Servicechain.io: Empowering Service Workers in Web3

DSC 180

John Mauricio, Vineet Tallavajhala, Anya Chandorkar, Kristina Shahatit

1. Abstract

In today's digital environment, building a credible professional reputation does not have many infallible platforms or applications. Especially in the service industry, it is hard to quantify tips, customer sentiment, and experience to carry on in a professional future. We created a user-friendly interface using a QR code integration to easily, digitally tip and rate using ethereum currency. We also created interactive timesheets for employees to enter hours worked. For the future, the usage of the blockchain's graph structure and timestamping attribute will allow us to easily query data and build a visualization dashboard for employees to view their rating, tipping, and work hour history and patterns. With the use of blockchain technology's immutable nature, network timestamps, and persistence based chaining, Servicechain.io presents a solution that can enforce a transparent and credible reputation as an employee. Our chain provides functionalities like tipping and rating on a scale of 1-5 for customers to leave honest feedback for service workers, as well as a segmentation to allow individual actors(companies) to create their respective contracts for their employees. Servicechain.io uses smart contracts to house core logic that interacts with the ethereum blockchain. In addition to the smart contract the app provides user-friendly mappings of names to the public hashes of users/organizations through a firebase backend. The usage of the blockchain's graph structure and timestamping attribute allows us to easily query data and build a visualization for employees to view their rating, tipping, and work hour history and patterns. For the future, the accruing of data points such as hours or tips will allow for the addition of NFT based goodwill rewards incentivizing higher service ratings.

2. Introduction

Over the past fifty years, the advent of service workers has grown more and more popular. In the United States specifically, tipping culture has increased and proven to be a steady flow of income for service workers. With the growing decentralization of society, it is imperative that a decentralized solution to migrate service workers' workflows exists. To provide a platform for automated tipping as well as monitoring job performance for service workers, we created Servicechain.io an application that will allow consumers to rate and tip service workers which are automatically stored on a blockchain so employers can identify and verify service workers' performance in a transparent and equitable manner. With a transparent and equitable blockchain specifically tailored for service workers, we envision a world in which an employee's reputation and credibility can be derived from these transparent chains. In order to further the utility of the application, we also added the ability for service workers to log their hours using Servicechain.io. With these hours automatically stored on the Servicechain, we also imagine that work experience can immediately be verified for the sake of experience validation when workers apply for other jobs. Although these services can all be easily attained currently within a centralized paradigm (through a mix of applications such as Square (tipping and payment), Yelp (rating), and Paycom (hours logging)), Servicechain uniquely decentralizes these various services through the use of smart contracts. Mixing together smart contracts (transaction protocols on the blockchain) with a React based front end, we aim to bridge together these various services and bring decentralization to the service industry.

3. Data Model

In the application of smart contracts, there is no data heavy process that involves cleaning, analysis, and modeling like many data driven projects. Many smart contracts do not rely on any data, and instead require data unique to users which will grow as more and more users use the product (the blockchain acquiring more data as time goes on). This process is also often tested through simulation and would not make use of a preset dataset to begin with. Within the Servicechain.io application, users data helps craft the blockchain from which much of the value of the application derives. That is, as more and more users use the application, more data regarding average tips, average ratings, and total hours worked are available to the user in the application. Creating user friendly interaction mechanisms, our front end bridges the gap usability and complexity of the blockchain.

4. Methods

For Servicechain.io, there are two primary smart contracts that interact with a react based front end to create the decentralized application (dAPP for short). The first smart contract is the *Factory* smart contract which enables companies to create individually deployed contracts in which their employees (service workers) can interact and input their respective data. For example, restaurant A, restaurant B, and restaurant C would each have individual deployed contracts. In objective oriented programming terms, one could think of the smart contract as the class and each restaurant as instances of this class. The second contract is the *Service* contract which enables the functionality for the various services offered on Servicechain.io such as tipping, rating, and logging hours. These contracts work with a custom built front end to enable the decentralized application. The front end is built using React.js, Next.js, as well as some basic Javascript. The backend to allow for user mappings to public addresses used firebase. Decentralized web applications are hard to transition to and because of this the app uses real time market data on ethereum to convert prices into USD in order to ease this transition by making use of aws Lambda and the Gecko API endpoint.

Below is a figure that illustrates the architecture of the application.

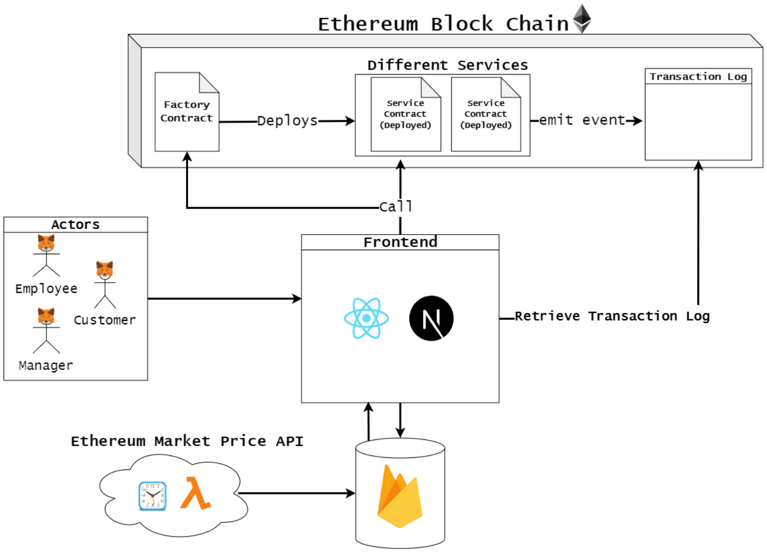

Fig 1. Servicechain.io Architecture

The actors in this architecture include the company, customer, and employee. The company represents the employer for an employee (in a service worker's case most often is a restaurant). The company has the primary responsibility of setting up the instance of the factory contract where most of the services will take place. After setting up the individual instance for their company, the company's only continual interaction is verifying the hours submitted by a given employee. The second actor is the customer. A customer scans a QR code presented by a service worker to access Servicechain.io. The customer is exposed to two primary functionalities that are exposed from the *Service* contract: the tip and rate functions. These functions allow the customer to rate and tip the last actor of the application: the employee (the service worker). The primary interaction for an employee actor is the Log Hour function, which allows employees to log their hours worked on the Servicechain.io, so their total hours worked is immutable and can immediately be verified for future jobs. The employee actor will also be able to view the tips as well as the ratings they have received using the Servicechain.io application. A more technical description of both the *Factory* and *Service* contract as well as their respective functions are expanded upon below.

In the Servicechain.io application there are two contracts that work together with the react front end that enable the functionality of the application.

The first one is *Factory*, a parent contract for the *Service* contract that creates individual factory instances. There is only one primary function within the *Factory* contract which allows for initialization.

- *createService* is a function that creates an instance of the *Service* contract for a given instance of the *Factory* contract. This allows the data for a specific "factory" (company) to be separate from other companies and enables the blockchain to be less cluttered with simultaneous amounts of transactions occurring.

The second contract is *Service* which enables the set of services that empower the application which includes the functions *submitTip, sendRatings, getRatings, enterHours, and getTotalHours.*

- *submitTip* is a function that allows customers to transfer money from their wallet to the wallet of the employee.

- *sendRatings* allow a customer to give a rating to an employee

- *getRatings* allow an employee to view the average ratings they have received from customers.

- *enterHours* allow an employee to enter their current hours worked for an arbitrary time frame

- *getTotalHours* allow an employee to view their total hours they have worked since they started using the system.

These six functions represent the core functionalities of our decentralized web application and allow us to create a service based application on the blockchain.

5. Results

Servicechain.io is an entry point to the power of blockchain technology and smart contracts in the service industry and a potential solution to the many flaws that come up within it by providing an ecosystem amongst customers, employees, and employers. Servicechain.io has the ability to deliver finances and data directly to the user without the need of a single authority by utilizing the distributed ledger provided by the ethereum network. This allows employers to see quantifiable impact by their employees by being exposed to low level details of their hours,ratings, and tips. It allows employees to not only be compensated but build a reputation of their hard work through the feature of ratings and tipping mechanisms built in the app. Customers are exposed to a more convenient way to impact businesses and transfer finances by utilizing direct customer to employee/employer deposits and ratings. All of this while being transparent and secure because of the blockchain network.