Zever

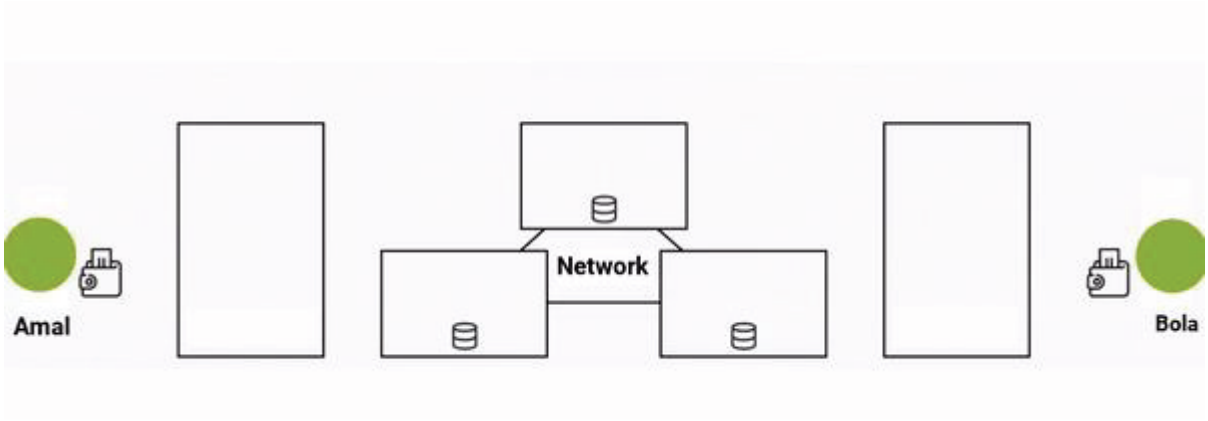
Zever
Network & Governing Council

1. Introduction

Zever chat application is built on a blockchain network, providing users with secure and private communication through end-to-end encryption. The application also functions as a view-only wallet, allowing users to manage their blockchain assets securely. This document outlines the technical architecture of the Zever chat application, detailing its key components and their interactions.

2. System Overview

At the core of the Zever application is a robust system of end-to-end encryption that ensures message privacy. Messages are encrypted on the sender's device and can only be decrypted by the intended recipient. This ensures that messages remain indecipherable to intermediaries, including the app's developers and service providers. The application uses the XMTP (Extensible Message Transport Protocol) to facilitate secure communication and key management. Wallet connectors are essential components that enable users to connect their cryptocurrency wallets to the application, allowing for secure authentication and interaction with the blockchain.



3. Key Components

1. **Network Client**
    - Facilitates the retrieval and sending of encrypted messages between network participants.
    - Utilizes blockchain technology to ensure that only the sender and recipient can read the messages.
    - Users log in using WalletConnect, signing in without involving private keys, thus maintaining security and privacy.
2. **Key Generation and Management**

© 2024–2025 Zever.

- TheApp supports key generation, allowing the establishment of secure relationships between users' blockchain accounts.
- Keys are used to encrypt and decrypt messages, ensuring secure communication.
- The client requests a wallet signature to sign the newly generated key bundle (only the first time) and to sign a random salt used to encrypt the key bundle in storage (every time the client is started).

3. **View-Only Wallet**
    - Provides an overview of the user's blockchain assets.
    - Enables users to send and receive payments, with each transaction requiring wallet approval.
    - Ensures no direct access to users' wallets, maintaining the highest level of security and privacy.
4. **Allow and Block Chat Feature**
    - Manages user consent preferences for communication with other blockchain accounts.
    - Consent preferences can be set to Unknown, Allowed, or Denied, ensuring user control over who can send messages.
5. **Identity Resolution**
    - Resolves identities using third-party services such as Ethereum Name Service (ENS) and Unstoppable Domains (UNS).
    - Supports both raw 0x wallet addresses and human-readable domain names associated with wallet addresses.

4. Technical Workflow

4.1 User Login

- **Login Mechanism**: Users log in via WalletConnect, signing in with their blockchain account.
- **Security**: No private keys are used during the login process; only public information is accessed, ensuring that the app does not gain any ability to execute actions on behalf of the user.

4.2 Message Encryption and Decryption

- **Encryption**: Messages are encrypted on the sender's device using keys generated by the App and XMTP network.

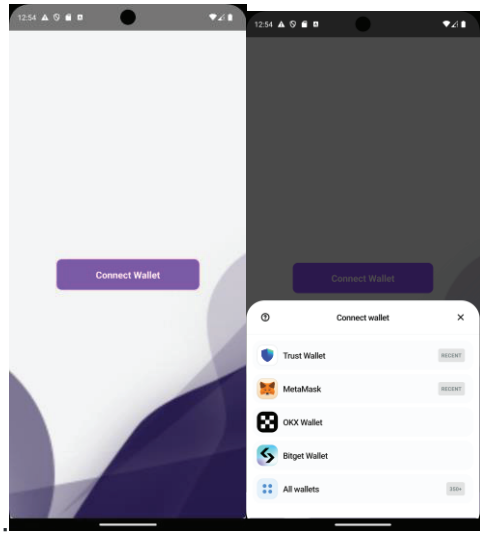
© 2024–2025 Zever.

- **Decryption**: Only the recipient can decrypt the messages using their corresponding keys.
- **Key Management**:
    - Upon login, the client checks the network for keys associated with the user's account.
    - If keys exist, the client retrieves them following the key retrieval flow.
    - If no keys are found, the client generates new keys and signs them with the user's wallet.

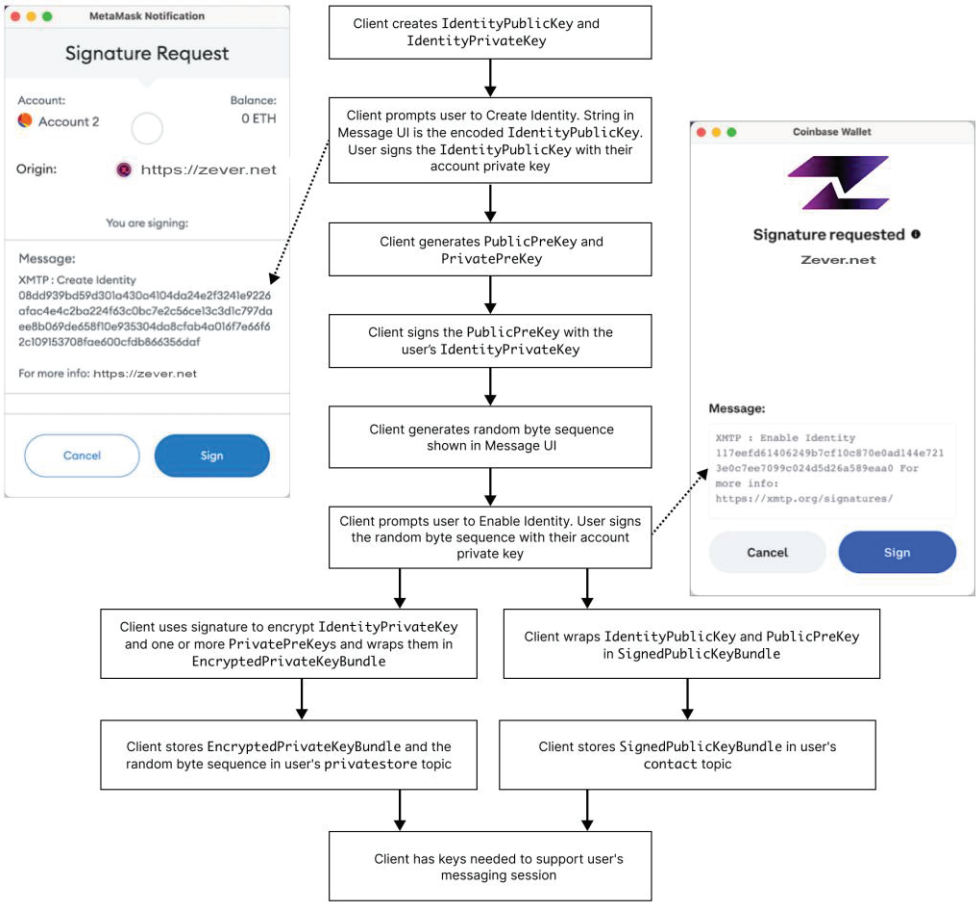
4.3 Key Generation Flow

6. **User Logs In**: The user logs into the app using WalletConnect.
7. **Key Generation**: The APP generates keys that are used to establish secure communication channels.
8. **Key Association**: These keys are associated with the user's blockchain account, ensuring that only the intended parties can access the messages.
9. **Key Signing**: The client requests the wallet to sign the newly generated key bundle (first time) and a random salt to encrypt the key bundle in storage (every

© 2024–2025 Zever.



time the client starts).

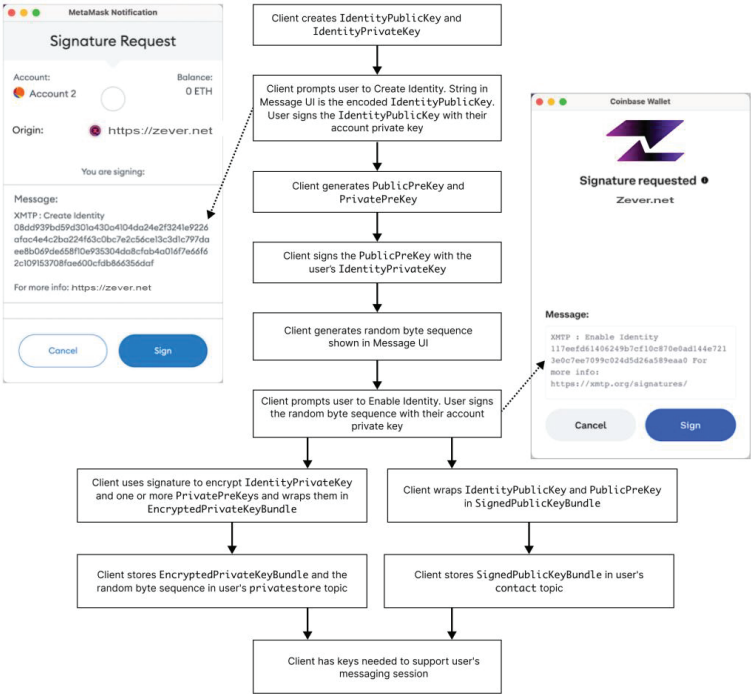


4.4 Key Retrieval Flow

10. **Key Check**: Upon user login, the client checks the network for existing keys associated with the user's account.

© 2024–2025 Zever.

11. **Key Retrieval**: If keys are found, the client retrieves them according to the defined key retrieval flow.
12. **Secure Communication**: Retrieved keys are used to decrypt incoming messages and encrypt outgoing messages.



4.5 Messaging Network Check

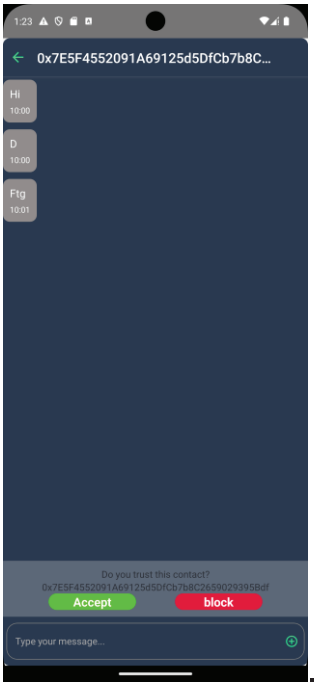
- **Network Check**: Before messaging, the app checks if the recipient's address is on the app network.
- **Invitation**: If the address is not found, the app sends an invitation to the recipient to join the network and enable communication.

4.6 Allow and Block Chat

13. **Consent Management**: The app requests and respects user consent preferences for communication.
14. **Consent Values**:
    - Unknown: Default state before user consent is set.
    - Allowed: User consents to receive messages.

© 2024–2025 Zever.

- Denied: User blocks messages from specific addresses

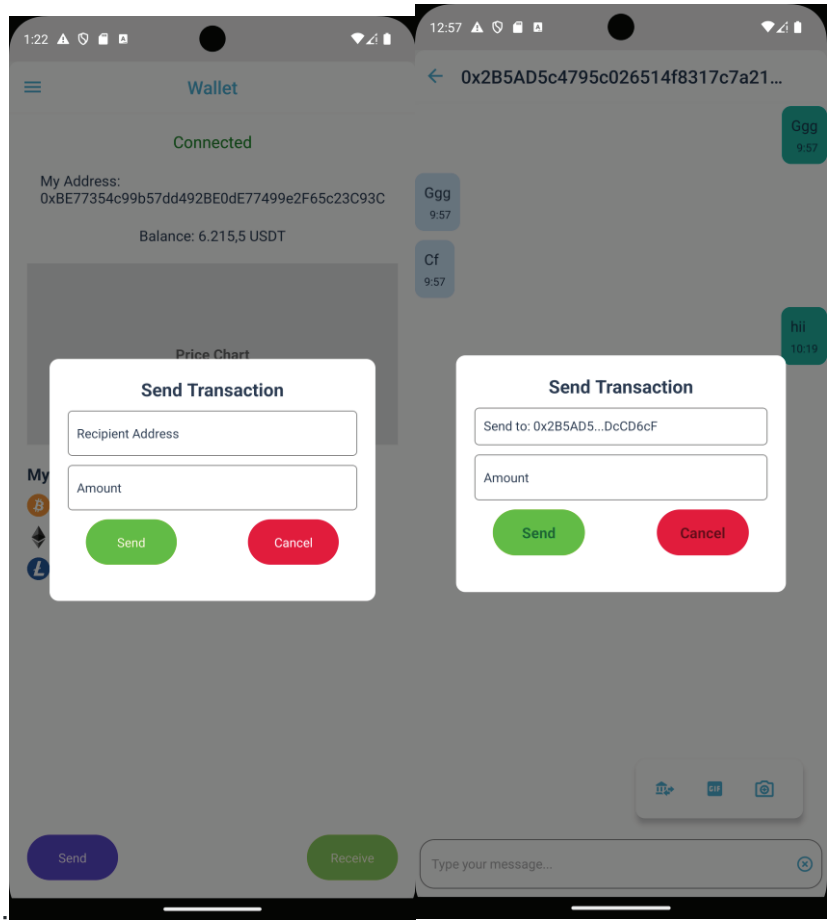


4.7 Token Transfer in Chat

- **Token Transfer**: Users can transfer tokens within the chat interface.

© 2024–2025 Zever.

- **Wallet Confirmation**: Transfers redirect users to their wallet for transaction confirmation or decline.



4.8 Identity Resolution

15. **Login and Identity Resolution**: Upon login, the app resolves the user's identity using popular services.
16. **Supported Identities**: The app supports resolving raw 0x addresses and human-readable domain names from ENS and UNS.

5. Security and Privacy Considerations

- **End-to-End Encryption**: Ensures that messages can only be read by the sender and the recipient.
- **No Local Databases**: Eliminates the risk of sensitive data breaches or unauthorized access.
- **Wallet Integration**: Uses WalletConnect for secure user authentication without exposing private keys.

© 2024–2025 Zever.

- **Transaction Approval**: Users approve each transaction within the app, ensuring control over their financial activities.
- **User Consent Management**: Allows users to control who can send messages, enhancing privacy.

6. Conclusion

The Zever chat application leverages blockchain technology and the difrent protocols to provide a secure, private, and user-friendly communication platform. By focusing on end-to-end encryption, robust key management, and comprehensive user consent features, the application ensures that users' messages and financial transactions remain secure and private.

© 2024–2025 Zever.