International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

DECENTRALIZED CHAT APPLICATION

[#1] E. Subramanian, [@2]Rohith S,[@3]Saiesh C B,[@4]Sujith R
#1Assistant Professor, Sri Shakthi Institute of Engineering and Technology, Coimbatore
#1esubramaniancse@siet.ac.in,
@rohiths22cse@srishakthi.ac.in SIET, Coimbatore,
@saieshcb22cse@srishakthi.ac.in SIET, Coimbatore
@sujithr22cse@srishakthi.ac.in SIET, Coimbatore.

Abstract - When we wanted to buy a product like a watchon the internet, then suddenly we see ads related to the product on Facebook, Instagram, Google, etc this means they are accessing our every information even the messages wesend to people on the Internet. There is a growing suspicionthat the web is betraying and spying on us. As communication is an important part of an individual's lifestyle, as each person communicates globally with the means of the internet every day and as in today's world, different chat systems are almost working on centralized systems i.e., all the data is in a centralized server. Therefore, a major problem is if the central server fails then the wholenetwork fails, and due to this major drawback is that therecan be a loss of user's data, information, and resources which is stored on the centralized server or even there can be a leak of user's chat information that is stored on the server.

Decentralized is the way to resolve this problem, it's an internet hosted via a peer-to-peer network. The information will be distributed and stored around the world on multiple devices like phones, laptops, and even smart appliances. To achieve this, we are using Gun.js, a decentralized graph database that is real-time has Low latency, and also has security, encryption, and authorization for the data.

Keyword- Gun.js, webRTC, node.js, Decentralized server, User Interface, Peer to peer network.

I. INTRODUCTION

In today's world chatting over the messaging platforms are a part of an individual's lifestyle. The most of the communication whether it is a confidential chat or normal chat are mostly happens over in social media platform. As we all know that most of the traditional chat application are working on a centralized server i.e., all the user data is storedon a central server. So, the major problem of this server is ifthe central server fails then the whole data collapses (breakdown) or even can leak the user information stored on the server. This is the major drawback of this system, but in today's world privacy makes a very important role in everyone's life and specially for those organizations that works on confidential tasks like for example the military purpose, they need a very secured privacy to handle their communication from the one end to the other end. So, to achieve this system our project had made the use of decentralized backend server, as the name suggests decentralized server it does not have any central server. It ispeer to peer network i.e., all the computers are linked together with equal permissions for processing data. To achieve this decentralized backend server, we used a library called as Gun.js, a decentralized graph database. In this, forfront end UI of our chat application, we used JavaScript library. Decentralized application consists of multiple nodes connected to each other in a mesh type of topology network. It's just like a peer-to-peer network that data is stored in such a way that it is almost impossible to view by an anonymous user.

II. PROBLEM STATEMENT

A. Data is stored in centralized server.

As to storing the user data, it has been stored in big servers as it also takes an ample amount of space. It has been said that if any one of the servers fails, the whole network collapse due to that loss of data, which is dangerous for the user as if their data has been lost. Due to that, you cannot retrieve your information back to overcome this issue.

So here comes our project in the picture, whereas the high-tech company stores their user in the server. At the same time, there is no guarantee that your data will be safe. Where data is stored in a hard disk somewhere in the cloud, which has been shared across multiple machines, whereas our project stores a small subset of data on each user based on the actual data that they consume via a peer-to-peer connection.

B. Lack of loyalty

As in today's fast-growing world, whenever we search for some product in a search engine after a while, we see a pop-up coming out of know ware showing the exact related product that you searched a while ago. Guess how they came to know that you were looking for this product. This may lead someone is monetizing your activity that your data is not secure, which means your data has been shared with a different company that you are not aware of, which means you have no longer control over your data where as in our

project the data is stored in a subset of data based on actual data. Which is much more secure than the centralized server.

III. SYSTEM COMPONENTS

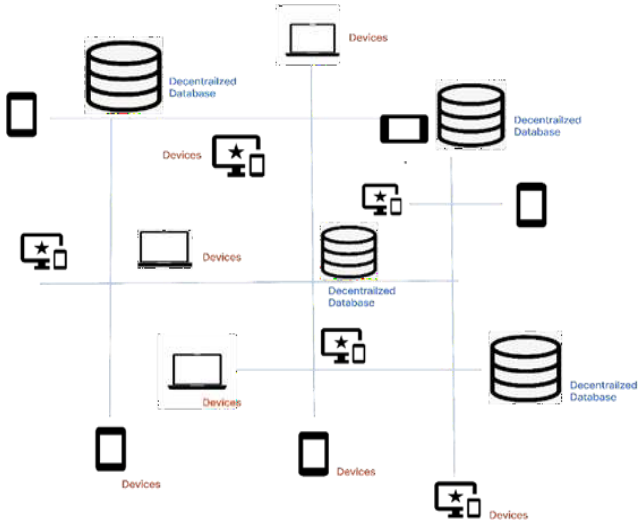

Fig.1: Diagram of Decentralized web

A. Gun.js: -

Gun.js decentralized graph database is a centralized database that which on a server hosted by big tech data in the gun is distributed across multiple peers or users throughthe use of webRTC and end-user like user requires a specific graphic data to use your app while another user requires anentirely different one with a few shared data points inbetween each peer can store the data it needs in the browsers local storage then sync changes with other peers inthe network a peer might be one of your end-users or a relayserver that you deploy to make the system more reliable thatmeans the entire database is the union of all graphs on the network, and no one machine controls the entire system a decentralized fire stored where you get real-time updates offline mode and the illusion of what latency the data itself ismanaged through a minimal graph API where each record is a node that contains a unique ID and any other custom datafor that entity in plain JavaScript any node on the graph can reference another node allowing you to create an infinite chain of circular dependencies. an individual node never contains a duplicate of another node just a reference to it allowing you to model complex relationships without the need for a schema.

Node js: Node.js allows developers to use JavaScript both on the client-side and the server-side, providing a unified language and ecosystem. This eliminates the need for context switching and enables code reuse between the front-end and back-end. This results in improved productivity and reduced development time.

Node.js has a vast and active ecosystem of modules and libraries available through the Node Package Manager (npm). This rich ecosystem offers ready-to-use tools and packages for various functionalities, such as web frameworks, database connectors, authentication, and testing frameworks. Developers can leverage these modules to accelerate development and enhance application functionality.

webRTC: -webRTC exchange real-time audio video streams with your friends entirely in the browser to capitalize on the work from home boom then webRTC is the API because it allows you to establish a peer-to-peer connection between two ormore browsers where they can exchange audio video mediadirectly without the need for a third-party server or nativeapp webRTC will make a series of requests to a stun server.

IV. SYSTEM DESGIN

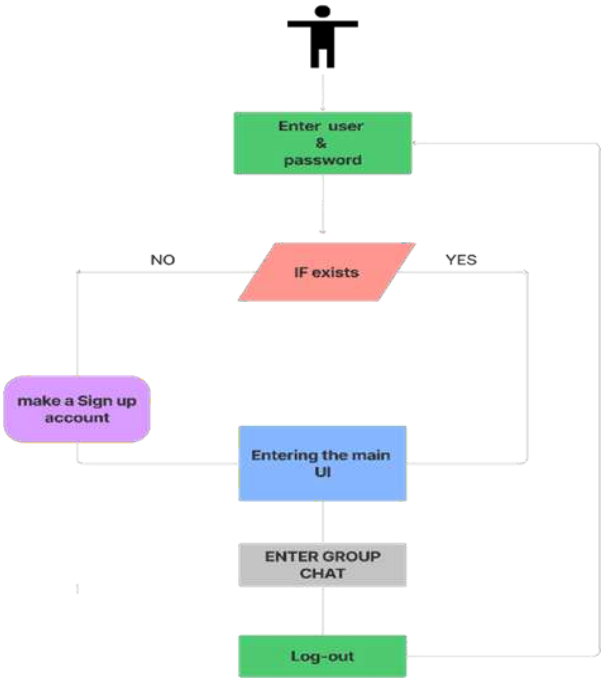

Fig. 2: User Flow Diagram

We made a chat web app where the data and infrastructure are not controlled by a big tech company instead it is decentralized across the entire user base using web technologies. We built this decentralized chatapplication using a library called gun.js and for the front-end UI, we have used figma (an open- source front-end compiler).

A gun.js is a decentralized graph database, a normal database that stores all the data in a hard disk somewhere in the cloud it may be shared across multiple machines, but for all intents a purpose we can imagine our entire data set on a single disk whereas gun.js also stores a small subset of data on each user based on the actual data they

consume in the application.

When a user makes a query for some data it will search across the network for the other users that have that data and sync up using webRTC which is a web real-time communication, so it is like an entire database as the union of all peers on the network. It's like a peer-to-peer network that data is stored in such a way that it is almost impossible to view as a very secure encryption function is used, this idea is very similar to a blockchain ledger where no individual has control of the entire network but it is not a blockchain technology which tends to be too slow.

For user authentication when we create a new user account this will generate secure key pair cryptographically and the key is associated with the public key so the past message can be found and the password is proof of work seed to prove and to decrypt access to an account's private keys.

V. SYSTEM IMPLEMENTATION

5.1 Data Storage and Replication:
The heart of Gun.js lies in its distributed graph database functionality. Data is stored in a decentralized manner across nodes, ensuring fault tolerance and resilience. Each node is both a data source and a potential data store, creating a distributed architecture that minimizes the risk of a single point of failure. Data replication mechanisms ensure consistency and availability, enabling a robust and scalable environment for social media data.
5.2 User Identity and Authentication:
User identity management in a decentralized environment is a critical aspect of our proposed solution. Utilizing public-key cryptography, users can securely authenticate themselves without relying on a centralized authority. This not only enhances security but also aligns with the ethos of decentralization, granting users complete control over their identity.
5.3 Content Distribution:
The nature of social media demands efficient content distribution. Gun.js facilitates this through its peer-to-peer communication model. Content is shared directly between nodes, reducing reliance on central servers and ensuring timely and consistent content delivery. Challenges such as network latency are addressed through optimized content propagation strategies.

VI. IMPLEMENTAION

6.1 Setting Up the Decentralized Network:
Implementing a decentralized social media network using Gun.js involves configuring nodes and establishing connections. The guide provided details the necessary software dependencies, network configurations, and steps to ensure a seamless deployment. A focus on simplicity and modularity aids developers in adopting the proposed solution effectively.

6.2 User Interface and Experience:
The user interface (UI) is designed with user experience at its core. Prioritizing simplicity and intuitiveness, the UI encourages user engagement. Additional features, such as customizable privacy settings and intuitive content sharing mechanisms, enhance the overall user experience. The goal is to provide a seamless transition for users familiar with centralized platforms.

VII. SECURITY AND PRIVACY

7.1 Decentralized Identity Management:
Security in decentralized social media begins with user identity management. Utilizing asymmetric encryption, users can securely authenticate themselves without the need for a centralized authority. This approach minimizes the risk of identity-related attacks, providing users with a level of security comparable to, if not surpassing, traditional centralized platforms.

7.2 Data Encryption and Access Control:
Privacy concerns surrounding user-generated content are addressed through robust data encryption and access control mechanisms. End-to-end encryption ensures that content is visible only to intended recipients, while access control mechanisms allow users to define who can interact with their content. These measures collectively contribute to a secure and private social media environment.

VIII. PERFORMANCE EVALUATION

8.1 Scalability:
A critical evaluation of scalability reveals the system's ability to accommodate an increasing number of users and content. Gun.js, with its decentralized architecture, exhibits a high degree of scalability, demonstrating the potential to support a large user base and accommodate growing social interactions.

8.2 Latency and Throughput:
Measuring latency and throughput provides insights into the efficiency of data transactions within the decentralized network. Comparative analyses with centralized platforms showcase the efficacy of the proposed decentralized solution. The decentralized nature of content distribution contributes to reduced latency and enhanced overall system throughput.

To user log in, we used the user auth method which

consists of three arguments, takes the username and password as first and second arguments and in the third argument, we defined a callback, in this case the callback is the value of the store that we set up earlier which will update automatically.
Now to sign up a new user is very similar to the previous but instead of calling auth we call that create method on the user and if the user creation is successful then we will log that user in automatically.
Now at this point, we have a working user authentication system.

For the main chat component which is where we query items from the database and which also gives the user the ability to send a message.

The chat component has two pieces of state, a string for a new message that the user will type into form input and an array of messages which will contain the message text along with the user who sent it and a timestamp.
To query messages, we set up the onMount lifecycle hook that will run whenever the component is first initialized, inside it, we used the database to make a reference to the chat node as for creating super chat we just give it a name of chat.
Then we just call map to loop over every single message in the chat and then call once to only read each message once, in this case each message is immutable so we don't need to listen to changes in real-time, then we defined a call-back that will be called on each new message that will give us access to the data and the id of that node.

Now if the data is defined, we format a message more suitable for the User Interface with the properties of who, what, and when.

1. Who - It is the user that sent the message and we can figure out who sent it by taking that data of the raw message and using it to get the alias,
2. What - It is the actual text of the user's message,
3. When - It is the property to get an accurate timestamp across all of our users, we're using Gun state as the final source of truth with the raw data, this gives us all the data we need for the User Interface.

Figma is a powerful tool for web design, offering collaborative features, real-time editing, and a robust set of design tools. Here's how you can use Figma for web design:
Wireframing: Begin by creating wireframes of your website layout. Figma's interface allows you to easily drag and drop UI elements like buttons, forms, and navigation bars to quickly sketch out the structure of your website.
Prototyping: Figma allows you to create interactive prototypes by linking different frames together. This enables you to simulate user interactions such as clicking buttons or navigating between pages, providing a realistic preview of how your website will function.
Responsive Design: Figma's responsive design features make it easy to design websites that adapt to different screen sizes and devices. You can create multiple artboards for different breakpoints (e.g., desktop, tablet, mobile) and design variations for each layout.
Assets and Libraries: Figma enables you to create reusable design components and build your own design libraries. This is especially useful for maintaining consistency across your website and ensuring that UI elements like buttons, icons, and typography styles remain uniform.

Overall, Figma offers a comprehensive set of features tailored to the needs of web designers, making it an ideal choice for creating high-fidelity web designs efficiently and collaboratively.

We made a form to submit them in the UI (User Interface), inthe form on the submit event will call the send message function. Inside that function, we use SEA to encrypt the actual message text and the key is the same as the one we used earlier to decrypt the messages. Now we can associate the message to the current user using the encrypted message as the value then we created a date to serve as the index for the message so it can be sorted properly at which point, we can reference the chat collection create a new node based on that index and store the message value.

IX. ADVANTAGES

1. For Government/Military officials: - It will be very useful for government security officials when they need to send their confidential messages or information regarding any work or to send their data to their intenders for some official reasons without any third-party involvement. As they don'tneed to trust any other application.

2. Users don't have to put trust in any Central authority.
    a. You don't need to trust any other chat application that is working on a centralizedserver or followed by the third parties that use or sell your data, information, photos to any adman who follows you around theinternet. As this decentralized chat application will be useful for securing your data you should be able to reduce or eliminate the trust that you're required to put into third parties.

X. CONCLUSION AND FUTURE WORK

CONCLUSION

Decentralized applications tend to form the interaction between two people more efficient and simpler. The chatting process nowadays features a mediating node, while our software doesn't have any mediating device/node i.e. Everyperson is connected by a peer-to-peer network. A Decentralized network can be a key technology that can solve privacy and confidentiality-related issues that exist in the Traditional or existing messaging system

FUTURE WORK

Customization and Personalization: Chat apps will offer extensive customization options, allowing users to personalize their interface, notification settings, chat themes, and other features according to their preferences

Real-time Translation: Advanced language translation capabilities will enable real-time translation of conversations, breaking down language barriers and facilitating global communication and collaboration.

Emotional Intelligence: Chat applications will continue to evolve towards greater emotional intelligence, understanding and responding to users' emotions through sentiment analysis, empathy, and context-aware responses.

AI Integration: Further integration of artificial intelligence (AI) will enhance chat applications' capabilities. AI-driven chatbots will become more sophisticated, offering personalized assistance, natural language understanding, and improved conversational experiences.

REFERENCES

[1] Sourabh, Deepanker Rawat,Karan Kapkoti,Sourabh Aggarwal,Anshul Khanna, "bChat: A Decentralized Chat Application",Inderprastha Engineering College, Uttar Pradesh, India,(IRJET).

[2] Peter Menegay, Jason Salyers,Griffin College,"Secure Communications Using Blockchain Technology",Milcom 2018 Track 3 - Cyber Security and Trusted Computing.

[3] Suman Kumar Das, Zenlabs, Zensar Technologies, Pune, India,"Secure Messaging Platform Using Blockchain Technology", (IJRES), ISSN (Online): 2320-9364, ISSN (Print): 2320-9356.

[4] Abhishek P. Takale,Chaitanya V. Vaidya,Suresh S. Kolekar,"Decentralized Chat Application using Blockchain Technology",Rajendra Mane College Of Engineering And Technology, Ambav, India.(IJREAM).

[5] Muhammed Kuliya, Hassan Abubakar,"Secured Chatting System Using Cryptography", Lovely Professional University, Punjab India,(