LedaCoin
(LDA)

www.ledacoin.com

Summer 2022

Last Update: July. 2022

Abstract

Smart contracts are going to change the world's businesses by providing an infrastructure for creating Dapps (decentralized applications) and DAOs (Decentralized Autonomous Organizations) in order to automate jobs based on blockchain technology. However, one should note that due to underlying consensus protocols, the blockchain on which smart contract is deployed and triggered, cannot access external data. In order to solve this problem, a kind of special technology was proposed which is called Leda. Ledas inject information of the real world into the blockchains whenever needed. Without Ledas, blockchains can be only used for tokenization purposes; however, by setting Leda technology on a blockchain, smart contracts can be programmed as required and perform as a decentralized autonomous organization.

Content

List of figures

1 Introduction

Smart contracts are a kind of application with tamperproof context and clauses which are deployed and executed on blockchain networks. It means that no party (and even creators) are able to manipulate the codes and functions after deployment. Thus, unlike traditional paper contracts or digital ones which are programmed on centralized platforms that are exposed to alteration, termination and deletion by a trusted party or a third person, smart contracts bring parties into agreement and generate a novel and powerful class of trust without relying on trust in any party or intermediaries. This feature turns smart contracts into a superior tool for realizing and implementing digital agreements.

It should be noted that smart contracts are trying to digitalize real-world agreements. Consequently, in order to carry out such a task, they need to access real-world data. However, due to special underlying consensus protocols, blockchains are avoided to connect external data sources, therefore they cannot access outside. Thus, smart contract developers encounter a connectivity issue according to which majority of smart contracts are not able to function practically. Without connectivity, smart contracts and blockchains are only tools and platforms, respectively by which tokenization of shares and assets of organizations are possible and they are not flexible enough to be used as a programming ecosystem.

Leda system is a vital requirement with which blockchain platforms acquire the flexibility of handling all kind of applications and digital autonomous organizations by accessing external data. Ledas are a kind of technology through which people are able to inject real-world data into their smart contracts. The most useful data source that is utilized as data feed in order to inject information, is HTTP/HTTPS API endpoint. As it was declared before, due to consensus mechanisms used by blockchains, they are unable to directly fetch such critical data.

Because making smart contracts to be externally aware and capable of accessing off-chain resources is a vital aspect, if they are expected to be substituted with manual and digital agreements in use today.

The features due which Leda Coin system shows off, are consisted of following clauses:

- The ability to access external data using various form of APIs and parsing helpers such as JSON, XML, HTML.
- The ability to add extra various data sources such as BTFS, WolframAlpha, Random, etc.
- The ability to add various kinds of proof to prove the authenticity of the injected data.
- The ability to develop a decentralized Leda service upon current infrastructure.
- The ability to add dedicated Leda data carriers for special enterprises in order to sell dedicated raw data.
- The ability to accept two various payment methods including native coin of BSC blockchain network (BSC) and project related token (LDA) with a special discount.

Before proceeding to the main project, examples of potential nextgeneration smart contracts and their needs to external data are presented which are expected to be substituted by their traditional equivalent:

- Normal Contracts: For example, contract of promoting a video to reach a certain number of views. When the views of the video reach to a certain number, promoter should receive money. Requirements: Accessing to number of views using scraping by xpath language.
- Securities: For example, bonds, interest rate derivatives, etc. Requirements: Accessing to APIs which report market prices and market reference data such as interest rate.
- Insurance: For example, if fire extinguisher system of a building gets the ability to record fire data, estimate the destruction and inject it into insurance smart contract, insurance contracts can pay the compensation. Requirements: Accessing to IoT (Internet of Things) data feeds.
- Trade finance: For example, when transit of goods is completed to a special region, freight is paid using smart contract. Requirements: Accessing to GPS data about shipment and alike information in order to fulfil contractual obligations
- Dapp: Generally, any kind of decentralized application which is operated by smart contract as backend codes. One can implement any kind of application using smart contracts in which any kind of information may be required.
- DAO: Generally, any kind of decentralized autonomous organization in which usage of any kind of information is possible.

The underlying reasons of using smart contracts for developing Dapps and DAOs are as follows:

- No need to backend codes on centralized servers.
- Handling security of contract by blockchain technology.
- Providing possibility of tokenizing your platform and create dedicated assets.
- Using peer to peer payment methods through native coin and tokens of the same blockchain network.
- Registering transactions publicly and transparently which can be used as receipt by customers etc.

Creating smart contracts without Leda technology is almost impossible and without Leda technology, blockchains use case is limited to be a tokenization platform. Although powerful enterprises are able to have dedicated Leda systems, people with limited budget, ability, time or any other reason are unable to prepare one. However, as you well know, in all over the world, small businesses are more important than strong organizations and great economic enterprises. Thus, providing an

infrastructure for attracting small businesses and implementing their businesses on BSC blockchain is critical and vital for progression and expansion of the network. For instance as emerging data centers and server services providers together with ready-to-use content manager systems including WordPress, Joomla, etc. led to spread of implementing small businesses on internet easily, creating a public Leda system on BSC network will cause people to be able to implement their local or small business on blockchain, as well and benefit from peer to peer payment methods of BSC network using native coin and tokens. In the following, Leda Coin is presented.

2 Leda Coin

2.1 Leda Structure

Leda Structure As it was said previously, Leda is a kind of technology with which blockchains are able to access real-world data. Simple schematic structure of Leda Coin system can be seenin **Figure 1**:

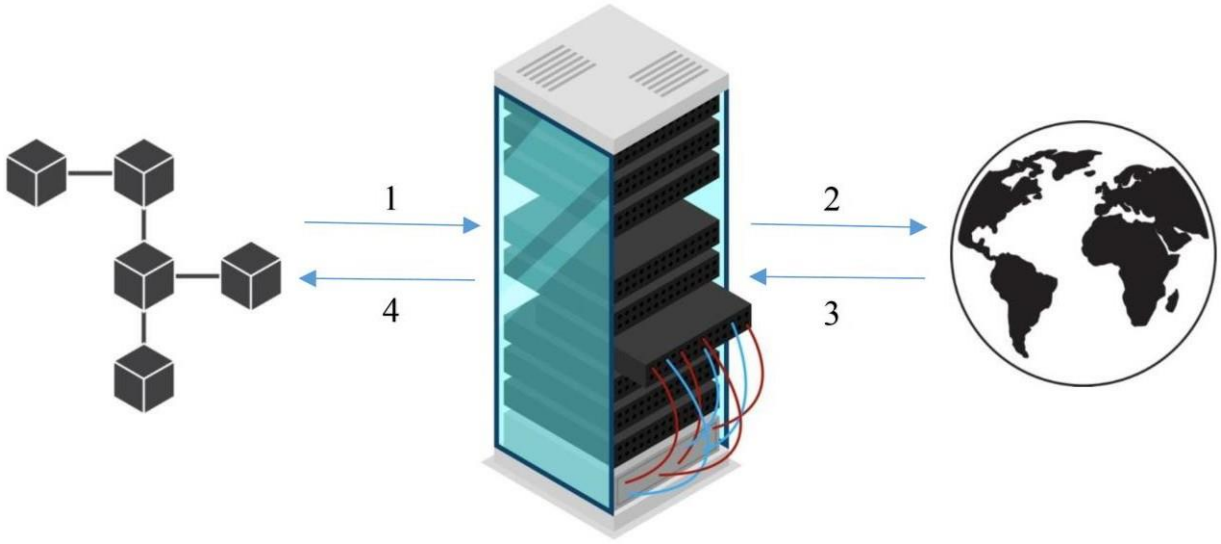

Figure 1: Simple Schematic Structure of Leda Coin System

2.1.1 Smart Contracts

Leda Coin is consisted of 3 various smart contracts that interact with each other. One of the contracts is the Leda API contract that client's smart contract should inherit from this contract through which client's smart contract can connect to Leda Coin and benefit from provided services. The other contract is Leda Coin address resolver which redirects client's requests to the correct service including public Leda system, decentralized Leda system, a specific enterprise Leda system, etc. The last contract is Leda Coin connector that receives client's request and process it in format of a query and emit specific data which can be read by watching Leda data carriers. Notice: Note that interactions between Leda's smart contracts are established in such a way that upgrading Leda Coin system to a higher version with more features and without occurrence of any disruption in previous users' contract is possible.

2.1.2 Data Carriers

Data Carriers After emitting information in blockchain by Leda Coin connector contract, Leda Coin data carriers that are watching connector smart contract, receive a group of information from which they recognize blockchain requirements of realworld data.

2.2 Leda Performance

Let's investigate how an Leda system works generally. By considering Figure 1, in the first step, blockchain emits special data using logging events that inspires its requirements of real world-data. In second step, Leda data carriers that are watching such events, understand blockchain's needs

of data and try to fetch them from real world by sending requests to various data sources using different kind of methods such as APIs, downloading files, asking questions, etc. In the third step, intended resource respond to the Leda data carriers' request and send requested data to the Leda data carriers. In the last step, Leda data carriers inject final result to the blockchain and eventually blockchain's need for external information is met. **Figure 2** shows interaction of smart contracts of Leda Coin system with together and with Leda data carriers in details. As it can be seen in Figure 2, in the first step, proper API contract should be imported into user's contract based on owner's requirements. There are various API contracts, each has a certain parameter written in it from which user's contract connects to the appropriate connector contract. After importing proper API contract to user's contract, in second step and before sending query to the proper contract, user's contract needs to know address of the appropriate connector contract in order to send the query. So, asking the address of the proper contract is carried out using the certain parameter whichis sent to Leda address resolver contract from utilized API contract. In third step Leda address resolver contract finds the appropriate connector contract address using a hash table that is provided in it. This hash table receives the certain parameter which mentions the proper connector contract, and returns the address of the connector to the user's contract. In fourth step, user's contract sendsthe query to the determined connector address on which a special kind of Leda data carriers including public, decentralized or enterprise are watching. In fifth step the connector contract sorts input data in a standard pattern and logs corresponding event based on user's needs and requirements. In sixth step valid Leda data carriers that are watching the connector contract, receives user's request in format of events. In seventh step valid Leda data carriers send request to the data source which is requested by user's contract and ask for required data. In eighth step requested data source responds to data carriers' requests and delivers appropriate data. In ninth step which is the last step of the procedure, Leda data carriers inject users' requested data into their smart contracts. Note that final results are injected to_callback function of users' smart contracts which are only allowed to be triggered by Leda's valid Leda data carriers in order to avoid injecting fake data by evil and fraud sources.

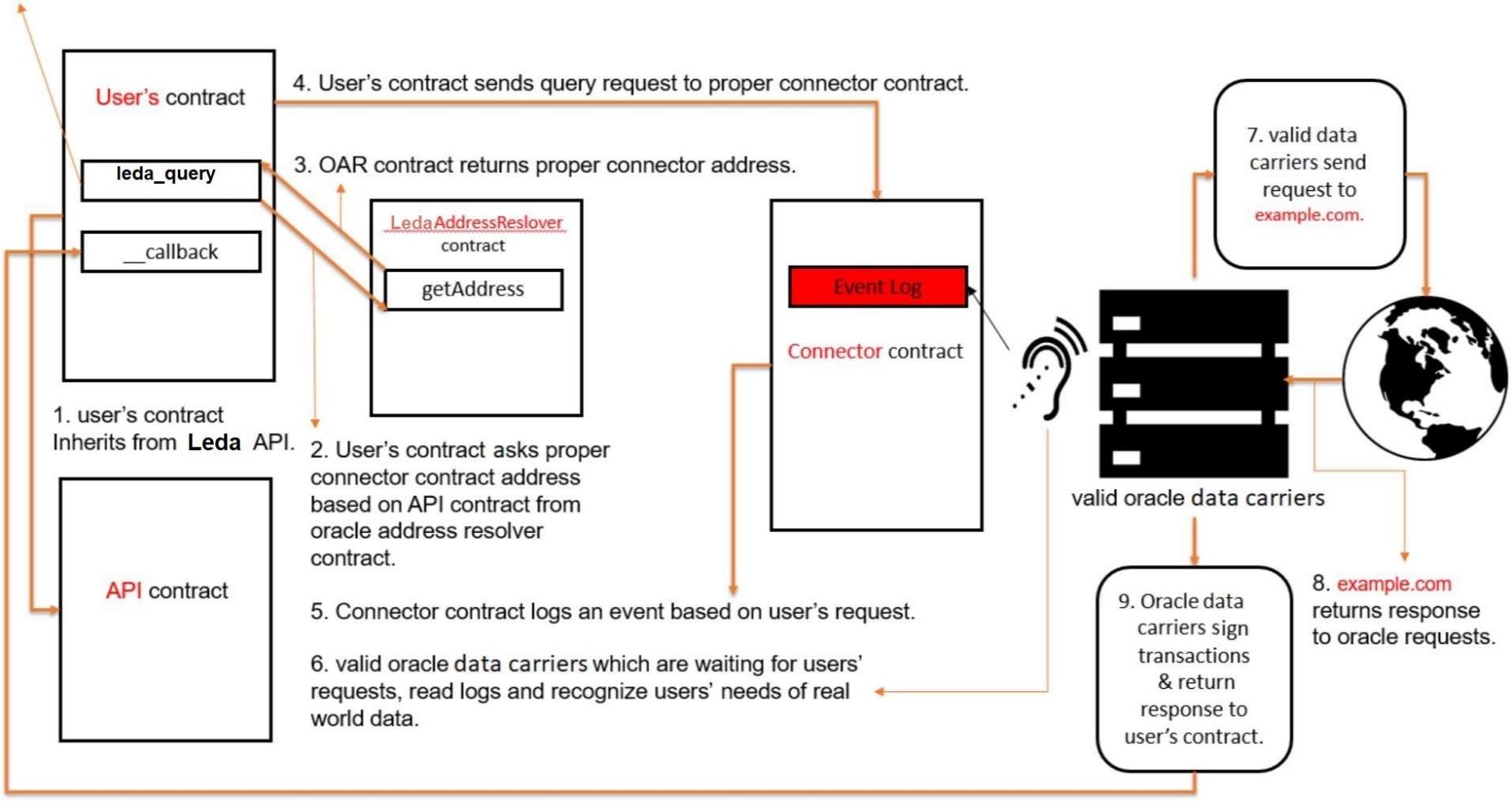

Figure 2: Interaction of Smart Contracts of Leda Coin System with Together and with Leda Data Carriers

2.3 Leda API Contracts

As it was said in previous section, proper API contract should be imported into user's contract based on owner's requirements. So, there are various API contracts in Leda ecosystem, each has special functionality. Various API contracts include:

- Public API contract
- Decentralized API contract
- Enterprise API contracts that are dedicated to each company or organization which is volunteer to sell data for LDA token It should be noted that using each of above contracts depends on user's requirements.

2.3.1 Public API Contract

Public API contract has been designed to simply obviate requirements from external data that the validity of data or proving the validity of data is not that important for user. Of course, in this method various kinds of proof will be provided to resolve proof of validity of data in a near future.

2.3.2 Decentralized API Contract

Decentralized API contract has been prepared for those users who need to implement a fully decentralized platform. Thus, they need to fetch external data using a decentralized Leda system as well. As developing such infrastructure is in our future plans, details of designing such Leda system will be reported in future.

2.3.3 Enterprise API Contract

Enterprise API contract has been provided for enterprise companies and organizations that own dedicated and special data which are not published and shared publicly. These companies and organizations can sell such data using Leda system and users who need such data can import that company's dedicated API contract. This API contract redirects users' requests to that company's connector on which company's dedicated Leda data carriers watch and respond users' contracts directly and inject dedicated data to their contract privately. Companies and organizations can exchange their dedicated information for LDA token which is paid by users' contracts. In fact, enterprise organizations can make money by selling data to Leda users. This is one of Leda system's business plans. Notice: Note that only public API contract is prepared yet and other Leda systems are going to be created in a near future based on roadmap plans.

2.4 Public Leda Data Sources

Data sources are various kind of trusted references such as a website or a web API from which required data is requested by Leda data carriers based on users' choice. One should note that each data source has a special use case and functionality and choosing a suitable data source is of great importance. There are several critical data sources that are expected to exist in Leda Coin system. Various data sources that are supported by Leda's public Leda system include:

- URL
- Complex URL

- WolframAlpha
- Random
- BTFS
- Nested
- Etc.

2.4.1 URL

This data source is a general data source that enables users to fetch every kind of data using http/https request APIs that related responses are in format of one of the following cases: 1. JSON 2. XML 3. HTML

2.4.2 Complex URL

URL data source is able to fetch a parameter from http/https request APIs that is consisted of a solo set of JSON/XML/HTML parameters. However, there are special cases in which one encounters an array of multiple set of JSON/XML/HTML parameters. For instance, an array consisted of multiple set of parameters that relates to matches of an event. Suppose that we need to know the winner of a match with a special matchID that exists in the array. In order to extract a data like matchWinner using a deterministic parameter like matchID, such data source is provided.

2.4.3 WolframAlpha

This data source redirects users' requests and questions to computational knowledge engine of WolframAlpha Company which is able to answer what one wants to calculate or know about. WolframAlpha computational intelligence answers the inquiries and corresponding responses are returned to the users' smart contracts.

2.4.4 Random

This data source generates random numbers and injects them into users' contracts. Random number has many critical use cases including statistical sampling, computer simulation, cryptography, completely randomized design, scientific calculations, etc.

2.4.5 BTFS

2.4.6 Nested

This case enables users to utilize the combination of different types of data source or multiple requests of the same data source that returns a unique result into users' smart contracts.

Notice 1: It should be noted that only URL data source is ready to use for now and other data sources are being developed to be available in a near future.

Notice 2: Off-chain architecture has been developed based on polymorphism provision according to which additional data sources can be implemented easily. Thus, in order to have more practical data sources in Leda Coin, users of all around the world can suggest new ones and Leda teamwill provide them in turn based on priority.

Notice 3: Note that above data sources are those references that are accessible publicly and data which are returned in such data sources are open to all. In these cases, Leda Coin is only a gateway to access to such public real-world data. However, in order to access dedicated data by special companies and organizations, people need to import company's dedicated API contract in

order to connect to their dedicated connector contract and Leda data carriers and receive their dedicated rare data directly from their data carriers in exchange for LDA tokens.

2.5 Responding Time of Requests

2.5.1 On Time Queries

By using this option in users' contract, responses will be returned by Leda Coin as soon as possible. When transactions of users' requests are confirmed and corresponding information is emitted in connector contract and they are received by Leda data carriers, the requests are processed right away and the results are returned immediately.

2.5.2 Scheduled Queries

Using this option enables users' contracts to submit queries in connector contract which should be responded in the future and after a certain period of time or on a certain time stamp. For example, users are able to ask Leda Coin to return price of BSC/USD after an hour or even tomorrow at 10'oclock through setting the exact time stamp.

2.5.3 Open Ending Time Queries

Open Ending Time Queries This part has been classified as both sections 2.4.7 and 2.5.3. This is due to the fact that this section is a distinct data source which has special use cases. Also requests that are sent using this feature, has a specific time-dependent functionality. For instance, consider a snooker game match. No one is able to predict ending time of the match exactly. Now suppose that a person wants to inject winner of the match and carry out some operations based on the winner. However, match is not finished to have a static result and no one can even guess the ending time of the match yet. In such cases there are three approaches based on user's opinion and intended server's capability:

1. Web socket protocol
2. Long polling method
3. Recursive HTTP/HTTPS request method

Users should utilize first method in cases that using web socket protocol to make connection to API link is possible. This method is an optimized approach and receiving response through this approach into users' contract is the most affordable method in comparison to the others. In this method, Leda Coin data carriers make a connection to the target link using web socket protocol and waitsfor a deterministic parameter to be responded. In above example, Leda Coin will wait for winner parameter of the snooker match to be fulfilled. As soon as the match is finished and Leda Coin receives intended response, it terminates the connection and injects received data to user's contract. Long polling method is a similar (inefficient) approach which is provided in old servers in instead of web socket protocol.

Now consider that a user needs to access a data from a link with similar condition; however, making a web socket or long polling connection to the link is impossible. In this cases recursive HTTP/HTTPS requests method should be utilized. Before using the method, user should estimate the ending time of the event. Then the user should determine the time step after each, Leda Coin ask the latest result from intended link for intended parameter. By setting these two elements in user's contract, process is started and Leda Coin tries to get the intended result in several efforts.

After a certain time, which was predicted by user, Leda tries for the first time to get the result. If Leda achieves the result, it will terminate the connection and returns the result. Otherwise it will hold the request and try after the time step again. This process will be continued until the Leda get intended result. As soon as Leda Coin gets the result in its latest query, it will terminate the job and injects received data to the user's smart contract.

2.5.4 Recursive Queries

This section is a special case that is not anything new in comparison to above time dependent functionalities; however, this part has a special use case that should be considered. This option has been provided for injecting any kind of data using various methods repeatedly and continuously based on an intended time step. For instance, one can consider a DEX which wants to have up to date price of BSC in its smart contract, continuously. In such smart contract BSC price is injected to blockchain every n seconds which is addressed as time step. The approach to resolve such cases is recursive queries. In order to investigate implementation of recursive queries in smart contract using Leda Coin, one can refer to Leda document in Ledacoin.com website.

2.6 Off-Chain System Architecture

In this section, off-chain system architecture is presented. **Figure 3** shows off-chain system architecture, schematically

.

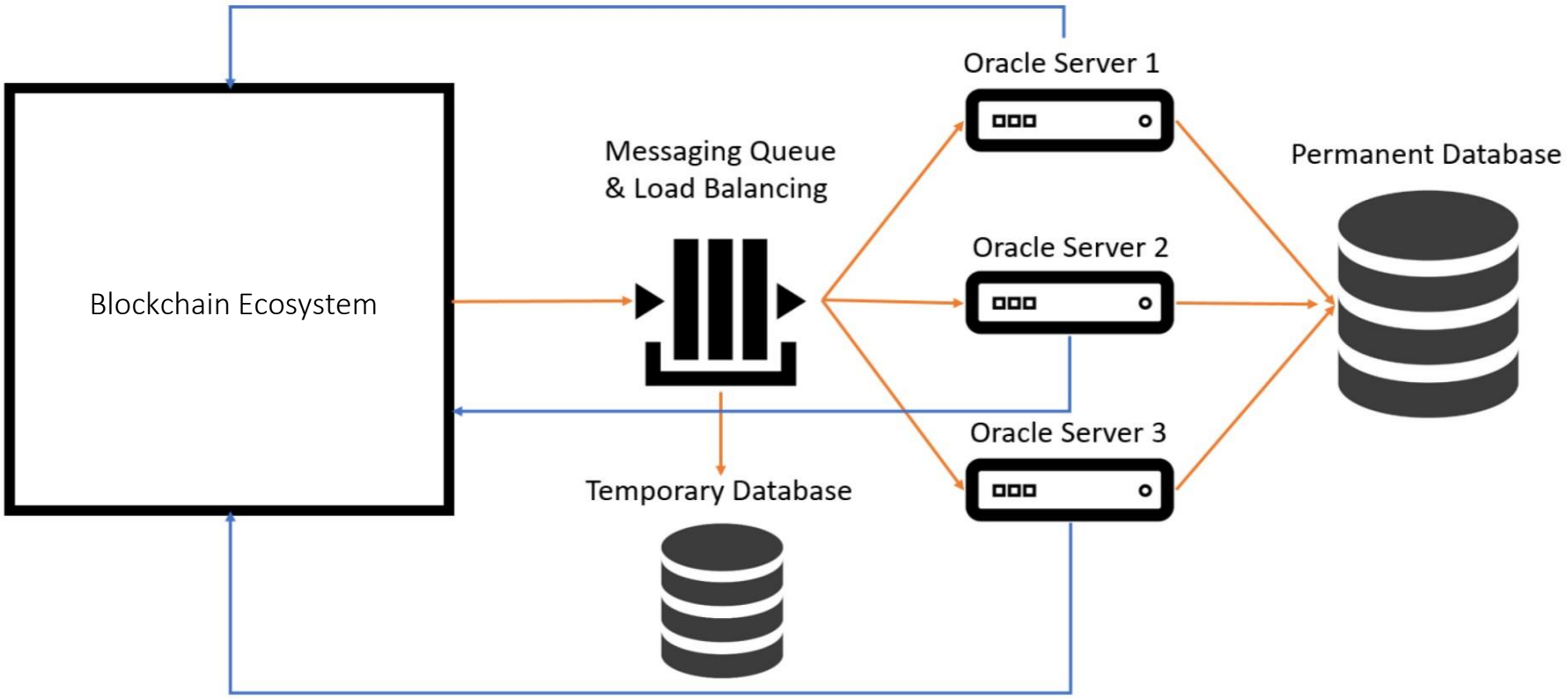

Figure 3: Schematic of Leda Coin's Off-Chain System Architecture

2.7 Payment Methods

Payment Methods In order to place a request of injecting data in Leda Coin system, users must pay the cost. Leda Coin provides two various payment methods for users to pay the cost of their requests including:

1. LDA (Leda Token)
2. BSC

Price of a request is calculated based on a certain algorithm and the amount is paid automatically from user's contract. In fact, in order to use Leda Coin system, user's contract is charged by LDAor BSC.

Before discussing payment process, note that price of the request is calculated based on BSC. Thus, in order to have the price based on LDA token, up to date exchange rate of LDA/BSC should be available every moment for our pricing system with which LDA-based price can be easily calculated. For solving this issue, a bot has been developed to watch LDA/BSC exchange rate continuously and compare it with the rate which was injected to Leda Coin's pricing system, previously. If the difference is more than 1 %, this bot updates the price and injects the new exchange rate to the blockchain and cost of next requests are calculated based on the new exchange rate.

In first step Leda Coin checks LDA balance of user's contract. If there is enough balance in user's contract according to the cost of the request, Leda will charge user's contract by LDA token, automatically. If there is not enough LDA balance in user's contract, in second step Leda Coin will check BSC balance of user's contract. This time if there is enough BSC balance in user's contract, Leda will charge user's contract by BSC. If there is not enough LDA and BSC balance in user's contract, user's request will be rejected and no response will be reported into user's smart contract by Leda Coin system.

Notice: Charging user's contract by LDA token include a percentage of discount in comparison with BSC payment method. In fact, paying LDA token is more affordable than paying by BSC for placing a request in Leda Coin system.

2.8 Roadmap

Due to complexity of the system and use cases, providing a prioritized roadmap with exact details is impossible; however general purposes are provided as following:

- Providing a feedback section in Leda Coin's official website, Ledacoin.com to receive us-ers' suggestions.
- Adding various data sources based on priority.
- Adding various kinds of proof in order to prove that users' query is not manipulated before it is delivered into their smart contract
- Providing complete compatibility with various social media apps such as Instagram, Telegram, etc. by adding dedicated data sources. Predicted sections includes:
    - Sending private post requests using user's encrypted access token that should be sent on behalf of user's identity.

    - Sending public post requests that doesn’t matter the owner of access token such as number of views for an Instagram post.
- Wide marketing in order to establish enterprise and dedicated Ledas for companies and organizations which are able to sell special and dedicated data.
- Developing a decentralized Leda system for both people who wants to participate as a node in Leda Coin and people who needs to have a completely decentralized smart contract.

3 Conclusion

Smart contracts are going to transform businesses and automate activities of an organization; however, without feeding real-world data into smart contracts, this case is impossible. Thus, a technical system is needed through which real-world data is injected into smart contracts that are deployed on decentralized blockchain networks. This technology is Leda system.

BSC network claims to have the best platform for developing Dapps and BSC foundation is promoting BSC network, continuously in order to attract more and more developers to increase the number of Dapps on the network and subsequently widen its ecosystem and community. Of course, BSC network stands in second place of blockchains ranking from number of Dapps viewpoint with almost 750 Dapps which is a good overall success; however, in Ethereum blockchain that stands in the first place of blockchains ranking, more than 3000 Dapps have been established and such a difference between number of Dapps of first and second placed blockchains is egregious. Such a big gap originates from an important deficiency which is felt significantly in BSC network. There should be a practical solution to fill the gap, based on which BSC is enabled to outpace Ethereum and stands on first place of blockchains ranking. The solution is providing a public Leda system. Unfortunately, there is no public Leda system in BSC network, thus public users cannot participate in BSC ecosystem as an active user and from ordinary users' point of view, BSC network is only a tokenization ecosystem. It can be easily said that creating Dapp for an ordinary user without a public Leda service is almost impossible.

As declared at the beginning of the context, in all over the world small businesses are more important than strong organizations and great economic enterprises. Thus, providing an infrastructure for attracting small businesses and implementing their businesses in BSC blockchain is critical and vital for progression and expansion of the network. Let's discuss an analogy in web-based projects. As emerging data centers and server service providers together with ready-to-use content manager systems including WordPress, Joomla, etc. that led to spread of implementing small businesses on internet extensively, creating a public Leda system on BSC network will have a similar effect and cause the ordinary users to be able to implement their local and small businesses on blockchain, as well and benefit from peer to peer payment methods of BSC network using native coin and tokens, adaptively.

Leda project is the public Leda system which has been developed to solve such issue and provide such services.

4 Appendix

4.1 LDA Tokens

Leda Coin has issued its dedicated tokens dubbed Leda Token (LDA) in BSC blockchain. The purpose of this token is creating a more affordable payment method for clients to pay for Leda Coin services with a significant discount in comparison to pay with BSC.

4.2 Token Distribution Policy

Token distribution will be carried out based on **Figure 4** The basic principle which has been observed in distribution policy is implementing maximum dispersion and avoiding centralization. In this procedure 20 % of the tokens will be sold and distributed between people from all over the world in an initial Exchange Offering (IEO) event which is being held on bw.com on 15th June 2022. It should be mentioned that 15 % of the tokens have been presold in private sale event phase. 10 %of the tokens are dedicated to founder team of Leda Coin and 5% of tokens are dedicated to advisors and consultants. Eventually, 10 % of the tokens have been dedicated to marketing and business development of Leda Coin, 10 % for software/hardware development, 10 % for opera- tions including establishing new branches, collaborations, etc. and at last 20 % for reserved staking pool for incentive related operations for decentralized Leda system.

Figure 4: LDA Tokens Allocation