AI for Bharat Hackathon

Powered by aws

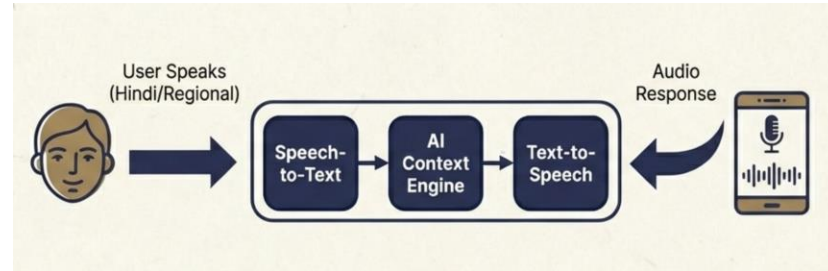**Team Name : ScriptDevs**

Team Leader Name : Aman Jaiswal

Team Members: Rohit Rathod , Bhavik Prajapati

User Speaks (Hindi/Regional) → Speech-to-Text → AI Context Engine → Text-to-Speech → Audio Response

Problem Statement : AI for Rural Innovation & Sustainable Systems

Brief about the Idea:

The Concept A mobile-first, voice-enabled super-app designed to bridge the gap in information, markets, and services for rural communities. The solution is specifically engineered for users with limited literacy and areas with low-bandwidth (2G/3G) connectivity.

Core Technology

- **Voice-First Interface:** Utilizes Speech-to-Text and Text-to-Speech to allow users to interact in Hindi, English, and **regional Indian languages without typing**.

- **AI Engine:** Centralized intelligence layer that maintains **cross-conversation context**, performs **intent detection**, and delivers highly targeted, **adaptive recommendations in real time.**

- **Memory:** Secure, persistent **long-term memory** per user that captures preferences, **behaviours**, and historical context to continuously refine personalization and **decision-making** over time.

- **Trust & Privacy:** Integrates **Digilocker for verifying** peer experts and a "Data Controller" to give farmers granular control over their data sharing.

The 5 Service Pillars

- **Agriculture:** Connects farmers directly to buyers (**supply chain**), offers AI-based pest/disease detection (precision farming), and provides **market price intelligence**.

- **Knowledge:** Facilitates peer-to-peer learning via **Twitter Space-like** groups and offers voice-based access to government schemes and skills training.

- **Economics:** Enables access to government loans, facilitates insurance claims, and uses AI to "nudge" farmers toward **smart savings during harvest**.

- **Health:** Provides **AI symptom pre-screening,** risk profiling, and connects users to telemedicine **portals like eSanjeevani.**

- **Infrastructure:** Offers actionable steps for accessing government **policies**, lodging complaints, and **promoting local businesses**.

Your solution should be able to explain the following:

1. How is it Different? (Differentiation)

• **Voice-First vs. Text-First:** Unlike **standard apps requiring literacy**, this **uses a Voice Interface** supporting Hindi, English, and **regional Indian languages**, enabling users to interact naturally via speech-to-text processing.

• **Holistic vs. Siloed:** Instead of focusing solely on agriculture, it **integrates 5 critical pillars**: Agriculture, Knowledge, Economics, Health, and Infrastructure into one unified super-app.

• **Verified Social Layer:** Unlike generic forums, the peer-learning feature ("**Twitter Space-like**") incorporates a Trust Layer using Digilocker to verify experts and prevent misinformation.

2. How will it Solve the Problem? (Mechanism)

• **Bridging Literacy Gaps:** The AI Engine and Language Transformer manage conversation **context in local supported languages**, allowing users to access complex services (loans, market trends) without reading or typing.

• **Economic Empowerment:** It solves **market access issues by connecting farmers directly to buyers** (supply chain) and using AI to "nudge" financial savings based on harvest patterns.

• **Health & Risk Access:** It **overcomes physical distance to healthcare** by offering AI pre-screening of symptoms and connecting users to telemedicine portals .

3. USP (Unique Selling Proposition)

• **Voice-First App:** The only platform combining livelihood, health, and finance accessible entirely through local language voice prompts.

• **Resilient Connectivity:** Performance on **2G/3G networks** with offline data synchronization.

• **Verified Community Knowledge:** A unique Peer-to-Peer Learning model that clusters users by goals and verifies advice sources, creating a safe environment for skill sharing.

List of features offered by the solution

Core Interface & Technical Features
- Voice-Based Interface
- Low Bandwidth Optimization
- Offline Functionality

Agriculture & Supply Chain
- Precision-based AI Farming
- Direct Market Access / Intelligence
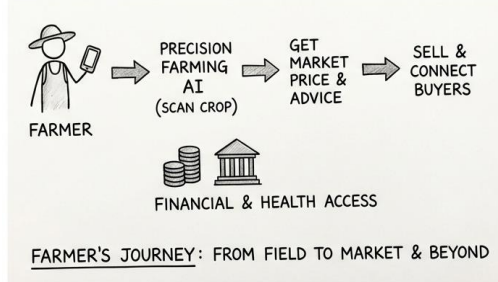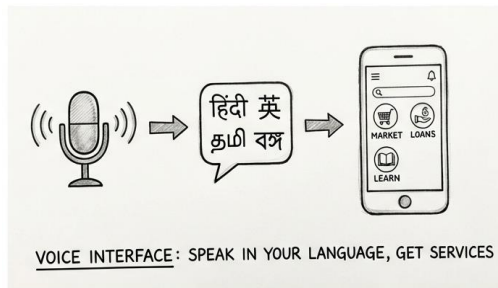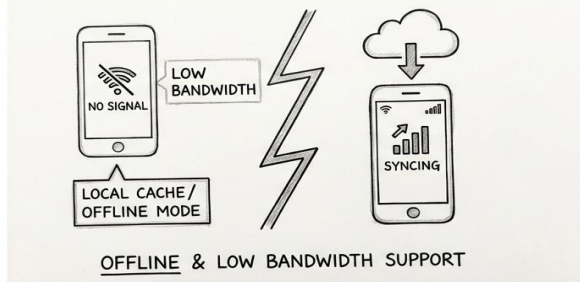- Logistic Coordination
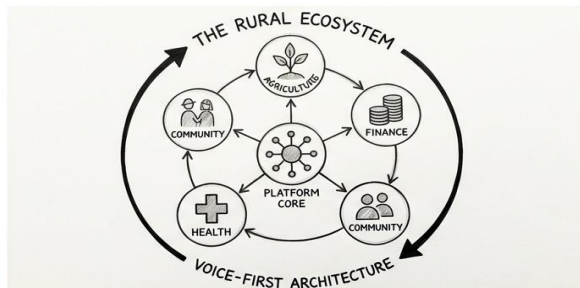
Knowledge & Community
- Peer-to-Peer Learning
- Trust & Verification
- Community Spaces
- Skill Training

Economics & Finance
- Loan Eligibility AI
- Saving Nudges
- Insurance Facilitation Assist

Health & Infrastructure
- Health Pre-screening
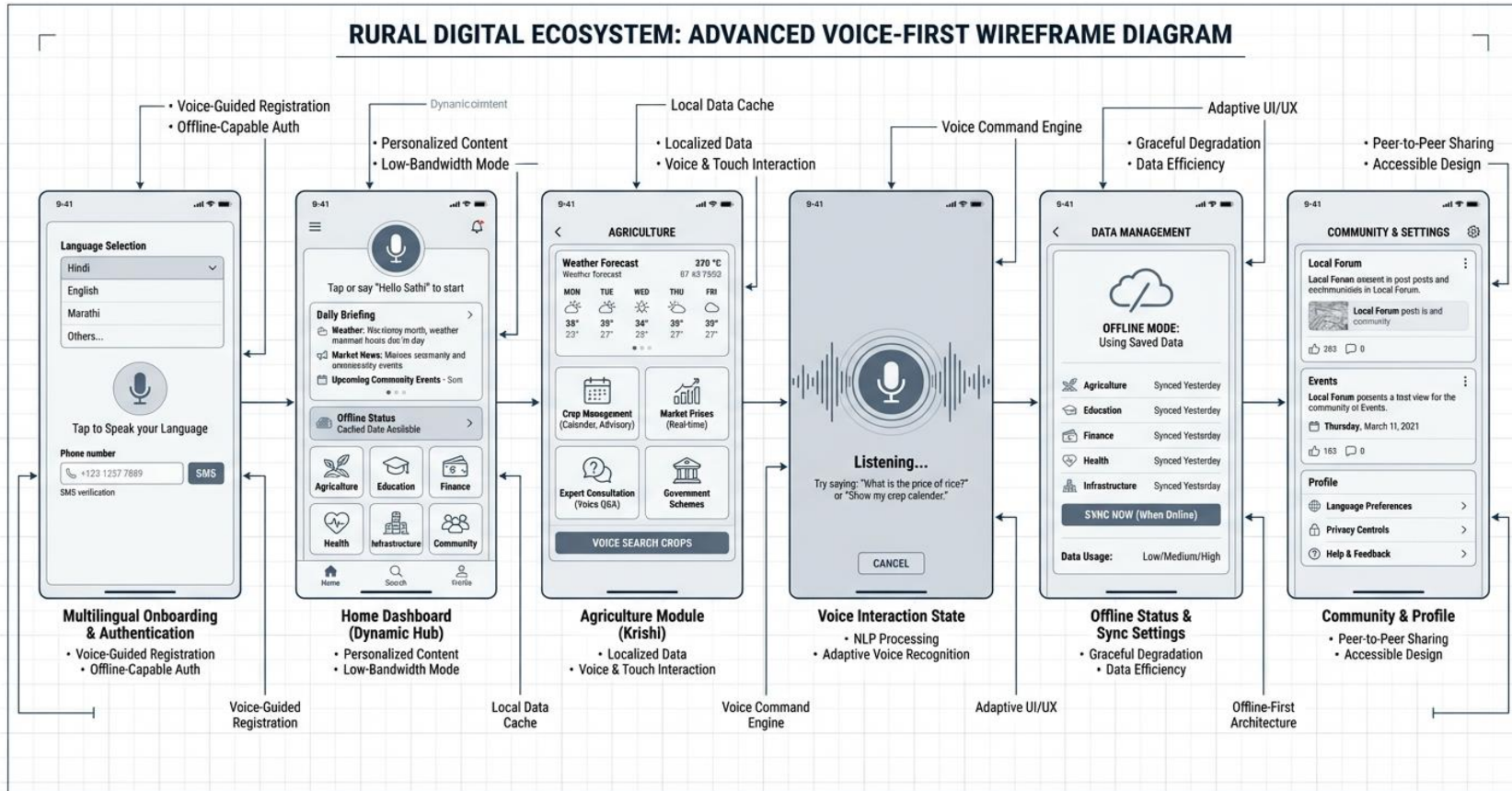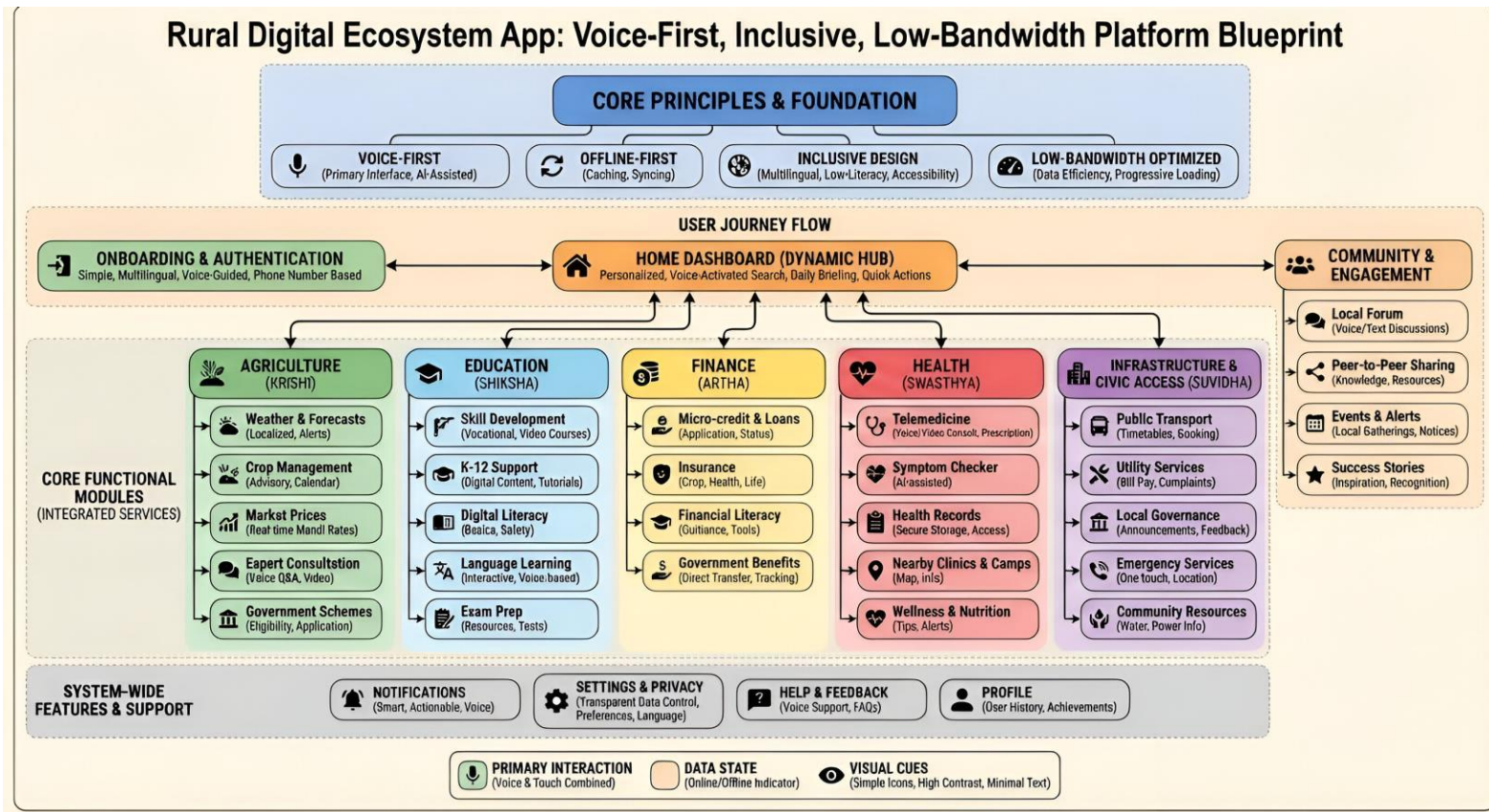- Teleconsultation Integration
- Governance Portal

THE RURAL ECOSYSTEM — VOICE-FIRST ARCHITECTURE

VOICE INTERFACE: SPEAK IN YOUR LANGUAGE, GET SERVICES

OFFLINE & LOW BANDWIDTH SUPPORT

FARMER'S JOURNEY: FROM FIELD TO MARKET & BEYOND

Process flow diagram or Use-case diagram
HIGH LEVEL SYSTEM FLOW

END USER (L1)
Mobile App (Voice-First Interface)
Low Bandwidth Optimized

↓

AI LAYER (L2)

| Speech-to-Text Engine | → | Context Management | → | Intent Recognition |

↓

LOCAL LANGUAGE TRANSFORMATION (L3)
Multi-lingual NLP | Regional Language Support

↓

SERVICE CATEGORIES

| Agriculture | | Knowledge | | Economics | | Health |

| Infra |

USER SPEAKS: "मेरी फसल में कीड़े लग गए हैं"

↓

AUDIO PROCESSING
• Noise reduction • Low bandwidth optimization

↓

SPEECH-TO-TEXT
• Regional language • Dialect handling

↓

INTENT RECOGNITION
Pest problem → Agriculture → Crop Precision

↓

CONTEXT MANAGEMENT
Profile • History • Session context

↓

SERVICE ROUTING → Pest Detection

↓

AI PROCESSING
Image request • Weather/Soil data • Patterns

↓

RESPONSE GENERATION
Aphids identified • Treatment • Prevention

↓

TEXT-TO-SPEECH: "यह एफिड कीट है। इसके लिए.."

↓

AUDIO PLAYBACK TO USER

Wireframes/Mock diagrams of the proposed solution (optional)

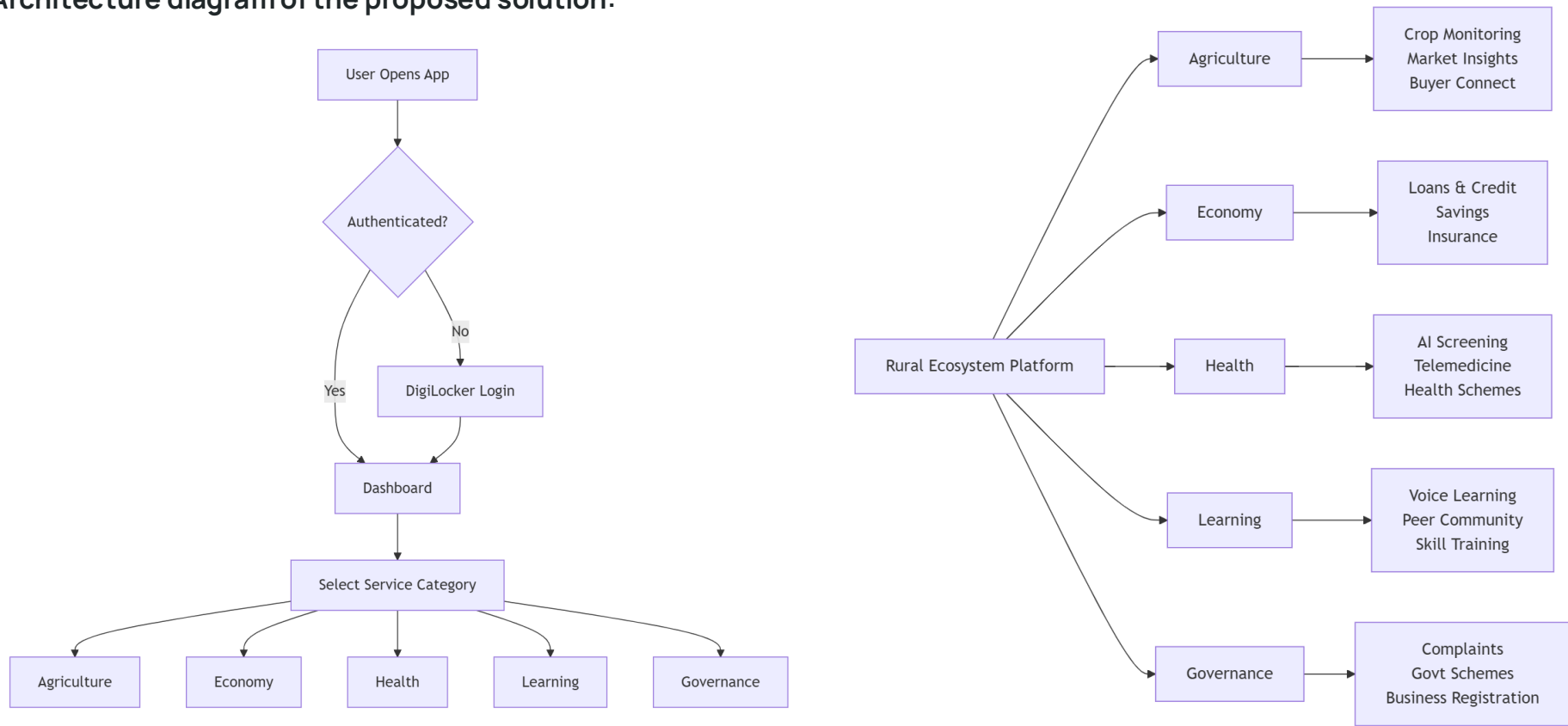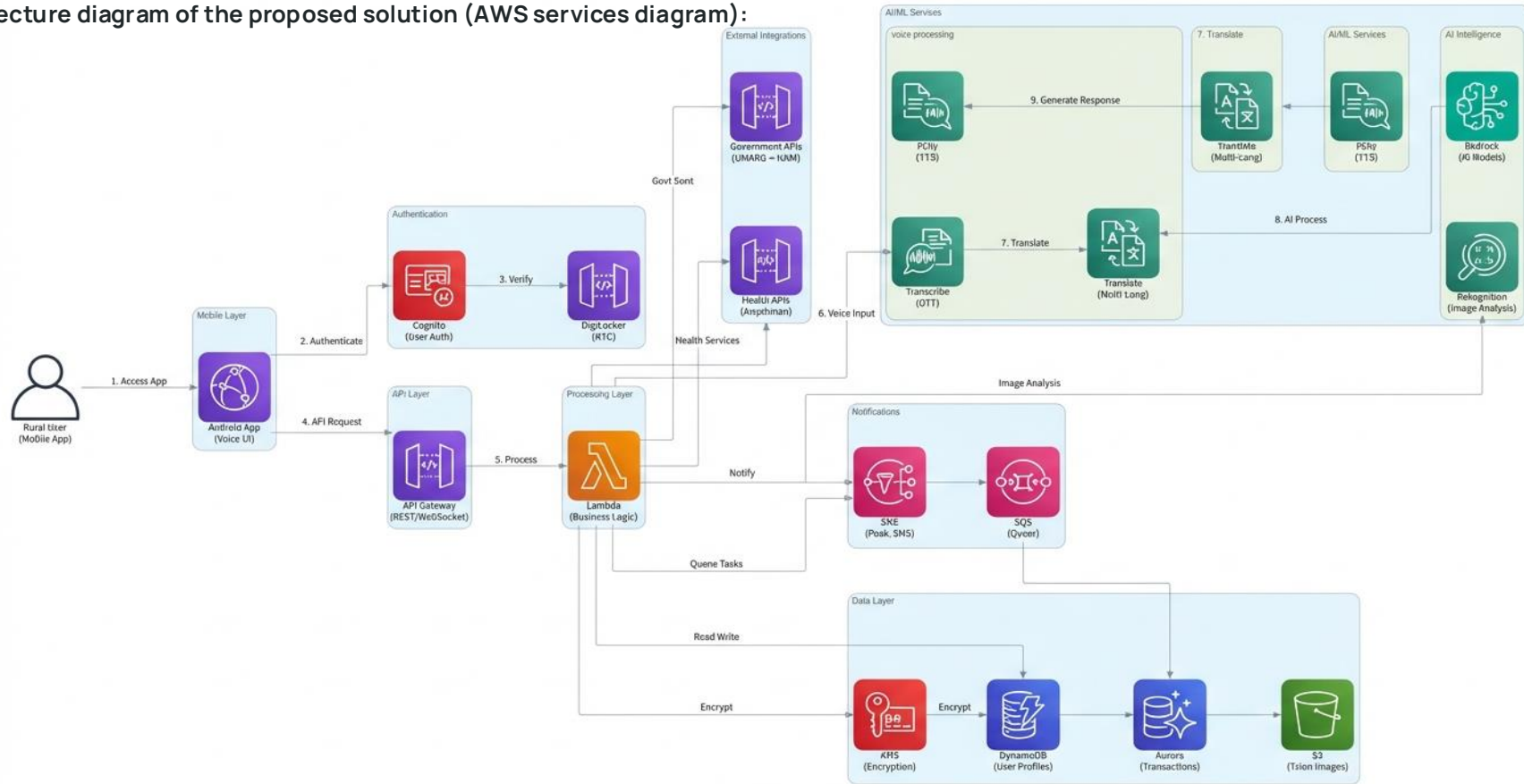**RURAL DIGITAL ECOSYSTEM: ADVANCED VOICE-FIRST WIREFRAME DIAGRAM**

Wireframes/Mock diagrams of the proposed solution (optional)

Architecture diagram of the proposed solution:

Architecture diagram of the proposed solution (AWS services diagram):

Rural Ecosystem Platform - AWS Architecture

Technologies to be used in the solution:

1. Cloud & Infrastructure
- **Cloud Provider (AWS / Azure / GCP compatible)** :- Core hosting environment
- **Docker :-** Containerization of Flask & Node.js services
- **API Gateway (Managed or Open Source)** :- Unified REST & WebSocket entry point
- **CDN (e.g., CloudFront or Cloud CDN)** :- Low-latency content delivery
- **Infrastructure as Code (Terraform / CDK)** :- Reproducible deployments
- **VPC & Load Balancer** :- Secure networking and traffic management

2. Data Layer
- **PostgreSQL (Aurora-compatible)** :- Transactional data (users, schemes, loans)
- **NoSQL Database (DynamoDB / MongoDB)** :- Session & conversation context
- **Object Storage (S3-compatible)** :- Voice recordings, images, documents
- **Redis (ElastiCache-compatible)** :- Caching and real-time pub/sub
- **Encryption & Secrets Management (KMS / Vault)** :- Secure key handling

3. Data Ingestion & Processing
- **Event Streaming (Kafka / Kinesis)** :- Real-time voice & IoT streams
- **Message Queue (SQS / RabbitMQ)** :- Asynchronous microservice communication
- **Workflow Orchestration (Step Functions / Temporal)** :- Multi-step processes
- **ETL Processing (Glue / Spark / EMR)** :- Data transformation & analytics
- **IoT Integration (IoT Core or MQTT Broker)** :- Farm sensor connectivity

4. Machine Learning & AI
- **Foundation Models (Bedrock / Open Models)** :- Conversational AI
- **Custom ML (SageMaker / MLflow + PyTorch/TensorFlow)** :- Crop disease detection / Yield prediction / Loan eligibility scoring / Health risk profiling
- **Speech Services (Transcribe / Open STT engines)** :- Voice recognition
- **Text-to-Speech (Polly / Open TTS engines)** :- Multilingual voice output
- **Computer Vision (Rekognition / OpenCV models)** :- Image analysis
- **Recommendation Engine (Personalize / Custom ML models)**

5. Automation & DevOps
- **CI/CD (GitHub Actions / CodePipeline)** :- Automated builds & deployments
- **Container Registry (ECR / Docker Hub)** :- Image storage
- **Monitoring (CloudWatch / Prometheus + Grafana)** :- Logs, metrics, alerts
- **Event Bus (EventBridge / Kafka)** :- Event-driven system design

6. Frontend & Mobile
- **React Native** :- Android-first mobile application
- **Redux / Context API** :- State management
- **Offline Storage (AsyncStorage / SQLite)** :- Offline-first support
- **Voice UI Components** :- Custom low-literacy interaction layer
- **Push Notifications (SNS / Firebase Cloud Messaging)**

7. Security & Privacy
- OAuth2 / JWT Authentication
- Cognito (optional) or Custom Auth Service
- Role-Based Access Control (RBAC)
- End-to-End Encryption (TLS + At-Rest Encryption)
- Consent & Data Control Framework (Privacy-by-Design Architecture)

Architecture Style
- Microservices Architecture
- Event-Driven Communication
- Offline-First Mobile Design
- AI-Augmented Service Platform
- Open Data Standards for Government Integration

Estimated implementation cost (optional):

Service / Component	Cost Driver (What you pay for)	Estimated Monthly Cost
Backend Hosting (EC2 / App Runner / VPS)	Single small instance (Flask + Node.js)	$20 – $40
PostgreSQL (Managed DB or small instance)	20–30GB storage	$20 – $50
Object Storage (S3-compatible)	Voice + image storage (30–50GB)	$5 – $15
Redis (optional, small instance)	Caching + sessions	$10 – $25
API Gateway (Basic REST + WebSocket)	1–2M requests	$10 – $30
Speech-to-Text (Transcribe or equivalent)	~3,000–5,000 minutes/month	$30 – $80
Text-to-Speech (Polly or equivalent)	Characters generated	$10 – $40
LLM / AI Processing (Bedrock or similar)	Token usage (light usage)	$40 – $120
Image Processing (Rekognition or CV API)	2k–5k images/month	$15 – $50
Push Notifications (SNS / FCM)	Mostly free	$0 – $5
Monitoring & Logs	Basic logs + metrics	$5 – $20
Security (KMS / Secrets)	Minimal usage	$0 – $10

AI for Bharat Hackathon

Powered by aws

Thank You