Cardano - What is it and how to Start Working with it

Sven Barac*, Ivica Botički*, Gabrijela Perković*, Vjekoslav Radošević* and Ivan Terzić*
* Faculty of Electrical Engineering and Computing, Zagreb, Croatia

Cardano is a popular blockchain platform known for its balanced approach to security and decentralization, as well as its use of smart contracts and scientific approach. In our research of the Cardano network, we analyzed its architecture and capabilities, including transactions, metadata, and wallets. Our research aimed to answer the question of the potential applications of the Cardano network in modern information systems. The result of our work is a publicly available API that enables data transfer to the Cardano network, allowing for implementation in other programs. Our research provides valuable insights into the possibilities and mechanisms of using the Cardano blockchain for modern information systems development.

Key words: Cardano, blockchain, software development

I. INTRODUCTION

Blockchain is a digital system for recording transactions and information so that it is too difficult or even impossible to alter or hack the system. It is a form of distributed ledger technology (DTL) - a protocol in which information is shared and duplicated across a network of computers in different locations, specified by its unmodifiable cryptographic signature known as hash. Its research started in 1991, but its became widely known after its first use for Bitcoin in 2009.

Blockchain is made of blocks linked together, encrypted and ran on several thousands of nodes all over the world. Blocks are compiled information stored on blockchain, with every block being duplicated and distributed across the blockchain network of computers using peer to peer network. Each block consists of its own cryptographic hash, the hash of the preceding block, timestamp of the moment of its creation, and transaction data. A hash is an immutable signature of a block generated by a mathematical function. By having stored the hash of the preceding block and the timestamps, a chain of all the blocks and its timeline can be made. Since blocks are formed in a chain, they are basically irreversible. Any alterations of one block would lead to alterations of all the other blocks.

Node is a core network stakeholder. It is a physical device that maintains the copies of the blocks [1]. Nodes keep all the copies in sync to prevent any malicious actions, store them onto own local disk, encrypt the data, take on new blocks and check their validity. They are the source of trust for this type of a system. They are characterized by redundancy, storing copies of the same information multiple times, and fidelity. When a node receives a block, it needs to verify it. Two of the most famous methods are proof of work and proof of stake [2]. In the proof of work method clients go through a mining process, which requires solving a complex math problem faster than other miners to add a block and get a reward. It is extremely energy consuming, and it could lead to centralization and manipulation due to miners gathering into mining pools. With the proof of stake method, a validator is chosen based on a specific algorithm. The validator gives a stake which they lose in case they try to manipulate the transaction data. This method consumes a lot less energy, ensures decentralization and invites more individuals, rather than “pools” to participate.

Decentralization is one of the most significant elements of blockchain. A common, but very flawed approach to storing data is one in which a single person, or a company owns a database, and the database is stored in one place. There is space for human malicious management over the database or an error in management. Data could also be mistakenly erased or rewritten. Even without taking into account the human impact, a simple power outage might lead to the entire system collapsing, while a natural disaster or a war could destroy it completely. In blockchain there are multiple copies of data all over the world, making such an outcome highly unlikely.

Hacking blockchain would require a different approach to database hacking. Modification of a block happens through consensus of the most. If one was to make an alteration of a single node, the blocks would cross-reference each other due to being synchronized, meaning that copies of the same block must have identical content at all times, and it would become apparent which node has incorrect information. To validate the alteration, hacker would have to control more than half of the copies of the block and simultaneously make changes of all of them [3]. That would be immensely expensive and most likely not feasible due to complexity. Even if the hackers managed to perform such a task, the other network members would notice such a big change in the system and react accordingly.

Blockchain has a variety of uses, including banking, healthcare, property monitoring, voting, and food production tracking for certificates of authenticity. It is used to manage more than 10 000 cryptocurrencies, including the most dominant Bitcoin and Ethereum. It works as decentralized banking system where each transaction over the currencies is stored in several blocks and verified. Transactions are transparent, meaning that they are traceable to everybody, but they are encrypted and anonymous. Moreover, one of the applications of blockchain are smart contract. They allow all participants to digitally send their portions of the deal, and upon gathering all the portions,

the contract would automatically be executed or the transaction would be stopped if its deadline expired or some other inconvenience occurred.

Despite blockchain being a promising technology, it does come with challenges. Firstly, it requires power for all the computers to run, especially when it comes to mining Bitcoin. Annual consumption of energy for mining Bitcoin amounts to somewhat more than the annual consumption of energy of Argentina. Secondly, the use of blockchain is limited due to low transactions per second rate and data storage capacity, so it could not be used as a general database. This system also provides a basis for illicit activities of dark web with not regulation currently. Moreover, since there is no single authority, regulation is even more uncertain. This leaves space for manipulation, as is the case with the mining pools mentioned previously.

II. CARDANO BLOCKCHAIN NETWORK

Cardano is an open source proof-of-stake (PoS) blockchain project that aims to provide a more secure and sustainable ecosystem for the development and execution of smart contracts and decentralized applications (dapps) [4]. Cardano uses Ouroboros, a unique proof-of-stake algorithm, which is the first proof-of-stake algorithm to be developed through peer-reviewed research [4].

Cardano network also supports the development and execution of smart contracts which were introduced in 2021. Smart contracts in the Cardano network are written in Plutus programming language/platform which is based on Haskell and Marlowe [5]; a domain-specific language (DSL) for writing and executing financial contracts [6]. In addition, Marlowe is embedded in both JavaScript and Haskell.

One of the main properties of the Cardano network is its use of multi-layer architecture. This design allows for a more flexible and modular system, as well as greater separation of concerns between different types of transactions and activities on the network. Figure 1 depicts the elements of the architecture such as the node, CLI, Daedalus wallet, GraphQL API server (queryable API for applications that use REST APIs to talk to other services [8]), and the database synchronizer (the component which is used to query historical information from the Cardano blockchain through the use of a Structured Query Language (SQL) relational database [9]). Nodes communicate via the Node-to-Node IPC protocol, which enables the exchange of blocks and transactions. On the other hand, applications on a single computer such as the wallet backends communicate with the network via the nodes using the node-to-client IPC protocol.

In the Cardano blockchain, addresses are used to identify the location where funds can be sent. They are derived from the public key of a user's wallet and are typically represented as a string of letters and numbers. Different kinds of wallets can be used on the Cardano network, some of which are software-based and others hardware-based. Software wallets can be accessed as mobile or desktop applications, providing users with access to their resources. Hardware wallets, on the other hand, are considered the most secure form of data storage and can be stored as a USB device, for example.

Transactions on the Cardano blockchain are used to transfer funds from one address to another. They consist of the sender's address, the recipient's address, and the amount of funds being transferred, along with some additional metadata such as the transaction fee. Transactions on the Cardano network are processed by nodes, which are responsible for validating the transaction and updating the blockchain's global state.

Metadata is an additional parameter or data that is part of every transaction which enables the creation of a clearer and better context for both the receiver and sender. (e.g. object with attributes in JSON file). Metadata is optional and can be added during the transaction submission, it can include information such as a message by the sender or a description of the purpose of the transaction.

Ada is the original cryptocurrency of the Cardano network with which users can receive and send their currencies and funds through transactions. In this way, users participate in the proof-of-stake consensus and contribute to greater security and validation of transactions and the network. When creating transactions, the fee for the transaction will be calculated and the term Lovelace will

TABLE I. CARDANO NETWORKS DIFFERENCES [14]

MAINNET	TESTNET
• „main network“ – live production version • Real transactions involving Adacryptocurrency and execution of smart contracts	• „Preview testnet” - get familiarised and play with cardano-node • “Pre-Production Testnet” - for those who are ready to run the mainnet but need to test it before running it • Uses „test Ada “ – have no real value (only for testing)
• The common multi-layer architecture • Plutus programming language for writing smart contracts • Open-source and actively maintained by the Cardano community	

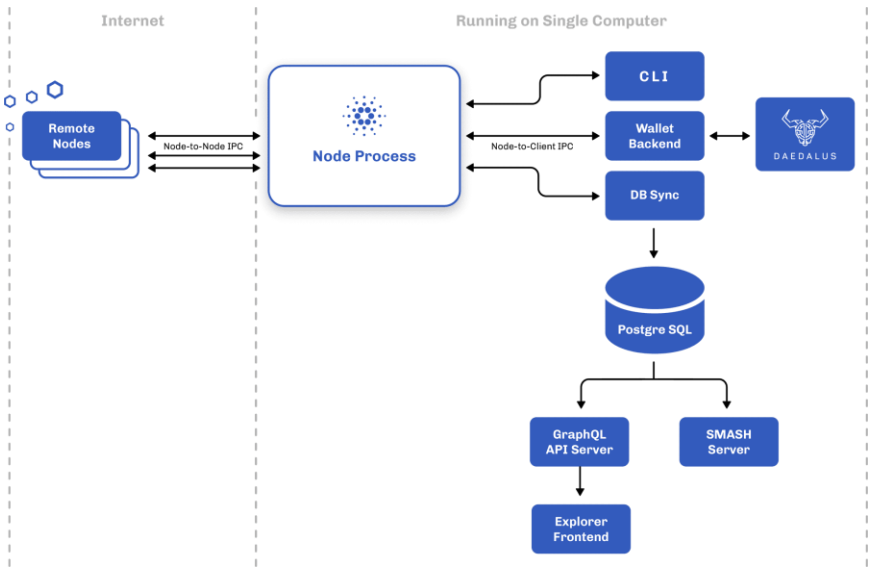

Figure 1 Cardano Arhitecture Overview [7]

be utilized. This may seem confusing, as the currency used was test Ada, and now there is a new term used. However, Ada can be expressed in smaller units named Lovelace, where one Ada contains a million of Lovelace units (similar to Satoshi and Bitcoin). In order to encourage users to actively contribute to the network and participate in the creation of new blocks, users can stake a share of their Ada currency and in return earn a reward on their stake. Staking is a way to support the network and help secure it, but also allows holders of Ada to earn a return on their investment. The total supply of Ada is limited and currently stands around 34 billion coins [10], which is the upper limit and cannot be exceeded. The mainnet and the testnet of the Cardano network refer to two different instances of the Cardano blockchain network.

In the Cardano network, the term epoch is the period of time in which new blocks are created and put on the network. Each epoch is divided into slots, where in each slot, there is a possibility of the creation of a new block. The duration of an epoch is approximately 5 days and contains approximately 432,000 slots, thereby ensuring the stability of the network [11]. At the end of the duration of each epoch, a slot leader is chosen to create a new block through the process of "epoch transition". The slot leader is elected by the stakeholders of the network proportionally to the amount of Ada they have staked. The more Ada they have staked, the more likely they are to be selected as a slot leader. This is to guarantee that the leaders are well distributed and also incentivize users to hold and stake their Ada. After a slot leader places a block on the blockchain, the process is repeated for each slot in the epoch.

III. Installing and Configuring a Cardano Node

When installing and running an instance of the Cardano node, the first thing that needs to be taken into consideration are the hardware specifications of the machine on which the node will be run. Cardano node installation turorial [12] suggests an AMD x86 or processor with at least two cores running at a minimum of 1.6GHz and on top of that, a minimum of 16GB of RAM along with 75GB of free disk space as prerequisites, and, in the case of using the mainnet network, these requirements increase. In practice, a machine with lower hardware specifications (e.g., 8GB of RAM) will struggle with compiling required binaries from source code but will still be able to run the node with sensible management of computer resources applied (it would be advisable to kill processes that are not needed and require a lot of memory to run, such as web browsers). While a machine with less than 16GB of RAM could run the node, it might not be able to run some tools and additional software associated with the node or the Cardano network, such as Daedalus Wallet [13]. Furthermore, while the recommended 75GB of disk memory might at first remain underutilized, the blocks can be filled in a short time interval. It is highly recommended to start with one of the testnet networks before switching to the mainnet and perform testing on the testnet before deploying on the mainnet.

A Cardano node can be run on the Windows operating system, various Linux distributions, and MacOS. In practice, running a node seems to be the easiest on a Linux distribution or on a Linux virtual machine. MacOS presents a solid option as it is based on Unix as well as Linux distributions, along with Ubuntu which is simple to set up and use. The official Cardano Developers site [12] did not have an installation guide for Windows at the time of preparing this report and finding Windows equivalents to Linux commands specified in the guide has proven to be a big challenge, so using a virtual machine, e.g. WSL2, is the best option should Windows be set as a target platform.

The installation process of a node is best covered in the official documentation [12]. As an alternative to going through the entire process of compiling the node binaries or should resources be limited, already compiled files can be directly downloaded. When compiling and further developing for Cardano, tools like Cabal, GHC, and Git are necessary. The process of compiling the binaries from the source code needs to be carried out with care by dully following the instructions as even the slightest mistake can lead to problems. Focus needs to be given to the required versions of the tools needed such as GHC or Cabal. The process also installs cardano-cli (command line interface), a key tool for interacting with the node allowing for a range of actions such as querying for a block at the top of the chain, creating key and address files, creating templates and signing the transaction.

Once installed the node needs to be run. The tutorial [14] on the Cardano Developers website contains almost all the instructions required to run the node. The node will need some configuration files which will give the cardano-node executable information on the kind of node the user wants to run [15], these files are necessary to run a node. Starting a node is basically just running a command in the terminal. As part of the startup, it is necessary to pick a folder for the database, decide on the port and IP address on which the node will be available, and submit the path to the configuration files. Writing a simple script that will start the node with the chosen arguments is also an option. Running the node in the terminal will display the node log, which will print out everything that the node is doing in the terminal.

The node will save new blocks to the database in the folder provided as an argument to the run command and this will be in a format similar to "Chain extended, new tip: 1e64e74bd7ac76d6806480a28017deb0aedd356fb61844ec95c429ae2f30c7c3 at slot 0". In a different terminal, the state of the blockchain, including the latest block in the node database and synchronization percentage, can be queried with cardano-cli commands (e.g., "cardano-cli query tip --testnet-magic 2" [14]). The testnet-magic flag signifies the testnet network (preview, preprod) on which the node is started (–mainnet flag is used for mainnet) [14]. Blocks, transactions, epochs, etc. for preview testnet can be explored via the Cardanoscan webapp [16]. To be able to further work with the transactions and submit them to the network, the node will have to be 100% synchronized with the network it is running on.

In addition to informing the user of successfully downloaded blocks, the node's log may also show some errors and quite a large number of notifications about peers. This is an integral part of how the Cardano network works using the Peer-to-peer (P2P) networking. Each node maintains its own set of peers classified into 3 different categories: cold, warm and hot peers. A newly discovered peer is initially added to a set of cold peers; they are peers without established network connection. A warm peer is a peer with an established connection, but no node-to-node protocols implemented, and only used for network measurements. The hot peer also has an established connection with the node, like the warm peer, and node-to-node protocols can use the connection between peers. Peers are managed by a P2P governor who maintains a certain number of peers of all types and promotes and demotes them depending on certain criteria so that there is always a certain number of peers available. A peer error can occur due to a node exception (e.g., an error in a node's file system), a network error (e.g., a failed TCP connection), or unexpected, adversarial behavior of the node [17].

Upon successfully starting the node, it is recommended to continue following the Cardano Developers documentation [18], [19] and install the Cardano Wallet. It should be downloaded and installed in a similar way to the node itself. While downloading or installing wallets, it is critical to download files from reliable and authorized sources in order to avoid theft of funds (much more important for the mainnet network). Although the Cardano Wallet is used for checking balance, sending transactions, getting transaction history details and more, it is not specifically needed to make a transaction on the Cardano network, or create a wallet for that matter.

The simplest way to create a wallet is using the cardano-cli executable [19]. Files required to build and sign a transaction are generated using commands address key-gen and address build. Running those commands with the required arguments will create three important files: a .vkey file, used to build an address, a .skey file, which is a private key used to sign transactions and approve them, both of them created by the address key-gen command. The third, an .addr file, is created from the address build command and is built from the .vkey file and contains only the wallet address, which is of this format on the preview testnet: "addr_test1vz95zjvtwm9u9mc83uzsfj55tzwf99fgeyt3gmwm9gdw2xgwrvsa5". With this address, test Ada can be claimed for free at [20].

IV. Developing software using Cardano-related frameworks

Upon successfully installing the Cardano node, the process of creating, signing and storing a transaction on an active Cardano block could be automated. Apart from using the comand line tools and scripts, one of the options for doing so would be to utilize a programming module that has all the functionalities the cardano-cli tool has. Cardanocli-js [21] is a JavaScript module that wraps every command of the cardano-cli for efficient interactions with the Cardano node. Installation of the cardano-cli-js is simple: the only necessary step is to download the package from the NPM repository using a Node package manager.

Since Cardano is not used for storing standalone data, every piece of information has to be integrated into a separate transaction as metadata to store it on the blockchain. Metadata is written in JSON format, with some restrictions including integer range 0 to 264 - 1 for values and $-(2^{64}-1)$ to $2^{64} - 1$ for keys. Floating point numbers, null values and boolean values are not supported and for that reason, they need to be transformed into another data type. Detailed description of metadata and transaction structures can be found in the official Cardano Docs guide [22].

Storing a transaction on the Cardano blockchain requires paying a fee which can be calculated using the cardano-cli tool. Using the cardanocli-js module, transaction fee can be calculated with a single function call while building the transaction. Following that, the transaction is ready to be signed which is done by using the peer's secret key. The secret key ensures that everybody on the network knows nobody tampered with the transaction after it was created. Formed and signed transaction is now ready to be stored on the platform. Storing the transaction is done by calling another cardanocli-js function which returns a unique transaction hash. That hash will be used later when querying the blockchain to find that specific transaction.

An important part of the interaction with the Cardano blockchain is fetching information from the network. These functionalities cannot be done using the cardano-cli-js, as they cannot be done in the cardano-cli command tool either. A solution to that comes with JavaScript support for Blockforst, an API that provides specific endpoints for querying the Cardano blockchain over HTTP [23]. In order to use the API, registered users have to request a Blockfrost token called "project_id" which is used as part of every HTTP request that will be sent to Blockfrost servers. The JavaScript module used for sending HTTP requests to Blockfrost servers is blockfrost-js [24]. Cardano mainnet, Cardano preprod and Cardano preview (testnet-magic-2) are the only supported Cardano networks from which Blockfrost can fetch the data. For displaying the remaining Cardano tokens, Blockfrost uses Lovelace currency instead of tAda (test Ada).

A great deal of information can be accessed using blockfrost-js module. The most important Blockfrost API endpoint covers fetching transaction data that also includes transaction metadata. To fetch the transaction data, an endpoint that accepts transaction hash as a unique value tied to every transaction (a cardanocli-js function return value) is used. Alternatively, metadata information can be fetched by targeting an endpoint that requires a metadata label identifying a metadata object, but its value is not globally unique so multiple JSON objects could be returned as a response.

Creating and storing transactions on the Cardano blockchain using the cardanocli-js module, as well as fetching transaction data using Blockfrost API are simple functionalities for interacting with the installed node and newly accessible network. Examples can be found as part of the CardanoPreviewTestnetUtils class which is packaged and uploaded to Npm repository as @tim1/cardano-testnet-tim1 [25]. The package can be downloaded and used on the systems that meet these requirements:

- installed Node.js version >= 12.19.0
- installed and configured Cardano node with version >= 1.26.1

V. Conclusions

Since its launch in 2017, Cardano has been one of the more widely used blockchain platforms available. Its unique mission is to establish an overall balance in the crypto and blockchain ecosystem by reworking the principles of security and decentralization, using capabilities of smart contracts, building a modular system through multilayer architecture and a scientific approach developed by IOHK (Input Output Hong Kong) for the Cardano network.

As an open-source program, its community is constantly expanding and working to improve the network. This community-driven development highlights the decentralized decision-making process and accelerates its innovation.

Decentralized application development (WEB3) is becoming increasingly utilized and since it is based on blockchain technologies, understanding basic concepts such as Peer-to-peer networking, proof of stake, proof of work, blocks and transactions in addition to having basic knowledge in development on blockchain network could prove to be beneficial. Cardano platform is a great way to introduce a developer to blockchain. Even if a programmer decides to use a different platform, the knowledge acquired while working with Cardano will be very valuable. Cardano has a big perspective and promising future since proof-of-stake is more environmentally friendly than proof-of-work because of the lower energy consumption which is becoming increasingly important in modern times. IOHK's blockchain platform has many uses and the mentioned code and usage is just the top of the iceberg.

References

[1] B. Becher, "What Are Blockchain Nodes and How Do They Work?", Sep, 2022 [Online]. Available : https://builtin.com/blockchain/blockchain-node. [Accessed Jan 10, 2023].

[2] A. Rosen, "Proof of Work vs. Proof of Stake: The Biggest Differences", Oct, 2022 [Online]. Available: https://www.nerdwallet.com/article/investing/proof-of-work-vs-proof-of-stake. [Accessed Jan 10, 2023].

[3] A. Hayes, "Blockchain Facts: What Is It, How It Works, and How It Can Be Used", Sep, 2022 [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp#toc-how-does-a-blockchain-work. [Accessed Jan 10, 2023].

[4] IOHK, "Why use Cardano?". [Online]. Available: https://docs.cardano.org/new-to-cardano/why-use-cardano. [Accessed Jan. 9, 2023].

[5] IOHK, "Plutus". [Online]. Available: https://developers.cardano.org/docs/smart-contracts/plutus. [Accessed Jan. 9, 2023].

[6] IOHK, "Marlowe". [Online]. Available: https://developers.cardano.org/docs/smart-contracts/marlowe. [Accessed Jan. 9, 2023].

[7] IOHK, "Cardano architecture". [Online]. Available: https://docs.cardano.org/explore-cardano/cardano-architecture. [Accessed Jan. 9, 2023].

[8] IOHK, "Cardano GraphQL". [Online]. Available: https://docs.cardano.org/cardano-components/cardano-graphql. [Accessed Jan. 9, 2023].

[9] IOHK, "About DB Sync". [Online]. Available: https://docs.cardano.org/cardano-components/cardano-db-sync/about-db-sync. [Accessed Jan. 9, 2023].

[10] Binance, "Cardano Price". [Online]. Available: https://www.binance.com/en/price/cardano. [Accessed Jan. 9, 2023].

[11] IOHK, "Slots and Epochs". [Online]. Available: https://developers.cardano.org/docs/stake-pool-course/introduction-to-cardano/#slots-and-epochs. [Accessed Jan. 9, 2023].

[12] R. Phair, "Installing cardano-node and cardano-cli from source," Dec, 2022. [Online]. Available: https://developers.cardano.org/docs/get-started/installing-cardano-node. [Accessed Jan. 9, 2023].

[13] IOHK, "Daedalus - Cryptocurrency wallet". [Online]. Available: https://daedaluswallet.io/en/. [Accessed Jan. 9, 2023].

[14] F. Bormann, "How to run cardano-node," Nov, 2022. [Online]. Available: https://developers.cardano.org/docs/get-started/running-cardano/. [Accessed Jan. 9, 2023].

[15] M. Szamotulski, "Understanding your configuration files and how to use them," Nov, 2022. [Online]. Available: https://github.com/input-output-hk/cardano-node/blob/master/doc/getting-started/understanding-config-files.md. [Accessed Jan. 9, 2023].

[16] Cardanoscan, "Preview Cardanoscan," 2022. [Online]. Available: https://preview.cardanoscan.io/. [Accessed Jan. 8, 2023].

[17] IOHK, "Peer-to-peer (P2P) networking". [Online]. Available: https://docs.cardano.org/explore-cardano/cardano-network/p2p-networking. [Accessed Jan. 9, 2023].

[18] T. Kammerer, "Installing cardano-wallet," Sep, 2022. [Online]. Available: https://developers.cardano.org/docs/get-started/installing-cardano-wallet/. [Accessed Jan. 9, 2023].

[19] T. Kammerer, "Exploring cardano wallets," Sep, 2022. [Online]. Available: https://developers.cardano.org/docs/integrate-cardano/creating-wallet-faucet/. [Accessed Jan. 9, 2023].

[20] IOHK, "Testnets faucet". [Online]. Available: https://docs.cardano.org/cardano-testnet/tools/faucet. [Accessed Jan. 10, 2023].

[21] alessandrokonrad, "Cardanocli-js", May, 2021. [Online]. Available: https://github.com/shareslake/cardanocli-js/blob/main/API.md. [Accessed Jan 10, 2023].

[22] T. Kammerer, "Build with transaction metadata", Sep, 2022. [Online]. Available: https://developers.cardano.org/docs/transaction-metadata/. [Accessed Jan 10, 2023].

[23] Blockfrost, "Blockfrost.io API Documentation (0.1.50)", Dec, 2020. [Online]. Available: https://docs.blockfrost.io. [Accessed Jan 10, 2023].

[24] blockfrost.io, "blockfrost-js", Dec, 2022. [Online]. Available: https://www.npmjs.com/package/@blockfrost/blockfrost-js. [Accessed Jan 10, 2023].

[25] tim1, "blockfrost-js @tim1/cardano-testnet-tim1", Dec, 2022. [Online]. Available: https://www.npmjs.com/package/@tim1/cardano-testnet-tim1?activeTab=readme. [Accessed Jan 10, 2023].