

Final Project



Resilient Resume About Me

Student: Chen-Yu Yu, Tsung-Chieh Chen, Howard Ku, Jessica Chen, Elliot Lin

Student ID: 921545136, 918374941, 921542663, 921523226, 921531687

Course: ECS 265: Distributed Database Systems

GitHub Link: <https://github.com/tobyyu007/Res-a-Me>

About Page: <https://tobyyu007.github.io/resame/>

1. Introduction

1.1 Motivation

In our analysis of contemporary resume-providing websites, such as LinkedIn [1], Glassdoor [2], and Indeed [3], has unveiled several critical challenges that need addressing. Firstly, the **Centralization** of these platforms poses significant risks, including potential data manipulation, restricted user access, and vulnerability to service disruptions. Such centralization concentrates control, often leading to issues in data management and reliability. Secondly, we noted concerns regarding **Data History and Integrity**. The restricted public access to a user's historical data and the risk of unauthorized data alteration by platform operators jeopardize the trust and accuracy of the information presented. Lastly, the issue of **Non-Standardization** stands out. The absence of a uniform resume format complicates the process for employers, who struggle to efficiently compare and analyze candidates' resumes due to varying structures and content layouts.

Furthermore, it is essential to acknowledge the pivotal role resumes play in our everyday lives. They are not just professional documents but repositories of personal information and career milestones, often containing sensitive data such as personal achievements, work history, educational background, and sometimes even personal contact details. The management and protection of this data are of utmost importance, not only to maintain privacy and security but also to ensure that an individual's professional narrative is accurately and securely represented. Therefore, the need for a robust, transparent, and user-centric platform for resume management and sharing is more critical than ever. The ideal platform would not only address these identified challenges but also respect and safeguard the personal nature of the information contained within resumes.

1.2 Objectives

Recognizing the prevalent issues in current resume-providing platforms, our objective is to create a decentralized solution that ensures the immutability of user resumes. Blockchain technology, which is increasingly being adopted in various sectors including finance [4], supply chain [5], and digital identity management [6], provides several key advantages for a resume-providing platform. Firstly, its decentralized nature effectively mitigates risks associated with data control and eliminates single points of failure. This aspect is crucial in creating a resilient and reliable platform. Secondly, blockchain brings transparency and immutability to the forefront. With a transparent ledger system, every data entry made on the platform is permanent and unalterable, significantly enhancing the integrity of user data. This feature ensures that a user's professional history is accurately and securely recorded. Lastly, blockchain technology facilitates the creation of a standardized resume format, which is a significant improvement over current platforms. This standardization not only streamlines the resume comparison process for employers but also enables the reliable verification of credentials and employment history. Through these features, blockchain technology stands as an ideal foundation for revolutionizing how resumes are created, shared, and verified in the digital age.

In response to these needs, we developed Res-a-Me, leveraging ResilientDB [7] as our blockchain backbone. Res-a-Me offers:

- **Decentralization:** A robust, decentralized network with multiple replicas to store user data, ensuring public accessibility and service fault tolerance.
- **Data History and Integrity:** Users' data is immutably stored on ResilientDB with a unique transaction ID for each entry, guaranteeing data integrity.
- **Standardization:** A standardized resume format simplifies the process for employers to compare and evaluate candidates effectively.

The integration of blockchain technology in Res-a-Me promises to transform the management, sharing,

and verification of professional credentials. This approach aims to make the process more transparent, secure, and efficient, benefiting both job seekers and employers.

The project's name, Res-a-Me, is a play on words, signifying 'resilient resume about me.' It encapsulates our vision of providing a resilient, reliable platform where individuals can confidently present their professional narrative. The name reflects our commitment to ensuring that users' resumes are not only secure and immutable but also easily accessible and verifiable. This aligns perfectly with our goal of building a resilient, user-centric platform that stands the test of time and technological changes, just as a resume should accurately and enduringly represent an individual's professional journey.

2. Application Architecture

2.1 ResilientDB Overview

ResilientDB is a high-throughput distributed ledger technology, designed on scale-centric principles to democratize and decentralize computation [7]. It revolutionizes the vision of a democratic and decentralized computational paradigm by embedding privacy, integrity, transparency, and accountability into its model. ResilientDB effectively addresses the shortcomings of classical fault-tolerant consensus protocols, which have struggled to satisfy the requirements of high-performance modern applications. It reimagines the blockchain framework, incorporating parallelism and deep pipelining at every layer, thus fully leveraging contemporary hardware and cloud infrastructure on a global scale.

Res-a-Me is fundamentally powered by ResilientDB, which plays a pivotal role in realizing our vision of deploying resumes on a blockchain-based platform. ResilientDB is not just the technological backbone of Res-a-Me but also the enabler of its key features. It brings to the table robust functionalities that are critical for implementing a decentralized, secure, and transparent system for resume management.

ResilientDB's primary functionality lies in its distributed ledger technology, which allows for secure and immutable record-keeping. This is instrumental in maintaining the integrity and authenticity of the resumes stored on Res-a-Me. Every transaction or data entry made on the platform is recorded on this ledger, ensuring that each user's professional history is preserved accurately and tamper-proof. This feature aligns perfectly with the need for reliability and trustworthiness in professional documentation.

Furthermore, ResilientDB's scalability and high throughput capabilities ensure that Res-a-Me can handle a large volume of user data efficiently, without compromising on speed or performance. This is crucial for providing a seamless user experience, even as the platform scales up and accommodates more users and data.

In Res-a-Me, we utilize ResilientDB's advanced consensus mechanisms to manage and verify user credentials and updates securely. This mechanism ensures that only authenticated users can make changes to their resumes, thereby protecting user data from unauthorized access or edits. Additionally, ResilientDB's decentralized architecture enhances the resilience of the Res-a-Me platform against potential system failures or cyber threats, ensuring continuous availability and reliability.

2.2 Architecture Overview

Our application's frontend is engineered using Vue.js [8], a cutting-edge JavaScript framework, complemented by Vuetify [9] for crafting user interface components. On the backend, we harness the power of Python for its versatility and integration with Flask [10], our chosen microservice framework. This combination not only enhances backend functionality but also facilitates effective interaction with ResilientDB [7], a key component of our architecture.

The Presentation Layer serves as the primary interface for user interaction with the application. Its design prioritizes intuitiveness and responsiveness, thereby ensuring an engaging user experience. In this layer, every user action triggers an authentication process to verify identity and permissions, reinforcing secure and reliable access.

Once authentication is confirmed, user requests progress to the Application Layer. This layer is integral, housing our Web Services API that forms a connection between the user interface and the backend processing units. It is responsible for orchestrating the application's logic, handling user requests, and interfacing with the Data Layer for data retrieval and storage.

The Data Layer is the foundation where our data is securely stored and managed. It leverages GraphQL for efficient data handling, interacting seamlessly with ResilientDB to ensure that data delivery to the frontend is both optimized and relevant. This layer is crucial in maintaining the application's high performance and responsiveness.

By integrating ResilientDB's advanced features, our application stands as a leading solution in the realm of digital technology, offering security, efficiency, and scalability. The overall structure, as depicted in Figure 1, is organized into distinct, well-defined layers to simplify comprehension and navigation.

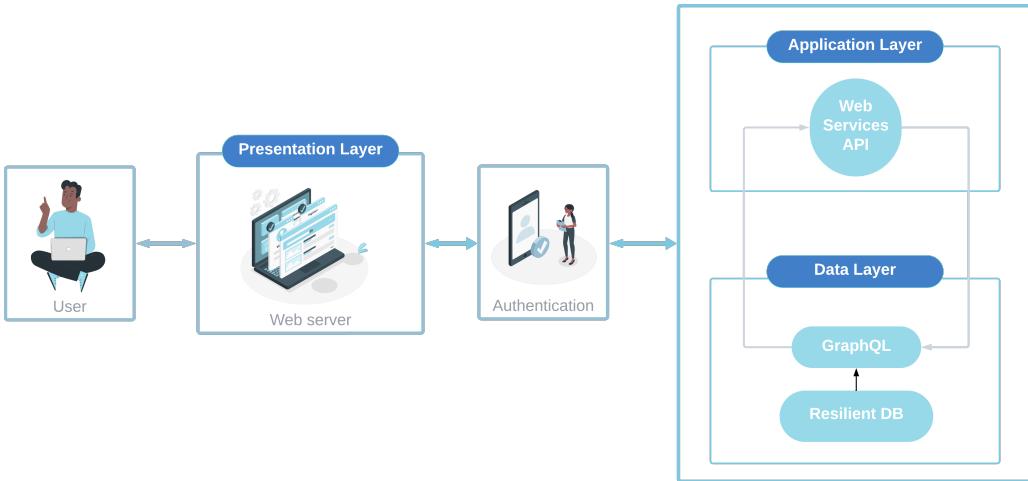


Figure 1: Architecture Overview

2.3 Authentication mechanism

Given that resumes contain sensitive personal information, it is imperative to prevent unauthorized editing. To ensure a secure and consistent experience on our Res-a-Me platform, we have implemented a token exchange workflow. In our Single-Page Application (SPA), the Authentication mechanism operates through an Authorization Server (AS). When a user successfully registers or logs in, the AS issues Access and Refresh Tokens. The access token grants entry to protected APIs on the Resource Server (RS), while the refresh token enables continuous user sessions. For example, if the access token expires, the SPA uses the valid refresh token to request new tokens from the AS. This method maintains session continuity, reducing the need for frequent re-logins. Figure 2 visually demonstrates this token flow, highlighting the interaction between the SPA, AS, and RS in the authentication process.

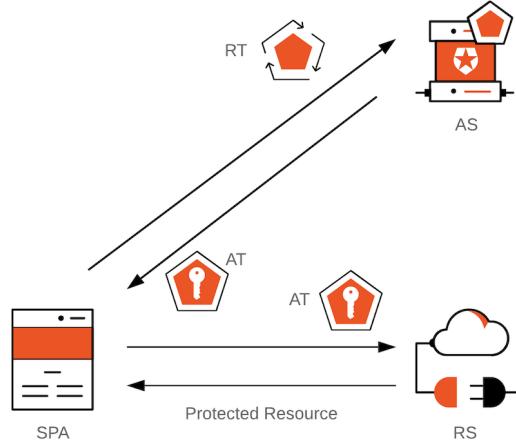


Figure 2: Token Exchange Flow

3. Frontend

The Res-a-Me platform is designed to offer a seamless and interactive experience through its various views: the Search View, Login/Register View, Resume View, and Edit View. Each view serves a specific purpose, enhancing the user's journey from registration to resume creation and networking.

3.1 Search View



Figure 3: Search View

As shown in Figure 3, our search view aims to enable users to easily locate individuals by searching their names, thereby enhancing community networking. To help users locate other individuals, we design a search interface with autocomplete functionality. The autocomplete suggestion bar displays each user's profile picture and job title, allowing for quick and accurate identification of the individuals being searched for. To further promote community networking, we implemented a "Make Connections" button. This allows users to view another user's resume, facilitating opportunities for making connections.

For logged-in users, the bottom section of the page features a "My Res-a-Me" button, providing easy access to their own resume page. Additionally, at the top right of the search view is a login/logout button, which allows users to log in to our Res-a-Me system or to be redirected to the Login/Register view.

3.2 Login/Register View

The Login/Register View, as shown in Figure 4, is the primary access point to the Res-a-Me application. Our design aims to provide a secure and user-friendly interface, ensuring a personalized experience for accessing Res-a-Me. This interface supports user authentication, enabling individuals to either log in with existing credentials or register for a new account. For returning users, the login view (Figure 4a) allows for easy credential entry and platform access through a 'login' button. New users can create an account via the 'Register' button, with a convenient toggle feature to switch between Login and Register Views. This toggle is especially useful for returning users who need to log in.

Additionally, the Register View (Figure 4b) incorporates checks to validate the format of the entered email and ensure password consistency, a key aspect in maintaining system security and safeguarding user credentials. Each user profile is securely linked to a verified individual, enhancing the overall integrity of the system.

Upon initiating a registration or login, the frontend sends a POST request with the user's email and password to our backend Flask server. Successful registration triggers the storage of access and refresh tokens in the session storage, followed by redirection to the Edit View. For a successful login, the frontend updates any existing tokens and redirects the user to the Resume View. Error handling mechanisms are also in place to guide users through common issues, such as incorrect password entry or login attempts with unregistered emails.

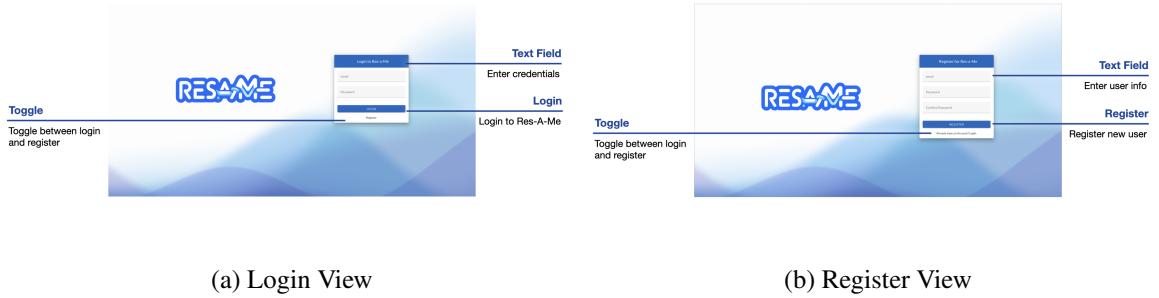


Figure 4: Login/Register View

3.3 Resume View

The resume view is a key feature of our platform, designed to present a detailed and engaging profile for each user. This view is segmented into five primary sections: About, Education, Skills, Experience, and Achievements, enabling comprehensive customization and display of a user's professional background. Figure 5 illustrates these sections. A sidebar on the left displays user information, including a profile picture and current position, along with quick navigation shortcuts to each section and the search feature.

In the About section (Figure 5a), users can write a brief introduction, giving others a snapshot of their professional journey. The Education section (Figure 5b) allows users to list their academic history, including institution names, degrees earned, graduation dates, and relevant descriptions. This information is presented in a timeline format, clarifying the chronological order of their academic milestones.

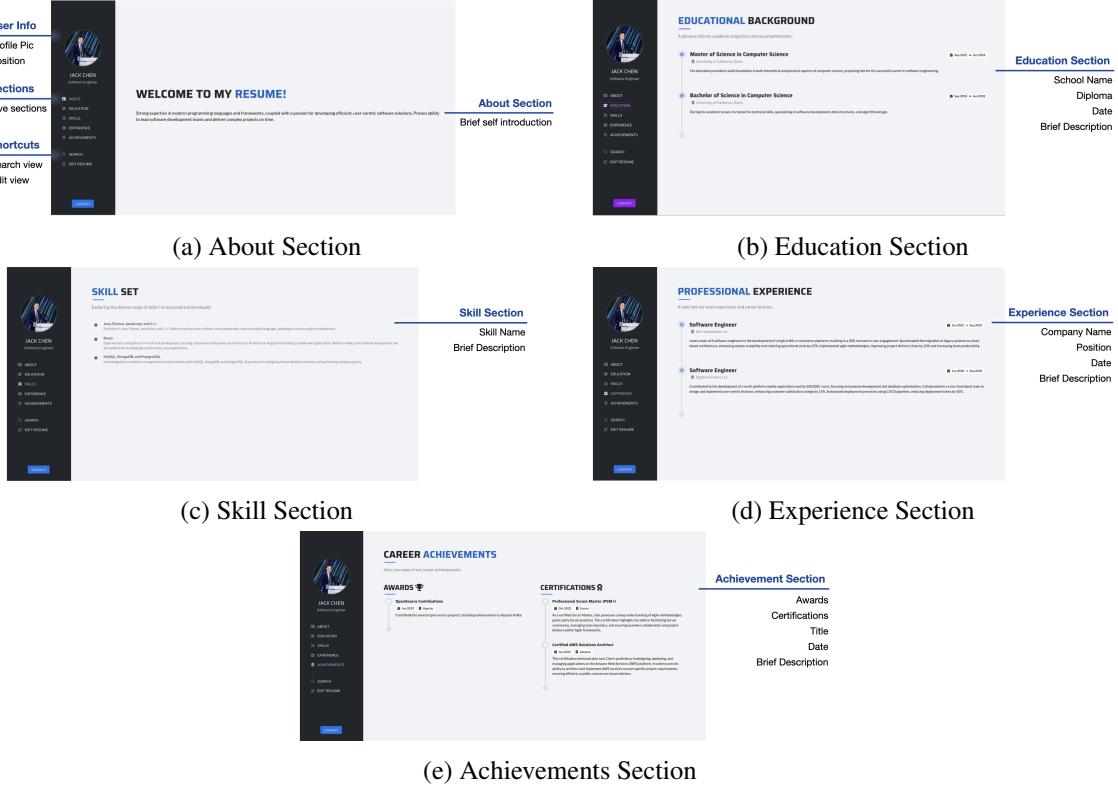


Figure 5: Resume View

The Skills section (Figure 5c) is for users to highlight their competencies, with space for both skill names and brief descriptions. Skills are listed in bullet points for a quick overview. The Experience section (Figure 5d) lets users detail their professional experiences, including roles, companies, employment duration, and responsibilities. Like Education, these details are arranged in a timeline for a clearer understanding of the user's career progression.

Finally, the Achievements section (Figure 5e) is dedicated to showcasing awards and certifications. Users can list achievement with its title, the date of receipt, and a brief description. This section also includes a timeline, emphasizing the chronological order of these achievements.

Our belief is that this structured, yet flexible design provides a comprehensive view of a user's history, offering recruiters a balanced and detailed perspective for evaluation. For updates or changes, users can easily access the edit view, where they can modify and refine their resume.

3.4 Edit View

Understanding that users' careers are continually evolving, we've designed the edit view of our platform to facilitate easy and intuitive updates to their resumes. Mirroring the resume's structure, this view consists of five corresponding sections: About, Education, Skills, Experience, and Achievements, with a collapsible design for streamlined navigation (Figure 6). Each section is pre-populated with the user's existing data for convenient editing (Figure 6a).

In the About section (Figure 6b), users can update basic information such as name, job, address, and a personal introduction. Additionally, users can upload and edit their profile picture, using our picture cropping tool (Figure 6c) for precise image adjustments. Once satisfied, users can save their changes with the 'save'

(a) Overview

Six Sections
Expand each section to start editing

(b) About Section

About Section
Name
Job Title
Address
Brief Introduction

Button
Choose profile image
Upload image
Save Edited Info.

(c) Crop picture

Button
Confirm the cropped image

Crop Picture
Draggable Resizable

(d) Education Section

Education Section
School
Diploma
Start/End Date
Brief Description

Delete
Delete unwanted info

Buttons
Add more sections
Save edited info

(e) Experience Section

Experience Section
Company
Position
Start/End Date
Brief Description

Delete
Delete unwanted experience

Button
Add more experience
Save Edited Info.

(f) Skill Section

Skill Section
Skill name
Brief Description

Delete
Delete unwanted skill

Button
Add more skill
Save Edited Info.

(g) Certification Section

Certification Section
Title
Start/End Date
Brief Description

Delete
Delete unwanted certification

Button
Add more certification
Save Edited Info.

(h) Award Section

Award Section
Title
Start/End Date
Brief Description

Delete
Delete unwanted award

Button
Add more award
Save Edited Info.

Figure 6: Edit View

button.

The Education section (Figure 6d) allows for updates to academic history, including institution names, degrees, dates, and descriptions. Recognizing the diversity of educational backgrounds, we've included options

to add or remove entries as needed. After editing, changes can be saved with a single click.

Similarly, the Experience section (Figure 6e) provides a format for updating professional history, with fields for company names, positions, dates, and role descriptions. This section also allows for the addition or removal of experiences, catering to varied career paths.

The Skills section (Figure 6f) is designed for users to list and describe their skills, with the flexibility to add or delete entries to match their evolving skill set. Likewise, the Certifications and Awards sections (Figures 6g and 6h) enable users to update titles, dates, and descriptions of their certifications and awards, with options to manage the number of entries. These two sections, while separate in the edit view, are combined in the Achievements section of the resume view for a cohesive presentation.

4. ResilientDB and Its Integration in Res-a-Me

As outlined in Section 2, ResilientDB forms the technological backbone of the Res-a-Me platform. ResilientDB, a high-performance blockchain database system, is pivotal in deploying resumes on a blockchain, enhancing security and integrity. This section will briefly describe ResilientDB's key functionalities and their integration within Res-a-Me.

4.1 Key Methods in ResilientDB

ResilientDB incorporates several methods vital for the operation of the Res-a-Me platform:

1. `generate_key_pair()`: This method is crucial for user security. It generates a unique public-private key pair for each new user, with the private key kept confidential and the public key stored in ResilientDB. This pair is essential for encrypting and signing transactions, thus securing user data.
2. `create()`: This function initiates a new user profile by creating a token with a blank metadata template. It assigns the creator and recipient as the user, and signs it with the user's private key to prevent unauthorized profile modifications. The function returns a transaction ID for future updates.
3. `transfer()`: Used for profile updates, this method retrieves and updates the user's token with modified data from the frontend. It secures the process by signing with the user's private key, ensuring data integrity. The new transaction ID is saved for future reference.
4. `get()`: A fundamental retrieval function, it fetches the user's token using the transaction ID. The token's metadata field contains the resume data, crucial for profile viewing and editing.

4.2 Utilization of ResilientDB in Res-a-Me

Each ResilientDB method is integral to different aspects of the Res-a-Me platform:

4.2.1 Search View

In the Search View, the `get()` function is utilized to retrieve a list of all users, which supports the autocomplete functionality of the search bar. This enables users to quickly find and view profiles of other users on the platform.

4.2.2 Register View

During registration, the `generate_key_pair()` function is used to create a secure public and private key for each user on ResilientDB. Subsequently, the `create()` function generates a unique token for the user. This

token, signed with the user's private key, ensures that only the user can modify their resume's metadata field, establishing a secure foundation for each user profile.

4.2.3 Resume View

Employs `get()` to access user data based on saved transaction IDs, allowing users to view their or others' resumes. The Resume View leverages the `get()` function to retrieve a user's data from ResilientDB based on their saved transaction ID. This feature allows users to view their own or other users' resumes, depending on the access rights and data available.

4.2.4 Edit View

In the Edit View, the `get()` function is initially used to fetch the user's existing data from ResilientDB. After modifications are made in the frontend, the `transfer()` function updates the resume in the metadata subfield and saves these changes back to ResilientDB. This ensures that any updates to the resume are securely stored and reflected in the system.

These ResilientDB methods underpin the functionality of the Res-a-Me platform, ensuring secure, efficient, and user-friendly interactions across various views. The methods not only maintain the integrity and security of the data but also provide a seamless user experience, from registration to editing and viewing resumes.

5. Backend

5.1 API Design

The API functions can be divided into three categories to satisfy our four views, as shown in Figure 7: User Management, User Listing, and Resume Interactions. The User Management APIs mainly focus on managing user accounts, which are used in the login view. The User Listing API returns all users that exist in our system. Finally, the Resume Interactions APIs are required to communicate with ResilientDB and can retrieve resumes through transactions. A full API list is given in Table 1.

1. User Management
 - (a) **Register:** Register a new user. This function handles the registration of a new user by receiving the email and password from the request's JSON payload.
 - (b) **Login:** Log in existing users. This function checks the user credentials against the existing users in the database.
 - (c) **Refresh:** Return new access token and refresh token if the input refresh token is still valid.
 - (d) **Logout:** Log out the currently authenticated user and redirect to the login page.
2. User Listing
 - (a) **User List:** Send the list of all existing users in the system.
3. Resume Interactions
 - (a) **Update Resume:** Retrieve the resume data for any user that is chosen. Since the resume page can show not only the user's resume but also the user they search on the search page, the resume data should include all five categories.
 - (b) **Load Resume:** Retrieve the resume data for the authenticated user according to the category. The content of the resume will be sent to the frontend.

- (c) **Edit Resume:** Get the revised data from the frontend and attach the new resume to the metadata of the user's token according to the category.
- (d) **Upload Image:** Upload the personal photo for the profile category.

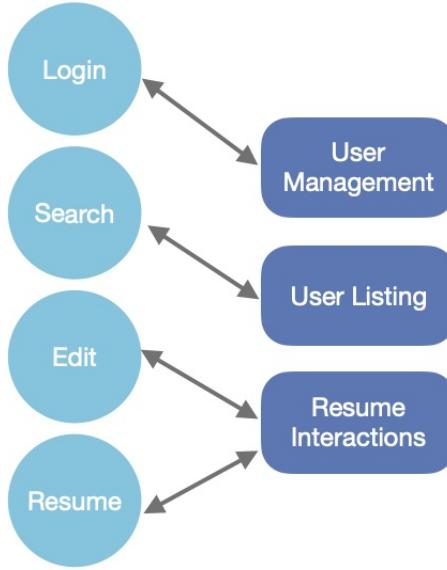


Figure 7: Relationship between API functions and four views

API purpose	Description
User Management	
register	Register a new user and create a token on ResilientDB
login	Check if the email and the password are valid
refresh	Return new access token and refresh token if the input refresh token is still valid
logout	Log out the currently authenticated user
User Listing	
userList	Send the list of existing users in the system
Resume Interactions	
loadResume	Retrieve the resume data for the authenticated user according to the category
editResume	Update the metadata of the user's token according to the category
updateResume	Retrieve the resume data for any user that is chosen
upload	Upload the image

Table 1: Full API List

5.2 Backend Services

To run our services, you need to set up three components. In Figure 8, the first one is the ResilientDB server, which deploys 4 replicas, 1 client proxy, and the Key-Value Service. Additionally, since we use HTTP for our main web service, ResilientDB_GraphQL should also be configured to initiate the Crow HTTP service. Finally, you can run Res-A-Me, which manages the API services for our application.

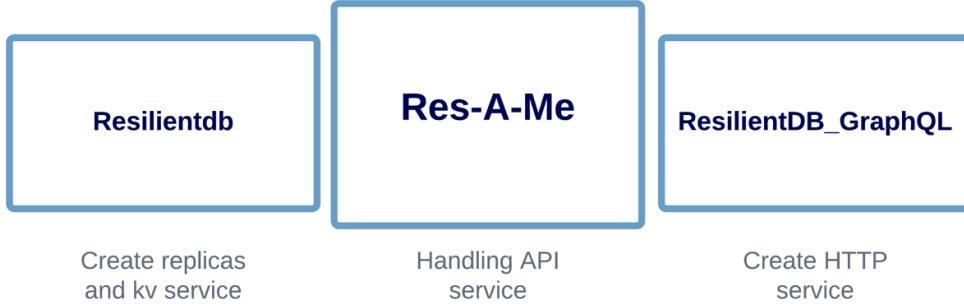


Figure 8: Three services are required to execute Res-A-ME

5.3 Res-A-Me Structure

Our main program will be divided into several parts: Main, controller, user management, and resume management. The Main part is used for frontend requests, while the controller part can call functions related to user management or resume management. The structure of our system is shown in figure 9

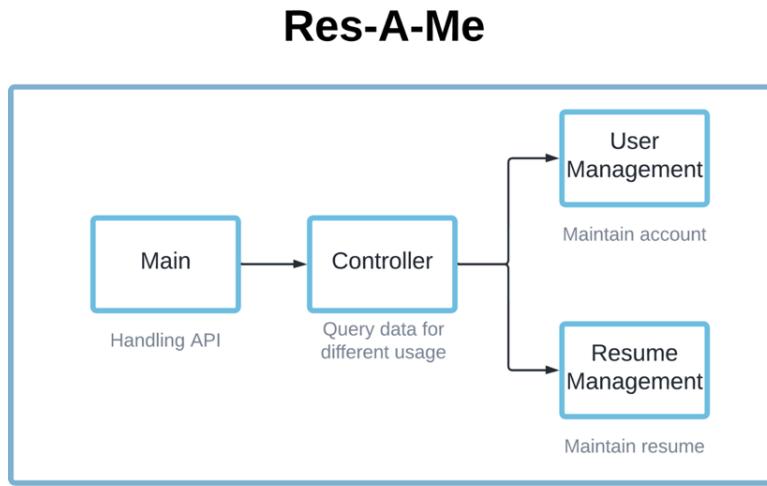


Figure 9: Three services are required to execute Res-A-ME

6. User Journey

The Res-a-Me platform is designed to accommodate a variety of user use cases, providing distinct experiences based on the user's interaction with the platform. Here are detailed journeys for different types of users:

6.1 New User

First-time users are welcomed by the Search View (section 3.1) upon visiting the Res-a-Me website. For those looking to join, the login button leads them to a registration form (section 3.2). This form features email verification to ensure correct format and a password confirmation step for security. Once registration is complete, users are directed to the Edit View (section 3.4). Here, they can fill in their personal and professional information, including job title, a brief self-introduction, and experience details. A notable feature is the option to upload and crop a profile picture in the About section for a perfect fit in the Resume View. After entering

all details and saving, users can navigate to the Resume View (section 3.3) to see their newly created profile. Finally, logging out allows users to explore other user journeys.

6.2 Returning User

For users who have previously registered, logging in (section 3.2) takes them directly to their personalized Resume Page (section 3.3). This page displays their profile information, and users can navigate through their resume using clickable shortcuts in the sidebar. The ‘Edit Resume’ shortcut offers quick access to the Edit View (section 3.4) for any updates or modifications to their information. Additionally, the ‘Search’ shortcut will navigate users back to the Search View (section 3.1), which enables users to find and view other profiles, equipped with an autocomplete feature for convenience. Logged-in users can also view their profile (Resume View) by clicking their profile image at the bottom of the Search page (section 3.1). Selecting and clicking on a profile’s Res-a-Me button allows users to view that individual’s full resume.

6.3 Non-Authenticated User

Our platform caters to curious visitors as well. Non-authenticated users, or those not logged in, can still search and view other people’s profiles (section 3.1). This functionality allows them to explore the platform (section 3.3) without the need for an account. However, such users won’t have access to the Edit Resume feature, maintaining the exclusivity of editing rights to authenticated users only. This distinction ensures the privacy and security of user data, allowing only authorized users to make changes to their profiles.

7. Future Work

Currently, Res-a-Me does not provide functionality for verifying the authenticity of a user’s resume. For example, users could potentially list false experiences, leading to challenges for employers during the candidate verification process. However, leveraging the blockchain technology underlying Res-a-Me, we have the potential to address this issue innovatively. By integrating smart contracts, we can automate the verification process whenever a user updates their resume. This process would involve confirming the accuracy of the information by directly contacting the associated company or educational institution. Implementing such a feature would greatly enhance the trustworthiness of the information on our platform, benefiting both recruiters and job seekers. Recruiters would be assured of the authenticity of candidates’ resumes, while job seekers could credibly showcase verified experiences and qualifications.

The implementation of this verification functionality is currently limited by the need for direct partnerships with companies and educational institutions. However, with the successful expansion of the Res-a-Me platform, establishing these connections becomes more feasible. In the future, this could evolve into a core feature of our platform, significantly enhancing the overall value and reliability of the service we offer.

Additionally, the current iteration of Res-a-Me is focused on viewing and editing resumes, without the capacity to directly apply for jobs. Introducing a job application feature could significantly enrich the platform’s functionality. This would allow users to apply for positions directly through Res-a-Me, streamlining the recruitment process. It would also enable recruiters to interact with and review applicants’ resumes more efficiently within the same ecosystem. Incorporating this feature would transform Res-a-Me into a more comprehensive career management and job application tool, offering a one-stop solution for both job seekers and employers.

In summary, as we look towards the future of Res-a-Me, our vision includes the development of advanced features such as resume verification and job application functionalities. These enhancements aim to provide a more robust, reliable, and all-encompassing platform for professional development and career advancement.

8. Conclusions

In conclusion, Res-a-Me, leveraging the robust foundation of ResilientDB, improves the realm of professional credential management on a blockchain-based platform. ResilientDB's distributed ledger technology underpins our platform, ensuring security, immutability, and transparency in managing resumes. Its scalability and high-throughput capabilities are key to handling extensive data volumes efficiently, offering a seamless experience as the platform expands.

The frontend of Res-a-Me, crafted with Vue.js and Vuetify, is a testament to our dedication to intuitive and interactive user experiences. The platform features various views including Search, Login/Register, Resume, and Edit, each tailored to enhance the user journey from registration to networking.

Significantly, we have considered various user use cases to ensure a comprehensive and user-friendly platform. Whether it's a new user discovering the platform, a returning user accessing their personalized resume, or a non-authenticated user exploring the platform's capabilities, each journey has been thoughtfully designed. This approach ensures that all users, regardless of their interaction level, can navigate and utilize the platform effectively.

Res-a-Me stands as an exemplary platform that marries advanced technology with user-centric design. It's not just about adopting blockchain technology, but about harnessing its full potential to redefine professional credential management. As Res-a-Me continues to evolve, it is set to become an essential tool in the professional world, addressing the dynamic needs of both job seekers and employers in the digital era.

In essence, Res-a-Me represents a platform that caters to diverse professional needs, demonstrating our commitment to innovation and user satisfaction in the digital landscape of career management.

9. Contribution

1. **Chen-Yu Yu:** Search view/Vue Integration/Flask Integration/Product Manager
2. **Tsung-Chieh Chen:** Login View/Register View/Flask and Vue communication
3. **Jessica Chen:** Edit View/Vue Integration
4. **Howard Ku:** Flask Integration/Resume Editing API/Name Searching API
5. **Elliot Lin:** Flask Integration/Register API/Login API/Resilient DB

References

- [1] LinkedIn. <https://www.linkedin.com/>.
- [2] Glassdoor. <https://www.glassdoor.com>.
- [3] Indeed. <https://www.indeed.com/>.
- [4] Blockchain for financial services. <https://www.ibm.com/blockchain/industries/financial-services>.
- [5] Blockchain for supply chain solutions. <https://www.ibm.com/blockchain-supply-chain>.
- [6] Blockchain for digital identity and credentials. <https://www.ibm.com/blockchain-identity>.
- [7] resdb. <https://github.com/resilientdb/resilientdb>.
- [8] Vue.js. <https://vuejs.org/>.
- [9] vuetify. <https://vuetifyjs.com/en/>.
- [10] Flask. <https://flask.palletsprojects.com/en/2.2.x/>.