**High Level System Design – ProtecTalk**

1. **Introduction**

ProtecTalk is a client-server application designed to enhance user security by analyzing 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 Overview**

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—analyzes 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. **Software Modules**
   1. **Frontend Modules (Mobile Application)**
      1. **User Interface Module**

**Purpose:** Managing user interface and interactions

**Components:**

* + - Intro Screen
    - Login/Registration screen via authentication
    - Home screen as a settings screen
    - Manage protection screen
    - Protection addition/remove screens

**Technology:** Flutter, Dart

* + 1. **Audio Capture Module**

**Purpose:** Capturing voice calls and handling system permissions

**Components:**

* Permission manager for requesting OS permissions
* Call state detector for unknown numbers
* Audio recorder

**Technology:** Android SpeechRecognizer API

* + 1. **Local Processing Module**

**Purpose:** Preliminary on-device analysis of audio

**Components:**

* Speech-to-text converter
* Scam pattern detector
* Local alert generator

**Technology:** Google Cloud Speech-to-Text API, on-device ML model

* + 1. **Notification Module**

**Purpose:** Handling incoming and outgoing notifications

**Components:**

* FCM integration for sending alerts to remote users
* Local notification manager for displaying real-time alerts to the user

**Technology:** Firebase Cloud Messaging

* + 1. **API Communication Module**

**Purpose:** Secure communication with backend services

**Components:**

* Authentication handler for login\registration
* Secure data transmission – Scam calls transcription, user data, protection requests

**Technology:** Dio/HTTP client, JWT tokens

* 1. **Backend Modules (Server)**
     1. **API Gateway Module**

**Purpose:** Managing external API requests and routing

**Components:**

* Authentication middleware
* Request validation

**Technology:** Express.js, TypeScript

* + 1. **User Management Module**

**Purpose:** Handling user accounts and relationships

**Components:**

* User registration
* Authentication
* Profile management
* Protector relationship management

**Technology:** Node.js, Firebase Authentication

* + 1. **AI Processing Module**

**Purpose:** Advanced analysis of call transcripts

**Components:**

* NLP processor
* Scam pattern detector
* Confidence score calculator
* Model trainer

**Technology:** Flask, Python, spaCy, NLTK, Transformers

* + 1. **Alert Management Module**

**Purpose:** Processing and distributing alerts

**Components:**

* Alert generator
* Notification dispatcher
* Alert history manager

**Technology:** Node.js, FCM Admin SDK

* + 1. **Data Access Module**

**Purpose:** Managing database operations

**Components:**

* Database connection pool
* Query builder
* Data models
* Caching layer

**Technology:** Mongoose ODM, MongoDB drivers

* + 1. **Monitoring and Logging Module**

**Purpose:** System health monitoring and error tracking

**Components:**

* Application logs
* Error reporting
* Audit trails

**Technology:** Winston, Prometheus, ELK Stack

* + 1. **Fraud Detection & Knowledge Base Service Module**

**Purpose:** Maintains and updates a live database of known scam patterns, keywords, and fraud behavior.

**Components:**

* Fraud Database Updater – Gathers new scam data from external sources and internal detections from user reports.
* Pattern Recognition Engine – Stores and refines AI-detected fraud patterns.
* Keyword List Management – Maintains a dynamic list of scam-related keywords.

**Technology:** Python (Flask), MongoDB, External APIs (for scam updates).

1. **Database Design**
   1. **User Data Database (MongoDB)**

**Purpose:** Stores user-related information, including profiles, relationships, devices, and activity logs.

**Key Collections:**

* Users – Stores user details and authentication data.
* Protectors – Manages user-protector relationships and notifications.
* Devices – Tracks user devices and messaging tokens.
* Call Logs – Records call history with timestamps and alert statuses.
* Call Transcriptions – Transcriptions of potential scam calls approved and sent by the user.
* Alerts – Stores scam alerts, confidence scores, and acknowledgment statuses.
  1. **Scam Pattern Database (MongoDB)**

**Purpose:** Maintains scam detection patterns and AI models for fraud analysis.

**Key Collections:**

* Scam Patterns – Contains known fraud patterns and severity levels.
* Fraud Scripts – Stores scam script samples and common phrases.
* Keywords – Lists important words and contextual weights for detection.
* ML Models – Manages trained machine learning models and performance metrics.

1. **Data Flow & Communication**

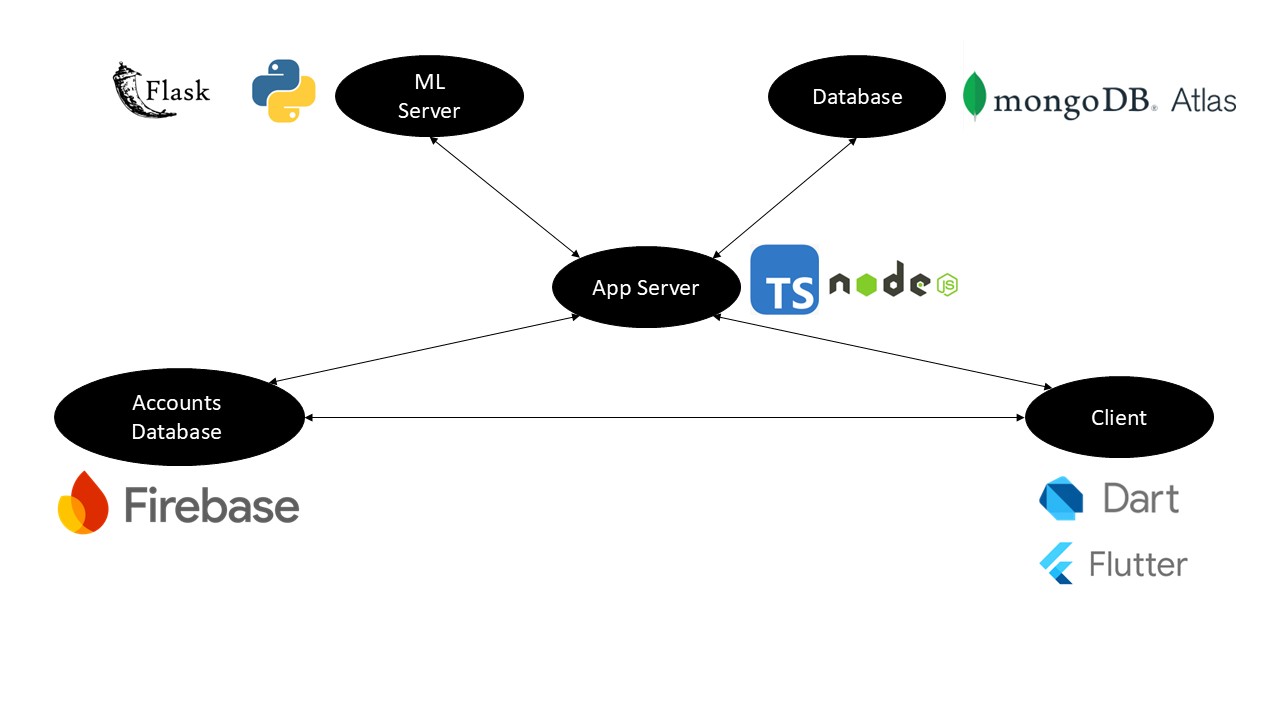
* **Call Detection & Notification:** The app detects an incoming call and begins monitoring for suspicious patterns.
* **Local Transcription & Scam Analysis:** The conversation is transcribed and analyzed in real-time using on-device AI.
* **Alert Generation & Logging:** If a scam is detected, an alert is generated locally and sent to the server for record-keeping (if user permits).
* **Server Processing & Notifications**: The backend stores metadata and triggers alerts to users and protectors via FCM.
* **Scam Intelligence Updates** – User reports and external threat feeds continuously refine scam detection models and patterns.

1. **Integration Points**

* Firebase Authentication: User authentication and authorization
* Firebase Cloud Messaging: Real-time notifications
* Google Cloud Speech-to-Text: Speech recognition and transcription
* MongoDB Atlas: Cloud-hosted database service
* External AI Services: Optional integration with third-party AI services for enhanced detection

1. **Security Considerations**

* End-to-end encryption for all communications
* Secure storage of sensitive user data
* Regular security audits and penetration testing
* Compliance with relevant data protection regulations
* User consent management for audio recording and processing

**Architecture Diagram**