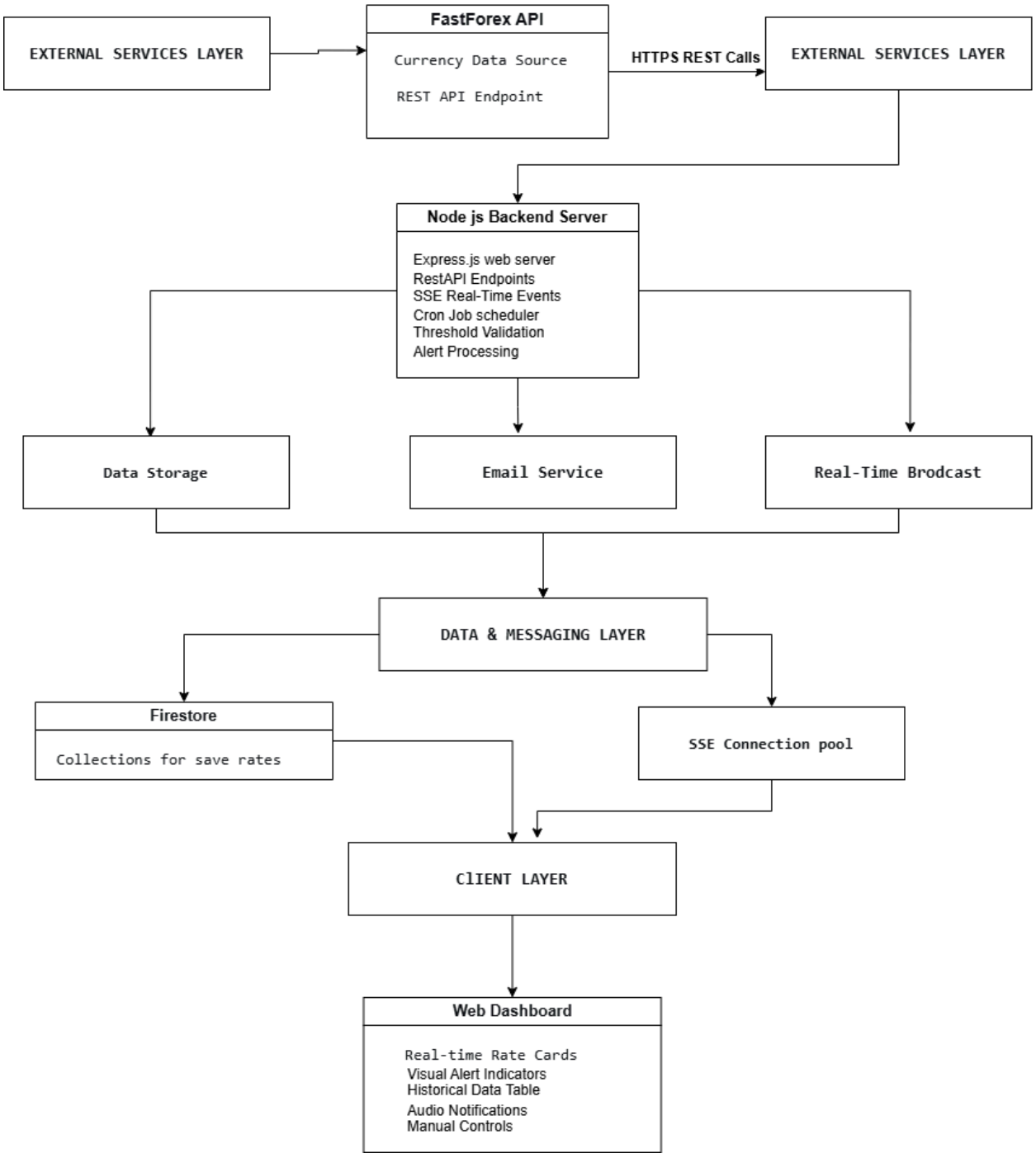
Currency Exchange Tracker

Solution Architecture Diagram



CLOUD COMPONENTS

1. EXTERNAL SERVICES

- **FastForex API**
  - URL: https://api.fastforex.io/convert
  - Purpose: Real-time currency exchange rates
  - Data: LKR to USD/EUR/GBP conversion

2. BACKEND SERVER (Node.js/Express)

- **Server Components:**
  - Web Server (Port 3000)
  - REST API Endpoints
  - SSE (Server-Sent Events) Server
  - Cron Scheduler
- **Services:**
  - currencyService.js - Rate fetching & storage
  - alertService.js - Threshold monitoring
  - emailService.js - Notifications
  - firebase.js - Database connection

3. DATABASE (Firebase Firestore)

- **Collections:**
  - exchangeRates - Current and historical rates
  - alerts - Threshold breach records
- **Data Structure:**
  - Document per rate entry with timestamp
  - Automatic 7-record limit

4. CLIENT (Web Browser)

- Single Page Application (index.html)
- Real-time UI updates via SSE
- Audio and visual alert system

COMPLETE DATA FLOW

1. DAILY RATE FETCH

Cron Job triggers (Everyday at 9 AM)
↓
Call FastForex API
↓
Process exchange rates
↓
Check against thresholds
↓
Store in Firestore
↓
Broadcast to all clients
↓
Update web dashboard

2. CLIENT STARTUP

User opens website
↓
Establish SSE connection
↓
Load current rates
↓
Display dashboard

3. MANUAL FETCH

User clicks fetch button
↓
Call backend API
↓
Fetch latest rates
↓
Update all clients

4. ALERT PROCESSING

Rate exceeds threshold
↓
Create alert record
↓
Store in database
↓
Notify all clients
↓
Play sound + show visual

5. HISTORY VIEW

User clicks history
↓
Get 7 latest records
↓
Display in table
↓
Highlight alerts

KEY FEATURES

- Real-time rate monitoring
- Automatic daily updates
- Visual threshold alerts
- Audio notifications
- 7-day history tracking
- Manual refresh option
- Live connection status

TECHNOLOGY STACK

- Backend: Node.js, Express, Firebase, google smpt
- Frontend: HTML, CSS, JavaScript
- Real-time: Server-Sent Events
- Scheduling: node-cron
- API: FastForex for rates