ResilientX: Decentralized Social Platform of the People, by the People, for the People

SHUJUAN (AMBER) CHEN, ZIYANG CHEN, TIANXIAO CHENG, JUNLONG TANG, and HAO-NAN ZHU,
University of California, Davis, USA

INTRODUCTION

Social platforms have become an integral part of our daily lives over the past several decades, largely due to the thriving of the internet. A social platform is a digital space where individuals, groups, and organizations can share, communicate, collaborate, and form virtual communities [5]. With the power of the internet, social platforms are meant to connect people regardless of geographical location, race, religion, color, gender, national origin, age, or disability. There are different types of social platforms, including social networking (e.g., Facebook), media sharing (e.g., Instagram), discussion forums (e.g., Reddit), and collaboration platforms (e.g., Slack).

While social platforms offer numerous benefits, they also have drawbacks [6]. The main concerns of social platforms include privacy issues, the threat of censorship and unwarranted banning, non-democratic dispositions, and monopolistic tendencies. Furthermore, emerging platforms bring more concerns about mental health implications due to addictive designs and spreading misinformation or fake news. It is crucial to address those issues as people increasingly engage their lives with social platforms.

Blockchain is believed to have a high potential for addressing the inherent issues of social platforms. Blockchain is a decentralized and distributed data storage technology. Unlike traditional databases that store information in a central location and are controlled by solely one party, blockchain disperses data across a network of nodes [7]. It is controlled by multiple parties. A consensus mechanism verifies each transaction made on the blockchain. Blockchain technology has the potential to bring about decentralization, transparency, and trust to existing social platforms while also enhancing their security. With blockchain, social platforms can reshape themselves to be more secure, transparent, and user-centric through enhanced privacy, reduced censorship for integrity, and empowerment of users.

In this project, we introduce ResilientX, a decentralized social platform empowered by ResilientDB [3]. ResilientX is an X-like social platform that allows users to share and broadcast their thoughts, activities, or lives. The main difference between ResilientX and X is that ResilientDB backs ResilientX. ResilientDB is a sustainable blockchain ecosystem that upholds privacy, integrity, transparency, and accountability. It presents a well-engineered and efficient system to meet the platform's decentralized data storage and processing requirements, which is composed of a three-tier architecture consisting of the Application, Consensus, and Network layers.

ResilientX is designed to uphold its main principles: decentralization, democratization, transparency, and privacy-preserving. In the following sections, we demonstrate our motivating example, core design principles, and market opportunity and risks, usage and evaluation of ResilientX.

Authors' address: Shujuan (Amber) Chen, csjchen@ucdavis.edu; Ziyang Chen, zynchen@ucdavis.edu; Tianxiao Cheng, txcheng@ucdavis.edu; Junlong Tang, jlotang@ucdavis.edu; Hao-Nan Zhu, hnzhu@ucdavis.edu,
University of California, Davis, California, USA.

MOTIVATING EXAMPLE: THE X SUPREMACY

In this section, we demonstrate the motivation behind ResilientX by illustrating how X (a.k.a Twitter) failed to observe core values including democratization, transparency, and privacy-preserving.

Suspension. X may suspend accounts temporarily or permanently from their service, and its use of suspension has always been controversial. For instance, between July and December 2021 alone, X suspended 1.3 million accounts and removed 5.1 million content [1].

Centralization. Due to the nature of centralization, X can make unilateral decisions regarding user accounts. For instance, after rebranding to X from Twitter, the handle `@x`, a personal account was taken over by X platform without justification [4].

Privacy Violation. X might violate user privacy without consent. Investigation initialized by the Federal Trade Commission uncovered that X contravened an accord with regulators in which it committed not to commercialize user data [2]. While this data was originally collected for security measures, it was later exploited to serve targeted advertisements to users.

DESIGN PRINCIPLES

ResilientX is designed to uphold its main principles: decentralization, democratization, transparency, and privacy-preserving.

Decentralization. Our core design concept is decentralization, an essential design principle that distributes control, authority, and decision-making away from any centralized authority. With ResilientDB as the backbone of ResilientX, every transaction and data entry undergoes a consensus verification process, ensuring integrity and authenticity.

Democratization. The cornerstone of ResilientX is democratization, for equal voice, power, and opportunity for all users. This is backed by the ability to distribute decision-making power to every single user from ResilientDB.

Transparency. ResilientX ensures transparency by design. With ResilientDB, ResilientX demystifies every operation that happens on the platform. This ensures users know how the platform works or how decisions are made.

Privacy Preserving. ResilientX prioritizes privacy-preserving and lets users have control over what they share, with whom, and how it is used. With the advantaged cryptographic technology and consent-based data practices, personal data remains secure, confidential, and accessed only with explicit permissions.

MARKET OPPORTUNITY

The market opportunity for ResilientX is significant, given the growing concerns and dissatisfaction with traditional social platforms. It tackles concerns related to privacy and data security, a prevalent issue in today's digital landscape. Users are worried about their personal information being mishandled or exploited for profit, and ResilientX's focus on privacy-preserving offers a solution.

Additionally, it addresses the frustration with centralized platforms making unilateral decisions, often leading to censorship and a lack of user agency. The democratization and transparency principles of ResilientX appeal to users who want to actively participate in platform governance and decision-making.

ASSOCIATED RISKS

The project also faces critical risks. First, there might be technical challenges and bugs, which can block us from completing the project. Second, limited resources, particularly the lack of servers, may challenge development and testing. Third, reliance on third-party dependencies introduces a risk of disruptions if these external services face issues or changes. These factors collectively pose significant challenges to the project's timely and successful completion.

THE DESIGN OF RESILIENTX

In this section, we describe the designed functionalities of ResilientX, as well as their underlying technical components. We first describe the ResilientX architecture, followed by the ResilientX components and their interactions. As shown in Figure 1, the core architecture of ResilientX is structured into three integral parts: the database, the backend, and the frontend, each playing a pivotal role in the system's overall functionality and user experience. The database of ResilientX is backed by ResilientDB with its KV service and Crow service for robust data management. For the backend, ResilientX utilizes the Python SDK of ResilientDB to build on top a ResilientX server that offers Python-based API services tailored for social media management. On the frontend, ResilientX features a Vue-based website, designed for easy accessibility and user-friendly interaction.

The following subsections provide a detailed breakdown of the functionalities and design specifics for each component.

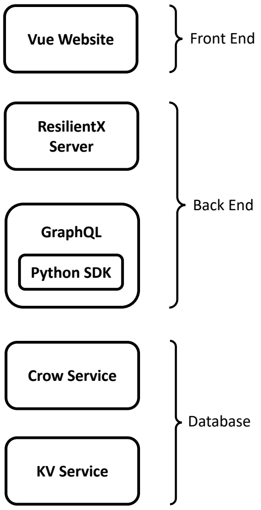


Fig. 1. Architecture of ResilientX

Database

The database of ResilientX utilizes a key-value store, incorporating the KV service and Crow services from ResilientDB. As it was mentioned earlier, the use of ResilientDB in ResilientX is critical for maintaining privacy, integrity, transparency, and accountability. The KV service provided by ResilientDB forms the core of the ResilientX database system, which is essential for efficiency and scalability for a decentralized social platform. The crow service acts as an endpoint, which provides an interface with the on-chain KV service.

API Endpoint	Descriptions
User Management	
`register(user_id, password)`	Registers a new user.
`check_login(user_id, password)`	Checks if user credentials are valid.
Social Interactions	
`post(user_id, post_id, content)`	Creates a new post.
`comment(user_id, post_id, content)`	Adds a comment to a post.
`get_user_posts(user_id)`	Retrieves all posts by a user.
`get_user_comments(user_id)`	Retrieves all comments by a user.
`get_user_info(user_id)`	Retrieves public profile information of a user.
`get_user_friends(user_id)`	Retrieves the friends list of a user.
`add_friend(user_id, friend_id)`	Adds a friend to the user's friend list.
`remove_friend(user_id, friend_id)`	Removes a friend from the user's friend list.
`get_all_posts()`	Retrieves all posts from all users.
`get_all_comments()`	Retrieves all comments from all users.
Helper Functions	
`get_value(key)`	Retrieves a value from the server using a GET request.
`set_value(key, value)`	Sends a value to the server using a POST request.
`initialize()`	Initializes the API.

Table 1. API Endpoints w/ Descriptions of ResilientX Backend

Backend

The backend of ResilientX provides Python-based API services for user management and social interactions. Table 1 lists the full functionalities of the ResilientX backend, which allows for user registration and authentication, post and comment creation, as well as the management of user profiles and friendships. The backend API interacts with the database through `GET` and `POST` requests to handle those functionalities effectively.

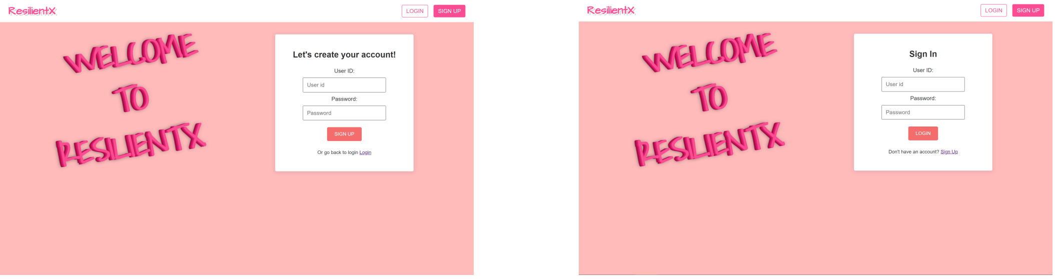
Frontend

The frontend of ResilientX is designed as a user-friendly webpage, offering an intuitive interface to its users. With Vue, a JavaScript frontend framework known for its adaptability and ease of integration, the frontend of ResilientX seamlessly interacts with the backend. With a tight integration from frontend to backend, ResilientX could ensure the best user experience by efficient data exchange and real-time updates, thus making itself both responsive and engaging for users.

DEMONSTRATION OF RESILIENTX

In this section, we showcase the functionalities of ResilientX. We first demonstrate the user registration and login process. Then, we walk through how users can set up their personal profiles, browse through a home page filled with posts from friends, and engage by creating their own posts and comments. Finally, we show how users can add friends and view their friends' profiles.

For each functionality, we also show the corresponding API calls that facilitate the underlying operations on ResilientDB.

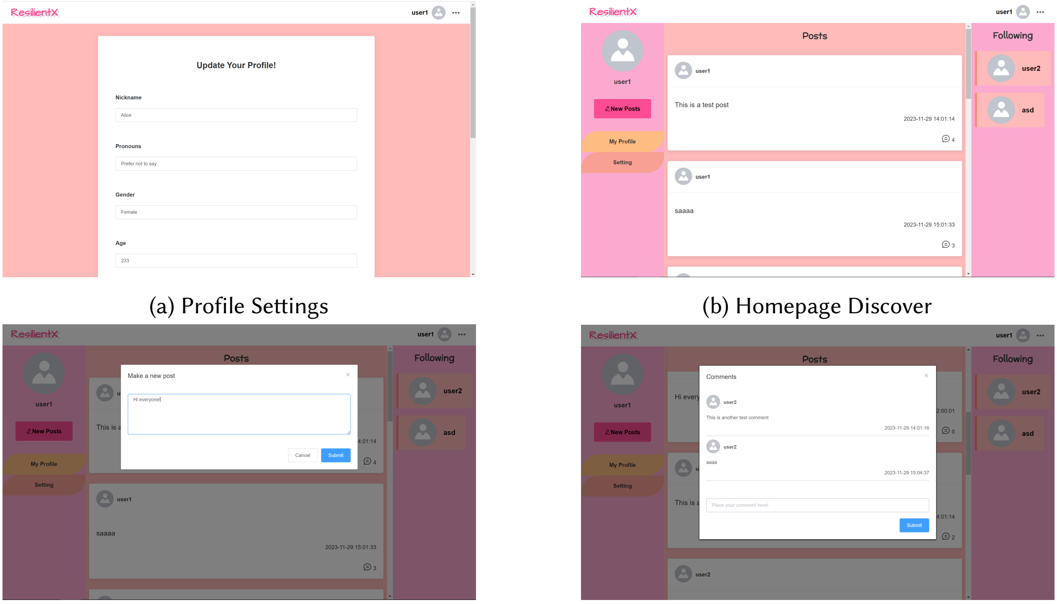


(a) User Registration

(b) Login

Fig. 2. Screenshots of the ResilientX user registration and login process.

User Registration and Login. Figure 2 showcases the pages for user registration and login in ResilientX. Specifically, Figure 2a corresponds to the user registration process, which is based on the `register(user_id, password)` API, a function that registers a new user. On the other hand, Figure 2b illustrates the login process, relying on the `check_login(user_id, password)` API, which is responsible for validating user credentials.



Nickname: Alice
Pronouns: Prefer not to say
Gender: Female
Age: 233

user1
This is a test post
2023-11-29 14:01:14
4

user1
saaaa
2023-11-29 15:01:33
3

(a) Profile Settings

(b) Homepage Discover

Hi everyone!

user2
This is another test comment
2023-11-29 14:01:16

user2
aaaa
2023-11-29 15:04:37

Place your comment here!

(c) Creating a Post

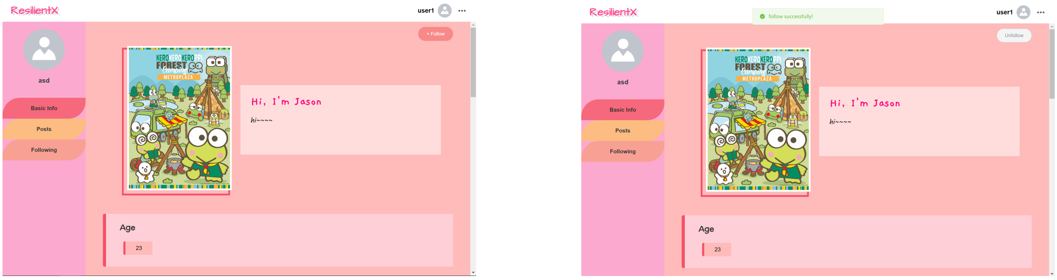
(d) Commenting on a Post

Fig. 3. Screenshots demonstrating various functionalities of ResilientX.

User Profile Setup. Figure 3a showcases the profile settings in ResilientX. In this page, users can personalize their profiles, which is supported by the `get_user_info(user_id)` and `set_value(key,` value) API calls. These API calls retrieval and updating of user profile information, ensuring a tailored and user-centric experience.

Homepage. The homepage, as shown in Figure 3b, allows users to explore a feed of posts from their network. This functionality is supported by the `get_all_posts()` API call, which retrieves a collection of posts from all users, ensuring a real time content stream.

Post and Comment. Creating and interacting with posts are fundamental functionalities of the social platform ResilientX, as shown in Figure 3c and Figure 3d. The posting feature relies on the `post(user_id, post_id, content)` API call for content creation, while the commenting functionality utilizes the `comment(user_id, post_id, content)` API call.



Hi, I'm Jason

hi~~~~

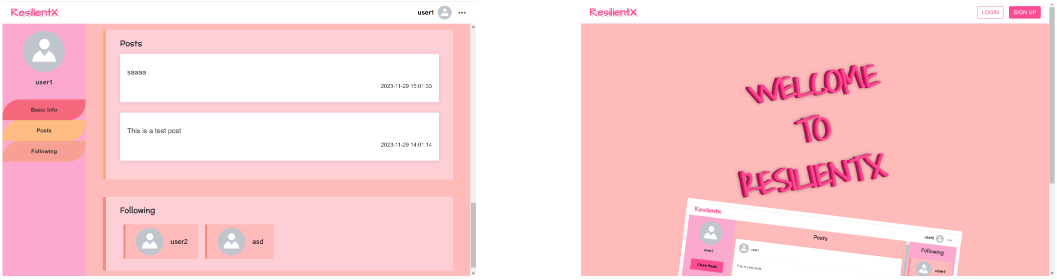
follow successfully!

(a) Following a User

(b) Unfollowing a User

Fig. 4. Screenshots demonstrating the friendship functionalities in ResilientX.

Friendships. The friendship management within ResilientX, crucial for building and managing user connections, are shown in Figure 4a and Figure 4b. The process of following another user, as shown in Figure 4a, is supported by the `add_friend(user_id, friend_id)` API call, which adds a user to the current user's friend list. Conversely, the unfollowing action, as shown in Figure 4b, utilizes the `remove_friend(user_id, friend_id)` API call to remove a user from the friend list.



saaaa
2023-11-29 15:01:33

This is a test post
2023-11-29 14:01:14

(a) User Personal Page

(b) Welcome Page

Fig. 5. Screenshots of the Welcome Page and User Personal Page in ResilientX.

Personal Homepage. Figure 5a shows the personal homepage for users, which acts as personal hub for each user on ResilientX. This page displays the user's posts and their following list with API calls such as `get_user_posts(user_id)` and `get_user_friends(user_id)`.

Welcome Page. Lastly, the welcome page of ResilientX, as shown in Figure 5b, is designed to enhance the user experience to be more friendly without relying heavily on API calls.

EVALUATION

In this section, we evaluate the performance of ResilientX. We first describe the experimental setup and then present the results of our experiments.

Experimental Setup

We evaluate the performance of ResilientX on a local machine with Intel Core i5-13600K as CPU and 32GB RAM. We use the Python SDK of ResilientDB to build the ResilientX backend, which is responsible for interacting with the ResilientDB database. The ResilientX frontend is built using Vue.js version `2.6.11`. We use the ResilientDB KV service to store user data and the crow service to interact with the KV service. We evaluate the scalability of ResilientX by answering the following research questions:

- **RQ1:** How does the number of users affect the performance of ResilientX?
- **RQ2:** How does the number of nodes affect the performance of ResilientX?

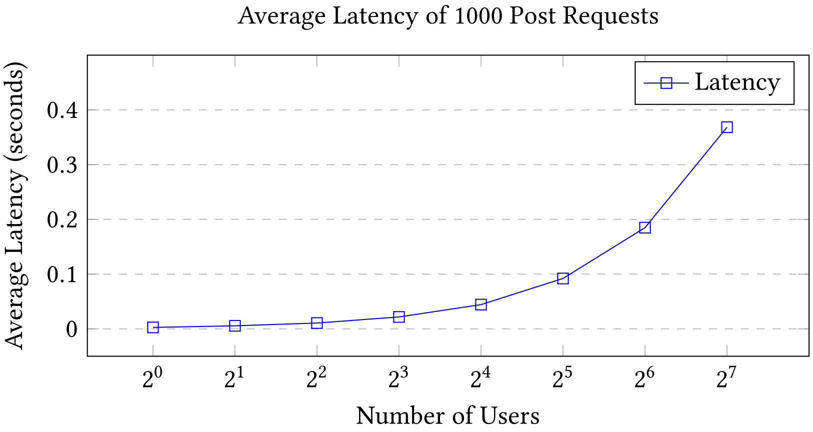


Fig. 6. Average Latency vs. Number of Users

RQ1: Scalability in terms of post workloads. To answer RQ1, we perform evaluation of performance of ResilientX under different post workloads. We vary the number of users making the post from 1 to 128 and measure the average latency to create posts. Figure 6 shows the results of our experiments. As we can see, the average latency to create a post increases. This is because when more and more users are using ResilientX, the backend database needs more time to handle results. However, the average latency is still very low, which is less than 0.4 seconds. This is because the ResilientDB KV service is robust and efficient enough to handle the increasing number of users.

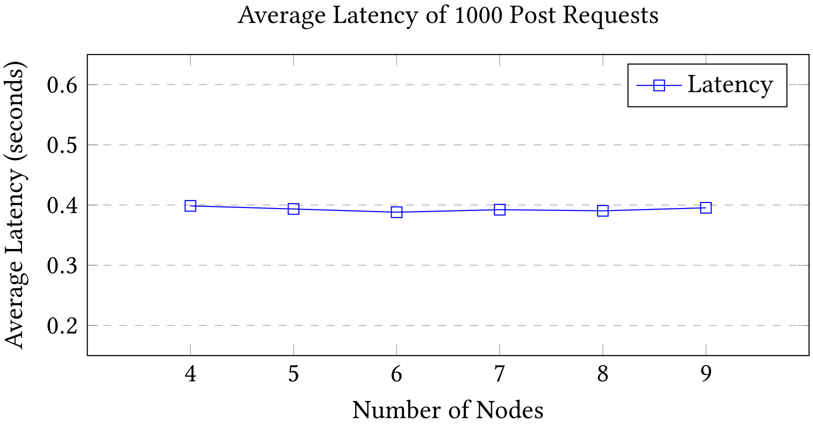


Fig. 7. Average Latency vs. Number of Nodes

RQ2: Scalability in terms of number of nodes. For RQ2, we perform evaluation of scalability of ResilientX in the scenario of increasing number of nodes. We vary the number of nodes from 4 to 9 and measure the average latency to create posts from 1000 posts. Figure 7 shows the results of our experiments. As we can see, the average latency to create a post keeps steady when the number of nodes increases. This is because the ResilientDB KV service is robust and efficient enough to handle when new nodes joining as the business grows.For our future work, it is worthwhile to conduct scalability test under controlled thread and/or core circumstances. The uniformity would help in isolating the impact of number of nodes from other variables, providing a confident insight into the scalability of the ResilientX system.

CONCLUSION

We build ResilientX, a decentralized social platform that holds core prioritizes for decentralization, democratization, transparency, and privacy preserving. ResilientX is built on the ResilientDB, which provides a secure and scalable infrastructure for KV service. ResilientX also provides friendly user interfaces so that it can be a platform for users to share their thoughts and ideas, and to connect with others. Our evaluation demonstrates the scalability and efficiency of ResilientX. ResilientX is a step towards a more democratic and transparent social platform that empowers users and protects their privacy.

REFERENCES

[1] Twitter Transparency Center. 2021 (accessed Oct, 2023). Rules Enforcement. https://transparency.twitter.com/en/reports/rules-enforcement.html#2021-jul-dec

[2] Federal Trade Commission. 2022 (accessed Oct, 2023). Twitter, Inc., U.S. v. https://www.ftc.gov/legal-library/browse/cases-proceedings/2023062-twitter-inc-us-v

[3] Suyash Gupta, Sajjad Rahnama, Jelle Hellings, and Mohammad Sadoghi. 2020. ResilientDB: Global Scale Resilient Blockchain Fabric. *Proc. VLDB Endow.* 13, 6 (feb 2020), 868–883. https://doi.org/10.14778/3380750.3380757

[4] NPR. 2023 (accessed Oct, 2023). Elon Musk takes @x handle away from its original owner. https://www.npr.org/2023/07/29/1190891082/twitter-x-account-owner-gene-hwang-elon-musk

[5] Jonathan A. Obar and Steven S. Wildman. 2015. Social Media Definition and the Governance Challenge: An Introduction to the Special Issue. *SSRN Electronic Journal* (7 2015). https://doi.org/10.2139/SSRN.2647377

[6] Jomon Aliyas Paul, Hope M. Baker, and Justin Daniel Cochran. 2012. Effect of online social networking on student academic performance. *Computers in Human Behavior* 28 (11 2012), 2117–2127. Issue 6. https://doi.org/10.1016/J.CHB.2012.06.016

[7] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017* (9 2017), 557–564. https://doi.org/10.1109/BIGDATACONGRESS.2017.85