McVenture Update whitepaper.md b92e9ef on Jan 23, 2017

2 contributors

402 lines (230 sloc) 23 KB

Table of Contents

1. Introduction

1. What is Rise

We are Rise, a crypto-currency and distributed application platform heavily emphasizing security and ease of use.

Our aim is to provide Graphical User Interfaces to Decentralized / Distributed application development, Smart Contract creation and sidechain, custom coin and token creation, all backed by our Distributed Proof of Stake system.

Build your own apps and rely on us to provide the security to your blockchain, instead of having to build your own.

2. Technical Background

Rise is being developed using Node.js[1] for the application backend, Go and shell scripts for the server, through Dokku and Docker.

Node.js was selected because of its highly efficient concurency model, due to the nature of Javascripts evented model.

The database in use is MongoDB[2]. MongoDB was selected based on the NodeJS application development framework that has been selected for use, Meteor[3].

The Meteor framework uses a SocksJS based data protocol layer called the Distributed Data Protocol (DDP). This protocol was designed by the Meteor Development Group to be a highly scalable, fast, and low latency distributed data protocol.

Dokku[4] and Docker[5] were selected because of its functional ability to create entirely separate application environments through the use of LXC[6] containers in Linux. A Dapp written in any language will be sandboxed by the LXC containers, exposed to the World only on port 80 by default, and will connect to the primary Rise Node application on that server through internal networking.

React[7] is a Javascript View (from MVC) library. It utilizes a Virtual DOM with Differential updates to make changes to the page, and controls data flow with one-way reactivity. If the model changes, the view will update to change. It also uses Components to template web pages, making complex application UI development streamlined. React's use of a Virtual Dom with Differentials updates allows changes to the UI to be made quickly on a user's click. Instead of having to rerender the entire page, we can rerender just the parts that need updating.

3. Key Innovation Factors

Security is of paramount importance to Rise. Rise will use a holistic approach to ensuring the security of the network, and the blockchain. Consistently running security audits on our infrastructure, hiring a dedicated vulnerability analyst, and working with the Node Security Project to keep up to date with vulnerabilities in NPM packages are just a few of the things Rise will do. A separate security whitepaper will be released prior to a launch of testnet. We will also be using BitHound to track security over time, and to ensure we do not release insecure packages.

All Dapps will be required to use SSL encryption on their public endpoints. This will be automatically provided by an integrated LetsEncrypt package with the Node installation. Every Dapp deployed to a node will automatically have an SSL certificate generated and assigned to it. They will be renewed every 60 days.

Linux Containers allow Rise to support more than just Node as a development platform. Development in any language that can run on Linux, and consume APIs is possible. We will also be fully supporting Javascript, Python, Ruby, C#, and Java as development environments, with special containers designed specifically for those languages to connect them directly to the Rise Node through a secure, internal networking channel.

Websocket based communication between nodes. Websockets is an HTTP technology that enables two systems to communicate in near real time. This will enable some enhancements to the consensus process, potentially speeding up the process to confirm transactions.

Delegate specific node will be a stripped down node designed just for use as a delegate. This, plus the websockets, will allow delegate nodes to be hosted anywhere, even behind firewalls, without difficulty.

GUI Smart Contract and Dapp Builders will enable anyone to build Smart Contract and Dapps, greatly reducing Time to Market, and increasing the number of people who can build apps for the Rise platform.

4. Rise Components

Nodes	Developers
Decentralized P2P hosting of dapps	Rise / Bitcoin API
Decentralized P2P storage for dapps	Rise Javascript SDK
Decentralized computing	Rise Python SDK
Secure containers for dapps	Rise Ruby SDK
Sidechain consensus for every dapp	Rise C# SDK
Easy node Installer	Rise Java SDK
Easy delegate Installer	Rise UI Framework

Nodes	Developers
GUI Smart Contract Builder	
GUI Dapp Builder	
Desktop Client	
Mobile Client	
Web Client	

2. Clients

1. Rise Web

The Rise Web client comes with all Rise Nodes. This enables users to manage their account, Dapps, RISE, and Smart Contracts from anywhere with an internet connection. Security is still maintained due to the use of a Single Page Application built in React, encryption happens on the client side and gets transferred over an SSL secured connection.

Running a Web client makes it so you do not have to have the entire blockchain downloaded. Of course, if you want the visibility into the blockchain that having your own copy allows, you can setup a full node on a spare machine, a vm, a VPS, or even a Raspberry Pi.

The web client is also the basis for the Rise Desktop and Mobile applications.

2. Rise Desktop and Mobile

Rise Desktop is built on Cordova[8], using the SPA built for Rise Web. This keeps the entire client on your machine, but enables feature parity across clients.

All clients will be capable of spinning up new nodes on various cloud hosting services. If you have an account, you will be able to sign into the hosting service and let the application spin up your Node for you.

Rise Mobile is built on Cordova as well. It will have the same features, with a Mobile focused UI, as the Desktop and Web clients.

The mobile client will be available for both iOS and Android and featured in the Apple and Google Play app stores.

3. Servers

1. Rise Node

The Rise Node is the basis for the entire infrastructure. It will be installable in three ways, through a shell script on a linux system, by downloading and running a packaged VM, and an automated process from the Rise Clients.

The Rise Node contains the Blockchain, Sidechains, Accounts, is a potential Delegate, and contains Rise API and DDP endpoints.

2. Rise Delegate Node

The Rise Delegate Node is a NodeJS application designed to run on any system, anywhere, to handle delegate responsibilities only. It contains a basic Web UI that enables setting adjustment. Only a single account can access the node, which is set on first run.

4. Consensus

Rise is based on the DPoS[9] (Delegated Proof of Stake) consensus mechanism. This method of consensus was originally created by the BitShares team.

DPoS is based on delegates creating blocks. Delegates are trusted accounts which are elected to be “Active Delegates”. The delegate accounts with the most votes create the blocks, up to a maximum of 101 delegates. Other delegates are listed as “Standby Delegates”, and can advance to the active list by receiving votes from the other Rise owners. All users of Rise have 101 votes available to elect their favorite delegates into the top 101 list. The weight of each of the 101 votes is proportional to the amount of Rise the user has in the wallet the votes are cast from. This total amount is shown on the delegate list as an “Approval”, and is listed as a percentage of the Rise available that is voted for that delegate.

Delegate promotion to the active or demotion to the standby list happens at the completion of the block generation cycle. Each cycle of blocks is created by the top delegates in random order. The block time is 10 seconds. Newly created blocks are broadcast to the network and added to the blockchain. After 6 to 10 confirmations, a block, along with its transactions, can be considered as confirmed. A complete block generation cycle's timeframe is dependent on network load and the number of delegates.

In DPoS, forks can occur, but the longest fork wins. Delegates must be online all of the time and have sufficient uptime. Uptime is used to catalogue the reliability of a node by logging each time that it misses a block that was assigned to it.

Delegates can be removed from the active delegate list if they miss blocks, or if they are not on the same blockchain as the majority of the rest of the delegates. When a delegate is removed, it must correct it's blockchain to the one of the majority, and have more votes than the delegate that replaced them, to be placed back in the rotation.

Delegates that are removed from the active list due to missing a block or being on a fork are placed in a third list called "Broken". Broken delegates will automatically redownload the blockchain from the last known matching block.

1. Delegates

The function of delegates is covered above in the Consensus section.

To be a delegate, a user needs to register a delegate account. This is accomplished from the client user interface in either the full or lite wallet. Keep in mind that block generation is only possible in the full wallet. This means that you can register a delegate in either version of the wallet but will only be able to perform the delegate functions from a full version of the client. The account number and username will be the same after the delegate registration. All Rise accounts are eligible to become delegates.

New delegates start as standby delegates. Standby delegates begin with an approval rating of 0% and will need to accrue votes from the Rise community in order to advance to be one of the top 101 delegates. Block generation is performed by the top 101 delegates only. If you are in standby status, you will not forge any blocks.

2. Network fees

All valid transactions in the network must be processed. Delegates process transactions and store them in new blocks. For this work, the delegates receive a fee. All transactions in the network must contain some type of fee as a spam countermeasure.

The default network fee for sending an Rise transaction is 0.1 Rise. For example, a 100 Rise transaction includes an additional fee of 0.1 Rise for a total transaction cost of 100.1 Rise.

The following is a list of fees for different types of transactions:

- 0.1 Rise of amount sent for a spend transaction
- 5 Rise for registering a second passphrase
- 5 Rise for registering a username
- 2500 Rise for registering as a delegate
- 1 Rise to add a contact
- 500 Rise to register a dapp
- 5 Rise per member for registering a multi-signature group.

Delegates receive the fees from all transactions of the last block cycle (Number of blocks in a cycle dependent on number of active delegates, which is variable based on network load).

Fees are split equally between all delegates who created a block in that cycle. Delegates who missed creating a block assigned to them during that cycle are not paid.

3. Peer-to-Peer

We are using a standard P2P network[10], which works on top of the http protocol, facilitated by websockets, and uses json formatted data as a method of data inter-change. The P2P module captures the following information about each peer:

- Version
- OS
- IP
- Port

5. Core Features

1. Usernames

Rise allows users to register usernames. Which act as an alias to your account. Other users can send transactions to this username and the linked account will then receive it. This eliminates the need to remember long account addresses.

The network fee for username registration is 100 Rise. Usernames may contain the following characters:

- Traditional Alphabet (Upper & Lower Case): A-Z, a-z
- Numbers: 0-9
- Special Characters: !, @, $, &, and .

Each username is unique. The length is currently limited to 16 characters. Currently, it is not possible to remove a username from your account.

2. Contacts

Rise allows users to maintain a contact or friends list. This feature can be used to store frequently used accounts, but can also be used as a reputation system. If an account has many confirmed contacts, it may be considered more reputable than one without.

Contacts work like followers on Twitter. A user is added to the contact list, which will then show as a pending contact request in the user's wallet. Regardless of whether or not the other user accepts the request, they will be shown in the contact list. Once the other user accepts the request, the requester will be added to his contact list as well. Both parties now have a new confirmed contact.

The network fee for adding a new contact or accepting an incoming request is 1 Rise.

3. Multi-signatures

Rise allows users to create a multi-signature group. A multi-signature group consists of several Rise users, called group members. Transactions from multi-signature groups can be configured to require some or all signatories for approval.

To achieve this a M of N multi-signature architecture is implemented. All members of a multi-signature group (N) are added, up to a maximum of 64 signatories, and then the required number (M) of signatures needed to approve a transaction is specified.

M must be greater than 1 and less than or equal than N. N is the number of members of the multi-signature group.

Once you initiate a transaction from the multi-signature group, all members will see this pending transaction and decide whether to approve or ignore it. Once the required number of confirmations has been collected, the group will allow the transaction and submit it to the blockchain.

The owners of a multi-signature group may change the rules of the group at any time with the approval of at least M of the signatories.

In addition to the above options, the following majority options will be added.

- Simple Majority (More than half of members)
- Super Majority (More than 2/3rds of members)

4. Smart Contracts

Smart contracts enable a new era of legal and business dealings. Connect your contract to external data sources through REST APIs, and allow them to execute actions automatically when criteria is met. Development of Smart Contracts will be handled through a Graphical User Interface, and will allow for currency transactions, actions via APIs, and more.

6. Decentralized Applications

1. Linux Containers

Rise Dapps are executed using Docker, a Linux Container based system. Containers isolate applications from each other and the underlying operating system, while providing a repeatable build process. Communication between Dapps is done through an internal network. Select the Dapp you want to connect to, and Rise will connect the Dapp to your Dapp internally, and allow you to communicate with the Rise API.

Upon launching a new Dapp, the Rise Node spins up a new container, and runs your Dapp in that container. The container can auto-detect the code you are trying to run, identify the language, and install dependencies.

If you so choose, you can also isolate your Dapp from all others, so that communication is restricted.

2. Dapps

A dapp is a decentralized application[11] that Utilizes the Rise SDK and runs in a container on a Rise Node. It works with the Rise Node using the Rise consensus algorithm. With current web technologies (HTML5/CSS3/JavaScript) the developer is able to create a powerful UI. Dapps can use any technology that runs on Linux in their development of the server side of the application.

Regular users can launch the dapps on any Rise Client.

3. Dapp Development

Developers write dapps in whatever language they want, using language specific SDKs, whichever UI framework they want (including Rise UI), or using Rise API endpoints.

Each Dapp runs in it's own container on a Rise Node, which reduces much of the potential attack surface for malicious actors. Each Dapp Container is connected internally to the Rise API endpoint, so API calls to Rise are all done internally to the Rise Node, instead of using the public Internet to send API requests.

The API includes a request/reponse mode, and a streaming mode, enabling realtime updates to Dapps from the Rise Node, and the Rise Node from the Dapps.

To open a dapp, the format: `https://dapp.<nodeName>.rise.vision/<dapp_id/username>` is used.

4. Dapp Computation

Built into the Dapp Store is a resource billing system. This will allow billing for system resources on the Rise Node, to host Dapps. Resources that can be billed for are CPU, RAM, Storage, and Network I/O. What is billed for, and how much, is up to the Node Owner.

Any Dapp registered on the Rise network will be available for Node Owners to use on their own Nodes.

Invoicing will be done in Rise, and automation will be available. If invoices aren't paid, automatic shut-downs, and deletion is possible. This is all customizable by the Node owner.

5. Dapp Consensus

Each Dapp has its own unique private side chain which operates in synchronization with the Rise block time and current block height.

Dapp sidechains are managed by a group of up to 101 master nodes, each of which have block generation enabled specifically for an individual dapp. The primary role of each master node is to process transactions and signify the validity of each block generated on the sidechain.

The signing of blocks by a master node against a given dapp is restricted by the dapp owners. Whom then approve individual Rise accounts as master nodes, which then are allowed to forge on the Dapp's side chain.

Sidechain consensus is maintained among the 101 master nodes using the same Delegated Proof-of-Stake (DPOS) method used to secure the Rise blockchain. This allows individual master nodes to collect fees from each transaction as reward for securing the dapp's side chain.

The motivations behind this form of consensus are to prevent unnecessary enlargement of the Rise blockchain and to retain individual sidechain autonomy, while at the same time, ensuring the integrity of each side chain is constantly upheld.

6. Dapp Master Nodes

Dapp master nodes are Rise nodes with an installed dapp and with block generation enabled specifically for that dapp. Dapp owners need to approve individual Rise accounts to be permitted to be a master node. The nodes process transactions and generate new blocks which are then secured by the Rise Blockchain, making them the core of the dapp system.

7. Dapp Storage

Dapp storage is handled by git repositories. When a Dapp is registered with the main network, it's master branch on the git repository is used whenever the Dapp is installed. Web hooks are available to push changes to all Rise Nodes that have the Dapps installed.

File Storage can be handled with local storage on in the container utilizing docker volumes, or can use file storage systems on the web served by cloud providers.

8. Dapp Deposits / Withdrawals

Developers can use either Rise or BTC in their dapps[12]. Users of a dapp may deposit or withdraw funds from any given dapp. When Rise or BTC are sent to a dapp address, the funds appear in the dapp account. The funds will then become available for use within the dapp. This works the same way for BTC deposits as it does with Rise. BTC is sent to a special dapp address and then appears in the dapp Bitcoin wallet.

Dapp accounts are a special type of account created by the network automatically for a dapp. All deposited Rise or BTC will be stored in the associated addresses. For security reasons, only the use of multi-signature dapp accounts with trusted signers is allowed.

Withdrawals from dapps are processed by master nodes. When a withdrawal request is sent, the dapp master node processes it and moves the funds to the specified withdrawal address in the Rise or Bitcoin blockchain.

9. Dapps Tokens

Developers may implement custom tokens in their dapps, and use these tokens as the main currencies within their dapps. These tokens may be used in the same way as Rise or BTC, but the tokens cannot be moved directly from one dapp sidechain to another dapp sidechain. They must only move via the Rise main chain.

7. Appendix

Written by

- Justin R. Donnaruma

Releases

- May 7th, 2016 (v1.0)

8. Sources

- [1] Node.js Organization
- [2] MongoDB
- [3] Meteor
- [4] Meteor DDP
- [5] Dokku
- [6] Docker
- [7] LXC
- [8] Blaze
- [9] Cordova
- [10] Bitshares DPoS.
- [11] Peer-to-Peer
- [12] David Johnston. Decentralized Applications.
- [13] Sidechains. Deposit/withdrawal sidechain.

Based on the Lisk Whitepaper v2.1