Architecture Documentation

ARCHITECTURAL STYLE(S) USED

- how does the architecture support various features of your application

1. Client-Server Architecture: The devices communicate with a central server that manages data and authentication services
   - Real-Time Data Exchange: Client-server allows for seamless real-time communication between homeowner and service personnel's mobile devices and the server
   - Scalability: Client-server supports adding new features or increasing the number of users without worsening performance
2. Microservices Architecture: The app uses different functions (location tracking, identity verification, user management) and they are handled by separate, loosely coupled services
   - Modular Development: Each feature is built as an independent service so updating each feature is easy without affecting the entire system
   - Platform Independence: Microservices can be deployed independently on both iOS and Android
3. Event-Driven Architecture: The app uses event-driven architecture to respond to triggers (location updates, verification requests) in real-time
   - Real-Time Tracking: Service personnel's location is constantly updated and streamed to homeowner through event-driven mechanisms
   - Immediate Notifications: When the app detects significant events (ex. the service personnel arriving at location), notifications are pushed to homeowner and supervisor

RATIONALE FOR YOUR ARCHITECTURAL STYLE AND MODEL

Architectural Styles

The centralization of the client-server architecture is essential for maintaining data consistency, integrity, and quick access across various user devices. This ensures that all data, including real-time location updates and authentication, is centrally managed and synchronized. The responsiveness of the event-driven architecture is crucial where real-time data is vital for accurate tracking and instant notifications which ensures that the app can handle real-time updates and immediate responses to specific triggers like location changes or verification requests. The modular approach of the microservices architecture ensures that the system remains flexible and can easily adapt to new requirements or scale to meet increasing demand for the system. Combining these architectural styles creates a balanced system that excels in unified data management, real-time responsiveness, event handling, modularity, and scalability. By leveraging the strengths of each architecture, the app can deliver a robust and flexible solution for real-time location verification. Essentially, these architectures work synergistically to create an app that is not only efficient and reliable but also adaptable and resilient. This integrated approach ensures that every aspect of the app functions smoothly and effectively, providing users with a seamless experience.

Architectural Model

The ControlsUI handles user commands and interactions, while GoogleMapUI integrates Google Maps for real-time location visualization. This ensures users can easily view and interact with location data, providing an intuitive user experience. The PermissionService enables the app to have the necessary permissions to access location data, complying with privacy and security regulations. The GeolocatorService provides the actual location data which is fundamental for the app's primary function of tracking and verifying locations in real-time. The MapController handles the maps functionality including receiving and processing location data while the PathRenderer visualizes the service personnel's path on the map. This ensures that users receive accurate and up-to-date location tracking which enhances the app’s reliability and effectiveness. The LocationHistoryManager maintains a record of past locations, allowing the app to offer insights and verification of historical data which is crucial for both transparency and security. Collectively, these elements of the model help create a cutting-edge app tailored to meet the demands of real-time tracking and verification.