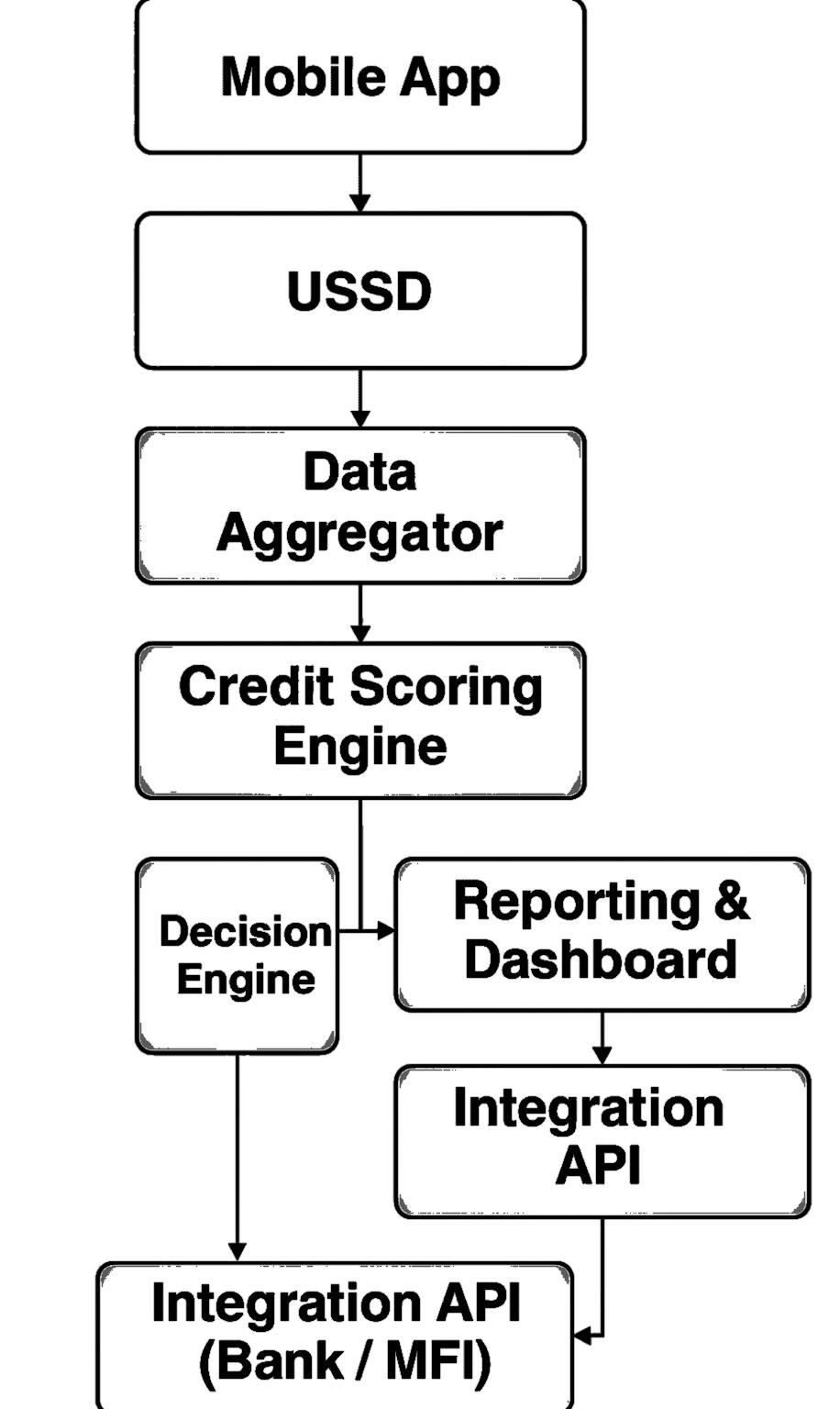
Technology Architecture and Design Document

# Architecture Document

**1.Architecture Diagram Overview**

**2. Flow of Information Between Blocks**

**1. Mobile App:** A smartphone-based interface for users to input data. Collects behavioral, transactional, and personal information. May allow photo capture of documents or survey responses.

### **2. USSD:** For users without smartphones or internet access.Offers a simple text-based menu to collect basic data.

### **3. Data Aggregator:** Centralizes and normalizes data from multiple channels (app/USSD).Cleans, filters, and formats incoming data. May combine with other sources like telco records, utility bills, or agent inputs.

### **4. Credit Scoring Engine:** Core intelligence that calculates the credit score. Uses machine learning or rule-based algorithms. Analyses behaviour, income patterns, repayment history, alternative data.

### **5. Decision Engine:** Makes automated credit decisions based on score and rules. Accepts/rejects applications. May suggest loan amount and repayment period.

### **6. Reporting & Dashboard:** Visual interface for analytics and monitoring. Real-time metrics (application volume, approval rate, etc.). Flags anomalies or fraud patterns. Used by administrators or financial institutions

### **7. Integration API:** Acts as a communication bridge between system and banks/MFIs.Sends credit decision, applicant data, and scores to financial partners. Enables programmatic loan disbursement or account creation.

**8. Integration API (Bank / MFI):** Final interface with the financial institution.Enables banks to access results and act on approvals.May allow updates back to the platform.

**Design Document**

**1.Technology Design**

* **Mobile App**: React Native (for Android/iOS)
* **USSD Gateway**: Africa's Talking / Twilio / Custom Telco Integration
* **Data Aggregator**: Node.js backend with Express.js
* **Credit Scoring Engine**: Python with Scikit-learn or Tensor Flow Lite
* **Decision Engine**: Node.js with custom rule engine
* **Reporting & Dashboard**: React.js + Chart.js / Grafana
* **Integration API**: REST API with JWT Author (Node.js or FastAPI)
* **Database**: PostgreSQL or MongoDB (depending on structure)

**2.Screen Design**

* User Registration
* Data input or survey
* Credit score result
* Admin dashboard
* Bank or MFI loan decision panel

**3.Data Base Design**

* User (Name, User Id, Mobile number, Gender, Address)
* Behaviour Data(Occupation, Income, Dependents)
* Credit score(User reference, Score, Risk Level)
* Decision(Decision Status, Remarks, Time of Decision)
* Integration log(Bank, Response Status, Interaction Time)

**4.Class Design**

1. User
2. Behavioural Data
3. Credit score
4. Decision
5. API Integration Log