the functions of the contract, the more their execution will cost.

The gas paid for performing functions is distributed to the nodes that maintain the protocol, thus providing a financial incentive to maintain the network.

So, when a user of the network wants to make a transaction, they will pay a certain amount of "gas" to the network of nodes that maintain it, that gas will be distributed proportionally, and the nodes will perform the transaction function.

This enables a fully decentralized network that does not have a single centralized point of failure, but rather the entire system verifies transactions.

Smart contracts allow that in addition to the classic transactions of transferring the Ether currency (the native currency of the Ethereum network), users can create their own applications through code, in the Solidity programming language, and once the application is published on the blockchain network, all other users can also access it and call its functions.

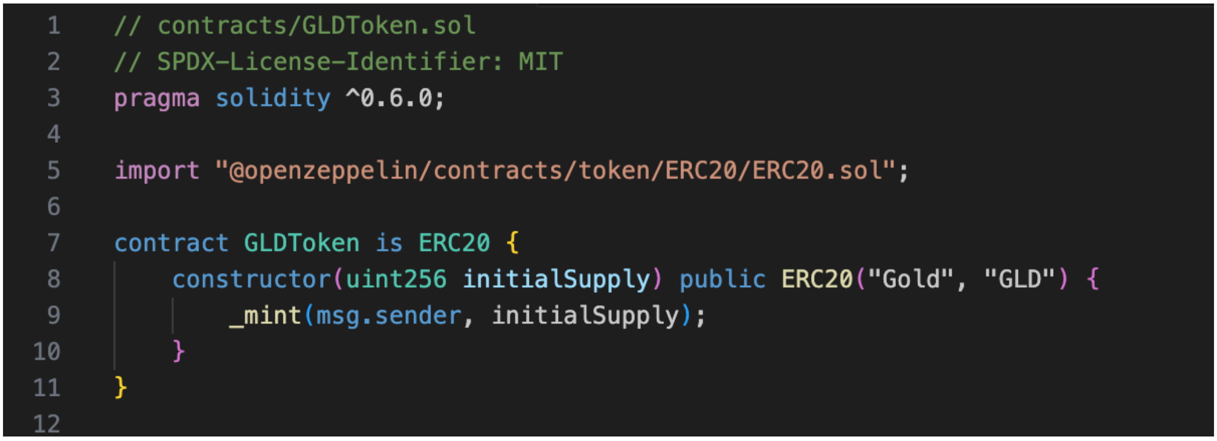
This possibility opened the door to a new type of cryptocurrency, the so-called "ERC-20" tokens, which are smart contracts themselves.

Anyone can create a smart contract containing functions like "transfer", "allowance", "approve", "balanceOf" or "totalSupply", and that smart contract will be its own ERC-20 cryptocurrency. "ERC" is an abbreviation for "Ethereum Request for Comment", so "ERC-20" is just the name of the standard for tokens that are not unique but "fungible", i.e. always of the same value. There are different standards, such as the "ERC-721" standard, which defines "non-fungible" tokens, i.e. tokens that are unique and always have different values.

An example of the difference between these two standards is that one USDT token, which is of the ERC-20 standard, is always equal to another USDT token, while two NFTs will always be different - even though they may in some cases have the same monetary value in the current market, they are different tokens because they have different unique addresses.

In addition to these two basic types of smart contracts that every modern blockchain needs, anyone can create their smart contract so those applications can do different things, like store tokens, exchange tokens and take percentages, transfer tokens from one blockchain network to another (bridge protocols), etc.

***Image 1**: An example of an extended ERC-20 smart contract written in the programming language "Solidity" for a cryptocurrency called "Gold", with the symbol "GLD", which supports the function of minting, i.e. sending tokens in the passed value "initialSupply" to the address of the wallet that published this smart contract on the protocol*



```
// contracts/GLDToken.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract GLDToken is ERC20 {
    constructor(uint256 initialSupply) public ERC20("Gold", "GLD") {
        _mint(msg.sender, initialSupply);
    }
}

```

Source: https://docs.openzeppelin.com/contracts/3.x/erc20 *(seen 3.10.2024)*

2. THE IMPORTANCE OF REAL-WORLD DATA IN BLOCKCHAIN SYSTEMS

2.1. CENTRALIZED EXCHANGES (CEX)

Centralized exchanges (CEX) are centralized platforms through which users can buy or sell cryptocurrencies. Some of the most popular in the market are Binance, Coinbase, Bybit, Bitfinex, Gate.io, and Kraken.

The process of exchanging cryptocurrencies at these exchange platforms is shown to users through a simple user interface, while the functionality itself is hidden in the background and the code is invisible, i.e. inaccessible to users because it is mostly located on the servers of the exchange platforms themselves and not on the blockchain. Parts of the architecture that reside on the blockchain protocol will often be hidden by exchanges and they will not publicly disclose what their smart contracts are, making it even more difficult for users to have any insight into how the software actually works.

Thus, the problem arises where the exchange itself dictates the price of cryptocurrencies in real-world values, ie. real currencies like dollars or euros. The price is generated using a concept called "market making".

- Market makers add liquidity to the "order book" and thus create a market, making it easier for other participants to buy or sell currencies when the order conditions are met. This happens when the purchase or sale is not done instantly, but buyers set conditions, such as a specific value of the currency - when the currency falls to a certain price, then it will be bought. For example, if users place an order to buy a currency when it is below market value, or place an order to sell a currency when it is above market value, those users may be called "market makers".
- Market takers are those who take liquidity from the "order book" by placing an order to be executed immediately, at current market prices, without waiting for conditions.

The mechanics of market creation at centralized exchanges represent a "black box", which means no one officially knows how the market is created and which algorithms were used when designing the system, so we can conclude that exchange offices can modify and set prices as they see fit.

As most tech-savvy cryptocurrency users are skeptical about using software that is unknown and not open source, there is a need for alternative solutions.

2.2. DECENTRALIZED EXCHANGES (DEX)

Decentralized Exchanges (DEX) are decentralized platforms like centralized exchanges, but which are completely written using smart contracts, which makes them completely decentralized, open, and transparent. Anyone can check the software of these exchanges and contribute by writing their own code because the software is open source.
Users can point out potential security flaws or bugs, which makes the project architecture itself much more serious - these exchanges must not have flaws because they would be immediately exploited since every line of code is public.
Some of the most popular exchanges of this type are Uniswap, 1inch, Balancer, and SyncSwap.

DEX exchanges are more complex because each part is created using algorithms and smart contracts, without the possibility of the exchange itself influencing anything in the background.
As they are built on blockchain protocols that support smart contracts, each transaction will have an added cost of "gas" that pays for the transaction itself, as well as a small percentage that is paid to the exchange. Users must have a blockchain wallet to interact directly with smart contracts when buying or selling.

There are three basic types of DEX exchanges:

- **Decentralized order books (Order book DEXs)** - which compile a record of all open orders for the purchase and sale of funds for certain currency pairs. Buy orders indicate that the trader is willing to buy or bid for a currency at a certain price, while sell orders indicate that the trader is willing to sell or ask for a certain price for the asset or currency. The spread between these prices determines the depth of the order book and the market price on the exchange. These types of exchanges have two types - decentralized order books that keep their order books on the protocol, and those that store them "off-chain", i.e. outside the protocol.
- **DEX aggregators** - platforms that use several different protocols and mechanisms to solve the liquidity problem. They essentially aggregate liquidity from several DEX exchanges to optimize exchange fees and token prices and offer users the best possible service in the shortest possible time frame.
- **Automated market makers (AMMs)** - systems that rely on smart contracts to solve the liquidity problem. The creation of these exchanges was largely inspired by the work of Vitalik Buterin, the creator of the Ethereum network and its whitepaper, in which he described how we can perform token exchanges using smart contracts that will store the tokens, thus creating liquidity.
  These exchanges rely on blockchain services that provide data from stock exchanges and other real-world platforms to determine the price of assets. Those services are called **blockchain oracles**.

Instead of linking buy and sell orders, these exchanges' smart contracts use pre-loaded funds called **“liquidity pools”**.

From these examples of centralized and decentralized exchanges, we can see the importance of real-world data, such as the price of assets, to the functioning of the entire system. Blockchain systems by themselves cannot supply data from the real world, but must solve that problem within the system, or use services like blockchain oracle.

2.3. THE BLOCKCHAIN ORACLE PROBLEM

The Oracle problem in the blockchain refers to the inability of the blockchain protocol to access data outside the protocol itself, that is, data from other decentralized or centralized systems, which makes blockchain networks isolated networks.
Solving this problem and connecting the blockchain network ("on-chain") with the outside world ("off-chain") requires an additional part of the infrastructure - **oracle**.

The Oracle problem is one of the most important obstacles to overcome for decentralization through smart contracts to achieve easier and faster adaptation in the wider market and to apply to different use cases for users.
Smart contracts on different protocols provide the ability to redefine how independent entities (individuals, firms, or organizations) can exchange funds or participate in mutually accepted agreements, but functioning separately from the internal economy of the smart contracts themselves represents a much broader digital economy without a blockchain system. which consists of all devices connected to the Internet.

The expansion of digital infrastructure, which is accelerating exponentially, also means an ever-expanding range of data and APIs that provide insight into real-world data, e.g. internet search results that represent popular topics for social discussion, IoT ("Internet of Things") sensors that show traffic patterns, or data sources that show the history of stock market trading.
Smart contracts based on blockchain protocols and traditional data from centralized APIs have the potential to combine their resources and services to create **hybrid smart contracts** that represent the future of automated data infrastructure.

The main question is **how to bring these two worlds together** - this question represents the essence of the oracle problem.

3. DECENTRALIZED ORACLE NETWORKS

3.1. BLOCKCHAIN ORACLE SERVICE DEFINITION

A blockchain oracle is an entity that connects the blockchain protocol to external systems, thus enabling smart contracts to be executed based on real-world inputs and outputs. As "real world data" we can take data from any "off-chain" systems, such as various web APIs, backend services of large companies, cloud providers, IoT devices, payment systems, other blockchain systems, etc.

Oracle systems play an important role in creating a "verifiable" web, connecting blockchain protocols with data outside the system, as well as establishing a connection between several different blockchain systems that are based on different protocols.

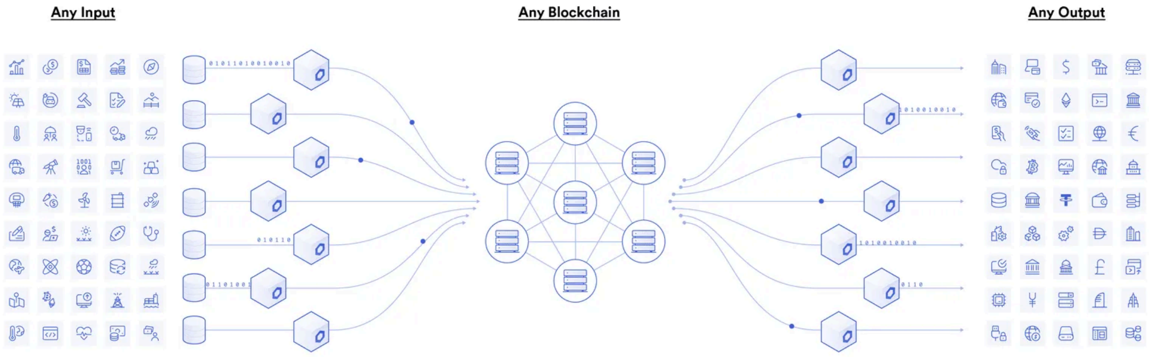
Some of the key features of the oracle systems are:

- **Listening** - monitoring the blockchain network to check for any requests from users or smart contracts for data outside of the protocol
- **Extracting** - fetching data from one or more external systems such as "off-chain" APIs placed on third-party servers
- **Formatting** - formatting data retrieved from external APIs into a format readable by the blockchain system, or formatting blockchain data into a format compatible with the external API. Example: smart contracts use WEI as a unit of measure (one Ether is equal to 10^{18} WEI) expressed in "integer" formats, so any data that comes in any other type must be formatted so that it is expressed in a unit that the smart contract can read
- **Validating** - generating cryptographic proofs to confirm oracle service performance using various combinations of data signing, blockchain transaction signing, TLS (Transport Layer Security) signature, TEE (Trusted Execution Environment) attestation, or zero-knowledge attestation
- **Computing** - performing "off-chain" computing for smart contracts, for example calculating an average value between multiple different oracle inputs, or generating a random number (this can be a useful case for applications that have game elements)
- **Broadcasting** - signing and broadcasting the transaction on the blockchain so that the data reaches the protocol and is received by the smart contracts
- **Output (optional)** - sending data to an external system after executing a program on a smart contract

Successful execution of these functions requires the oracle system to run on the blockchain system as well as off the blockchain system at the same time.

The components on the blockchain protocol serve to establish a connection, i.e. listening to requests, broadcasting data, sending confirmations, extracting blockchain data, and, if necessary, performing computational operations on the blockchain. Off-blockchain components serve to process requests, collect and format external data, send blockchain data to external systems, and perform computational operations off-protocol for easier scaling and greater privacy and security.

***Image 2**: An example of Oracles connecting blockchain systems with input and output data from external sources, i.e. from the real world*



Source: https://chain.link/education/blockchain-oracles *(seen 11.10.2024)*

3.2. OBSTACLES OF BLOCKCHAIN PROTOCOL IN SOLVING ORACLE PROBLEMS

Blockchain systems are secure because of the basic principles on which their protocols are based - they require a simple form of consensus using data already recorded in a list of transactions, i.e. in blocks.

Each transaction from the list is verified and confirmed by various participants in the network before being written into the block, which means that if a transaction is written into the block, it is definitely correct (except in specific cases where the entire blockchain protocol is compromised or attacked). The transaction list is correct and trustworthy because it uses the power of decentralization to validate all transaction data using "nodes" that are active on the network.

Also, decentralization plays a role in the execution of various consensus mechanisms (such as "Proof of Work" or "Proof of Stake" algorithms), thus achieving security and integrity that allow changing the protocol and its rules only under conditions in which the majority of participants in the network agree on the same decisions.

These properties of the blockchain system guarantee that the protocols will be deterministic and impenetrable.

Although they solve the problems of safety and correctness of objective data within the system, such decentralized protocols are not a good solution for questions of correctness of data that are subjective or require additional external information, i.e. context that is not accessible to nodes in the network.

Simple questions like the market price of Bitcoin currency or the weather forecast for a certain region can produce a wide range of different answers depending on the data sources the participants have access to. Then the question arises - which answer is correct, and how can its correctness be confirmed?

For the sake of the functioning and security of the blockchain system itself, subjectivity must not be implemented at the basic level of the protocol - this would create numerous security, privacy, and decentralization problems.

When removing subjectivity and striving to use objectively accurate data in the protocol, one of the potential challenges is the quality of the data itself. Even a simple request to fetch the price of Bitcoin currency can be challenging because the public data on sites or individual exchanges can be very different from the data we can get from a paid API provider or from another aggregator service that is much better at filtering data and transactions. and has a greater financial benefit if it provides accurate, high-quality data.

It is difficult to guarantee the quality of data coming from "off-chain" sources by blockchain nodes, because the concept of decentralization allows anyone to implement their node relatively anonymously and provide data from different sources, even if those sources are not e.g. paid API services that guarantee high data quality.

If data quality checks were to be enforced, the blockchain would reduce the level of decentralization because the cost of getting up and running nodes for all oracle computing operations would increase, which in turn would cause fewer people to be able to run nodes, thus affecting the security of all other applications that run on that blockchain protocol.

Another challenge is scalability - every time a new data source needs to be added to the network, i.e. if it is necessary to change the functioning of an already existing data collection method, the coordination of all nodes in the network is needed in order to agree and accept the software change, so such changes lead to greater social demands, slower implementation and development of core protocol components, and slow down the innovations that can be made on the protocol. The more logic and complexity that is implemented at the base level of the blockchain, the more chances there are for potential attacks and security breaches for both the blockchain and any applications that are built on top of it.

Even decentralized applications that do not use oracle services can be exposed to attacks if the protocol itself fails due to oracle problems.

Due to these obstacles, oracle systems must never be implemented at the base level of any blockchain network, but must be built as separate networks, which allows blockchain systems to do their work, that is, to focus on issues of consensus and correctness of transactions, while oracle networks will have greater flexibility to implement deterministic data from subjective "off-chain" data sources without introducing potential security flaws that can compromise other applications.

3.3. CENTRALIZED BLOCKCHAIN ORACLE SERVICES

The purpose of smart contracts is to achieve determinism through technological implementation and execution of contract terms as opposed to probabilistic execution performed by human participants.

To achieve this goal, the blockchain must not have any central point of failure. This feature must be respected when implementing the oracle system if we want the deterministic nature of smart contracts to be fully maintained.

The economics of smart contracts are not simple, and they are by no means small - some of the larger applications and organizations are storing hundreds or tens of millions of dollars in smart contracts.

If smart contracts that store multi-million dollar asset values, which otherwise operate completely decentralized, were to include a centralized oracle system - this could lead to major problems and financial losses. A centralized oracle service that could affect the functioning of the outcome of a smart contract must never be implemented if we want to maintain decentralization and security.

Even in specific cases where organizations implement their own centralized blockchain systems (for example, for needs within the organization itself, where they do not need decentralization but other qualities of the blockchain protocol), a centralized oracle system can be a problem. Although the participants controlling the oracle may have the best intentions, they are still subject to all the problems that centralized systems are subject to - such as DDoS (Distributed Denial of Service) attacks, hacker attacks, server shutdowns, or random failures. Each of these issues poses a major risk to the system and potentially compromises the end user's assets.

Such a model is not scalable or secure, and does not fit in any way with the philosophy of decentralization, the idea of which promotes the entire decentralized infrastructure as key to the safe and reliable execution of transactions.

3.4. DECENTRALIZED ORACLE NETWORKS

Decentralized oracle networks (DON) are distributed systems of oracle services designed to securely supply external, "off-chain" data to the blockchain system without relying on a central point of failure.

Unlike centralized oracle services, decentralized oracle networks use multiple independent nodes for data aggregation to ensure greater security, transparency, and verifiability. This approach is crucial for maintaining the basic principle of blockchain - decentralization.

The basic components of decentralized oracle networks are:

- **Nodes** - independent participants working within the network to source, verify and transmit data. Each node can request data from various external sources, and in some networks, they are incentivized to act fairly through reward mechanisms, e.g. through "staking" mechanisms or transaction fees
- **Data aggregation** - information from different nodes is aggregated to form a single reliable data collection. By collecting data from various sources, decentralized oracle networks reduce the risk of error-prone, inaccurate, or malicious data. Common methods of aggregation include taking the average mean for numerical data, majority consensus for binary outcomes, or using cryptographic evidence to verify data accuracy
- **Consensus mechanism** - similar to blockchain consensus algorithms, decentralized oracle networks often use various mechanisms to ensure that most oracle systems provide accurate data. These mechanisms may include reputation systems, where nodes with a positive history are prioritized, or financial incentives, such as reduction of fee rewards in case of malicious behavior, or expulsion from the network
- **Infrastructure** - building an infrastructure that supports decentralized oracle networks involves implementing the following components:
    - **Smart contracts** - these contracts enable the interaction between the blockchain and the oracle system. They determine how the data should be retrieved, how many oracle systems are needed, and how the data should be aggregated and verified

- **Middleware** - many oracle networks rely on infrastructure located on the layer between the blockchain and the "off-chain" world. This layer creates an abstraction over the complex operations of extracting, verifying, and formatting data before it reaches the blockchain protocol
- **Incentive structures** - to maintain a decentralized model, oracle networks often employ economic incentive models where nodes are rewarded with tokens or fees for providing accurate data. Misbehaving nodes may be penalized with reduced fees or fines

Oracle networks implement security and trust mechanisms such as redundancy and fault tolerance to reduce the risk of manipulation or omission - this is done by sending requests for the same data across multiple different nodes. Even if some nodes are compromised, the rest of the network should provide accurate data.
Using cryptographic evidence and other techniques, oracle networks ensure that the data being transmitted can be verified and that no one has influenced that data at any step of the process.

The benefits that decentralized oracle networks bring are reducing the need to trust a single entity that is subject to errors, corruption, or other problems, resistance to censorship because there is no central authority that controls the network, and transparency because the functioning of the oracle system can mostly be seen on blockchain, which gives participants the ability to control how data is aggregated.

3.5. EXAMPLES OF DECENTRALIZED ORACLE NETWORKS

Some organizations provide ready-made infrastructure of decentralized oracle networks for ease of use and faster development of applications.
Users can always create their own protocols and systems, but as in the case of blockchain, where it is much easier to use already existing infrastructure such as e.g. Ethereum or Solana protocol, and in the case of oracle networks it is much easier to use already existing solutions.

CHAINLINK

One of the leading organizations specializing in oracle network services. Chainlink enables developers to create secure and reliable connections between "off-chain" data and decentralized blockchain-based applications through its network of independent oracle systems. This organization supports a large number of different use cases and different services for each of them.

- **Cross-chain** enables connection between blockchain protocols, sending messages and transactions, sending funds, and creating "cross-chain" applications that are not tied to one blockchain ecosystem, but support several different protocols
- **Data streams** service offers the possibility of quickly obtaining data from the DeFi (Decentralized Finance) market thanks to its architecture, and the data is visible and verifiable thanks to the transparent and decentralized infrastructure. This service is suitable for applications that need data that changes at high speed, such as decentralized exchanges where the value ratio of different cryptocurrencies can change in milliseconds

- **Market and data feeds** provide a safe, reliable, and decentralized source of "off-chain" data to enable unique smart contracts for DeFi use cases, and many others. Users are facilitated with access to a decentralized and reliable infrastructure, high-quality data sources that are resistant to manipulation, and nodes that are resistant to attacks and verified. They also support all blockchain networks which means all smart contracts are supported on every protocol
- **Proof of reserve** offers autonomous, reliable, and secure tracking of resources and assets using cryptographic evidence. This tracking can be performed for funds stored in "off-chain" vaults, as well as for verification of funds between different blockchain protocols. When the values of assets in the reserve ("off-chain" or "on-chain" treasury) are changed, the Chainlink network will send that information, and immediately transmit it to the smart contract on the blockchain that is responsible for confirming the reserve (Proof of Reserve Contract) and thus inform the protocol about any changes
- **Functions** is a platform for retrieving data from any API and performing computational operations without the need for a server. This platform works by the end user running the Chainlink function embedded in the dApp (decentralized application), then the dApp creates a request consisting of the API endpoint and encrypted credentials to the **Chainlink Functions** smart contract, the decentralized oracle network constantly listens to the smart contract and when it accepts to send request, each node independently performs the collection of external data and any necessary computation on it and returns the result. Then the nodes reach a consensus on the final answer, and one node is chosen to return the results to the blockchain. In case the chosen node fails to send the result, the next node is chosen to upload the data to the blockchain. The end result thus has high reliability and security
- **Automation** is reliable, high-performance smart contract automation that allows developers to rapidly scale their operations in a decentralized and efficient manner. An example of automation is the execution of a selected function of a smart contract at certain time intervals, or in case of a predetermined condition
- **VRF (Verifiable Random Function)** provides a cryptographically secure source of randomness for blockchain-based applications. The problem of randomness is generally a problem for blockchain systems because in computing there is no such thing as complete randomness - behind every random number generation function there is an algorithm. This is especially a problem in blockchain systems, where all the code is public and accessible. If everyone has access to the randomization algorithm code, then anyone can always calculate what the random number will be based on that algorithm, which automatically makes that number non-random. The VRF platform helps with this use case by providing randomness using a decentralized oracle network that uses its own randomization algorithms that are unpredictable by the blockchain system and then passes the results to a blockchain protocol that can use random values in that case

BAND PROTOCOL

Band Protocol is an oracle platform that supports communication between blockchain protocols and aggregates and forwards real-world and API data to smart contracts.
These are some of the services they provide:

- **Cosmoscan** - a platform for tracking and viewing transactions, blocks and validators on the Bandchain oracle network. One can also inspect the data sources and APIs used to retrieve them, view the scripts used for various functions, and download datasets used in the oracle network
- **Band Price Feeds** - facilitates the development of dApp platforms by providing the Band Standard Dataset - sets of real-time price information for over 200 symbols (relationship between currencies, for example "BTC/ETH") that include cryptocurrencies, forex, and commodity values. These oracle systems provide data for the Ethereum, Binance Smart Chain, Fantom, and Near Protocol networks
- **Band VRF** - as with Chainlink, this VRF service addresses the fundamental issues that threaten the integrity of various blockchain use cases by providing random results along with proof of authenticity that validates the random number generation process
- **Band Integration Tools** - these tools provide developers with a wide range of options for integrating applications with the BandChain network. "**cw-band**" library enables the connection of smart contracts from the **CosmWasm** network through the IBC Relayer service with real-world data - smart contracts send requests to the relayer, which picks them up and forwards them to the **BandChain oracle service**, which prepares and aggregates data from various sources and returns them to the relayer, which then forwards those results to the contracts on the CosmWasm Network. The Pricefeed module works the same way, but for smart contracts that are on the **Cosmos** blockchain. These functionalities are also supported for the Falcon network.

API3

This platform is also one of the leading oracle service providers. API3 builds solutions that bridge the path between "off-chain" data and blockchain applications with maximum security and safety.

Some of the services that API3 offers are:

- **dAPI** - Decentralized APIs represent data sets that are decentralized on the blockchain, and which are obtained directly from oracle services that are owned by the API providers themselves, using the **Airnode** service. This data is constantly updated, and owners of decentralized applications can always read data from the blockchain and see the values of any dAPI in real-time
- **Airnode** - this service is an oracle that forwards "off-chain" API data to the blockchain via data feeds (dAPI) or using its **"request-response" protocol (RRP)**. Airnode operates off-blockchain, as a back-end component of the API3 ecosystem. Essentially, Airnode allows API providers to set up and maintain their own oracle nodes - providers can publish data feeds (dAPIs) to any decentralized application, allowing smart contracts to be written that can interact with "off-chain" data
- **QRNG** - like other platforms, API3 offers its service for generating random numbers. Unlike most PRNGs (pseudorandom number generators), the QRNG service uses the **QRNG (quantum random number generation)** method, which is based on quantum phenomena. There are several ways in which quantum randomness can be implemented, but all of them boil down to making the generated numbers completely random due to the outcome of quantum events that are theoretically uncertain but with well-defined characteristics. Taking these methods into account, QRNG is the gold standard for generating random values

- **DAO** - in addition to the services they provide, API3 delegates a large number of its decisions to DAO (decentralized autonomous organization) members, who can vote and influence decisions about the direction in which the organization will move, about the next steps, or the implementation of new software components. API3 aims to become a fully team-based and democratized environment where each participant will be able to influence the next steps through the token system - anyone who owns a certain number of selected API3 tokens can influence organizational changes
- **OEV Network** - this platform solves the problem of all major protocols spending large amounts of money every year on data extracted through oracle services. The OEV network is a "layer 2" solution based on the **Arbitrum Orbit** protocol that creates a financially efficient market for collecting oracle data. This is accomplished by data claimants bidding for valuable oracle data, and the proceeds from such auctions are programmatically funneled back to the decentralized applications that generate that valuable data.

DIA

DIA (Decentralized Information Asset) is a decentralized open-source oracle platform that focuses on collecting transparent and verifiable financial data generated by the community and providing that data to blockchain protocols and applications, most often in the domain of decentralized financial systems.

The services they provide are:

- **Token Price Feeds** - data feeds for token prices, collected from various centralized and decentralized exchanges. This service supports prices for over 3000 tokens. While traditional oracle providers depend on data sources from a third party or some external organization, which are often non-transparent and can be manipulated, DIA provides fully transparent data sources that guarantee data resistant to manipulation and corruption
- **NFT Floor Price Feeds** - this service provides access to transparent market data for over 18,000 NFT collections. DIA uses the results of transactions and sales from numerous NFT stores and markets to generate customized data feeds for NFT prices
- **Random Number Generator** - random number generator designed for decentralized gaming platforms, NFT collections, market predictions, and other decentralized applications
- **Fair Value Price Feeds** - oracle pricing systems for liquid derivatives, i.e. funds that are "staked" on the blockchain. DIA **xLSD** (Liquid Staked Derivatives) feeds provide price feeds of such derivatives that are collateral-checked and fairly priced

REDSTONE

RedStone provides modular oracle systems for all layer 1 and layer 2 blockchain networks based on the Ethereum Virtual Machine, and certain "non-EVM" networks. The infrastructure design allows for independent and robust modules, and signed data packets are broadcast to the DDL (Data Distribution Layer) and stored on the Arweave protocol. The

Arweave protocol is known for its decentralized way of storing data and files that, once written into blocks, are permanently saved for further use.
In addition to services such as randomization, price feeds, data processing and formatting, and NFT data feeds, RedStone provides three different models:

- **RedStone Pull Model** - this model enables the automatic insertion of data into user transactions, thus achieving maximum gas price efficiency. This approach is easy for users to implement because the entire process fits into one transaction. The Pull Model significantly reduces the costs of decentralized applications that collect "off-chain" data
- **RedStone Push Model** - the push model is designed for applications that need a more traditional approach to oracle systems, for data that is sent to the blockchain protocol at wider time intervals, and such oracle models provide full control over data sources and update conditions.
- **RedStone X Model** - designed to meet the requirements of advanced protocols such as perpetuals, options, and derivatives. By providing price feeds in the first subsequent block after user interactions, the X model removes any risk that may occur at the beginning of the process

4. DETERMINISM AND OFF-CHAIN DATA SYNCHRONIZATION

4.1. DETERMINISM IN BLOCKCHAIN

In blockchain, "determinism" can refer to the property of blockchain that guarantees that any input query that has the same parameters will result in the same output. So, no matter how many times we repeat the query without changing the parameters, the answer must always be the same without any changes. This property is crucial to the integrity and trustworthiness of the blockchain system.

These are some of the components of the blockchain system where determinism plays an important role:

- **Consensus mechanisms** - the algorithms that determine the agreement of participants in the blockchain network, consensus mechanisms, rely on deterministic rules to ensure that all nodes in the network come to the same conclusion about the state of the network. For example, if we provide the same set of transactions and environment state, every node on the network should end up with the same final state after computation
- **Smart contracts** - they must be deterministic so that their functions are executed the same way on every node. If smart contracts produce different results on different nodes, this could lead to consensus failure, thus breaking the integrity of the entire network
- **Transaction processing** - when a transaction is processed on the blockchain, it must have the same result on every node that executes it. This means that the order of transactions, the state of the blockchain, and the logic defined on the smart contract must be deterministic

- **Hash functions** - those used in blockchain systems, such as SHA-256 in the Bitcoin network, are deterministic. Given the same input, these functions will always produce the same output values, which is a fundamental property for checking data integrity and makes the blockchain more secure
- **Block creation** - the process of creating new blocks involves solving cryptographic problems (in the "Proof of Work" mechanism) or reaching consensus around a decision (in the "Proof of Stake" mechanism)

The rules governing these processes are deterministic to ensure that all data in the blockchain is always correct and that all participants have agreed on the accuracy of that data.
Determinism guarantees consistency, reliability, and security within blockchain networks, thus allowing them to function as intended without conflicts or disagreements.

However, while it provides some of these benefits, the deterministic nature of blockchain creates challenges when trying to implement data outside of the blockchain itself, as this data is often unpredictable and can change over time.
Since smart contracts require deterministic input data, data that is fixed and does not change, off-chain data such as real-world events, stock prices, or weather conditions cannot be directly passed to the protocol without the risk of potentially compromising or collapsing consensus. If one node receives different data from another node (due to a different time of receiving the message or a different data source), the results may vary, leading to an inconsistent state on the network.
This lack of synchronization creates the risk of "forks" being created (certain parts separating from the primary and adding their changes to the state), which undermines the decentralized nature of the blockchain system.

To solve this problem, oracle systems need to source, verify, and deliver consistent data off-chain to blockchain protocols in a way that can ensure that all nodes receive the same information at the same time, to preserve determinism.

4.2. CHALLENGES WITH OFF-CHAIN DATA SYNCHRONIZATION IN DETERMINISTIC SYSTEMS

When a node joins the blockchain network, it must synchronize with the current state of the entire blockchain network by downloading all historical transactions. This process involves validating every transaction, from the first block in the chain onwards, to ensure that a node reaches the same state as other nodes on the network. Since blockchain networks are deterministic, the result of each transaction must be the same across the entire network, which guarantees that every participant in the network will share the same transaction history and the same data.

The process of "rewinding" historical data on the chain allows new nodes to verify the authenticity of all data they synchronize with and adhere to protocol rules. However, the inclusion of "off-chain" data in this process creates challenges because it introduces unpredictable external data into an otherwise deterministic system.
Out-of-system data supplied by external APIs or real-world services presents a unique problem for blockchain determinism. As blockchain nodes by design cannot access or verify data offline themselves, they must rely on oracle services to deliver this information to them. Without a quality mechanism for obtaining and verifying data off-chain, it is impossible for

nodes to "rewind" such data deterministically. Each node could access a different collection of data at a different time or from different sources, leading to different transaction outputs.
This unpredictability and lack of synchronization with off-chain data conflicts with the deterministic nature of blockchain systems, making the use of reliable data very challenging.

One of the most critical risks when synchronizing in a deterministic environment is data inconsistency. If nodes rely on offline data, they may receive data that is out of date due to the time it takes for the network to access the data or API, or due to manipulation of the sources from which the data is obtained.
This can cause nodes to get different computation results and come to different conclusions about the same transaction, thus creating discordance in the network. These discrepancies make the network inconsistent with the basic principles of consensus, where all nodes must agree on a single version of the "truth". In addition, such discrepancies can also lead to exploitation by hackers or bad network actors who can manipulate transaction outcomes and thus create security vulnerabilities for applications that rely on data from the outside world.

If nodes receive "off-chain" data that differ from each other, there is a risk of the network splitting into "forks", where different groups of nodes process transactions based on different external information. Forking occurs when nodes disagree about the state of the network, potentially creating multiple versions of the blockchain.

Such forking or duplication on multiple networks can lead to big problems because some nodes can continue working on one set of transactions, while other nodes process other transactions, which leads to confusion, double cost problems, and loss of data integrity. Forking not only threatens the security of the blockchain but also complicates the logistics of the work of all participants, especially for decentralized applications that rely on accurate and uniform external data.
Maintaining consistency and reliability when synchronizing external data is critical to solving and preventing fork problems, and preserving network integrity.

Forking a network does not always have to be a security breach, but in some cases, it can be intentional if network participants disagree on some decisions.
For example, we can take the Bitcoin blockchain fork that happened on August 1, 2017, exactly at 478,559. block. Until the previous block in the network, there was one Bitcoin network, but after the block in which the fork occurred, the Bitcoin network split into two networks - Bitcoin and Bitcoin Cash. The fork occurred because the participants did not agree on the changes and innovations of the technical implementations, so the Bitcoin network remained the same as before, while Bitcoin Cash introduced certain technical improvements, such as increased block sizes, which allowed the Bitcoin Cash network to process more transactions per second versus Bitcoin. This fork was consensual and planned when the developers cloned the Bitcoin code and modified it, so this cannot be classified as a security breach.

4.3. STORING EXTERNAL DATA ON-CHAIN FOR CONSISTENCY

Oracle systems serve as key intermediaries between "off-chain" data sources and blockchain systems, allowing decentralized applications to communicate with real-world information. Since blockchain systems, by design, cannot access external data, oracle systems represent trusted intermediaries that source, verify, and transmit data coming from outside the blockchain environment. These systems collect data from various sources and transfer it to the blockchain to be used by smart contracts. Oracle systems fall into two main divisions:

centralized, which relies on a single data source, and decentralized, which aggregates input data from many different sources to increase trust and reduce centralized points of failure. Without the oracle system, blockchain networks would not be able to use external data, which would greatly limit their use in real-world applications such as decentralized finance, insurance, or supply chains.

Before external data is stored on the blockchain, it must undergo a process of aggregation and verification to ensure its reliability and accuracy. Decentralized oracle networks (DONs) play a key role in this process by pulling data from multiple disparate sources and aggregating it to reach a consensus on the correct data.

Aggregation from multiple sources ensures that no single oracle can provide false or manipulated information, and the blockchain can be trusted to write and store data.

Also, oracle systems often integrate cryptographic evidence to guarantee that data has not been altered in transit - only when the data has been verified will it be written to the blockchain, enabling deterministic execution of smart contracts.

Once the data is written to the blockchain, "**data finality**" is achieved, which means that the data becomes an immutable part of the blockchain's history. This allows all nodes to reference the same, deterministic version of "off-chain" data when processing transactions or executing smart contracts. The permanence of data on the blockchain guarantees consistency across the entire network and thus prevents discrepancies that may occur if different nodes access different data at different time intervals.

Data finality is especially important in applications like DeFi applications, where price feeds, interest rates, or loan statuses must be consistent to prevent financial losses or security breaches. By storing external data on the blockchain, oracle systems ensure that every node in the network has access to the same reliable data, thus preserving the deterministic nature of the blockchain.

4.4. SOLVING THE CHALLENGES OF DETERMINISM AND SYNCHRONIZATION ON THE EXAMPLE OF THE CHAINLINK COMPANY

Chainlink is a leading network of decentralized oracle systems that enables smart contracts to securely communicate with real-world data, APIs, and other "off-chain" computational operations. By bridging the gap between blockchain environments and external data sources, Chainlink extends the functionality of decentralized applications outside the blockchain environment itself, enabling the integration of different user experiences, such as asset market prices, weather forecasting, or IoT applications. Chainlink's decentralized architecture ensures that no single individual can control the flow of data and thus ensures the secure nature of the blockchain where trust is not required while facilitating reliable integration with "off-chain" information.

Chainlink solves the challenge of deterministic synchronization of external data by aggregating data from many different independent oracle systems. Instead of relying on a single provider, which introduces central points of failure, Chainlink integrates a network of oracle systems where each oracle obtains data from different sources. That data is then aggregated into a unique and verifiable data set through consensus mechanisms, ensuring that no single node can influence the outcomes of these processes. By combining data from multiple sources, Chainlink significantly increases the reliability and security of the data it provides, reducing the risk of manipulation, mismatches, and errors when synchronizing "off-chain" data.

The Verifiable Random Function (VRF) provides a secure way to introduce randomness into blockchain applications while maintaining determinism. Traditional blockchain systems generally have difficulty with random values, as any unforeseen data off the network can throw off the deterministic nature needed to reach consensus. VRF generates cryptographic evidence that verifies that the randomness is tamper-resistant and verifiable by all nodes on the network.
This allows each node to receive the same random value without compromising the deterministic nature of the system. The VRF service is particularly useful in applications such as gaming applications, NFTs, or games of chance, where a random value is required to determine the outcome of the program without being a target for manipulation.

To ensure data integrity, quality, and reliability, Chainlink has developed economic incentive models for oracle system operators. Oracle systems are compensated for their work in the LINK token, the native token for the Chainlink network, as a reward for providing accurate data. They may lose their token portions if they provide wrong or malicious information. This system creates a strong financial incentive for oracle system operators to consistently provide good data, while the decentralized nature of the Chainlink network ensures that no single oracle can manipulate or download data feeds, as oracle systems are selected and rewarded based on their performance and reliability.

As an example of using the Chainlink oracle network, we can take price feeds, which are widely used by DeFi applications. DeFi platforms, such as Aave or Synthetix, rely on the price Chainlink oracle networks to obtain accurate real-world asset prices and use them to execute smart contracts related to lending, loans, or collateralization. Decentralized data aggregation ensures that price feeds cannot be manipulated and are constantly synchronized between all nodes on the network, thus preventing discrepancies that can lead to problems such as incorrect values related to the liquidation of assets or incorrect prices.
These examples show how Chainlink maintains determinism while synchronizing external data, preserving the accuracy and reliability required for critical financial applications.

5. TYPES OF EXTERNAL DATA IN BLOCKCHAIN SYSTEMS

5.1. DATA FEEDS

Data feeds are a type of "off-chain" data that provides consistent, real-time numerical information so that it can be used on the blockchain. One of the most common examples of such data is **price feeds**, which are essential for the functioning of DeFi applications. Price feeds aggregate price data from numerous exchanges to provide correct valuations of assets in the market - such as cryptocurrencies, oil prices, or the price ratio of various foreign currencies. Using decentralized oracle systems, blockchain networks can obtain this numerical data in a safe and secure manner.

Consensus for such data feeds is most often achieved through the aggregation of data from multiple sources.

5.2. CHAINLINK FUNCTIONS

Chainlink functions extend the capabilities of the oracle system by allowing smart contracts to access more external data through HTTP requests. While traditional oracle systems specialize in numeric data like price feeds, Chainlink's functionality allows smart contracts to communicate with external APIs, allowing a large amount of non-numeric data to be retrieved and stored on the blockchain. Applications may include providing time data for a smart insurance contract parameter, checking the status of user accounts, or getting any other real-time information from an API.

Through Chainlink functions, smart contracts can make direct requests for specific external data, thus expanding the capabilities of decentralized applications.

These features are especially useful in cases where smart contracts need real-time input that is dynamic rather than just constant numerical values. For example, a DeFi application can use Chainlink functions to check external regulatory data or legal statuses before executing a transaction. This flexibility makes it easy for blockchain systems to integrate with the outside world while still guaranteeing the security and accuracy that decentralized oracle systems provide.

6. COST AND BENEFIT ANALYSIS USING REAL DATA

6.1. RISKS

Implementing real-world data in blockchain systems presents both opportunities for innovations and challenges. While external data expands the potential use cases for decentralized applications by allowing smart contracts to communicate with external systems, it also introduces several risks, costs, and operational challenges. Such integrations must always take into account the trade-off between the benefits of implementation and the downsides, such as oracle manipulation, security risks, and financial costs related to accessing external data, as well as technical difficulties in system development and guaranteeing accuracy.

It is necessary to maintain a balance between decentralization and system efficiency and make strategic decisions when integrating real data in blockchain networks.

6.2. ATTACKS BY MANIPULATION OF THE ORACLE SYSTEM

One of the most significant risks related to the use of "off-chain" data in blockchain systems is the potential for an attack to occur by manipulating the oracle system. Oracle systems serve as a bridge between smart contracts and data sources outside the blockchain, so since they introduce external data into otherwise deterministic systems, they become a high-value target for exploitative attackers. If an attacker can manipulate the data that the oracle provides, they can potentially alter the outcomes of smart contracts in their favor, which can lead to huge financial losses or other undesired outcomes.

A prominent example of oracle system manipulation is the **price feed attack** on DeFi platforms. In cases of this attack, attackers will attempt to manipulate the price of an asset or cryptocurrency on a specific oracle system or exchange to cause miscalculations of collateral or liquidation. This can then result in the wrong initiation of liquidation or unfair chances for buying or selling assets. Decentralized oracle networks such as Chainlink attempt to mitigate this risk by aggregating data from various independent oracle systems, reducing system reliance on a single data source. However, achieving complete security is challenging, and manipulation is always a risk, especially if a large number of oracle providers are compromised.

6.3. TRANSACTION COSTS AND ORACLE FEES

Integrating external data also comes with significant financial costs and prices, primarily in the form of **transaction fees** and **oracle service fees**. Every time external data reaches the blockchain, a transaction needs to be made to be written on the blockchain, which involves paying gas for the cost of the transaction. These fees can vary depending on network demand, making regular data updates expensive, especially on high-demand networks like the Ethereum network.

In addition to transaction costs, there are fees for using decentralized oracle services like Chainlink. Oracle operators must be paid to provide accurate and precise data, and these costs are often covered by developers and users of decentralized applications. For applications that require frequent data updates, such as DeFi platforms that rely on real-time asset prices, these fees can accumulate and represent a significant operational cost.

Although some projects may opt for centralized oracle systems to reduce costs, this often comes at the cost of losing decentralization and security. The trade-off between cost efficiency and the "trustless" nature of decentralized oracle systems is a key consideration when deciding how to collect external data for each decentralized application.

6.4. THE CHALLENGE IN ENSURING DATA ACCURACY

Another critical challenge when using real-world data in blockchain systems is ensuring data integrity and timeliness. For smart contracts to work correctly, the data they rely on must be reliable and up-to-date. Any delay in updating the data or irregularities in the supplied data can lead to improper execution of smart contracts and potential losses.

For example, in the context of price feeds, an update delay could mean that smart contracts rely on outdated prices, which would lead to inefficient or inaccurate "decision-making" by smart contracts in DeFi applications. Latency in data delivery can be particularly problematic in markets with fast-changing and updating data, where price movements move quickly and prices fluctuate. Similarly, inaccuracies in the data provided by oracle systems, whether due to technical errors, compromised sources, or manipulation, can cause application misbehavior and erode user trust.

Oracle systems that collect data from multiple sources aim to mitigate these challenges by providing redundant data and verification mechanisms, but achieving real-time accuracy while minimizing delays remains difficult, especially in highly decentralized networks where transactions are not instantaneous.

The fundamental trade-off in using real-world data is between decentralization and efficiency. The concept of the **blockchain trilemma** talks about the three main pillars of blockchain, **security**, **scalability,** and **decentralization**, and the great challenge of achieving all three simultaneously. As an example of this problem, we can take any decentralized application - if it has a high degree of decentralization and a high degree of security, it is difficult to scale because decentralization involves a large number of participants, i.e. nodes, and security involves demanding computing operations, so scaling would be either expensive or ineffective. If we have an application that is efficient and easy to scale and wants a high rate of decentralization, it must somehow potentially compromise security by making it easier for nodes to perform easier computing operations.

Like the **blockchain trilemma**, using real-world data runs into a similar problem. Decentralization is the basic principle of a blockchain system that promotes security, trustworthiness, and resilience by distributing authority across a network of many participants. However, this comes at the cost of speed and efficiency, especially when interacting with off-chain data.

Decentralized oracle networks such as Chainlink prioritize security and trust by sourcing data from multiple independent oracle systems and using aggregation mechanisms to reach a consensus on accurate data. While this approach reduces the risk of manipulation and ensures that data is not controlled by a single entity, it can lead to delays and increased costs due to the need to pay fees for multiple providers and manage consensus protocols. This slows down the speed at which data can be updated on the chain and increases the complexity of the system.

On the other hand, centralized oracle systems offer greater efficiency and lower costs by relying on a single source or a small number of trusted data providers. This reduces data delivery latency and minimizes the complexity of consensus mechanisms, but sacrifices decentralization and introduces individual centralized points of failure, making the system vulnerable to manipulation and attack.

For serious applications that have a large amount of funds in their reserves, such as DeFi applications, the trade-off between decentralization and efficiency is crucial because users must calculate the benefits of faster and cheaper data acquisition against the risks posed by centralization.

Integrating real-world data offers significant advantages and benefits, but it also introduces numerous risks, costs, and challenges. While decentralized oracle networks like Chainlink help alleviate many of these problems through data aggregation and economic incentives for network participants, manipulation attacks, high operational costs, and the need to ensure timely and accurate data remain significant issues. Furthermore, developers must constantly weigh the trade-offs between decentralization and efficiency and balance the security of decentralized oracle systems with the cost and speed of more centralized networks. As blockchain technology continues to evolve, finding a balance between these factors will be critical to enable the adoption of the technology by the general public and to see the technology accepted in more real-world applications.

6.5. EXAMPLES OF REAL ATTACKS USING FLAWS IN ORACLE SYSTEMS

In blockchain systems, attacks are more common than in other closed systems because it is much easier for attackers to see security flaws due to the open nature of the blockchain protocol. This means that when designing the system architecture and its implementation, we must pay extra attention to security so that such attacks do not occur.

An example of one such attack that can be carried out by manipulating the oracle system is the flash loan attack. The attack got its name because it targets DeFi applications that offer flash loan opportunities, i.e. a new type of uncollateralized loans executed on smart contracts. This algorithm was created by the Aave protocol, one of the leaders in the DeFi category.

Traditionally, there are two types of loans - secured loans that require some collateral, and unsecured loans that require no collateral. An example of an unsecured loan is a cash loan from a bank, where some banks offer loans based on transaction history and creditworthiness, without the need for collateral. However, if the loan amount is large, such as a loan for an apartment, the bank will ask for collateral, for example, a mortgage for an apartment, vehicle, etc. to reduce the risk.

Flash loans are essentially unsecured loans that require no collateral, no credit checks, and no limit on how much you can borrow, as long as you can repay the amount in the same transaction.

Flash loans allow users to borrow as much as they want even though they have no equity. For example, if we need a large amount of Ether tokens, the protocol will pay us the entire amount, but this does not mean that it is automatically ours. We must do something with the borrowed funds to pay the loan back to the protocol and potentially take the remaining surplus for ourselves.

For this to work, the process must happen quickly, and the debt must be paid on time, otherwise the transaction will be aborted and reversed. Decentralized "lenders" do not require collateral because there is an agreement to disburse funds that are secured by the blockchain. Attackers of such systems find ways to manipulate market data while still adhering to the rules of the blockchain protocol.

One such attack occurred on the **PancakeBunny** platform in May 2021, which caused the token price of this platform to fall by more than 95% from its previous value.

The attacker initially borrowed a large amount of BNB tokens through **PancakeSwap** and used this to manipulate the price ratio of USDT/BNB and BUNNY/BNB tokens on the PancakeBunny liquidity pools. This allowed the hacker to secure a large amount of BUNNY tokens, which he then suddenly released into the market, thus causing the price to drop. He then repaid the debt through the **PancakeSwap** platform. The data suggests that the hacker stole nearly $3 million in profits this way, leaving the wounded protocol behind.

The largest flash loan attack occurred in 2021 in February when the **Alpha Homora** protocol lost **$37 million** using the **Iron Bank** lending platform.

The hacker constantly borrowed the sUSD token from the Iron Bank platform through the Alpha Homora decentralized application, doubling the borrowed value each time. This was done in a process that involved two transactions where the hacker loaned back his funds to the Iron Bank platform, allowing him to earn **Yearn Synth sUSD** (cySUSD) tokens in return.

Then, he borrowed **1.8 million USDC Coin** (USDC) tokens from the Aave protocol through a flash loan and exchanged them for sUSD tokens using the Curve protocol. The sUSD token was used to back the flash loan and the loan to Iron Bank, allowing it to constantly borrow and repay more and more funds and earn a proportional value in the cySUSD token each time. The hackers repeated this process many times, which allowed them to steal large amounts of Creamy cyUSD tokens which they would then use to borrow other cryptocurrencies from the Iron Bank platform, thus borrowing 13 thousand Wrapped Ethereum (WETH) tokens, 3.6 million USDC tokens, 5.6 million USDT tokens, and 4.2 million DAI tokens.

As we can see, the processes in Alpha Protocol hacking can be very complex and require a series of steps that must be performed very quickly, which only shows how far hackers are willing to go.

7. EXAMPLES OF BLOCKCHAIN SOLUTIONS FOR REAL-WORLD CHALLENGES

7.1. SECURE DATA SHARING

Blockchain, with its properties of decentralization, transparency, immutability and security, offers robust solutions to real-world problems. By enabling secure data sharing, improving transparency, and reducing the need for intermediaries, blockchain provides significant benefits in sectors such as supply chain management, healthcare and identity verification.

Blockchain's role and application in secure data sharing lies in its decentralized architecture that ensures that data can be securely shared without relying on a single trusted authority. Traditional centralized systems often have security vulnerabilities such as hacking and data theft, manipulation, or unauthorized access. Blockchain solves these problems by distributing data across a network of nodes, where each node contains an identical copy of the data, and changes are made only when consensus is reached. This makes blockchain ideal for scenarios and situations that require a high level of data security, such as financial transactions, medical records, or personal identity information.

One use case can be presented in inter-organizational collaboration - blockchain is particularly useful for inter-organizational data sharing, where multiple parties require access to the same information but may not fully trust each other. For example, in the energy sector, blockchain allows energy producers, consumers, and regulators to securely share information about energy consumption and production without relying on a centralized third party.

In the example of "Delicata" brand chocolate, which can be found in domestic stores, we can see the real application of blockchain technology in the real world. This chocolate is part of the "Tony's Open Chain" movement, which aims to improve the process of collecting cocoa beans and making chocolate from start to finish and to prevent the exploitation of workers. They achieve this by recording the entire supply chain process, from the collection of the first cocoa beans to the end result on the blockchain through **ChainPoint** technology.

7.2. TRANSPARENCY AND ACCOUNTABILITY

One of the most significant advantages of blockchain is its inherent transparency. As all data on the blockchain is publicly visible and permanently recorded, it is almost impossible to alter transaction records without leaving a trace. This makes blockchain ideal for industries where transparency is key, such as public governance, corporate audits, or monitoring social impact.

In supply chain management, blockchain allows stakeholders to track products from their origin to their final destination in real-time, as exemplified by ChainPoint technology. This transparency ensures accountability at every step, preventing fraud, counterfeiting, and inefficiency. For example, in the food industry, blockchain can track the entire product lifecycle, ensuring food safety, regulatory compliance, and ethical sourcing of materials. The immutable nature of blockchain records ensures that all participants, including consumers, have access to the same accurate, real-time information about a product's journey.

However, although this is a great benefit for some systems, there is a trend of implementing the blockchain protocol in more and more systems. Blockchain is a specific solution for certain systems and industries, but that doesn't mean it solves all problems. In some systems, privacy is required, and the property of blockchain to store all transactions so that they are publicly visible prevents the use of this technology for certain purposes.

Although blockchain wallets are inherently anonymous, if the identity of the wallet is somehow linked to the identity of a person in the real world, all privacy is lost, and anyone can see all of that person's financial transactions.

Therefore, if we are building a system where privacy is key and we don't need a high degree of decentralization, like internal company systems for processing payments, we don't need technology like blockchain, but in that case, centralized systems would be quite enough.

7.3. REDUCED RELIANCE ON INTERMEDIARIES

Blockchain reduces reliance on intermediaries by enabling direct **peer-to-peer** transactions. In traditional systems, intermediaries such as banks, brokers, or insurance companies often have to establish trust between parties for transactions to be confirmed. However, this also adds cost and complexity, which often leads to inefficiencies. Blockchain eliminates the need for intermediaries, allowing parties to transact directly using smart contracts, where predefined rules are automatically applied without human intervention.

Blockchain-based identity systems allow individuals to have full control over their data, reducing reliance on centralized authorities like government agencies or social platforms for identity verification.

Blockchain can securely store personal data, allowing users to share only necessary details with third parties. For example, a person applying for a loan can prove their identity without having to reveal sensitive information such as their full address or social security number. This decentralized identity verification model improves privacy while maintaining trust.

The way it can be achieved is through the technology "**Zero Knowledge Proofs**" - proving without knowledge, which allows some information to be confirmed without revealing it to the other party. An example of this would be issuing a loan, where the bank needs to know if the customer has enough funds to repay the loan. Using Zero Knowledge Proof mathematical algorithms, the user can prove to the bank that he has enough funds, without revealing the amount of funds.

7.2. BLOCKCHAIN IN HEALTHCARE

In healthcare, blockchain technology can revolutionize medical record management by providing a secure, tamper-proof system where patients have complete control over their data.

Traditionally, medical records have been segregated across institutions, creating inefficiencies, potential omissions, and data loss. The decentralized and immutable nature of the blockchain ensures that all medical data is stored securely and accessible only to authorized individuals, such as doctors or patients. Patients can also grant access to data to other healthcare professionals, ensuring continuity of care while preserving privacy.

Blockchain is also increasingly being used to ensure the integrity of medical supply chains. By tracking pharmaceutical or medical devices from production to distribution, blockchain helps prevent counterfeiting and ensures that all products are sourced ethically and safely.

This transparency is particularly critical for vaccines and similar drugs, where any discrepancies in procurement or storage can have life-threatening consequences.

8. CONCLUSION

The integration of real-world data into blockchain systems represents a key evolution in the development of decentralized technologies. As blockchain systems strive to expand beyond the digital environment into industries such as finance, healthcare, supply chain management, and more, the ability to reliably integrate off-chain data becomes indispensable. To achieve successful integration, it is necessary to explore the various challenges, technologies, and solutions involved in enabling real-world data access to blockchain networks in a decentralized, secure, and efficient manner.

The history of decentralized financial systems and the philosophy behind blockchain technology underscore its core promise: trustless, immutable, and transparent systems that remove the need for centralized authority. Smart contracts represent a key innovation, enabling digital agreements on the blockchain. However, the inherent limitations of blockchain systems, and especially their inability to natively communicate with external data, represented a fundamental obstacle to fully realizing their potential and extending it to a wider mass of users. This is where oracle systems step in as a solution, playing the role of a bridge between on-chain and off-chain data.

The importance of real-world data is great - for blockchain systems to scale and solve real-world problems, they need to be able to communicate with external information, whether that information is price feeds, weather forecast data, or stock market information. However, guaranteeing that this data is accurate, timely, and resistant to manipulation is a complex challenge, especially when maintaining the decentralized nature of the blockchain.

The introduction of decentralized oracle networks (DONs), such as Chainlink, has helped address the risks of centralized oracle systems. By distributing trust among numerous participants, decentralized oracle systems ensure the safe and reliable delivery of external data to blockchain networks, mitigating individual points of failure.

The key challenge when integrating this data is blockchain determinism and synchronization of "off-chain" data. For the transaction record to remain consistent, all nodes must agree on a single "truth", which becomes difficult with changing external data. Oracle Networks ensures that this data is accurately recorded and stored on the blockchain protocol while preserving determinism across all nodes.

This paper also describes various types of external data ranging from numeric data feeds, such as price feeds for DeFi applications, to complex API responses. Each type introduces its challenges in terms of achieving consensus and secure data access, with solutions such as Chainlink functions facilitating data availability.

A cost-benefit analysis of implementing these systems highlights the trade-offs when using real-world data. While the benefits include opening up new opportunities for decentralized applications, risks such as manipulation of the oracle system and costs such as transaction fees must be considered. Striking a balance between decentralization and efficiency is a key challenge, because the more decentralized systems are, the slower and more expensive they become, and harder to maintain and innovate.

Finally, real-world data use cases in supply chain management, healthcare, or identity verification show how blockchain systems integrated with off-chain data can solve real-world problems. The ability of the blockchain protocol to ensure secure data sharing, transparency, and reduced reliance on intermediaries offers significant advantages to all industries that benefit from such systems.

We can therefore conclude that, while there are still challenges with determinism and decentralization, decentralized oracle systems have made significant progress in solving these issues. The ability to securely integrate real-world data into blockchain systems will be critical to the expansion of distributed systems and decentralized applications that will solve real-world problems from various sectors. Even security flaws that arise along the way, and hacker attacks, contribute to this system development even more to prevent them in the future, so innovations in the design of such protocols are constantly happening.

The future of decentralized systems is optimistic.

9. LITERATURE

[1] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2009.

[2] Steve Ellis, Ari Juels, Sergey Nazarov, “ChainLink, A Decentralized Oracle Network”,
2017.

[3] Vitalik Buterin, “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform”, 2014.

[4] Website “https://ethereum.org/en/foundation/”

[5] Website “https://chain.link/education”

[6] Website “https://docs.redstone.finance/docs/introduction/”

[7] Website “https://docs.diadata.org/”

[8] Website “https://www.tonysopenchain.com/our-approach/our-processing-models”