SmartKYC Technical Documentation

1. Project Overview

SmartKYC is an AI-powered, API-first identity verification platform designed to reduce customer onboarding time from days to under 10 minutes. By orchestrating a multi-modal AI pipeline—combining Optical Character Recognition (OCR), Document Forensics, Biometric Liveness Detection, and Risk Analytics—SmartKYC delivers a secure, compliant, and "zero-touch" verification experience.
The system is designed as a state-aware workflow engine that guides the user through verification steps (Document Upload → Biometric Check → Risk Analysis) while maintaining a strict audit trail of every decision made by the AI.

2. Technology Stack

The SmartKYC platform relies on a modern, cloud-native stack optimized for speed, security, and scalability.

Backend & API Layer

- **Language:** Python 3.9+
- **Framework:** Django & Django REST Framework (DRF)
- **Architecture:** RESTful API with State-Machine Logic
- **Data Handling:** JSON-based Document Storage (Prototyped for NoSQL scalability)

Mobile Frontend (Android)

- **Language:** Kotlin / Java
- **Networking:** Retrofit (for API communication)
- **UI/UX:** Jetpack Compose / XML Layouts designed for "Context-Aware" user guidance.

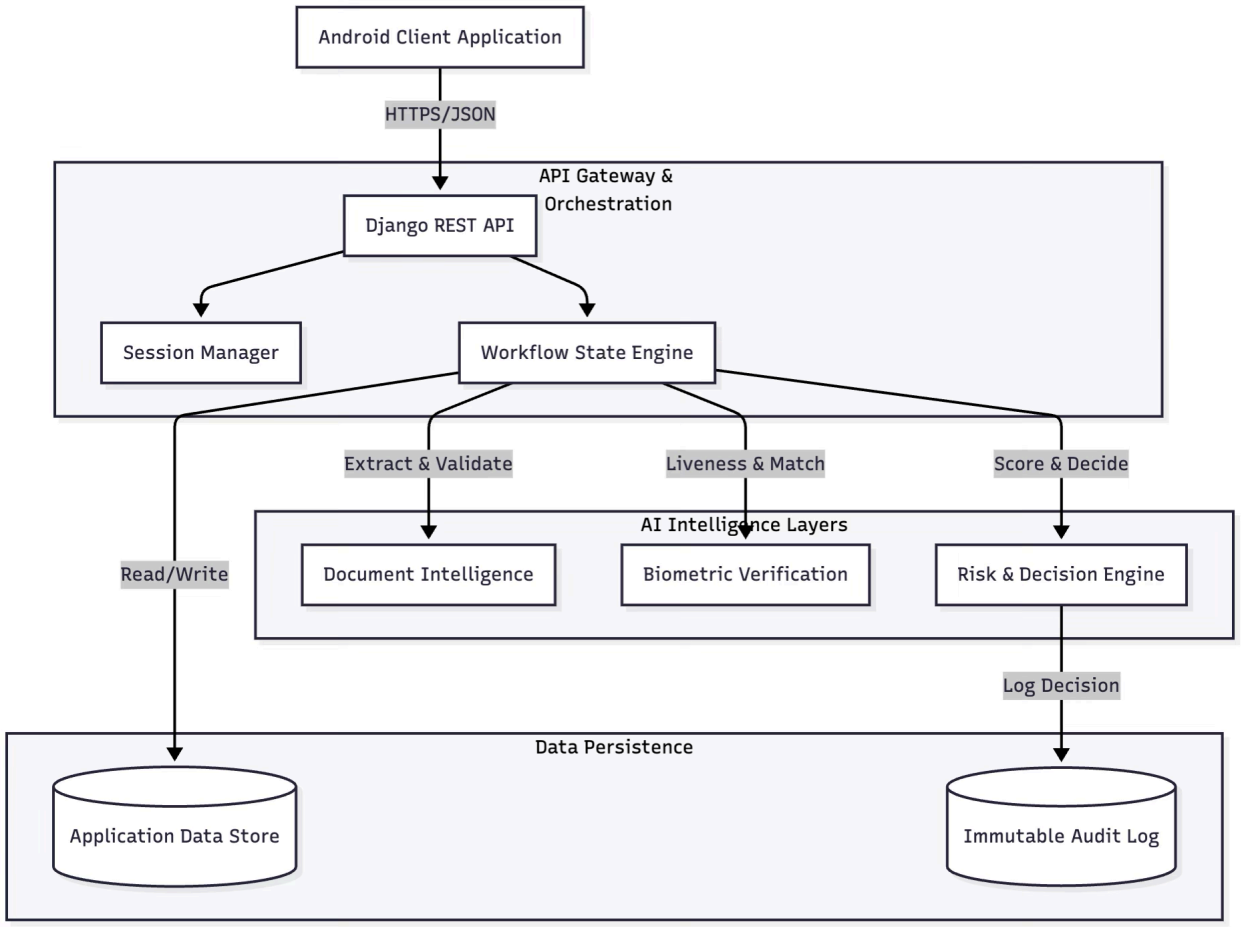
AI & Machine Learning (Simulated Core)

- **Document Intelligence:** Transformer-based OCR (e.g., TrOCR) for text extraction; CNNs for image forensics (tamper detection).
- **Biometric Verification:** CNN-based Face Matching (e.g., FaceNet) and Passive Liveness Detection.
- **Risk Engine:** Gradient Boosted Decision Trees (XGBoost) for risk scoring.
- **Explainability:** SHAxP (SHapley Additive exPlanations) for generating human-readable audit logs.

3. System Architecture

SmartKYC follows a **Layered Service Architecture**. The backend acts as an orchestration layer that manages the user session state and coordinates calls to specialized AI modules.

High-Level Architecture Diagram



Architectural Components

1. **The Orchestrator (Django Backend):**
    - Acts as the central nervous system.
    - Exposes four primary endpoints (/start, /document, /selfie, /analyze).
    - Enforces the specific sequence of the KYC journey; a user cannot proceed to Selfies until Documents are validated.
1. **The Workflow Engine:**
    - State-machine logic embedded within the API.
    - Automatically transitions application status (e.g., from PENDING_DOCUMENTS to PENDING_RISK_ANALYSIS) based on AI outputs.
1. **The Data Layer:**

- Utilizes a flexible JSON-document model. This allows the system to adapt to different ID types (Passports vs. Utility Bills) without requiring rigid schema migrations.

4. Data Model & Storage

SmartKYC utilizes a **Document-Oriented Data Model**. Instead of rigid relational tables, application data is stored as flexible, hierarchical JSON structures. This approach allows the system to adapt to varying KYC requirements (different ID types, local regulations) without schema migrations.

4.1 Core Data Entity: The Application Object

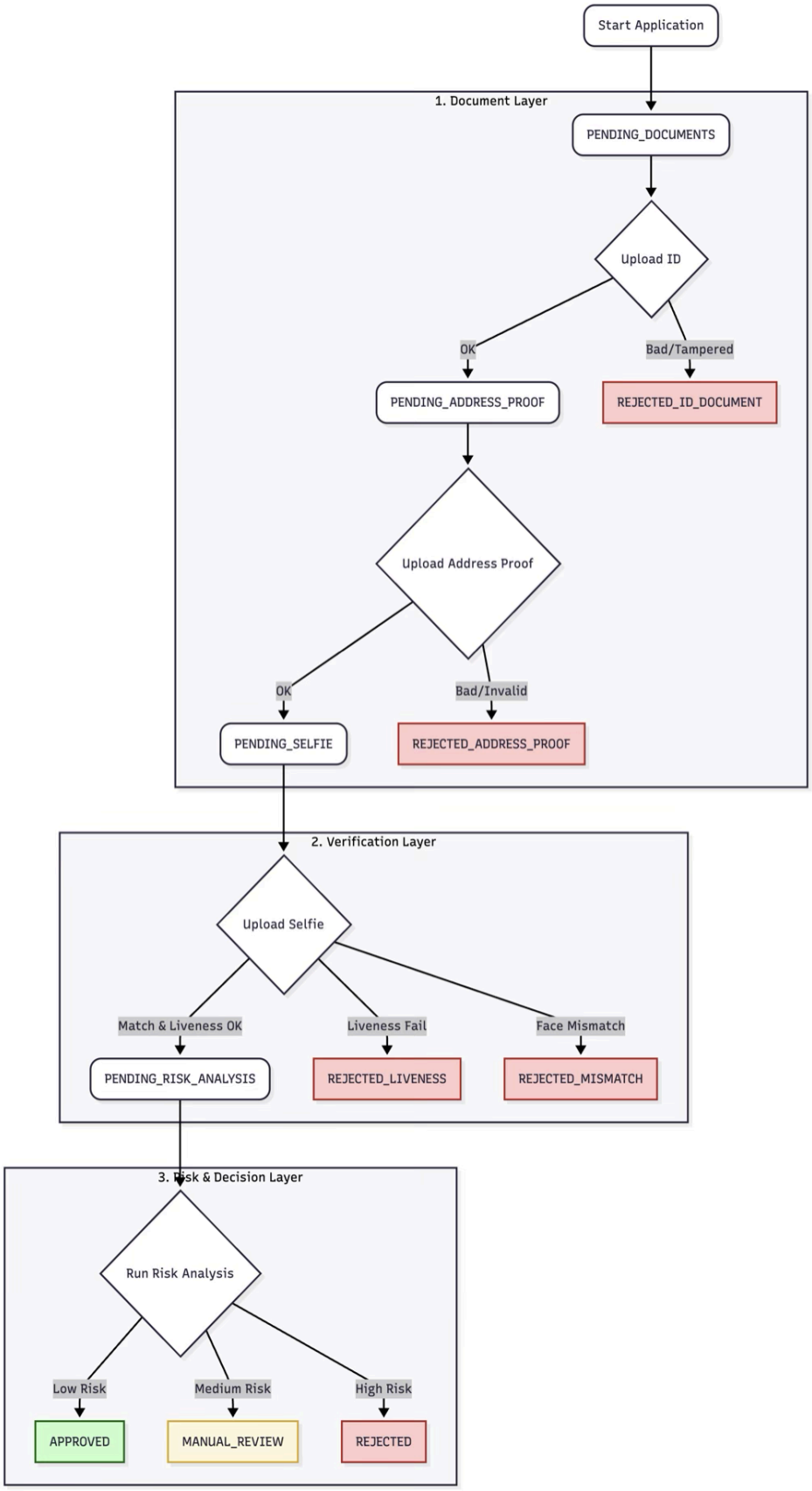
Every user journey is encapsulated in a single Application object, identified by a UUID. This object aggregates raw inputs, AI inference results, and the overall workflow state.

JSON Schema Structure:

```
{
  "application_id": "UUID (Primary Key)",
  "status": "String (e.g., PENDING_RISK_ANALYSIS, APPROVED)",
  "created_at": "Timestamp",
  "updated_at": "Timestamp",
  "documents": {
    "id_document": {
      "type": "PASSPORT | DRIVER_LICENSE",
      "file_path": "URI",
      "forensics": {
        "is_tampered": "Boolean",
        "confidence": "Float"
      },
      "extracted_data": {
        "name": "...",
        "dob": "..."
      }
    },
    "address_proof": {
      "type": "UTILITY_BILL",
```

```
      "extracted_data": {
        "address": "...",
        "provider": "..."
      }
    }
  },
  "selfie": {
    "file_path": "URI",
    "liveness_check": {
      "status": "REAL | SPOOF",
      "score": "Float"
    },
    "face_match": {
      "score": "Float",
      "decision": "MATCH | MISMATCH"
    }
  },
  "risk_analysis": {
    "risk_score": "Integer (0-100)",
    "decision": "APPROVED | REJECTED | MANUAL_REVIEW",
    "xai_explanations": [
      "List of human-readable reasons for the decision"
    ]
  }
}
```

4.2 KYC Application Lifecycle


Start Application
1. Document Layer
PENDING_DOCUMENTS
Upload ID
OK
Bad/Tampered
PENDING_ADDRESS_PROOF
REJECTED_ID_DOCUMENT
Upload Address Proof
OK
Bad/Invalid
PENDING_SELFIE
REJECTED_ADDRESS_PROOF
2. Verification Layer
Upload Selfie
Match & Liveness OK
Liveness Fail
Face Mismatch
PENDING_RISK_ANALYSIS
REJECTED_LIVENESS
REJECTED_MISMATCH
3. Risk & Decision Layer
Run Risk Analysis
Low Risk
Medium Risk
High Risk
APPROVED
MANUAL_REVIEW
REJECTED

5. AI / ML / Automation Components

The "Smart" in SmartKYC is driven by a modular AI pipeline. Each module operates independently but feeds data into the central workflow engine.

5.1 Document Intelligence Layer

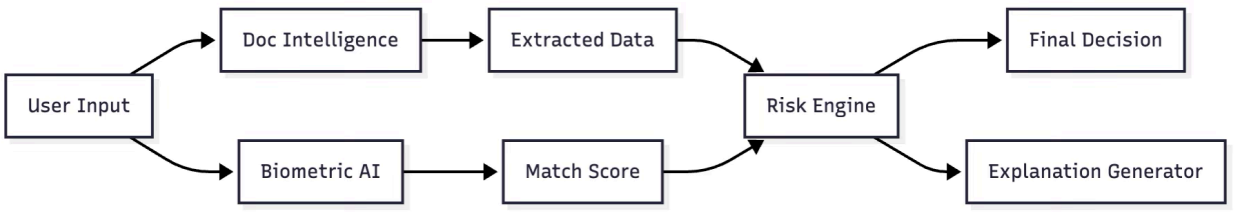
- **Function:** Extracts PII (Personal Identifiable Information) and validates document authenticity.
- **Technique:**
    - **OCR (Optical Character Recognition):** Utilizes Transformer-based models (e.g., TrOCR) for high-accuracy text extraction from noisy images.
    - **Forensics:** Uses Convolutional Neural Networks (CNNs) to detect pixel manipulation, font inconsistencies, and digital tampering (e.g., "Photoshop" detection).

5.2 Verification Layer (Biometrics)

- **Function:** Ensures the user is present and matches the ID document.
- **Technique:**
    - **Face Matching:** Embeds the ID photo and the selfie into vector space (using models like FaceNet/ArcFace) and calculates the Euclidean distance to determine similarity.
    - **Liveness Detection:** Analyzes the selfie for depth cues, texture analysis, and screen reflections to prevent "spoofing" attacks (using photos of screens or printed masks).

5.3 Risk Intelligence & Explainability Layer

- **Function:** The final decision maker.
- **Technique:**
    - **Scoring:** An XGBoost classifier aggregates hundreds of signals (e.g., "IP address location," "DOB mismatch," "Low liveness score") to generate a risk probability.
    - **Explainability (XAI):** We utilize SHAP (SHapley Additive exPlanations) values to reverse-engineer the AI's decision. Instead of a "Black Box" rejection, the system outputs: *"Rejected because: Address on ID does not match Utility Bill."*



6. Security & Compliance

Given the sensitive nature of Identity Verification, security is architected into every layer.

6.1 Data Security

- **Encryption in Transit:** All API communication is secured via TLS 1.3.
- **Encryption at Rest:** Stored JSON documents and images are encrypted using AES-256 standards.
- **PII Redaction:** The logs generated by the application automatically redact sensitive fields (like ID numbers) to prevent data leakage in server logs.

6.2 Regulatory Compliance

- **Audit Trails:** The system maintains an immutable append-only log of every state change. If an application moves from PENDING to APPROVED, the timestamp, the AI score that triggered it, and the specific logic used are recorded.
- **GDPR/Data Privacy:** The architecture supports "Right to be Forgotten." Because data is structured by application_id, deleting a user's data is a precise, single-operation deletion of their JSON record and associated assets.

7. Scalability & Performance

The architecture is designed to scale from a hackathon prototype to a production workload handling millions of verifications.

7.1 Asynchronous Processing

While the API is synchronous for the user (HTTP Request/Response), the heavy AI lifting is designed to be offloaded to worker queues (e.g., Celery/Redis).

- **Benefit:** The API server remains responsive. The user receives a "Processing" status immediately, while the GPU-intensive AI models run in the background.

7.2 Stateless Microservices

The backend is stateless. The application state is stored entirely in the database (or JSON file in the prototype).

- **Benefit:** We can horizontally scale the API layer by simply adding more server instances behind a Load Balancer without worrying about session stickiness.

7.3 Modular AI Updates

Because the AI layers are decoupled via the Workflow Engine, we can upgrade the OCR model (e.g., v1.0 to v2.0) without rewriting the business logic or taking down the Biometric service.

8. Conclusion & Future Roadmap

Conclusion

SmartKYC reimagines the traditional Know Your Customer process by shifting from a manual, friction-heavy workflow to an **intelligent, automated, and transparent** experience. By leveraging a microservices architecture and a multi-modal AI pipeline, the platform addresses the "Trilemma" of Identity Verification: balancing **Security**, **User Experience**, and **Regulatory Compliance**.

The system successfully demonstrates that onboarding does not need to be a black box. Through **Explainable AI (XAI)**, we empower users to correct their own mistakes (e.g., "Image too blurry") and provide compliance officers with the "Why" behind every decision. This results in higher conversion rates, lower operational costs, and a robust defense against fraud.

Future Roadmap

While the current implementation serves as a comprehensive "walking skeleton" and functional prototype, the roadmap for production includes:

1. **Live Database Integration:** Replacing the JSON storage with a distributed NoSQL database (e.g., MongoDB or DynamoDB) for global scale.
2. **Blockchain Identity Ledger:** Storing a hash of the verified identity on a private permissioned blockchain to create a reusable "Digital ID," allowing customers to onboard at other partner banks instantly without re-submitting documents.
3. **Video Liveness Integration:** Upgrading the passive selfie check to an active video challenge (e.g., "Turn your head left, then blink") to defeat advanced deepfake attacks.
4. **Global Watchlist Screening:** Integrating real-time API calls to Interpol and OFAC sanctions lists during the Risk Analysis phase.

SmartKYC is not just a tool for verification; it is the foundation for a secure, inclusive, and efficient digital financial ecosystem.