DataMatch

Zixin Wei z5wei@ucsd.edu
Maxwell Fang m3fang@ucsd.edu
Yanhao Guo yag007@ucsd.edu
Aditya Melkote avmelkote@ucsd.edu

Nolan Sheffield
Sheffield.Nolan@franklintempleton.com

Abstract

Data scientists require vast amounts of data to run complicated models. Centralized marketplaces exist to provide data, but their methods contain privacy risks. Therefore, we propose a decentralized, Ethereum-based data marketplace that allows anyone to buy or sell data securely without intermediaries. **Our platform leverages smart contracts to automate transactions—including licensing and payment conditions— and incorporates data anonymization techniques like Personally Identifiable Information (PII) removal, allowing training without exposing raw sensitive data.** We plan to implement a token economy to incentivize the provision of high-quality data by rewarding honest stakers and punishing dishonest ones. Off-chain storage solutions like IPFS are used to store large datasets under blockchain-based mechanisms. This approach resolves storage issues by enhancing data security, reducing costs, and maintaining transparency.

Code: https://github.com/Fangtastic7/DSC-180A-B16---Group1

1 Introduction

In the digital age of information, data is everywhere, and it is one of the most important assets to any organization. Companies produce large amounts of data each day to help provide insights and discoveries for their customers, industries, and ultimately the world. Currently, there are existing data cloud marketplaces that provide data scientists, analysts, and other professionals with the convenience of accessing data from a wide variety of sources. This can be viewed as beneficial, where it helps to reduce time and analytics costs; nevertheless, there are many risks associated with having a centralized storage. Data leakage, hardware failure, and large-scale data transmissions can result in low efficiency, high costs, and insufficient use of data Li et al. (2020). However, a decentralized, Ethereum based data platform would help to implement privacy-preserving techniques like secure-multiparty-computation or homomorphic encryption, allowing analysis without exposing sensitive data. One could argue that cloud marketplaces can ensure the trust factor of users through verification, but this requires information to be shared by the user. With blockchain technology and cryptography techniques, many issues, regarding user privacy and trustworthy transactions, can be resolved in a peer-to-peer network.

Due to Ethereum having a proof-of-stake sybil resistance mechanism, the Ethereum blockchain and by extension any smart contract hosted on the blockchain by result is a secure yet transparent method of executing transactions. By employing Ethereum smart contracts, users can license, ensure proper usage rights, and define payment conditions without help from an authority or third party. To prevent attacks on the network, Ethereum utilizes a proof-of-stake based consensus mechanism to prevent single-point failures, where all nodes must agree on the state of the blockchain. If an attacker were to attempt to take control of the chain, they must destroy a massive amount of ETH. Essentially, the system incentivizes individual stakers to behave honestly and disincentivizes stakers from acting maliciously.

Blockchains are robust and reliable, but there is one challenge that it faces: scalability. Considering there are vast amounts of data, blockchains are not feasible in "terms of computational resources and cost." da Silva Vanin et al. (2022) Many studies consider only storing transaction history information in blockchains while the real data is "stored in off-chain infrastructures, such as cloud service providers or health institution premises." da Silva Vanin et al. (2022) The issue with cloud service providers or health institution premises is that data can be easily leaked or compromised. In early 2024, UnitedHealthcare was hacked by a Russian ransomware group, where 100+ million personal health records were compromised. The Russian group ended up securing 22 million dollars and causing 3 billion dollars of damage. There are many more cyberattacks happening today and the number will continue to rise. One solution to ensuring safer data storage is the use of the InterPlanetary File System (IPFS), which decentralizes data in a peer-to-peer network. With IPFS, users can hide their networks, encrypt their data, and control gateway utilization to further secure their information. Ultimately, we propose that smart contracts provide a better option than third party marketplaces and off-chain decentralized storage systems maintain greater data privacy than cloud service providers.

2 Overview

Our decentralized data marketplace project aims to create a secure platform for buying and selling datasets while preserving user privacy. Over the two quarters, we successfully built a marketplace integrating Ethereum-based smart contracts, IPFS for off-chain storage, and private information (PII) removal. The marketplace facilitates trustless data transactions, allowing sellers to freely list their datasets and buyers to securely purchase them. The platform's foundation is a smart contract deployed on the Amoy test network, which manages licensing, payment terms, and access control without intermediaries. This ensures data authenticity, transparency, and privacy.

Remix IDE was utilized to develop and deploy the smart contract onto the test network, allowing us to test the functions for listing, buying, and delisting data. Pinata is a gateway to interact with IPFS which enables the ability to store files in a decentralized manner and generate unique CIDs for each. Next.js served as both the frontend and backend framework, connecting users with the blockchain, handling file uploads, and managing interactions with Pinata and the smart contract. MetaMask was used to enable users to connect wallets and sign transactions seamlessly.

By combining blockchain technology and an off-chain decentralized storage system, we have built a scalable, secure, and privacy-preserving data marketplace. The project focused on establishing the core infrastructure, the integration of many tools, and successfully demonstrating critical functionalities like file uploads, secure storage, and trustworthy transactions. The project lays the foundation for further applications, including public deployment, recommendation algorithms, and a token economy to incentivize high-quality data contributions.

3 Literature Review

In the paper "A Blockchain-Based IoT Data Marketplace," the authors discuss the decentralized platform leveraging blockchain technology and smart contracts to enable secure and trustless data trading among the Internet of Things (IoT) devices Sober et al. (2022). Traditional decentralized data marketplaces often face challenges of high infrastructure costs and trust issues. With blockchain technology, the platform can address concerns through transparency, immutability, and decentralization. The immutability feature of blockchain is featured in the security of transactions, where everything is permanent and traceable. Smart contracts, on the other hand, can automate and enforce the rules of data exchange, facilitating seamless interactions between buyers and sellers of data. Through optimizing smart contracts, the costs of transactions can be lowered. Components, such as proxies and brokers, the platform can manage data trading efficiently. By addressing the practical considerations of cost, scalability, and security, the study presents a robust framework, demonstrating a blockchain-based marketplace model that is both feasible and adaptable across various IoT-driven industries.

One effort to reduce some of the shortcomings of previous iterations of decentralized-based

marketplace projects, such as the Origami Network, is to remove a centralized point of storage using a Swarm decentralized database to eliminate single points of failure. In addition to employing blockchain technology and an automated escrow process, Jorge Cacho et al. Fonseca, Dahal and Kim (2020) addressed that the problem of high gas fees as well as slow execution speed for querying and iterating through the database can be resolved by storing the data in Swarm and simply performing a match with data on the blockchain. As each transaction consumes a certain cost, the Swarm approach greatly reduces the net gas fee per user transaction and speeds up the transaction to a rate that is competitive with traditional transaction services.

4 Infrastructure

4.1 High-Level Infrastructure

- *Remix-Ethereum:* Remix-Ethereum is the IDE where we write our smart contract code. In addition, it allows us to deploy our contract. Any updates or changes that are done here have to be redeployed.

- *MetaMask Wallet:* To make a transaction (buying, listing, and delisting), the buyer/seller must connect to a crypto currency wallet. In this case, we utilized Metamask, where the currency is either ETH, MATIC, or POL, depending on the testnet. Each account has a public address that is used to send and receive ETH as well as for verifying funds.

- *Amoy Test Network:* This is a test network for our contract to operate on until our contract is ready for production. It is essentially a simulated Ethereum Virtual Network (EVM) that uses POL as its currency. The currency is considered fake crypto currency, so that we can run as many tests as we want without incurring any costs.

- *OKlink.com:* OKlink is where we can interact with our smart contract. We run our smart contract, verify it (ensure the bytecode for contract execution is the same as when we deployed it), publish it, and make transactions with it. This process requires the connection of Remix and Metamask.

- *Amazon Web Services:* This facilitates pinning and unpinning data from the IPFS without exposing sensitive information. First a file will be uploaded to Amazon S3 to handle large file uploads, pinned to Pinata IPFS using Amazon Lambda Functions and then removed from Amazon S3.

- *Pinata.cloud:* An internet file API used to upload data. This connects our smart contract to the IPFS system, which allows uploading and retrieving data. When a seller wants to list their data, they will upload a file, and the file will be transmitted into Pinata cloud, tagged with an unique CID. When a buyer buys data, they will be able

to attain the file via an internal dataID (or unique key). The CID is used to fetch specific data in the cloud, but is not accessible by the user, who can only access it through the dataID.

- *Next.js:* A React-based web framework that serves as both the frontend for building an intuitive user interface and the backend for managing interactions between users, APIs, and the blockchain system. It enables tasks like file uploads, dataset browsing, and data purchasing while seamlessly integrating with MetaMask for wallet connectivity and transaction signing. Using API routes, Next.js processes requests from the frontend, acts as an intermediary to services like Pinata and smart contracts, and facilitates blockchain interactions via Ethers.js. By simplifying complex blockchain operations and providing real-time updates, Next.js ensures a user-friendly, scalable, and high-performance platform, making it essential for this data exchange system.

- *spaCy:* An open-source Natural Language Processing (NLP) library designed for efficient, production-ready text analysis. It provides pre-trained models for multiple languages and supports key NLP tasks such as tokenization, named entity recognition (NER), part-of-speech (POS) tagging, dependency parsing, and text classification. Unlike other NLP libraries, spaCy is optimized for speed and scalability, making it suitable for real-world applications like chatbots, information extraction, and text summarization.

4.2 Design Logistics

To ensure user protection and data privacy, data must be encrypted before being listed by the seller (Also remove any Personal Identifiable Information). Once the data is uploaded into IPFS, the data cannot be accessed without the CID. The only way for someone other than the seller to obtain the CID of a specific data file is to purchase it on the marketplace, see Figure 1. The purpose of PII removal is to hide sensitive user information, like patient information of a medical dataset. Prior to purchasing data, buyers have the ability to preview the description or specifications of a data file. The description or overview of the data file shows a snapshot of information to any users on the market, which includes the title, description, file type, file size, and number of purchases. When a buyer purchases a listing, they are granted permission to download the data file. Downloading data out of the marketplace may ruin the privacy of the data, but this issue is addressed with a disclaimer that warns the user of the potential consequences of sharing data illegally.

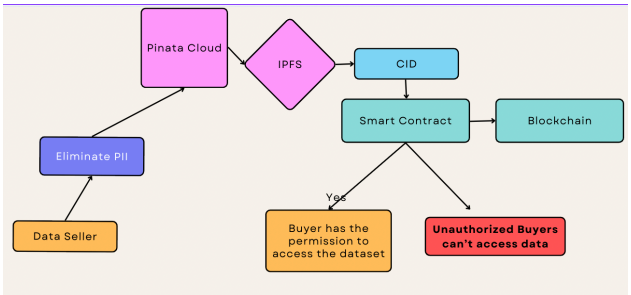

Figure 1: *Diagram of Proposed Decentralized Data Marketplace*

5 Methods

5.1 Setting the Smart Contract in Motion

The smart contract operates as the backbone of the data marketplace by managing user interactions, enforcing permissions, and maintaining transactional integrity. It is deployed on the Amoy Test Network using Remix IDE, where it is compiled, connected to MetaMask, and deployed. Deploying the contract requires gas fees in ETH, after which users with MetaMask accounts can interact with it. Each user has a unique public address, allowing them to function as a buyer or seller, depending on their actions within the marketplace.

Sellers use the listData() function to upload a logo, file, title, price, and description, allowing some data to be available for public viewing via shared get functions. These get functions allow all users to access dataset descriptions and details. Buyers, on the other hand, use the buyData() function to purchase datasets, provided they meet criteria such as sufficient funds and eligibility. Sellers retain exclusive control over their listings and can use the delistData() function to remove their items, while buyers are restricted to functions relevant to purchasing.

The permissions and roles are represented in Figure 2, where sellers can write and call vertical functions, such as listData() and delistData(), while buyers interact horizontally, accessing public methods like get and buyData(). Some functions require specific conditions to be met before execution, ensuring secure transactions and preventing unauthorized actions. This structure enforces clear boundaries and functionality for buyers and sellers while supporting public visibility of listing details.

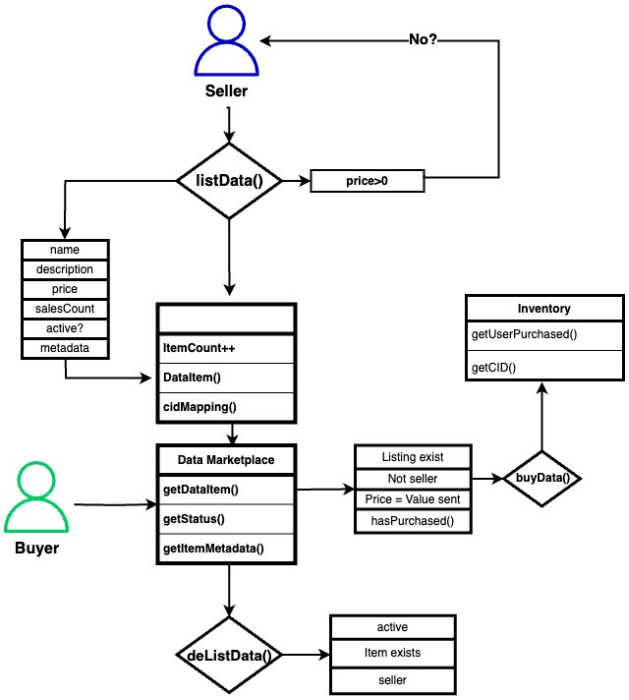

Figure 2: *Low-level implementation of the smart contract*

5.2 PII Removal

The implementation of personally identifiable information (PII) removal in our system is designed to ensure that sensitive user data is protected before being uploaded to decentralized storage. Our approach leverages the spaCy NLP library to detect PII from structured datasets. When a user uploads a file, it is first processed through a temporary directory, where the raw data is analyzed for potential PII, including names, email addresses, phone numbers, and other identifiable details. The system uses pre-trained Named Entity Recognition (NER) models to classify sensitive entities, which are then either masked or replaced with placeholder values. After successful PII redaction, the cleaned file is saved and used for further processing, ensuring compliance with data privacy regulations.

To automate this process, we have integrated the PII removal function into the backend of our application, which is triggered upon file upload. The backend, built using Next.js and TypeScript, executes a Python script using child process execution (execAsync), passing the uploaded file through the PII detection model. Any identified PII is logged, and a sanitized version of the file is returned for storage. Any identified PII is logged, and a sanitized version of the file is returned for storage. Error handling mechanisms ensure that if the PII removal fails, the system prevents unprocessed files from being uploaded. Once processed, the cleaned data is uploaded to Amazon S3, and a reference to the file is stored on IPFS for decentralized access. This implementation ensures that all datasets shared on our platform remain free from personally identifiable information while maintaining usability for data analysis and machine learning applications.

5.3 IPFS Integration

File storage is decentralized, and handled through the InterPlanetary File System (IPFS) via Pinata Cloud for gateway access to the IPFS network. When the listData function is called in the smart contract and the user uploads a corresponding file, it is first sent to Pinata, which stores it on the IPFS network. The IPFS network, through Pinata's gateway provides a CID (Content Identifier) for the file. This unique id will be used to fetch data from the IPFS network with the CID when the buyData function is called in the smart contract and can be fetched when calling the get methods in the inventory. The CID is protected, so only the purchasing user can view the data. The smart contract calls the javascript functions via the Ethers JavaScript library.

5.4 Marketplace Deployment

To deploy the marketplace, our team utilized Vercel as the platform. Vercel is an efficient cloud platform optimized for Next.js applications, providing automatic builds, continuous deployment, and global edge network distribution. Instead of integrating with GitHub, we opted for manual deployment from our local machine due to the large size of our project files. Using the Vercel CLI, we were able to deploy directly from our development environment by running 'vercel –prod', which initiated the build process and deployed our application to a globally distributed network. This approach allowed us to retain control over our deployment workflow without relying on an external repository.

During deployment, we configured environment variables through Vercel's dashboard to securely store sensitive keys, such as our blockchain RPC URL and smart contract addresses. Since "env.local" files are not automatically included in production, manually adding these variables was crucial for maintaining secure and functional blockchain interactions. We also leveraged Vercel's logging and debugging tools to monitor API calls and blockchain transactions, ensuring smooth functionality post-deployment. Through this approach, we successfully deployed our decentralized marketplace, allowing users to interact with the platform in a scalable and globally distributed manner.

6 User Interaction

6.1 Data buyer (Purchase data file)

In our data marketplace, buyers can preview the information about the data file before they plan on purchasing it. To buy a dataset, the buyer will click on the "Buy Now" button which calls the buyData function, providing the data's unique identifier (dataId) and their address. The function checks that the buyer is not the seller, does not already own the item, and has sufficient funds, including gas fees required for the transaction. Once completed, the item is added to the buyer's inventory, accessible through the get functions in inventory, and the seller receives the payment. It is essential that buyers must account for

gas fees in addition to the dataset price. The platform ensures uniqueness of listings using a unique ID system, preventing duplication while allowing sellers to delist and relist items with consistent identifiers. (Figure 3)

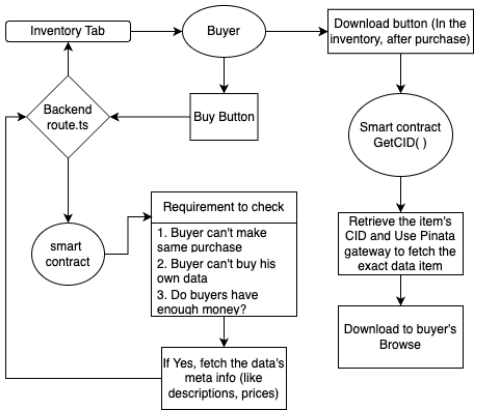

Figure 3: *Procedure Diagram for data buyer*

6.2 Data seller (File upload)

In the data marketplace, sellers upload files through the platform by providing details such as the file, price, and description. AWS S3 and Lambda are used to handle large file uploads and interact with Pinata without exposing sensitive information to the user. The backend processes the upload by sending the file to Pinata, which generates a unique CID using IPFS. This CID, along with the seller-provided details, is passed to the smart contract through the listData function. The contract stores the CID and metadata on the blockchain, emits a DataListed event, and makes the dataset available for buyers to view and purchase. Check Figure 3 below for the detail procedure:

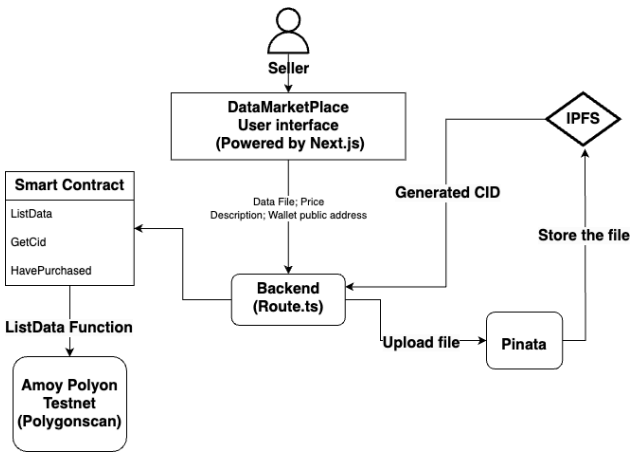

Figure 4: *Procedure Diagram for file upload*

7 Video Demo

Here is the video demo: https://www.youtube.com/watch?v=qdlgu8o1eWU.

8 Discussion

We successfully created a blockchain data marketplace that allows for buyers and sellers to make transactions securely without intermediaries. Buyers can view all listings in the marketplace and make purchases. Sellers can list data by uploading it with a small gas fee. In regards to uploading, it is important that a free version of AWS and Pinata is used to upload files into IPFS. Upload rate cannot be sped up due to the limitations of what is provided, potentially due to throttling speeds. For larger files, it is important to acknowledge that it may take a couple seconds longer to process. In the upload process, there is also a feature of PII that can screen for private information in only comma-separate values (CSV) files. Given only a certain amount of time, we could only achieve screening CSV files with a decent performance. One thing to note is that if PII fails to process a CSV file, it will not be uploaded to IPFS (a design choice we made to prioritize privacy at the cost of potentially false flagging data sets). Ethics is another issue that came across the project. One of the hardest challenges is preventing the spread or misuse of downloaded data once it has been bought. If there was more time, we could have incorporated an encryption feature on the downloaded file to maintain greater data protection. The temporary solution we came up with was adding disclaimers for uploading and downloading data. The last issue that has been haunting transactions for the entire quarter is having successful transactions being pushed onto the test network. Every time we have the user pay a certain gas fee, the Metamask transaction will fail if the fee is too low. Initially, we thought that the cache would fill up, so clearing it would resolve the issue, but there was no consistency. After a couple of weeks, it was discovered that a low fee would not lead to success but a fee above a certain threshold. We can always ensure that transactions will be successful by increasing the priority (35 GWEI) and base fees (40 GWEI).

9 Future Work

Future work can be done to improve the overall buyer and seller experience through new functionality to enhance trustworthiness within the marketplace and introducing improvements on top of existing functionality. As speed is currently a downside, an enhancement can be made to increase the speed of user uploads. A comprehensive recommendation system can be introduced to better recommend suitable data sets automatically, allowing for a smoother experience for potential buyers. Additionally, a preview could be introduced when attempting to purchase a dataset for a small fee, allowing users to more comprehensively evaluate the integrity of a dataset before investing in purchasing the entire dataset. On the seller's side, PII techniques could be introduced to support a variety of data types,

as well as improving the existing system to more comprehensively detect PII. Furthermore, another feature that could be introduced is a rating system or customer feedback that is public for listings. This way, community consensus can enforce good actors and implicitly punish bad actors, prioritizing showing users only high quality data sets. Lastly, the most important attribute would be adding encryption techniques to prevent illegal usage and misuse of data.

References

Fonseca, Jorge, Binay Dahal, and Yoohwan Kim. 2020. *Decentralized Marketplace Using Blockchain, Cryptocurrency, and Swarm Technology*.: 865–882. [Link]

Li, Junqing et al. 2020. "A Blockchain-Based Educational Digital Assets Management System." *IFAC-PapersOnLine* 53 (5): 47–52. [Link]

da Silva Vanin, Fausto Neri et al. 2022. "A Blockchain-Based End-to-End Data Protection Model for Personal Health Records Sharing: A Fully Homomorphic Encryption Approach." *Sensors* 22 (24). [Link]

Sober, M., G. Scaffino, S. Schulte, and S. S. Kanhere. 2022. "A Blockchain-Based IoT Data Marketplace." In *Proceedings of the 20th International Conference on IoT and Blockchain Applications*. Springer. [Link]