**High Level System Design - ProtecTalk**

1. **Introduction**

ProtecTalk is a client-server application designed to enhance user security by analysing incoming calls for potential scams. It leverages AI-driven NLP to detect suspicious patterns in audio captured by the mobile OS, alerting users and their protectors in real time via Firebase Cloud Messaging (FCM). Key goals include real-time monitoring, robust speech recognition, and a secure, scalable architecture.

1. **System Architecture**

ProtecTalk employs a client-server model where an Android mobile app built with Flutter and Dart handles user interactions, audio capture, and notifications. A cloud-hosted backend—using Node.js for API services and Flask for AI processing — analyses call data with AI/ML tools to detect scams. User data, call logs, and alerts are managed via MongoDB (Atlas) and Firebase Authorization, while FCM delivers instant notifications.

1. **Main Components**

**Frontend (Mobile Application)**

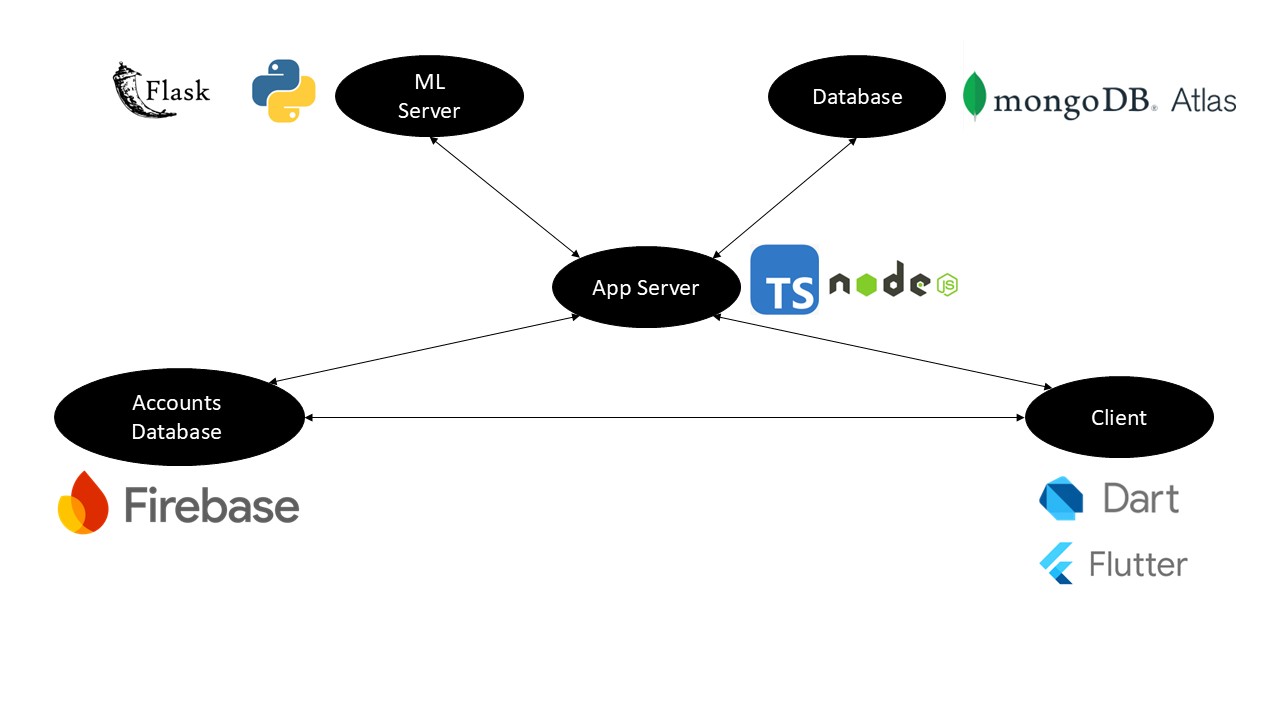
* **Operating System:** Android
* **Programming Language:** Dart
* **UI Framework:** Flutter
* **Speech-to-Text:** Android built-in SpeechRecognizer API / Google Cloud Speech-to-Text
* **Real-time Communication:** Firebase Cloud Messaging (FCM) for instant alerts
* **Custom Communication with Backend:** Direct interaction with the server for additional security and controlled message flow

**Backend (Server)**

* **Programming Languages:** TypeScript (Node.js) and Python (Flask for AI)
* **Frameworks:** Node.js for core API services, Flask for AI-powered processing
* **AI/ML:**
  + **NLP:** spaCy, NLTK, Transformers
  + **Scam Detection:** Custom model / Google Cloud AI / FastText
* **Database:**
  + **Firebase Authorization:** User authentication and session management
  + **MongoDB (Atlas):** Storage for user profiles, call metadata, call logs, and alerts history

**4. Data Flow & Communication**

1. **Call Detection & Notification:** The app detects an incoming call and begins monitoring for suspicious patterns.
2. **Local Transcription & Scam Analysis:** The conversation is transcribed and analysed in real-time using on-device AI.
3. **Alert Generation & Logging:** If a scam is detected, an alert is generated locally and sent to the server for record-keeping (if user permits).
4. **Server Processing & Notifications:** The backend stores metadata and triggers alerts to users and protectors via **FCM**.

**Architecture Diagram**