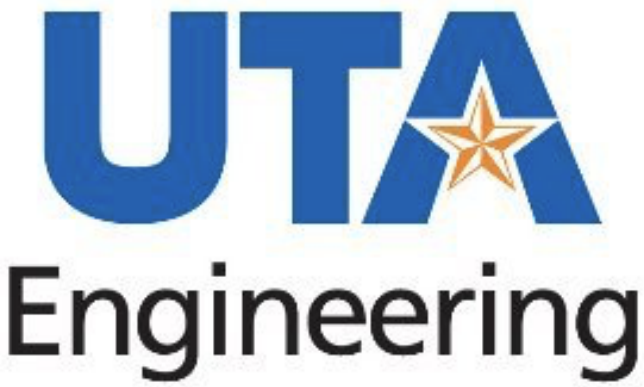
Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4316: Senior Design I
Spring 2024



Web3 Enjoyers
Smart Pay

Sarah Akin
Edward Alkire
Zeynep Erguven
Abhisek Kumar Jha
Kerry Pacheco

Revision History

Revision	Date	Author(s)	Description
0.1	2.20.2024	AJ	Document Creation
0.2	2.28.2024	SA	Final Edit of First Draft

CONTENTS

List of Figures

1 Problem Statement

In any financial system the relationship between participants is naturally adversarial. Lack of transparency and trust requirements are precarious compromises in such a system. Before the invention of decentralized blockchain technology, transparency and trust-minimization have been too costly or impractical. Compromising solutions have been brand based reputation, government authority and license schemes, and the threat of costly legal or violent intervention. Consider how opaqueness is exploited in the prevailing financial system as seen in the form of politician insider trading, especially charitable organizations or otherwise that mislead and betray the interests of their contributors, deception in selling a brand, "advertising", as opposed to market participation earned reputation, and the imbalance of power between labor and capital.

2 Methodology

We are going to build a web app and smart contracts to enable any organization to deploy a smart contract for public and transparent treasury management, as well as trust-minimized salary agreements. The web app will support deployment of smart contracts, along with various pages to display organization information, relevant status and on-chain events information. We will also utilize blockchain indexing infrastructure to support aggregating history on-chain event historical data.

3 Value Proposition

The goal of this project is to enable organizations to offer a degree of transparency and trust-minimization previously impossible, without smart contracts, cheaply and easily in their funding, salary agreements, and other financial arrangements. With an organizations historical and current financial information made transparent and public in a non-falsifiable way, this increases confidence between market participants by minimizing the amount of trust required of participants. A charity might benefit having a transparent treasury so that donors know what they're funding and how their donation is spent. Employees might benefit from smart contract based salary agreements by not having to trust that they will be paid what they are owed, or the terms of how their salary agreement will be managed.

4 Development Milestones

- Project Charter first draft - March 3rd 2024
- System Requirements Specification - March 2024
- Architectural Design Specification - April 2024
- Detailed Design Specification - May 2024
- CoE Innovation Day poster presentation - April 2024
- Final Project Demonstration - August 2024

5 Background

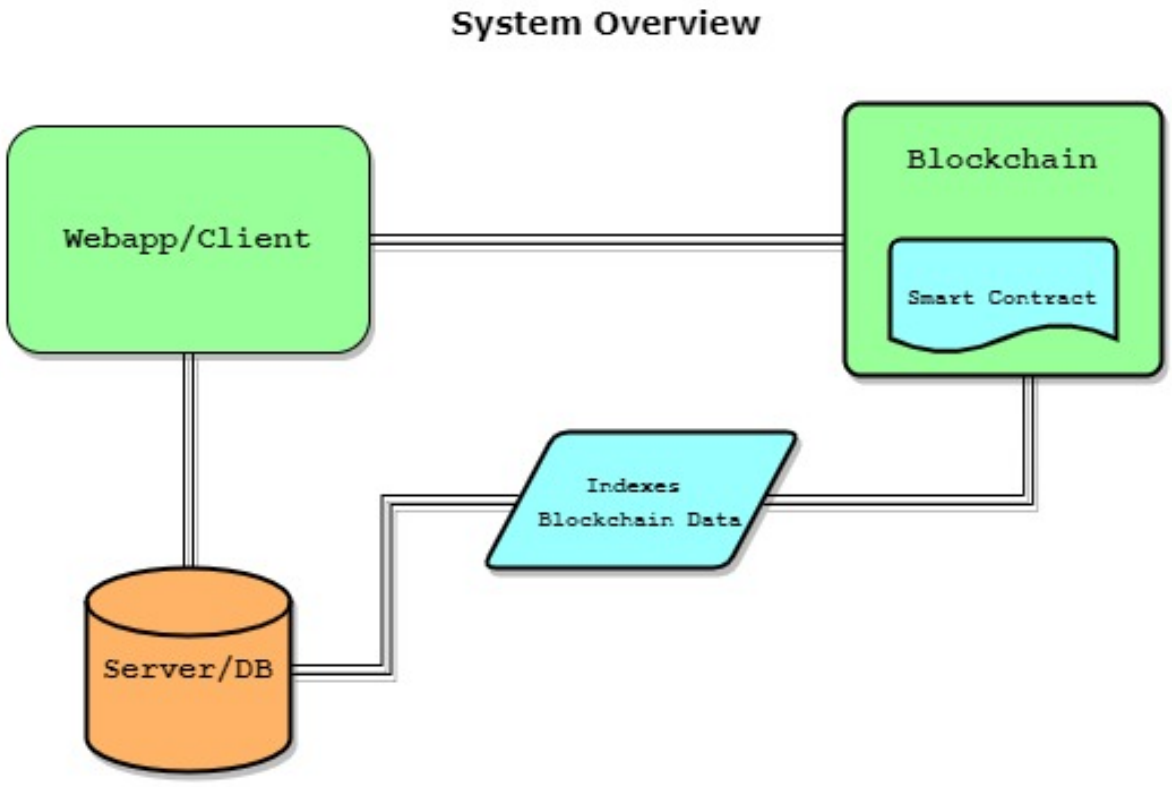
The problem with treasury management and salary agreements in the prevailing financial system is opaqueness and the entailing trust assumptions.
Consider an non-profit organization like Wikipedia, which misleads users into thinking they wouldn't exist without regular donations, meanwhile their operational costs are relatively tiny and they have hundreds of millions of dollars of assets, enough to operate forever without another donation.
More credible charities could benefit from transparent treasury management to attract donations when they can prove their need and history of appropriate management of funds.
Salary agreements are another area where trust is abused in the prevailing financial system. Employees have to trust their employers brand in that they will compensate them as they expect. This becomes a problem in the case of companies which mismanage funds or go bankrupt, and employees aren't duly compensated. Relying on reputation is also a problem for startups which want to attract talent. Even if a startup offers a competitive salary, prospective employees are being asked to take a risk trusting a company with no reputation, or trusting that the startup will still have funds left to pay them for very long.
Traditional salary agreements may be prone to human error and human emotions, potentially leading to strife and legal intervention. For example, the average wrongful termination settlement is $40000. [1]
Traditional salary agreements are burdened by unpredictability and unnecessary costs, whether they be legally related or due to human nature.

6 Related Work

The landscape of smart contract-based web applications has evolved significantly over the past few years, with advancements across various sectors including finance, supply chain, and digital identity. Despite these many solutions, there remains a gap between existing offerings and the specific needs of specialized applications. The current most popular type of DeFi protocol for salaries can be described as "salary-streaming". An example of such protocol is Sablier, which facilitates the streaming of salaries to employees in real-time. This approach allows employers to deposit funds into a smart contract, which then streams the salary to employees on a per-second basis [2]. Similarly, Stream Protocol offers real-time salary payments through a comparable mechanism, enhancing the flexibility and efficiency of wage distribution [3]. Another similar platform is Superfluid, providing real-time finance capabilities on the Ethereum blockchain and supports salary streaming, enabling financial assets to flow through networks in real-time [4]. These platforms provides a foundation for understanding the state-of-the-art in smart contract based salary distribution mechanisms. However, while promising, these solutions are simple and don't offer the features we seek to implement in this project. Though, they might not deliver these requirements due to possible constraints like network fees, scalability issues, and smart-contract security risks that come with increased smart contract complexity.

7 System Overview

At a high-level this project will have the following major components:
A web interface will allow a financial institution to create smart contracts and their specifications; this institution can also deploy the smart contract and manage the treasury and expenses. The company employees will have access to their information and can view their status an on-chain events pertinent to their account.
In addition to the organization and employees, any user of the application can view the information and financial status of public organizations. This transparency enhances trust among its parties and allows for independent verification of financial data ensuring accountability of the network.
A database server platform that can perform different tasks such as user account authentication, user profile storage, payment history and in general manage user sessions and other storage that is not suitable for the blockchain.
The database will also interact with the blockchain by indexing data extracted from it. It will fetch the requested data, process it, and store it in the appropriate format. Data indexing improves the performance of the application by reducing the need to query the blockchain directly.
The blockchain will work as a financial decentralized ledger with the ability to create and store and execute smart contracts. The smart contract itself contains all the financial legal terms and specifications of the contract translated into programming code. This means all the business logic and the definition of all the rules and operations of the agreement.



8 Roles & Responsibilities

Our stakeholders are all team members, the professor, Chris Conly, University of Texas at Arlington, and our customers, both the company and their employees. The point of contact will be Edward Alkire. Edward Alkire will also be the product owner, he has the most knowledge about what the requirements are and came up with the idea. We will have rotating scrum masters, changing every sprint. The schedule goes as follows:

- Sprint 1: Abhisek Kumar Jha
- Sprint 2: Kerry Pacheco
- Sprint 3: Sarah Akin
- Sprint 4: Zeynep Erguven

Edward Alkire has a background in Java, Spring Boot and Solidity, and since he understands block chain the best, he will be working on smart contracts. Sarah Akin has a background in Java and React, and she will be working on both smart contracts and front end development. Abhisek Kumar Jha is knowledgeable with React, so he will be working front end as well as smart contracts when needed. Kerry Pacheco is knowledgeable in Java, Mongo database, and XML, and she will be working on the front end with Zeynep Erguven who also has a background in Java as well as React and CSS.

9 Cost Proposal

The approximated spending will be used on tools to help us build the website. Hostinger comes with a domain, SSL certificate, and many more features. [5] This is important due to the trust our clients will need in knowing our site is secure as well as easily accessible to them. Our clients trust is necessary because we will be handling their funds.

9.1 Preliminary Budget

Details	Estimated Amount	Actual Amount
Hostinger - Domain Name & SSL	59.88	-
Hardware for Ethereum node	500	-
Total Projected Materials Cost (Sprint 1)	559.88	-

9.2 Current & Pending Support

Details	Funding Amount
CSE Department	800
Total Projected Funding	800

10 Facilities & Equipment

This software project requires the following facilities: and office or home office space that is quiet and free of interruptions. This will vary depending on the developer's personal preferences; the main objective is to find a comfortable space where team members can be productive.
The equipment needed for this project includes computer monitors, mouse and keyboard, laptops, headphones, and speakers. A comfortable desk chair and a desk are essential to create a suitable environment. Other useful devices the team can use depending on the requirements of the project are a printer,

external hard drive or cloud storage, and a docking station to connect multiple devices.

11 Assumptions

The following list contains critical assumptions related to the implementation and testing of the project.

- Platform chosen for deploying smart contracts will remain stable and support the application's transactions without significant changes in gas fees and transaction processing times throughout the project life-cycle.
- The development team will have access to and proficiency in the necessary programming languages, development tools, and name of the product by the 2nd sprint cycle.
- External data sources required for the smart contracts to function correctly will be reliable and available by the 4th sprint cycle.
- Users' adoption of digital wallets and familiarity with blockchain operations will be sufficient to interact with the application without extensive training or support.
- Regulatory and legal compliance requirements relevant to the blockchain and cryptocurrency operations in the target market will remain favorable and not introduce unexpected requirements on the project's scope.
- The integration of the smart contract with the front-end application will allow seamless interaction without significant latency or user experience issues, assuming adequate network conditions and infrastructure compatibility.

12 Constraints

The following list contains key constraints related to the implementation and testing of the project.

- Final prototype demonstration must be completed by May 1st, 2024 to align with the strategic goals of the organizations.
- Customer installation site will only be accessible by development team will be limited to off-peak hours to minimize the impact on network performance and costs associated with deploying and testing smart contracts.
- The project budget is capped at $800, covering all the expenses including development tools, and cloud hosting fees.
- All data obtained from customer site must be reviewed and approved for release by the Information Security Office prior to being copied, affecting how user data is collected, stored, and processed.
- Deployment of the web application and its associated smart contracts must be compatible with the blockchain protocol to ensure redundancy and avoid platform-specific risks.
- The project will rely solely on internal development and testing resources to ensure intellectual property protection and reduce external dependencies.

13 Risks

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

Risk description	Probability	Loss (days)	Exposure (days)
Competition from similar smart contract web applications	0.70	14	9.8
Limited experience developing smart contract applications	0.50	20	10
Long hours of working shifts outside of project	0.50	15	7.5
The team's dependence on the project mentor and other members	0.45	10	4.5
Iteration deadlines on the same date as major deadlines in other CSE classes	0.45	10	4.5
Unforeseen changes in project requirements	0.40	9	3.6
Different moving parts make the integration of different software tools critical to avoid major refactoring and delays	0.25	15	3.75

Table 1: Overview of highest exposure project risks

14 Documentation & Reporting

14.1 Major Documentation Deliverables

14.1.1 Project Charter

The initial version of the project charter will be delivered on March 3rd, 2024. It will be updated in case of any critical modifications made to the project. The final version will be delivered in August 2024.

14.1.2 System Requirements Specification

The initial version of the system requirements specification will be delivered in March 2024. This document will be updated in case any requirement change or modification is made to improve the project. The final version will be delivered in August 2024.

14.1.3 Architectural Design Specification

The initial version of the architectural design specification will be delivered in April 2024. This document will be updated in case any requirement change or modification is made to improve the project. The final version will be delivered in August 2024.

14.1.4 Detailed Design Specification

The initial version of the detailed design specification will be delivered in May 2024. This document will be updated in case any requirement change or modification is made to improve the project. The final version will be delivered in August 2024.

14.2 Recurring Sprint Items

14.2.1 Product Backlog

The items will be added to the Jira product backlog from the SRS. Jira is an agile project management tool used by teams to plan, track, release, and support software. It will be used to maintain and share the product backlog with the team members and stakeholders. The items will be prioritized by a group vote of the team members.

14.2.2 Sprint Planning

Each sprint will be planned using Jira by creating individual tickets assigned to the team members by a group vote. There will be eight sprints in total to complete the project.

14.2.3 Sprint Goal

The team will share their opinions on the sprint goal and vote to make a decision. There will be feedback and exchange of ideas from the customer's side.

14.2.4 Sprint Backlog

The team will vote to decide which product backlog items make their way into the sprint and the scrum master verifies the final decision. The Jira software will be used to maintain the backlog.

14.2.5 Task Breakdown

Each team member will voluntarily claim their work from the sprint backlog in the first two days of the sprint. If there remain any unassigned tasks after two days, the team will hold a meeting to decide on which members to assign the rest of the tickets. The time spent on the tasks will be documented using the tool provided by the Jira software.

14.2.6 Sprint Burn Down Charts

The scrum master will be responsible for generating the burn-down chart for each sprint. Since each team member has their account created in Jira, it will not be a problem to access the total amount of effort expended by each member. The burn-down chart will use the "issue count" format.

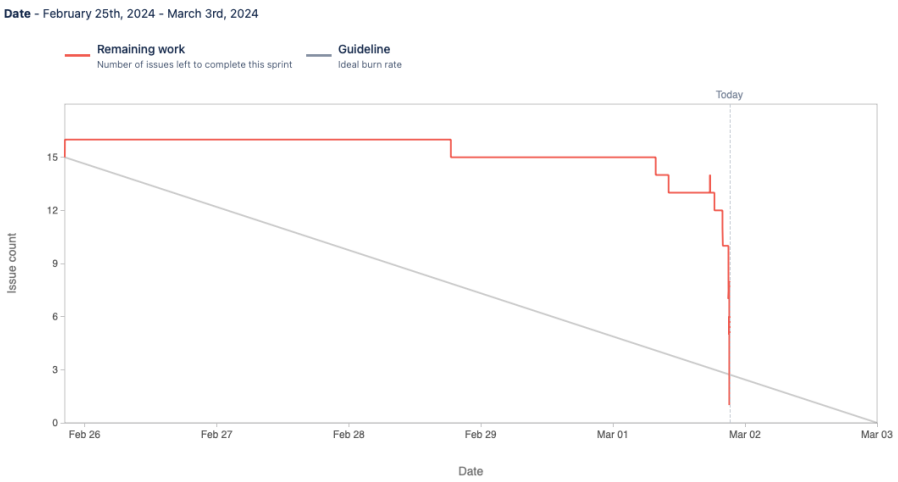


Figure 1: Sprint 1 burn down chart

14.2.7 Sprint Retrospective

The sprint retrospective will be separated into three items and will be reviewed after each sprint as a team. The team will discuss the retrospective with the help of the items; what went well, what to improve, and action items. Each member will document their opinion on these items which later on will be discussed in a meeting as a group. This documentation will be due at the end of each sprint.

14.2.8 Individual Status Reports

Each member will document their status by using the items discussed above. It will be reported every sprint. If there exists any critical and urgent status of an item, it will be discussed immediately.

14.3 Closeout Materials

14.3.1 Web Page

On the project web page, organizations will be able to deploy a smart contract for their organization and manage treasury and expenses. Employees will be permitted to view their status and on-chain events pertinent to them. Any user will be permitted to view information on the status of public organizations.

The project web page will be accessible to the public. It will be delivered in August 2024, provided at closeout.

14.3.2 Demo Video

The demo video will show how to navigate throughout the web page and how the system works as a whole. The videos will be 2-3 minutes long. It will cover the topics about how to deploy a smart contract for a given organization and manage treasury and expenses. It will also cover how to view the status and events of an employee.

14.3.3 Source Code

This project will adopt the git version control system. The source code will be maintained and modified in GitHub. Source code will not be provided to the customer, and it will not be open-sourced to the general public.

14.3.4 Source Code Documentation

Documentation will be provided in the form of a user guide page on the website.

14.3.5 Installation Scripts

The service will enable deployment of smart contracts to a blockchain.

14.3.6 User Manual

The user will be provided with a digital manual. Similar to the demo video, there will be a two-minute setup and navigation process.

REFERENCES

[1] "Average wrongful termination settlements and verdicts [2024] — wrongfulterminationsettlements.com," https://www.wrongfulterminationsettlements.com/average-wrongful-termination-settlements/, [Accessed 03-03-2024].

[2] "Sablier | Token Distribution — sablier.com," https://sablier.com/, [Accessed 03-03-2024].

[3] "Home - StreamPay — streampay.io," https://www.streampay.io, [Accessed 03-03-2024].

[4] "Superfluid | Stream Money Every Second — superfluid.finance," https://www.superfluid.finance, [Accessed 03-03-2024].

[5] Hostinger, "Web Hosting | A Fast and Secure Platform for Your Website — hostinger.com," https://www.hostinger.com/web-hosting, [Accessed 03-03-2024].