Software Requirements Specification (SRS)

Project/Initiative: Client Onboarding Wealth Management System

Prepared by: Business Analyst (Prashant Gavhane)

September 2025

Version 1.0

1 Document Revisions

Date	Version Number	Document changes
05/09/2025	1.0	Initial Draft

2 Approvals

Role	Name	Title	Signature	Date
Project Sponsor				
Business Owner				
Project Manager				
System Architect				
Development Lead				
User Experience Lead				
Quality Lead				
Content Lead				

3. Introduction

3.1 Purpose, Scope, and Audience

- **Purpose:** Define the detailed system-level requirements for a digital client onboarding solution in wealth management.
- **Scope:**
  - Automates onboarding from lead creation → KYC/AML → risk profiling → e-sign → account creation.
  - Integrates with CRM, Core Banking, Compliance APIs, DMS, and e-Signature providers.
  - Ensures compliance with KYC, AML, FATCA/CRS, GDPR.
- **Audience:** Business Analysts, Product Owners, Development Teams, QA/Testers, Compliance Officers, and Stakeholders.

3.2 Assumptions and Constraints

- **Assumptions:**
  - Clients have internet access and a digital device.
  - APIs from compliance/e-sign providers are available.
  - RM and Ops teams use web dashboards for exceptions.
- **Constraints:**
  - Must comply with local and international regulations.
  - Onboarding SLA ≤ 2 days for standard clients, ≤ 5 days for HNWI.
  - Only supported browsers: Chrome, Edge, Safari (latest 2 versions).

4. Overall Description

4.1 Product Perspective & Overview

- A web + mobile responsive platform.
- Modular architecture: Forms Engine, Document Management, Compliance Screening, Risk Profiling, Consent, Account Creation.
- Cloud-hosted, scalable microservices.

4.2 System Environment & Dependencies

- **Environment:** AWS cloud, Linux servers, Postgres DB, React front-end.
- **Dependencies:**
  - Compliance APIs (Refinitiv, World-Check).
  - e-Sign APIs (DocuSign/AdobeSign).
  - Core Banking (SOAP/REST APIs).
  - CRM (Salesforce REST API).

5. Functional Requirements

5.1 Detailed System Functionalities

- **UC-001:** Capture personal/financial data (multi-language, save/resume).
- **UC-005:** Upload & validate documents (OCR + duplicate check).
- **UC-008:** Perform AML/KYC screening via APIs.
- **UC-011:** Conduct risk questionnaire → auto risk score.
- **UC-013:** Capture digital consent + GDPR agreement.
- **UC-014:** Enable e-signatures for agreements.
- **UC-015:** Auto-create accounts in core banking.
- **UC-017:** Sync client data to CRM.
- **UC-018:** Ops exception dashboard for monitoring/escalations.

5.2 Interactions with External Systems

- **CRM:** Store client profiles, history.
- **DMS:** Store signed documents, KYC files.
- **Compliance APIs:** PEP/sanctions checks.
- **e-Sign Providers:** Digital signatures.
- **Core Banking:** Account creation.

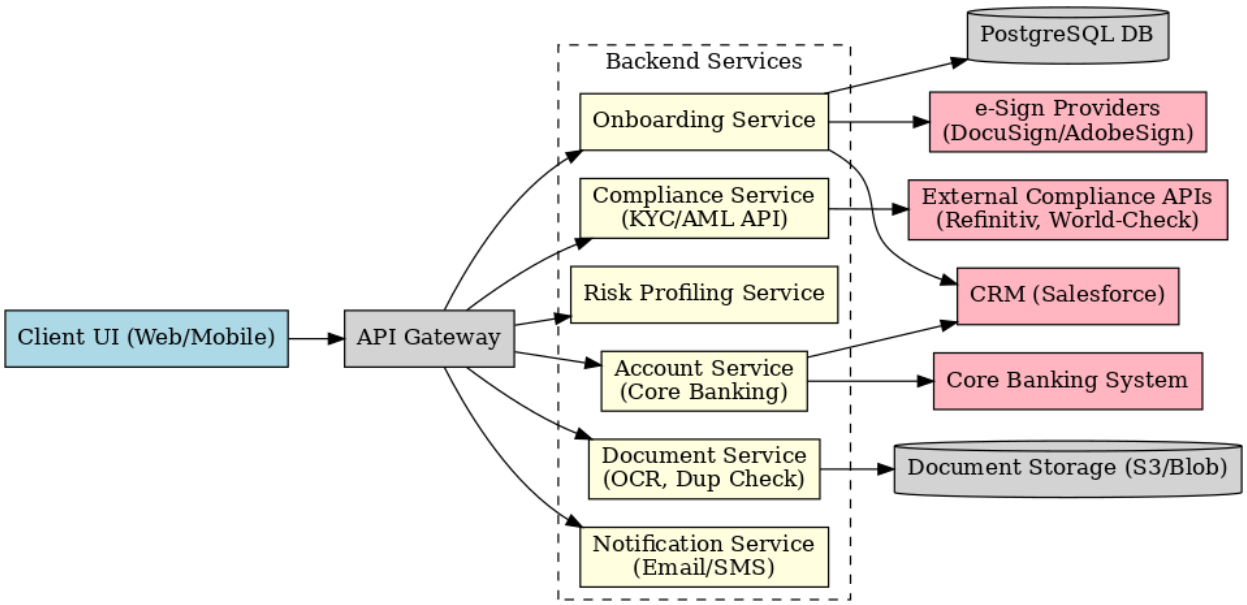
6. Non-Functional Requirements

- **Performance:** Handle 500 concurrent onboardings; average response ≤ 3 seconds.
- **Scalability:** Support growth to 1M+ clients.
- **Availability:** 99.95% uptime.
- **Security:** AES-256 data encryption; TLS 1.3; role-based access.
- **Compliance:** GDPR, FATCA, CRS, AML/KYC.
- **Auditability:** All actions logged with timestamp/user ID

7. System Architecture & Design

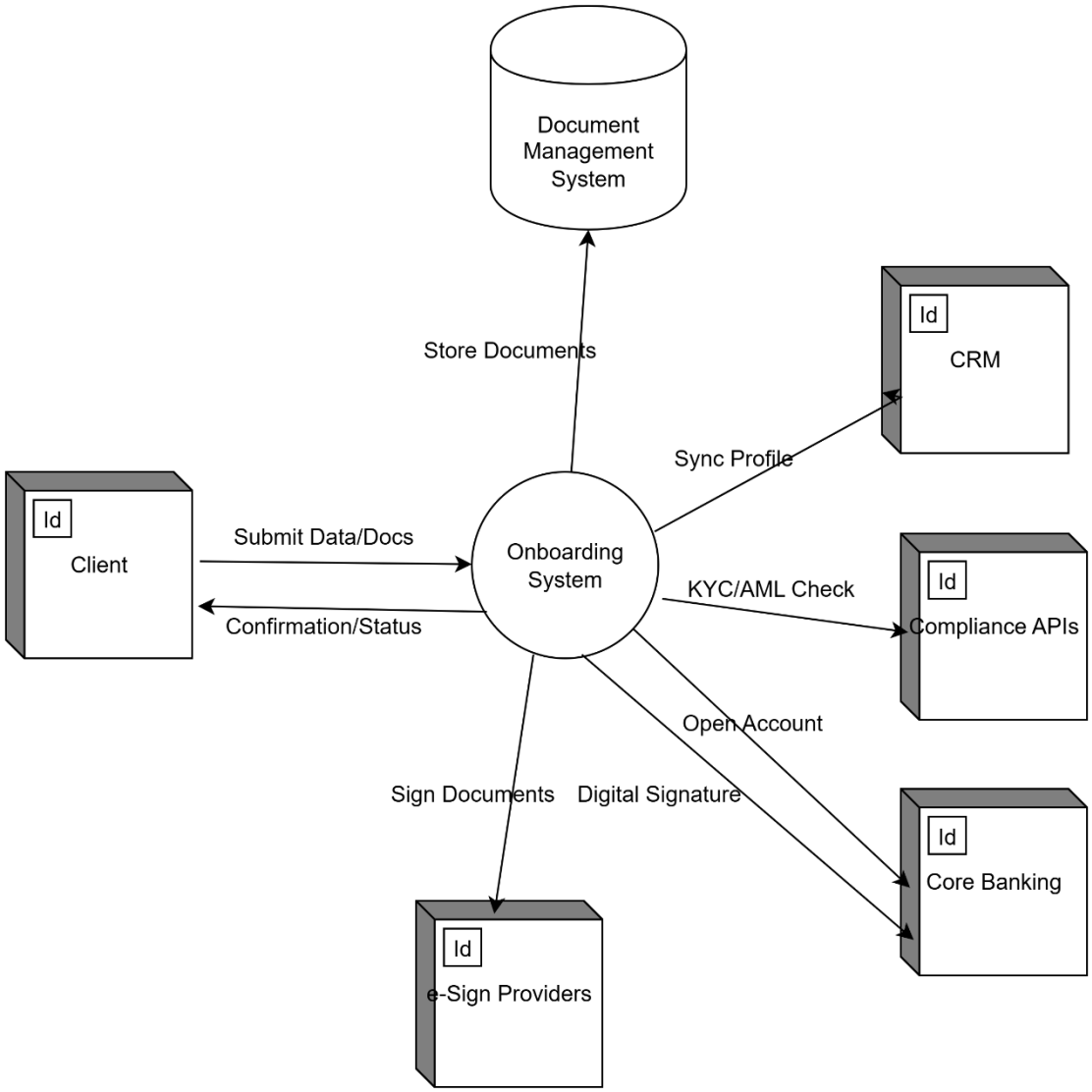
7.1 High-Level System Architecture Diagram

- **Frontend:** React/Angular → API Gateway
- **Backend Services:**
  - Onboarding Engine (forms, rules)
  - Compliance Service (AML/KYC APIs)
  - Risk Service (scoring engine)
  - Document Service (OCR, duplicate check)
  - Account Service (core banking integration)
  - Notification Service (email/SMS)
- **Database:** PostgreSQL
- **Storage:** S3/Blob storage for documents



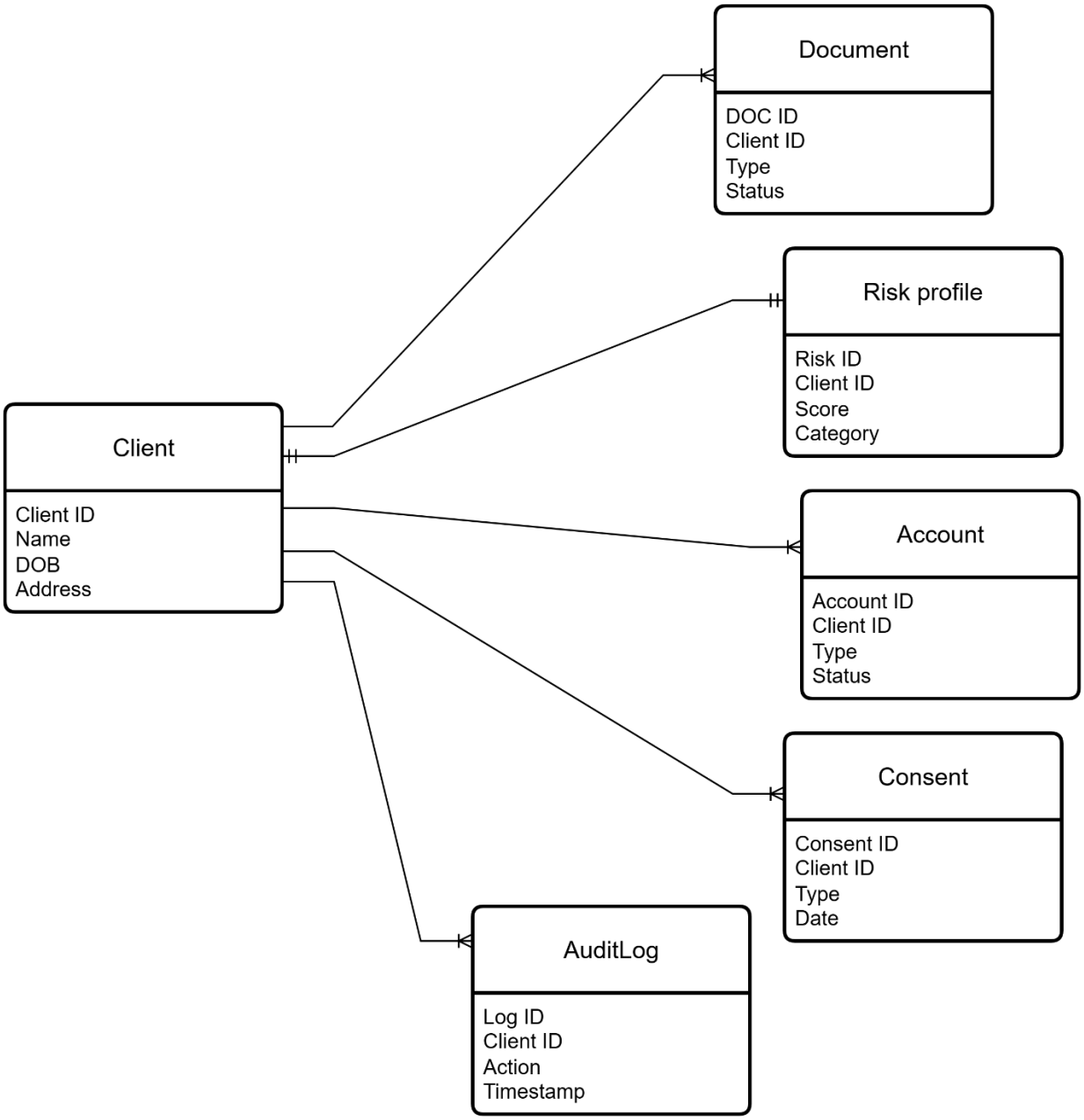
7.2 DFD (Data Flow Diagram – Level 1)

- Client submits data → Onboarding Engine → Compliance/CRM/Core Banking APIs → Response → Confirmation to Client.



7.3 ER Diagram (Entities)

- **Entities:** Client, Document, RiskProfile, Account, Consent, AuditLog.
- **Relationships:**
  - Client → Documents (1:M)
  - Client → RiskProfile (1:1)
  - Client → Accounts (1:M)
  - Client → Consents (1:M)



7.4 Database Schema (Sample)

- **Client Table:** ClientID, Name, DOB, Address, RiskID
- **Document Table:** DocID, ClientID, DocType, Status
- **RiskProfile Table:** RiskID, Score, Category
- **Account Table:** AccountID, ClientID, AccountType, Status
- **Consent Table:** ConsentID, ClientID, ConsentType, Date

7.5 API Endpoints (Sample)

- POST /api/onboarding/client – Create client profile
- POST /api/onboarding/documents – Upload document
- GET /api/onboarding/kyc/{clientId} – Run KYC screening
- POST /api/onboarding/risk – Submit questionnaire
- POST /api/onboarding/esign – Send agreement for signing
- POST /api/onboarding/account – Open account

8. Constraints & Dependencies

- **Technical Limitations:**
  - Only digital document formats (PDF, JPG, PNG).
  - Mobile devices limited to Android/iOS apps with offline support.

Dependencies: External APIs for KYC, e-Sign, Core Banking must be available 24/7

9 Performance & Scalability

- Must support **500 TPS (transactions per second)** during peak onboarding.
- Horizontal scaling on Kubernetes cluster.
- Asynchronous queuing for API retries (RabbitMQ/Kafka).

10. Testing & Validation

- **Unit Testing:** API-level validations.
- **Integration Testing:** External API contracts.
- **System Testing:** End-to-end onboarding flows.
- **UAT:** With compliance & RM teams.
- **Performance Testing:** Load & stress tests.
- **Security Testing:** Penetration + vulnerability scanning

11. Approval & Sign-Off

- **Stakeholder Review:** Draft circulated to Product, Compliance, Tech, Ops.
- **Final Sign-Off:** Head of Wealth Management, Compliance Officer, CTO