Renstside

Release v0.0.1

Apr 26, 2020

Contents

1 Introduction **1**
1.1 Abstract . . . 1
1.2 Shared Economy . . . 2
1.3 An Example . . . 2

2 QuickStart Guide **3**
2.1 The Web Application of Rentside . . . 3

3 Market Opportunity **5**

4 Case Studies **7**
4.1 Rentside in Farming . . . 7

5 Architecture of Rentside **9**
5.1 The peer to peer network . . . 9
5.2 The model . . . 10
5.3 Deep dive to Rentside's Solution Architecture . . . 15

6 Technical Documentation **21**

7 Initial POC **23**

8 About **25**
8.1 Founder and Developer . . . 25
8.2 Contact . . . 25

CHAPTER 1

Introduction

Rentside info website click here

Rentside is a distributed application which serves the purpose of shared economy built on Ethereum blockchain. This gives the economy to connect various sellers and buyers directly. The economy can be anything ranging from a small grocery shop to mid side vehicle rental service to a large factory owners. All the connections are made without the use of any central system. It uses a decentralised method to form a peer to peer network of the sellers and buyers, rather producers and consumers. Producers are the ones who generates business and consumers are the ones who consumes business. The application is built on top of ethereum blockchain network. It uses this network to make a distributed ledger of various parties (nodes) connected in a peer to peer fashion. This gives an opportunity to the seller to directly contact a buyer and vice versa, in turn reduces the middleman charges and time to market of the seller's product. Once both of them are connected and agreed on a condition, the smart contract executes and creates a block which further gets chained to the ethereum public network. This happens by spending a small fee in terms of ether and the selected gas value.

1.1 Abstract

The tech world is one of the most dynamic segments of this universe. For the one moment everyone is behind technology and the next moment the technology becomes obsolete. Each day a new framework and technology is released but still, there needs to be something which should have the power to carry the mass together. D-Apps are the decentralised way of communicating to each other using mobile and web applications. In this time of world, where data becomes crucial part of any human being , we can't risk to give it to any central authority to govern and control. D-Apps solves this problem, here you hold your data and you are the sole person to make it secure using your cryptographic keys. Proof of work is one of the most important feature of blockchain, this tells us that what ever you do in your business is recorded and stored in blockchain. So if you are a musician and you play guitar and publish your music in the blockchain network, it's always yours. The proof fo work algorithm of Ethereum preserves your work. Rentside is also built using the similar ideologies of D-Apps. It is true that in today's time its impossible to completely decentralised something but we can still tend towards maximum decentralisation.

1.2 Shared Economy

To understand shared economy lets understand the fact of P2P (peer to peer) and a very important statement "What's mine is yours, for a fee" . A peer to peer network is a kind of handshake where each one of them is directly connected to each other. There is no central system, not governing power and no policies. This type of network connects owners to users with their assets which is named as P2P renting.

- Access instead of ownership: rather than buying an asset, the seeker rents it from someone else.
- A platform brings together owners and seekers and facilitates all processes between them.
- The business / platform itself does not possess any of the assets on offer: Airbnb does not own a single room, Zoom-car not a single car

1.3 An Example

To start with an example lets compare how we can leverage the features of three major E-commerce giants in Rentside using de-centralisation.

- E-Bay - It is an online E- commerce website which provides you an opportunity to list your products on its dashboard and if some buyer shows some interest they can connect to each other and proceed with the deal. You can list any kind of electronic products, second hand products, vehicles etc.
- Soundcloud- You can upload your music, share it and listen to other's music.
- Zoom-car - You can list you car if you are not using it for others to rent and then earn money.

These platforms provide you a great solution to today's major problem statements but in a centralised way. You don't have the access to the whole data fragment and 90 percentage of the data is stored in the central servers which is controlled by E-Bay, SoundCloud and Zoom-car. This leads to 2 major problems.

- Origin of products [Transparency] - You may not know from where the product is originating. May be it is a fake product or a tampered product. You lack knowledge of origin.
- Proof of work - You don'y get your reward for your genuine effort. In case of Soundcloud , you many not get your copyright if someone uses it again and again.

Rentside solves these 2 major problems by making the full framework decentralised. As the system is decentralised you have - * Data residing on public ledger- Since the data is communicated to all the ledgers connected by the peers, everyone has access to it. It is not hidden. A node can use its private key mixed with the public key to fetch the data. This solves the transparency problem which was in the centralised system. * Ethereum Proof of Work- The ethereum's proof of work algorithm keeps track of every block in the blockchain. Since, your data (your music in case of sound-cloud) is generated as a block and a miner has invested its computing resource to solve it's existence, it gets connected to the blockchain network and hence its resides there forever. This ensures you're copyright to your work is maintained.

CHAPTER 2

QuickStart Guide

2.1 The Web Application of Rentside

Rentside comes with Web application which can be accessed using any web browser. You need to install Metamask extension to your browser. These are the features which will be provided to Users-

[As a seller] * Identify yourself as a seller or a buyer, you can create multiple accounts if you wish to have both roles. But you need to have separate private keys for all the accounts. * Use your google account linked to Rentside. * Create your profile/portfolio. * Add your products/ showcase what you do and want to sell. * Advertise it. * View potential buyers in your area.

[As a buyer] * List the products you want to buy. * Create your profile/ portfolio. * Connect to potential sellers in your area. * Communicate with them using Rentside chat. * Find a deal and move forward in blockchain.

CHAPTER 3

Market Opportunity

CHAPTER 4

Case Studies

Let's discuss some of these interesting case Studies.

4.1 Rentside in Farming

CHAPTER 5

Architecture of Rentside

5.1 The peer to peer network

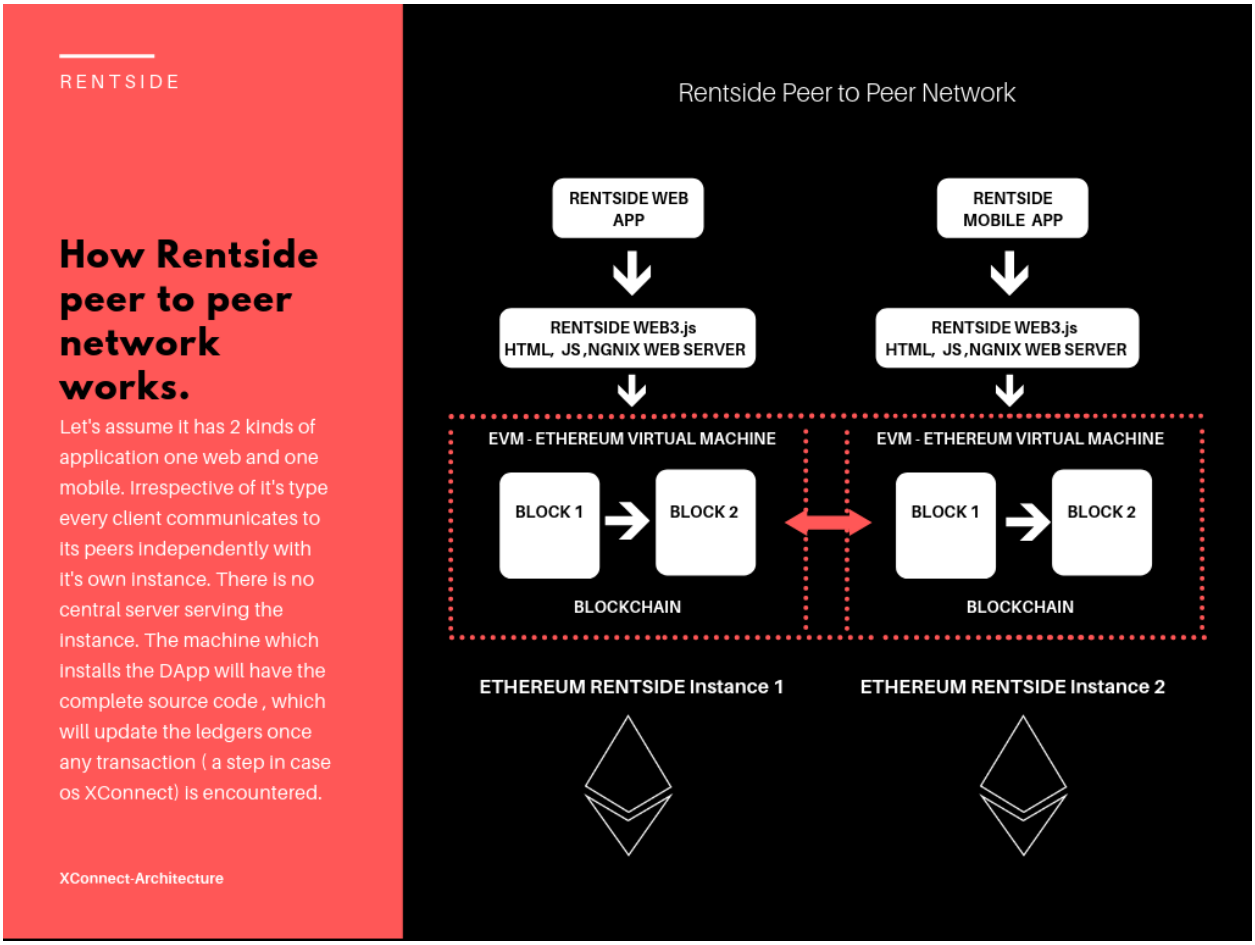
Rentside's peer to peer model works very similar to other Ethereum based decentralised applications. Each instance which runs on top of the Ethereum framework runs individually on each devices which tends to use this application. Since there is no central server, none of the instance is relied on it. The machine which uses it has all the source code. Once any transaction takes places , which means any step in Rentside , example user registration, it updates the ledgers on every node of the network.

Here we don't rely on any central system controlling and governing everything. This satisfies the first and the foremost law of the blockchain ecosystem. Let's deep dive more into this network.

Rentside web app is basically an angular based application with a backend of python and node. This needs a server to run, and as stated we don't have any central server in this model we assume that the machine which runs this application has all the dependencies installed, or more precisely a docker based platform to run the app.

So the major components of this architecture are -

- Database - All the transactions which happens around the application are stored in the Ethereum blockchain as a package of blocks which are linked together. Let's take and example - Assume someone placed an order using Rentside platform and after few days cancelled it and asked for a refund, then all these steps will be recorded in the blockchain ledgers and shared to everyone. Hence this becomes tamper less. Ethereum uses and algorithm of proof of work to ensure the integrity of data pushed into the blockchain.
- The Algorithm and Code- So where does all the code base resides ? So first of all, every login in Rentside is written in Solidity in the form of Smart Contracts. The buy, sell , refund, dispute etc is a specific smart contract individually. Each one of them is deployed in Ethereum Virtual Machine, it converts the code into Ethereum byte code and then deploys it into the ethereum blockchain.



5.2 The model

Here, the model of Rentside's architecture is discussed in terms of top view, where the three main parts of the application is shown, one is the front-end which is the end users tab, the backend APIs, the ether payment gateway called as metamask and the EVM (Ethereum Virtual Machine). Later we will discuss about its core event transactional model.

5.2.1 Top View

- The Decentralisation

When we talk about decentralisation of any distributed application, we need to understand that we cannot isolate the application from the world to make it decentralised. Rather we divide the application into many parts and then make the core decentralised. Core refers to the parts where the logics of blockchain resides. The goal is to fetch maximum decentralisation, not complete decentralisation.

- Front-End

The frontend of Rentside is build on Angular 7 with all the source code packaged into modules. This is hosted on a

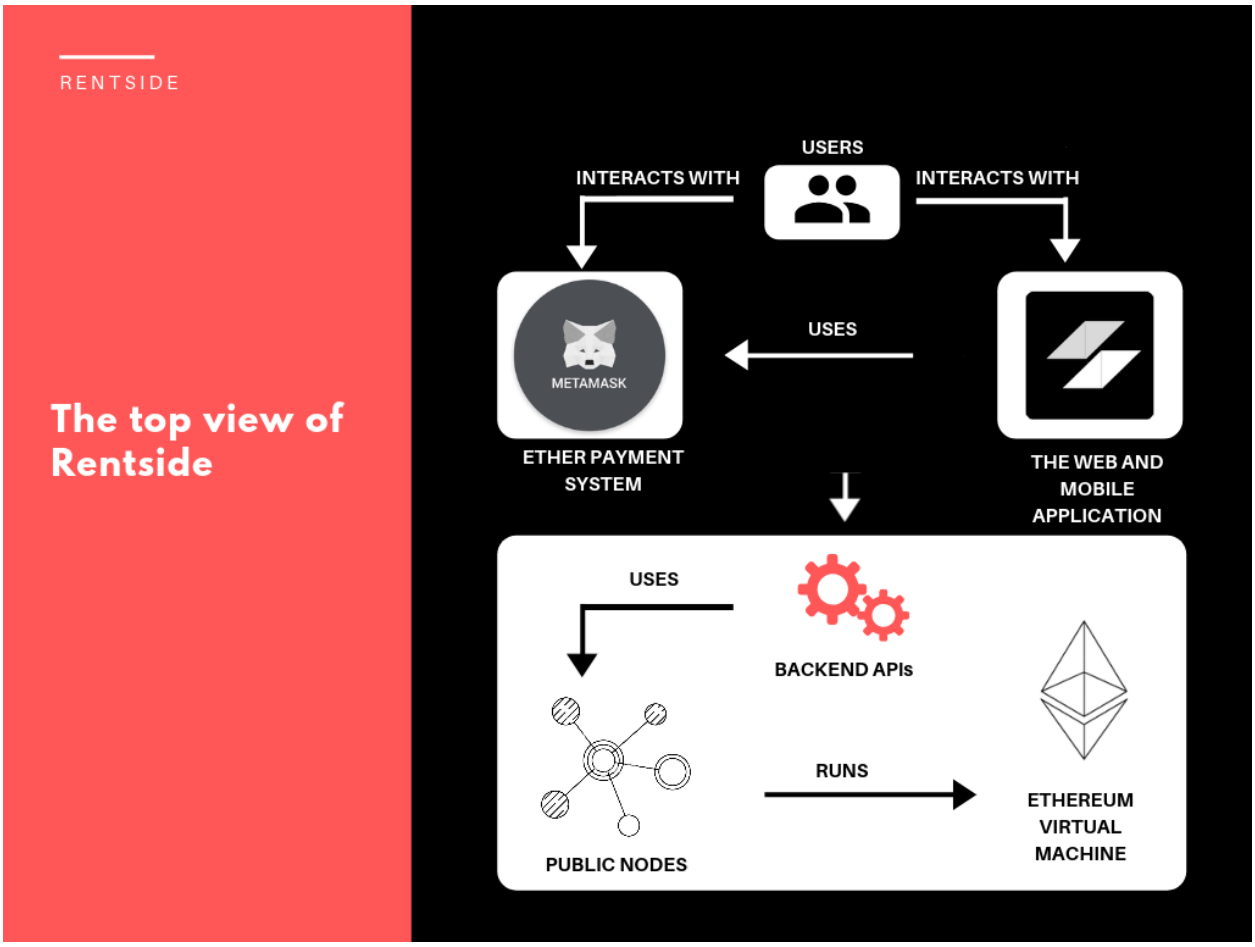
central server which connects users to its backend using Python APIs. These APIs are capable of connecting to the Ethereum network which is completely decentralised.

- Backend

The backend consists of the wrapper APIs connected to Solidity smart contracts to connect to public nodes and finally runs on ethereum virtual machine.

- Metamask

Metamask is a bridge that is used to connect the application to ethereum public network. Rentside web uses metamask chrome extension and the mobile app uses metamask mobile client.



5.2.2 Rentside's Transactional Model

- The Transaction

In Rentside transaction is referred to each step which takes place in lieu of ethers. A transaction is defined as a service which this application provides in turn of some amount of ethers. A profile creation is a transaction which will have some ether value, an advertisement is a transaction and a dispute management service is also a transaction in Rentside.

- Crypto Identify

In Rentside , Metamask is used as a crypto wallet which uses public and private key to generate tokens to push any transaction to the blockchain network. The private key gives you an identity and one can only act on behalf on this. This is always stored in your local machine which never goes online making no one able to control the identity except the owner. With the combination of the private and the public key one can perform any allowed action on the centralised and de-centralised network. This means Rentside stores all of your transactions and actions cryptographically secured so that nothing is tampered.
So as it has been already stated that due to limitations of scalability in networks we combine different approaches to achieve maximum decentralisation. Rentside uses both, decentralisation and some part of centralisation. Both plays a vital role in the architecture of Rentside. This is explained in the below topics.

- Backend

From technical point of view, there has to be a bridge that connects the smart contracts to the web/mobile applications. In today's fully de-centralised applications, where the clients interacts with smart contracts directly the bridge is narrowed down to JSON RPC API capabilities of public APIs , which in turn are forced to exist because of the fact that not every device can run and support its individual network node.

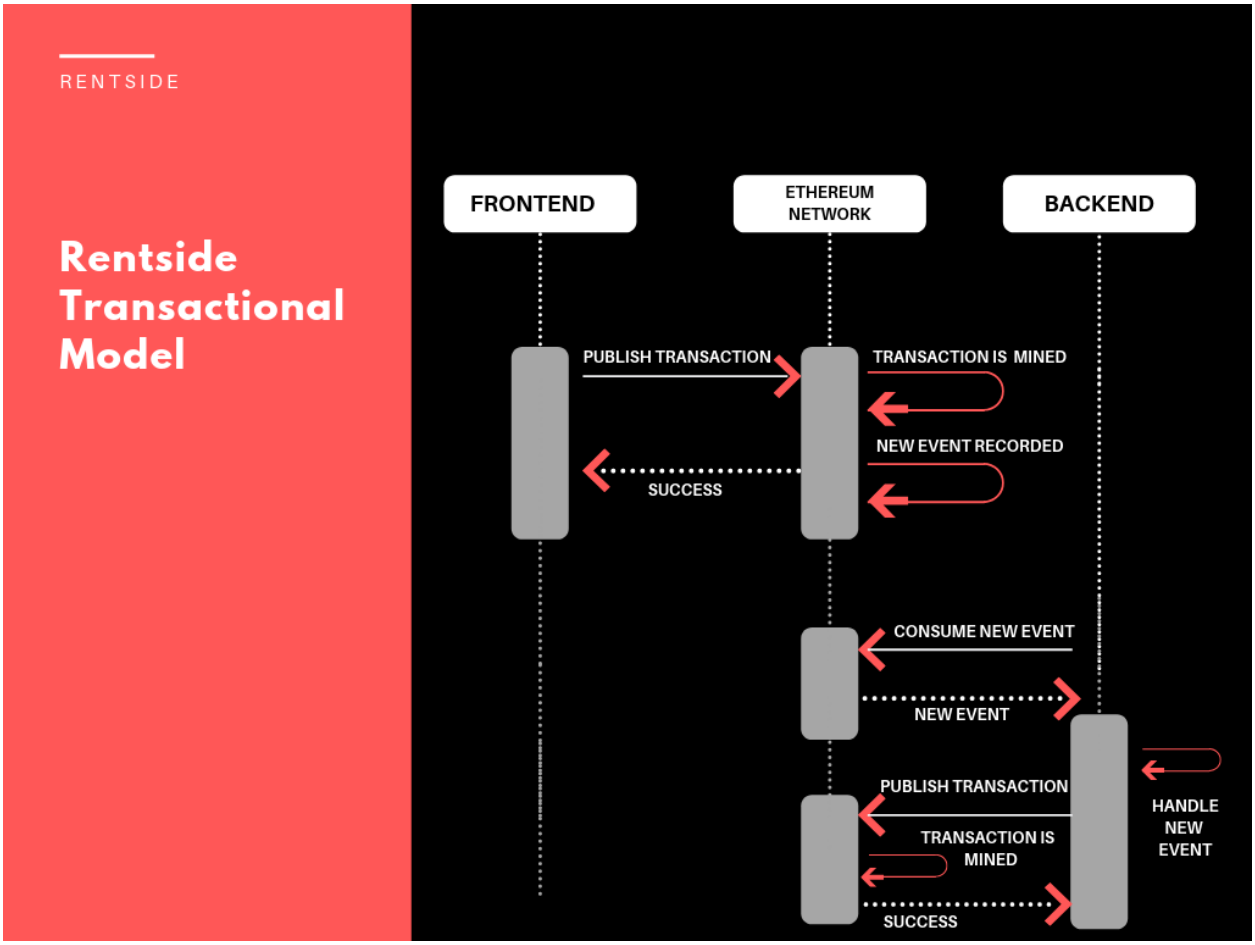
There can be two major interaction in a decentralised network.

- Listening to the network events (like token transfers) / reading the network state.
- Publishing transactions- invoking state-changing smart contract functions like token transfer

In Rentside from the backend point of view these happens-

- We listen to a particular network event by continuously polling the network.
- Once we get an event, we perform some business logic and then decide to publish a transaction in response.
- Prior to publishing the transaction, we want to ensure that it will likely be mined (in Ethereum, the successful transaction gas estimation means there are no errors against the current network state). However, we can't guarantee that the transaction will be mined successfully.
- Using a private key, we sign and publish the transaction. In Ethereum we also have to determine the gas price and gas limit of the transaction.
- After publishing the transaction, we continuously poll the network for its status.
- If it takes too long and we can't get the status of the transaction, we have to re-publish it or trigger a "fail scenario". Transactions can be lost for various reasons: network congestion, dropping peers, network load increase, etc. In Ethereum, you can also consider re-signing a transaction with a different (actual) gas price.
- After we finally get our transaction mined, we can perform more business logic if needed. For example, we can notify other back end services about the fact of the transaction being completed. Also, consider waiting for a

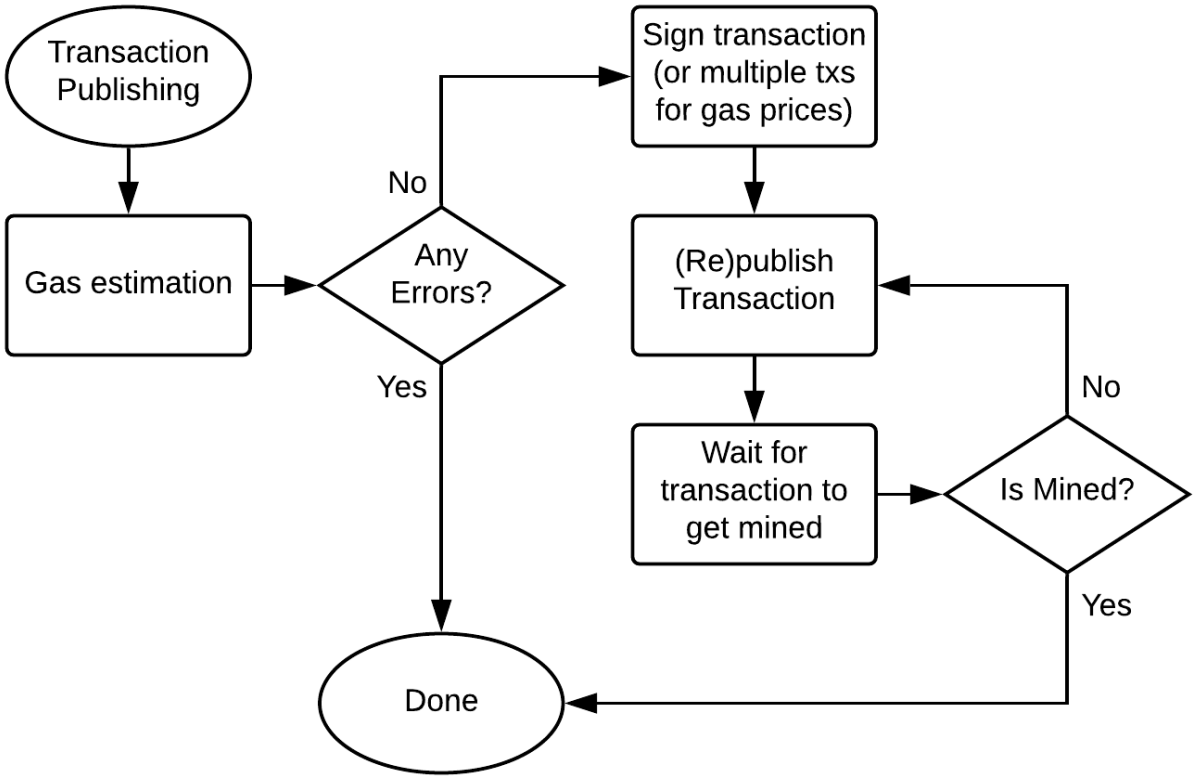
couple of confirmations prior to making final decisions regarding the transaction: the network is distributed and hence the result can change in a matter of seconds.



- Publishing Transaction

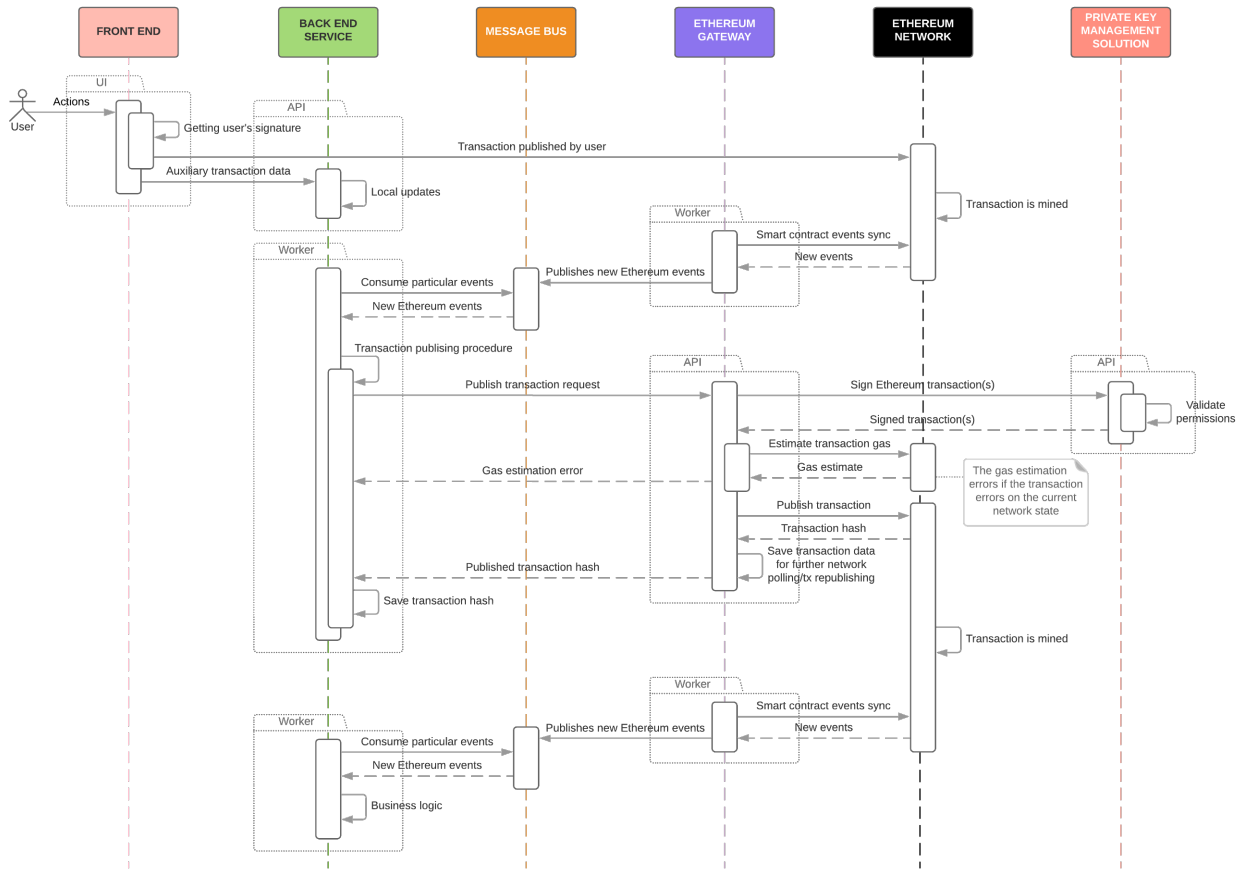
Steps

- Preparing the transaction. Along with transaction data, this step implies requesting the network state in order to find out whether this transaction is valid and is going to be mined (gas estimation in Ethereum) and the transaction's sequential number (nonce in Ethereum).
- Signing the transaction. This step implies the usage of the private key. Most likely, here you'll want to embed the custom private key assembly solution.
- Publishing and republishing the transaction. One of the key points here is that your published transaction always has a chance to get lost or dropped from the de-centralised network. For example, in Ethereum, the published transaction can be dropped if the network's gas price suddenly increases. In this case, you have to republish the transaction. Moreover, you may want to republish the transaction with other parameters (at least with higher gas price) in order to get it mined as soon as possible. Thus, republishing the transaction can imply re-signing it, if the replacement transaction wasn't pre-signed before (with different parameters).



Using the above reproach in Rentside we determine the transactional system. This is explained below.

- The user executes a function in a smart contract, which ultimately allows the back end to perform a successful charge transaction.
- A back end service responsible for a particular task listens to the event of charge allowance and publishes a charge transaction.
- Once the charge transaction is mined, this back end service responsible for a particular task receives an event from the Ethereum network and performs some logic (including setting the next charge date).



5.3 Deep dive to Rentside's Solution Architecture

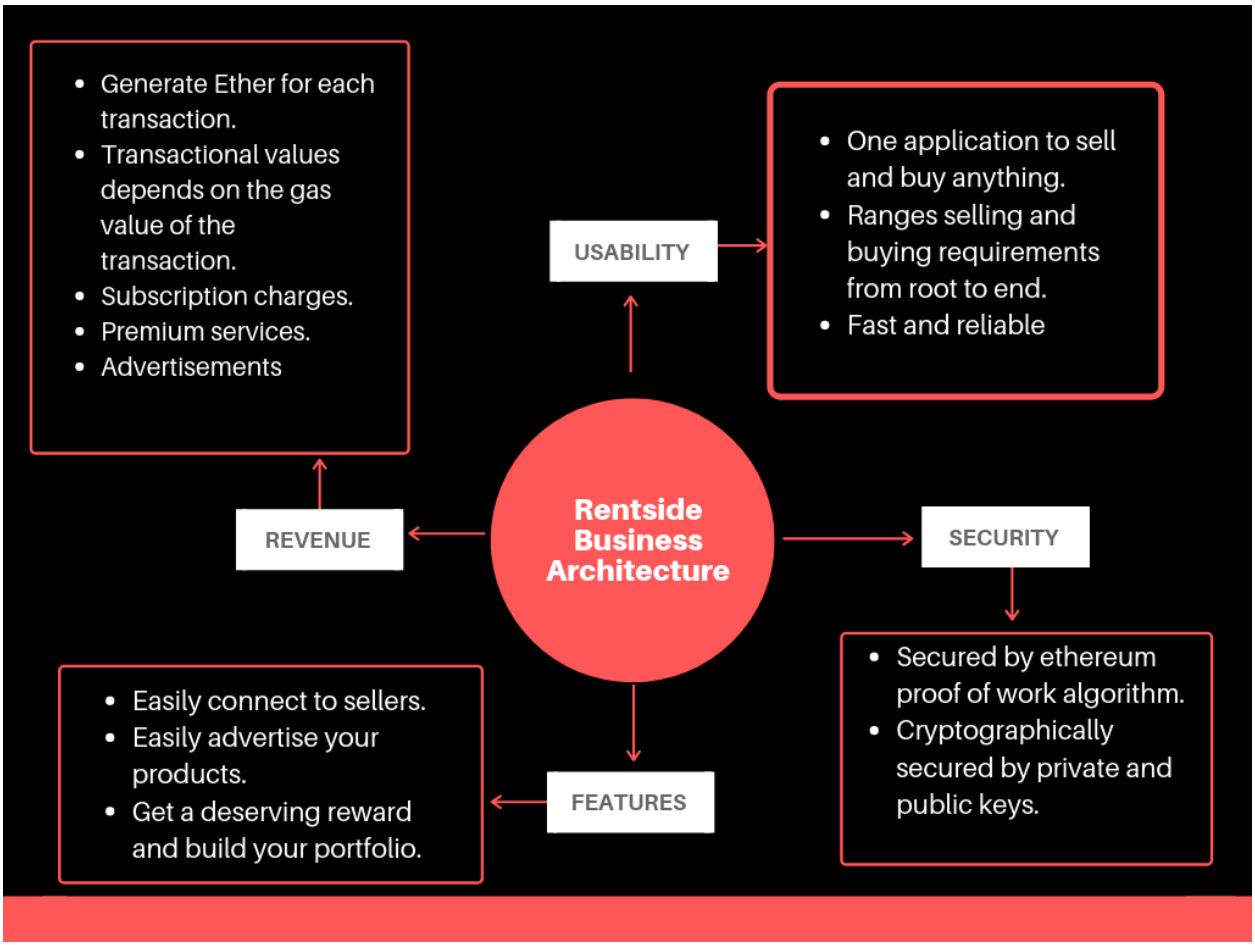
5.3.1 Business Architecture

Rentside's business architecture shows the four major areas of which can enhance the business growth of Rentside in the public blockchain market. As it has been already stated that the audience of this application not only involves the big market monopoly leaders such as Reliance industries but also includes small businessmen who make money by connecting to small backyard farmers or small fishermen. Let's discuss each of the major areas in detail.

1. Usability - This depicts the functional usability of Rentside. We claim X~ tends to everything. You can advertise any product you own or have on this platform. Even a buyer who wants to buy anything can buy it from the platform by connecting to the particular seller. You don't need to have a specific - * System to sell only groceries (don't need big-basket) * System to sell electronics. (don't need an E-Bay) * System to rent out my vehicle (don't need a Zoom-car and Drivezy) | Rentside gives you a decentralised platform where you can do all these easily. One application that perform everything.

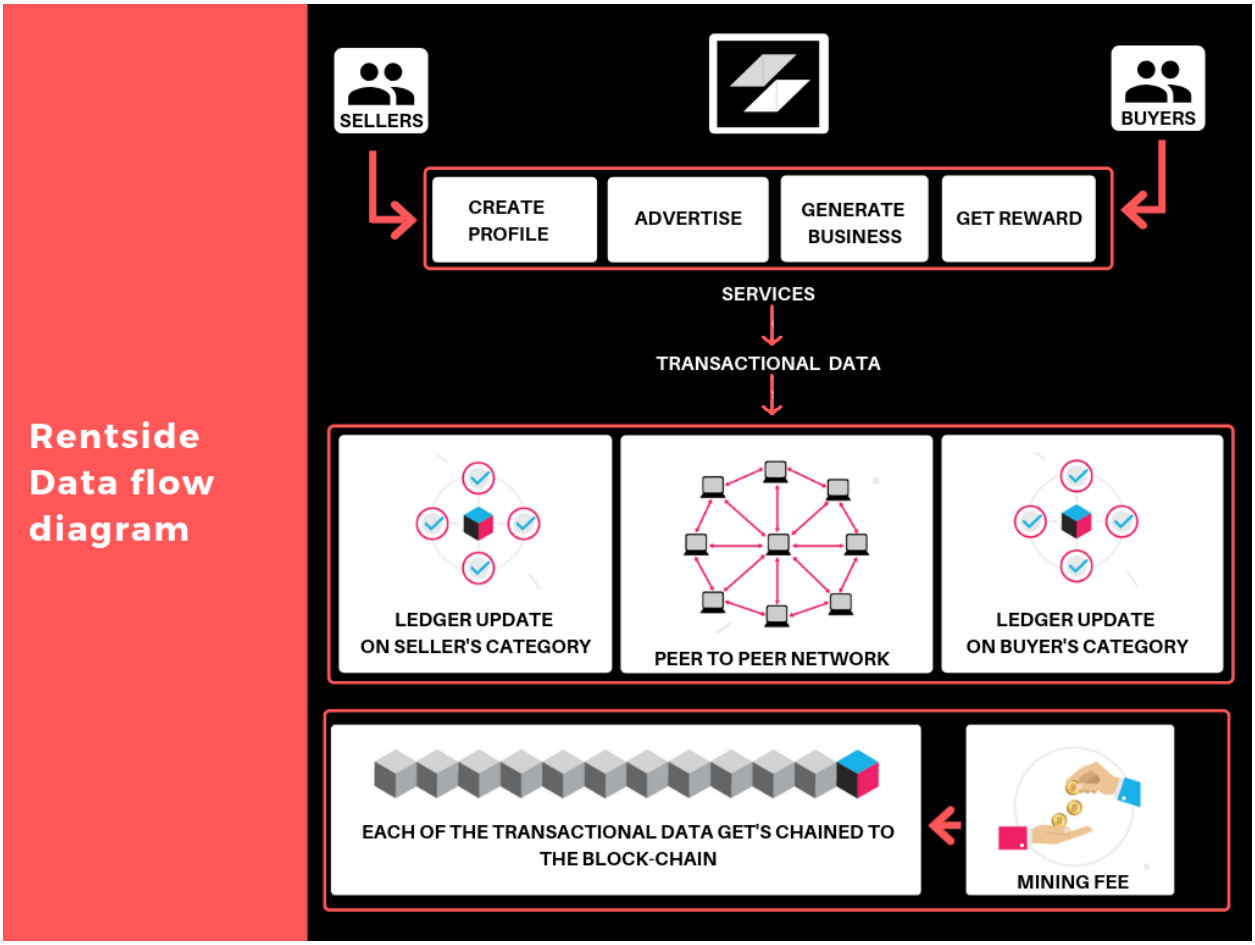
2. Revenue - The revenue model of Rentside is designed in a way that it keeps the blockchain stiff every time. The gas values of the transaction is one of the main part which governs generation of revenue. Let's discuss the features of Rentside which generates money- * Transactional value for each transaction. For every services in Rentside the customer has to pay a small amount in the form of ether. For creating an account to advertise the products or services you provide, all remains ether chargeable. * Subscription Charges - Rentside has premium features which is extra chargeable. For example- in a free subscription you can only connect to 50 sellers/buyers a month. In premium versions you can connect to more consumers and producers. For now this is kept out of scope. * Advertisements- This also generates a huge revenue. Per advertisement it is ether chargeable.

3. Security- The security concerns of Rentside is ensured by various cryptographic algorithms using public and private keys. To proceed with any transaction, one need to register to Metamask, which is a gateway to ethereum blockchain network. You need to create your private key which needs to be kept secure. | The other most important concern of blockchain is credits and copyrights. This is achieved by Ethereum's proof of work algorithms.



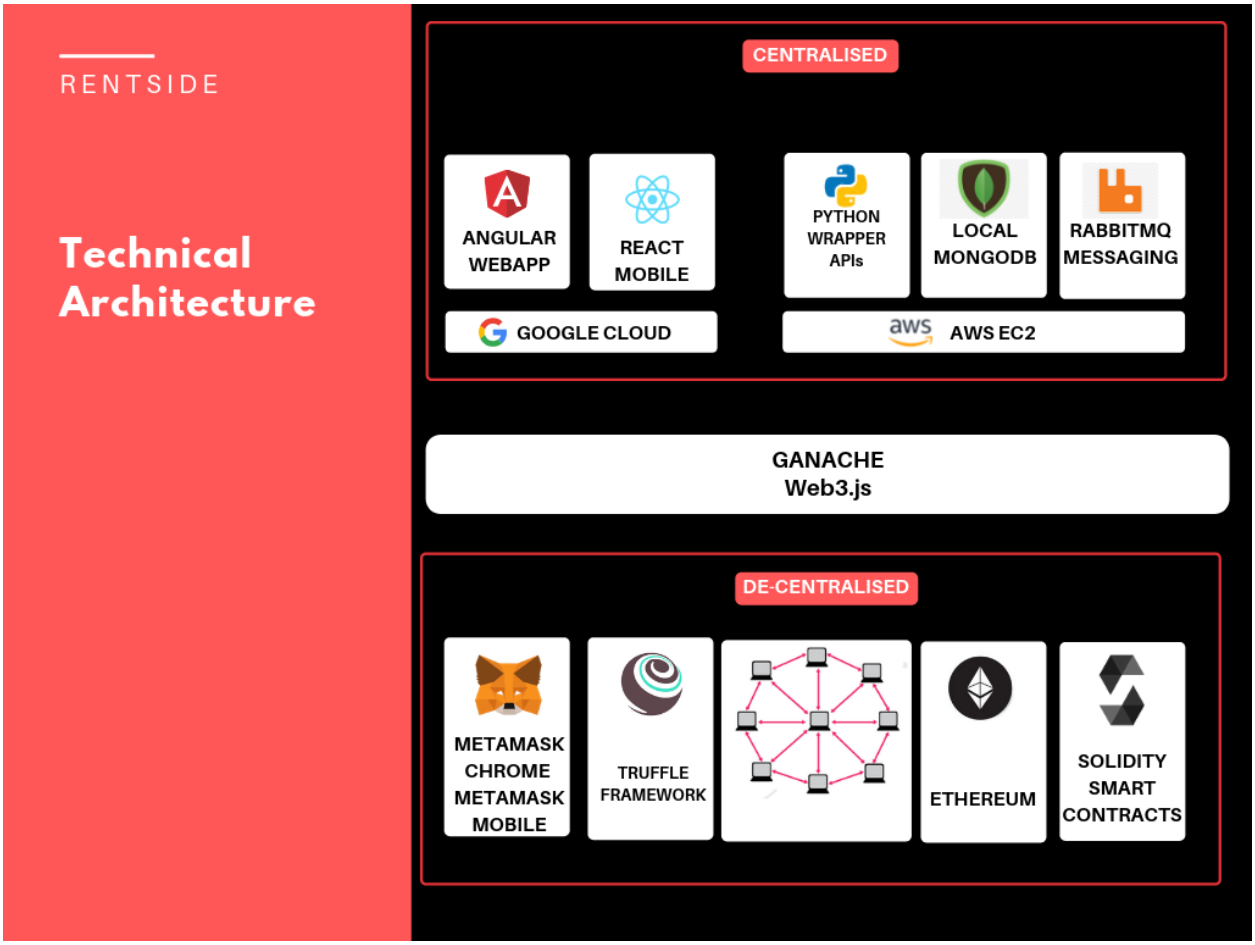
5.3.2 DFD (Data Flow Diagram)

In Rentside the data flow depends on the services we use. Typically there are four service which generates data following nested services. So each time the user uses Rentside it generates data , a block gets created and the public ledgers get updated for all. Once the block is ready it depends on the gas value to mine the block and chain it to the blockchain.



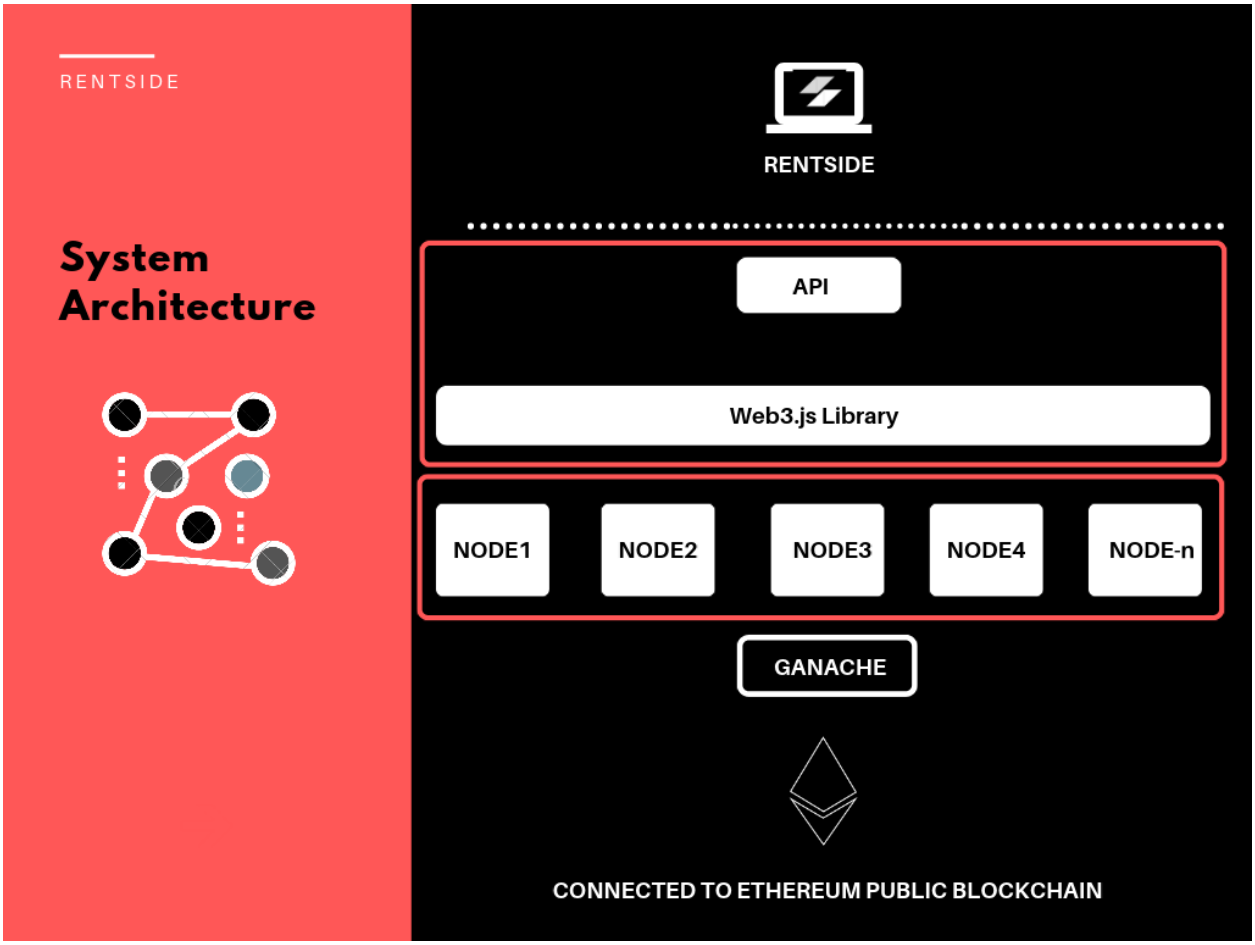
5.3.3 Technical Architecture

1. The Central Part | In terms of technical architecture, as Rentside comes in two variants, one as a web application and other as a mobile application to serve the purpose on every kind of available devices. The former is built on Angular 7 and the mobile version is built using React native. Both of them are packaged as an image on a docker container and hosted on Google cloud platform. Further , these front ends are connected to the decentralised part using python wrappers. The local data is stored in MongoDB which sometimes serves as a local cache. In some cases where you want data fetching speed to be high and can't rely on block-chain network Rentside uses its local DB to make processing faster. Further, these wrappers are connected using RabbitMQ messaging services which works on a publish-subscribe fashion. These web hooks provide an asynchronous connection of the front end of Rentside and the blockchain network. All these wrapper services, local database and messaging queues resides under Amazon EC2 instance. 2. The Decentralised Part | Until now, all the technical dependencies were hosted on Google cloud or AWS, which is completely centralised. The decentralised part contains Ganache-Truffle-Web3.js framework which allows the central part to connect to the decentralised world. Ganache is a in memory blockchain which provides test ethers to develop any application. In current scope of Rentside, Ganache is used as a blockchain network. Web3.js is a very powerful javascript library which allows any javascript based framework to connect to Ethereum blockchain. Rentside is based on Web3 and Truffle. For the core part, which is smart contracts Rentside uses Solidity.



5.3.4 System Architecture

The system architecture extends the technical architecture and explains how Rentside interacts with the lower layers of its decentralised services. The Rentside's package is connected to the public blockchain layer using Web3.js library wrapped by python APIs. This combines together to bring up the communication between the nodes .... Node1 to Node n. These nodes resides on the Ganache local arena for the scope of the this application.


RENTSIDE
System Architecture
RENTSIDE
API
Web3.js Library
NODE1
NODE2
NODE3
NODE4
NODE-n
GANACHE
CONNECTED TO ETHEREUM PUBLIC BLOCKCHAIN

CHAPTER 6

Technical Documentation

- APIs
- Solidity as Core
- Ganache Local Arena
- Using Web3.js
- Rentside to public blockchain

Note: This part will be coming up once the Rentside MVP is completely ready. Before 18th June :)

CHAPTER 7

Initial POC

Rentside Release Timeline	
Release v0.0.1	Documentation with solution architecture and product explanation. https://rentside-introduction.firebaseapp.com/
Release v0.0.2	The complete MVP with all the features quoted in the documentation.
Release v0.0.3	Integration of chat services with peers and predictive analytics

CHAPTER 8

About

8.1 Founder and Developer

Rahul Gorai
Software Engineer
Block-Chain Developer

Rentside, A Conduent Hack 2019 Product

8.2 Contact

Email- gorai.rahul2022@gmail.com