WKforce – Software Requirements and Technical Design Document

Version: 1.0
Date: TBD (Post-July 2025)
Prepared for: Tenyne Inc

---

1. Executive Summary

The WKforce Software Requirements and Technical Design Document provides a comprehensive specification for the development of a scalable, compliant, and user-friendly workforce orchestration platform. This document translates the strategic business goals outlined in the BRD into actionable engineering deliverables.

WKforce aims to enable project owners to assemble human talent, AI agents, and robotic tools in under 7 days—drastically reducing the time-to-productive-team metric. This aligns with Tenyne Inc's vision of merging human creativity with machine efficiency.

WKforce is not just a platform but an infrastructure layer designed for the future of work. It blends marketplace dynamics, AI-enhanced sourcing, and robotic process orchestration in one unified cloud-native platform. The software system will be the backbone for talent lifecycle management, from initial request to final delivery, while enforcing compliance and ensuring data privacy.

Intended Audience

This document is written for:

- Product Owners defining scope
- Engineering teams designing the platform
- QA and DevOps ensuring performance and compliance
- Stakeholders including legal, finance, and operations
- Pilot clients and partners evaluating feature delivery

Document Overview

Each section of this document reflects an evolving scope from concept to concrete specification:

- Sections 1–3 define the core business and functional objectives
- Sections 4–8 detail architecture, technology choices, and compliance
- Sections 9–14 support long-term operations, QA, scaling, and partner alignment

---

2. System Overview

WKforce serves as a next-generation workforce marketplace integrating freelancers, boutique agencies, AI agents, and (eventually) robotic service providers. It enables project managers to issue briefs, receive matched talent suggestions, initiate contracts, and manage deliverables on a unified platform.

The platform is designed for the SMB and mid-market sector, prioritizing ease of onboarding, budget predictability, and compliance automation. The system includes an employer wizard, AI-powered role card generation, smart matching logic, and an agent-pluggable delivery hub.

Key Goals

- Reduce team formation cycle to < 7 days
- Deliver seamless cross-border contracting and payments
- Offer AI and robotic integration from day one (via agent sockets)
- Achieve 50%+ gross margin through SaaS + credit monetization

Vision

WKforce is a hybrid workforce orchestration platform that enables businesses to assemble and manage productive teams by integrating humans, AI agents, and future robotic process units. The system is designed to be composable, extensible, and compliance-aware.

It is intended to be the "Workforce Cloud" for SMBs and mid-market enterprises worldwide. Within one platform, users can:

- Submit a business need via a plain-language brief
- Get structured team recommendations
- Contract and fund human or agent resources
- Track deliverables in a unified workspace

Market Position

WKforce differentiates itself from Upwork (human-focused marketplaces) and emerging AI agent stores (bot-only) by offering:

- Multi-agent orchestration
- Integrated compliance and escrow
- Human-in-the-loop fallback
- AI-powered job specification and matching

Core Platform Capabilities

1. **Orchestration Engine:** Enables AI+human hybrid delivery
2. **KYC & Compliance Layer:** Automates identity and license checks
3. **Smart Matching System:** Uses vector search and rules-based filters
4. **Delivery Interface:** Kanban + chat + AI agent sockets
5. **Payment System:** Credit wallet + escrow + ledgering

Stakeholder View

- **Employers:** Expect fast, compliant workforce formation with one invoice
- **Freelancers/Agencies:** Want reliable, recurring income and easy onboarding
- **AI Vendors:** Seek monetization, dashboards, and throttling visibility

- **Compliance Teams:** Need ledgers, consent logs, and traceable workflows

---

3. Functional Requirements

F-01: User Registration & KYC

Description: Users must register as individuals or teams. KYC documentation must be uploaded and verified for compliance.

User Flow:

1. User selects role (Employer, Talent, Agent)
2. Inputs profile info
3. Uploads identity and legal documentation
4. Team owners may invite members
5. System validates data automatically and flags for manual review

System Features:

- Email/magic-link login
- ML-based KYC pre-screening
- Parent-child account linking (for teams)

Acceptance Criteria:

- 95% of users auto-approved within 5 minutes
- Admin panel shows pending/approved/rejected status
- All uploaded KYC docs are encrypted and time-stamped

F-02: Employer Wizard (Brief to Spec)

Description: Converts natural-language project briefs into formal specifications ("role cards") using LLM and input prompts.

User Flow:

1. Employer submits project need (text prompt)
2. System asks clarifying questions (budget, timeline, goals)
3. LLM generates a structured set of deliverables and roles
4. Employer approves or refines suggestions

System Features:

- Embedded LLM (OpenAI or Anthropic)
- Spec templates per role type (e.g., data analyst, RPA bot)
- AI-human hybrid editing interface

Acceptance Criteria:

- 90% of submitted briefs result in usable specs in < 3 minutes
- Generated specs must include: goals, roles, timeline, budget
- Employers can edit and save specs to drafts

F-03: Smart AI Chat (People | Agents | Hybrid)

Description: Users interact with a smart assistant that can answer queries, assign tasks to agents, or escalate to humans.

Modes:

- **People:** Queries routed to available human specialists
- **Agents:** Fully handled by LLMs or robotic APIs
- **Hybrid:** AI starts task; humans intervene at milestones

User Flow:

1. User opens chat with project bot
2. Chatbot interprets request and classifies it
3. Routes to human/agent/hybrid based on request type

System Features:

- NLP-based intent classification
- RAG (retrieval-augmented generation) to avoid hallucination
- Escalation triggers (e.g., confusion, failure, ambiguity)

Acceptance Criteria:

- Median response time < 1 sec
- Users can view agent vs. human origin for each reply
- Escalations logged and audited by admin panel

F-04: Budget & Credit Management

- Splits payments between USD and internal credits
- Enforces milestone payments with ledger visibility
- Shows real-time credit consumption

F-05: Matching Engine

- Uses rule-based ranker for humans and agents
- Displays top-20 ranked profiles with latency < 150 ms
- Offers filters for location, skill tags, rate, and availability

F-06: Contract Generator & E-signature

- Generates SoW + legal clauses
- Integrates with DocuStub for secure e-signature
- Supports redlining and approval loop pre-signature

F-07: Escrow & Ledger

- Stripe test-mode escrow for MVP
- Displays pending/released payments by milestone
- Immutable ledger records per transaction event

F-08: Delivery Hub

- Kanban board for task tracking
- Embedded chat and file vault (100MB)
- Webhook interface for agent status updates
- Time tracking for compliance and invoice verification

---

4. Non-Functional Requirements

This section outlines the non-functional attributes of the WKforce platform. These ensure the system is performant, secure, reliable, maintainable, and scalable. Each sub-section describes technical benchmarks, architectural expectations, real-world scenarios, and design implications.

4.1 Performance Requirements

Overview: WKforce must support a responsive user experience with minimal latency, even under load. Given the AI and human hybrid workflows, real-time updates are critical.

Targets:

- API latency: p95 < 300 ms (under 100 RPS)
- UI click-to-response time: < 1.5 seconds for 90% of actions
- AI chat completion time: < 2.0 seconds (avg for short tasks)

Stress Testing & Load Handling:

- K6-based stress testing simulates up to 500 concurrent users
- Load-balanced API gateway supports horizontal scaling
- Caching strategy using Redis for session and match state

Scenario: An employer posts a brief and receives ranked candidates. The backend LLM generates role specs in real time while the frontend loads cards in < 2 seconds.

4.2 Availability and Reliability

Objective: WKforce must be highly available and resilient to single point-of-failure scenarios.

Uptime Commitment:

- MVP SLA: 99.5%
- Post-GA SLA target: 99.9% with failover

Architectural Approach:

- Redundant service containers with zone replication
- Stateless API pods with GCP Cloud Run auto-scaling
- DB backups every 15 mins (Qdrant + SQL), retained 30 days

Disaster Recovery (DR):

- Warm failover region in Frankfurt for EU users
- Recovery Time Objective (RTO): < 1 hour

- Recovery Point Objective (RPO): 15 minutes

Scenario: A sudden spike in AI queries triggers GCP autoscaler to provision 3 new pods. The LLM response latency remains under SLA.

4.3 Security Requirements

User Data Protection:

- TLS 1.3 enforced across all public endpoints
- OAuth 2.0 + device-based verification for login
- Role-based access control (RBAC) with least-privilege default
- Data-at-rest encryption using AES-256-GCM

Platform Security Practices:

- Penetration testing scheduled bi-annually
- Automated security scans via GitHub Actions + Snyk
- Audit logging for all role elevation, escrow actions, and AI prompts

Secure KYC Flows:

- Document hashes stored separately from file blobs
- Self-destructing links for identity proof delivery
- Admins access PII via consent-controlled views

Scenario: A freelancer uploads their passport for KYC. The file is stored encrypted, indexed with a hash, and viewable only through an admin-guarded audit trail.

4.4 Scalability

Design Goals:

- Horizontal scaling of APIs, background workers, and AI functions

- Modular codebase for adding new workforce types (e.g., robotics)
- Separation of compute for AI and core system logic

Traffic Planning:

- 150 active clients at MVP with a goal to support 1,000+ by year 2
- Event-driven architecture with pub-sub queue for task updates

Database Scaling:

- Use of GCP AlloyDB (Postgres-compatible) for transactional data
- Vector data (Qdrant) hosted in sharded mode with cache layer

Scenario: A marketplace event triggers 300 job match requests. The match engine horizontally scales across 6 nodes, with sub-200ms response sustained.

4.5 Usability and Accessibility

UX Commitments:

- Guided workflows with wizard interfaces (for employers)
- Mobile-first responsive layout using Tailwind
- Clear action hierarchy and undo-friendly interactions

Accessibility Benchmarks:

- WCAG 2.1 AA compliant forms and buttons
- ARIA labels for all interactive components
- High-contrast mode toggle

Localization Support:

- English (default), with pilot-ready translations in French and Yoruba

- UTF-8 full international character support

Scenario: A user with low vision accesses WKforce in high-contrast mode. All major functions, including chat, remain keyboard-accessible.

4.6 Maintainability & Observability

CI/CD Approach:

- GitHub Actions deploy to dev, staging, and production
- Infrastructure as Code (IaC) using Terraform and Helm charts

Monitoring & Logging:

- Centralized logging using GCP Logging and Prometheus
- Alerting rules for API error rates, latency spikes, and disk I/O

Service Metrics:

- Per-service dashboards (Grafana)
- Uptime robot pings every 60 seconds
- Tracing (OpenTelemetry) for inter-service latency

Code Quality Metrics:

- Linting and formatting enforced on pre-commit
- Unit test coverage minimum: 85% (services)
- Static analysis via SonarQube

Scenario: An update to the job matching engine fails pre-deploy tests due to a dropped field in the role card spec. Linter + coverage gate block release to staging.

---

5. Technical Architecture

This section presents the complete technical architecture for the WKforce platform. The architecture is designed to be modular, scalable, cloud-native, and AI-augmented. It supports high throughput, real-time interaction, secure document and identity handling, as well as agent integration via pluggable sockets.

5.1 High-Level Architecture Overview

WKforce is built on a microservices architecture, running in a containerized environment managed by GCP Cloud Run. It employs API gateways, load balancers, and scalable backend services to support thousands of concurrent workflows.

Core Layers:

- **Presentation Layer:** Web-based UI and mobile-responsive frontend
- **API Layer:** RESTful endpoints, GraphQL (Phase 2), and WebSockets for real-time updates
- **Service Layer:** Stateless backend services (auth, job matching, contracts, messaging)
- **AI Orchestration Layer:** LLM prompt pipelines and RAG-based content generators
- **Persistence Layer:** SQL (AlloyDB), Vector DB (Qdrant), Object Store (GCS)
- **Compliance Layer:** Consent logging, GDPR/NDPA hooks, audit trails

5.2 Component Breakdown

1. Frontend Client (Web App)

- React with TypeScript
- Tailwind CSS for design consistency
- Hooks-based state management with React Query
- WebSockets for live chat and board updates

2. Authentication Service

- OAuth 2.0 + device binding
- JWT tokens with refresh support
- Rate-limited endpoints for auth spam protection
- Magic link fallback login

3. KYC and Identity Service

- Document parser via OCR & AI
- Third-party KYC provider fallback (e.g., Veriff)
- Encrypted blob store (GCS + hash index)

4. Job Matching Engine

- LLM-to-role-card transformer
- Embedding generator for job vectors
- Rule-based filter and score weight configuration
- Top-20 candidate retrieval in <150 ms

5. Escrow & Ledger Service

- Integrates Stripe, Paystack, and Adyen
- Tracks escrow balances per milestone
- Generates ledger entries for audits

6. Delivery Hub Service

- Task board (Kanban)
- Embedded threaded chat
- File vault (100MB file limit)

- Agent Webhook API (REST & WebSockets)

7. Admin Console

- View pending/approved KYC entries
- Manually override job matches or disputes
- Access audit logs, consent trails, and system flags

5.3 Microservices and Interactions

Microservice	Language	API Protocol	Dependencies
Auth Service	Python	REST	PostgreSQL, Redis
KYC Service	Python	REST	OCR AI, GCS, Pub/Sub
Matching Engine	Python	REST	Qdrant, LLM API
Escrow Service	Go	REST	Stripe SDK, Postgres
Delivery Hub	Node.js	WebSocket	Redis, Firestore
Agent Gateway	Python	REST	OpenAI, Webhooks

Services are deployed using Docker and orchestrated in Cloud Run with autoscaling. Services share common libraries for logging, metrics, and tracing (OpenTelemetry).

5.4 Data Flow Diagrams

Diagram A: User Registration and KYC

1. User submits profile and documents
2. Auth service creates token and user ID
3. KYC service triggers OCR parsing and stores hash
4. Approval status returned to user UI

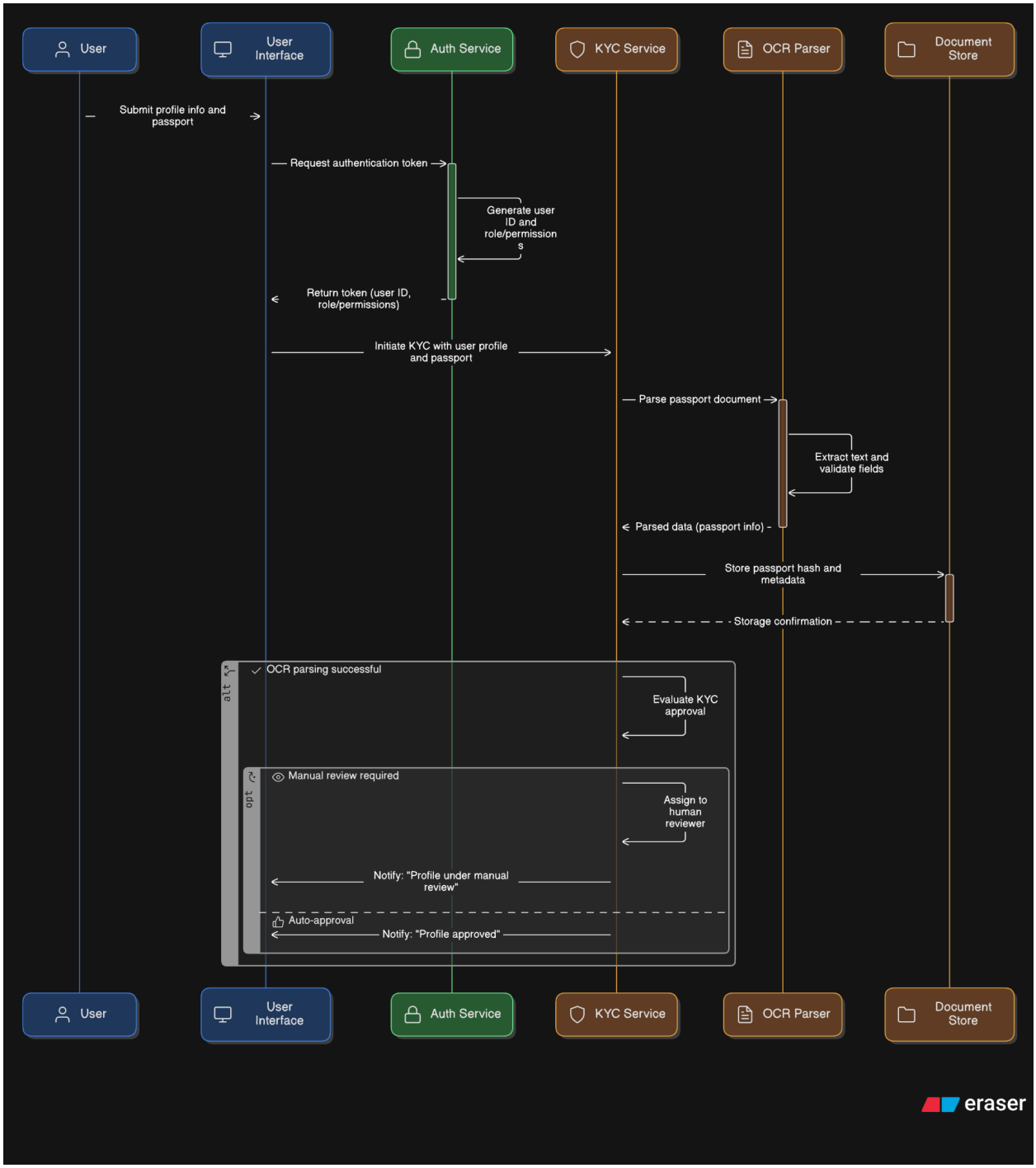


Diagram B: Job Posting to Match

1. Employer submits a plain-text brief
2. LLM transforms text to role cards

3. Job vectors generated and stored
4. Matching engine retrieves top candidates via Qdrant
5. UI renders candidate list

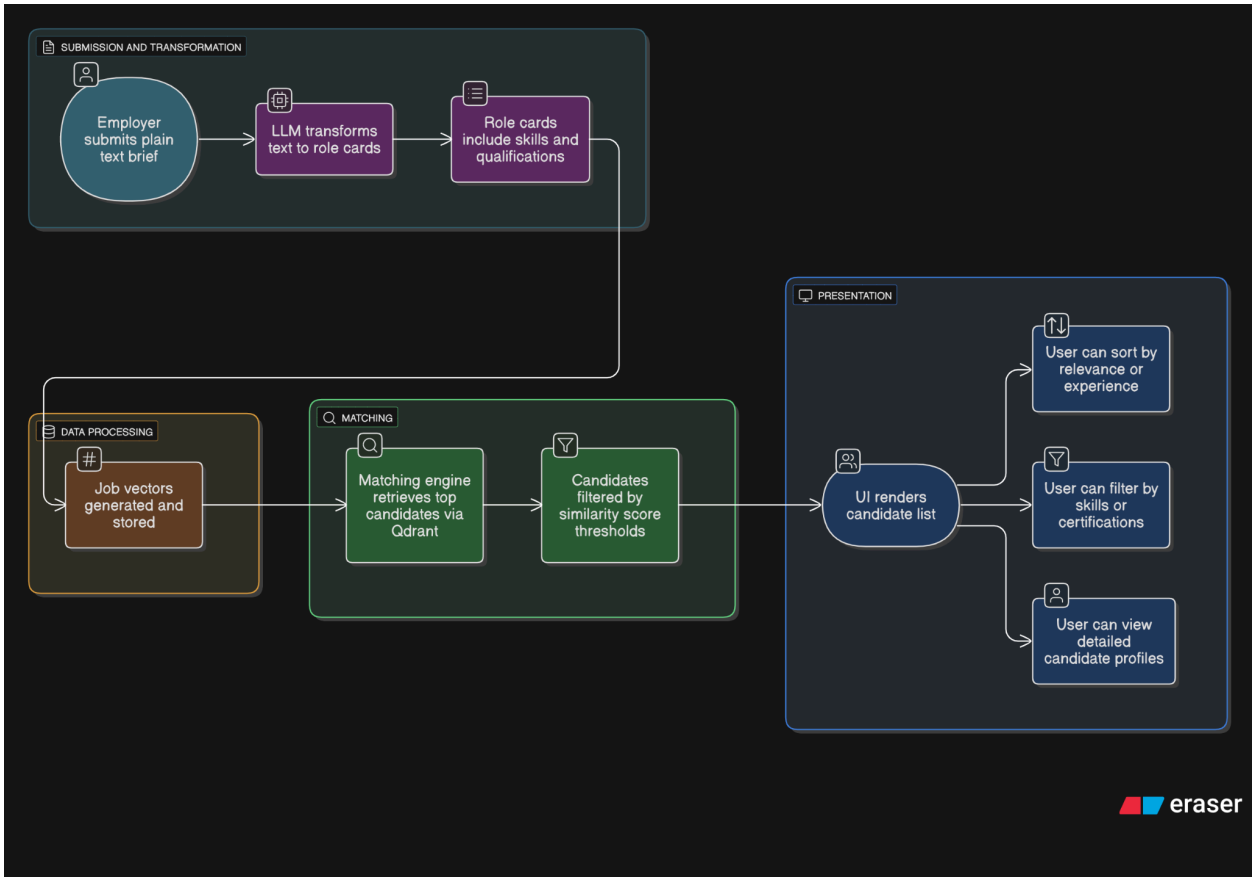
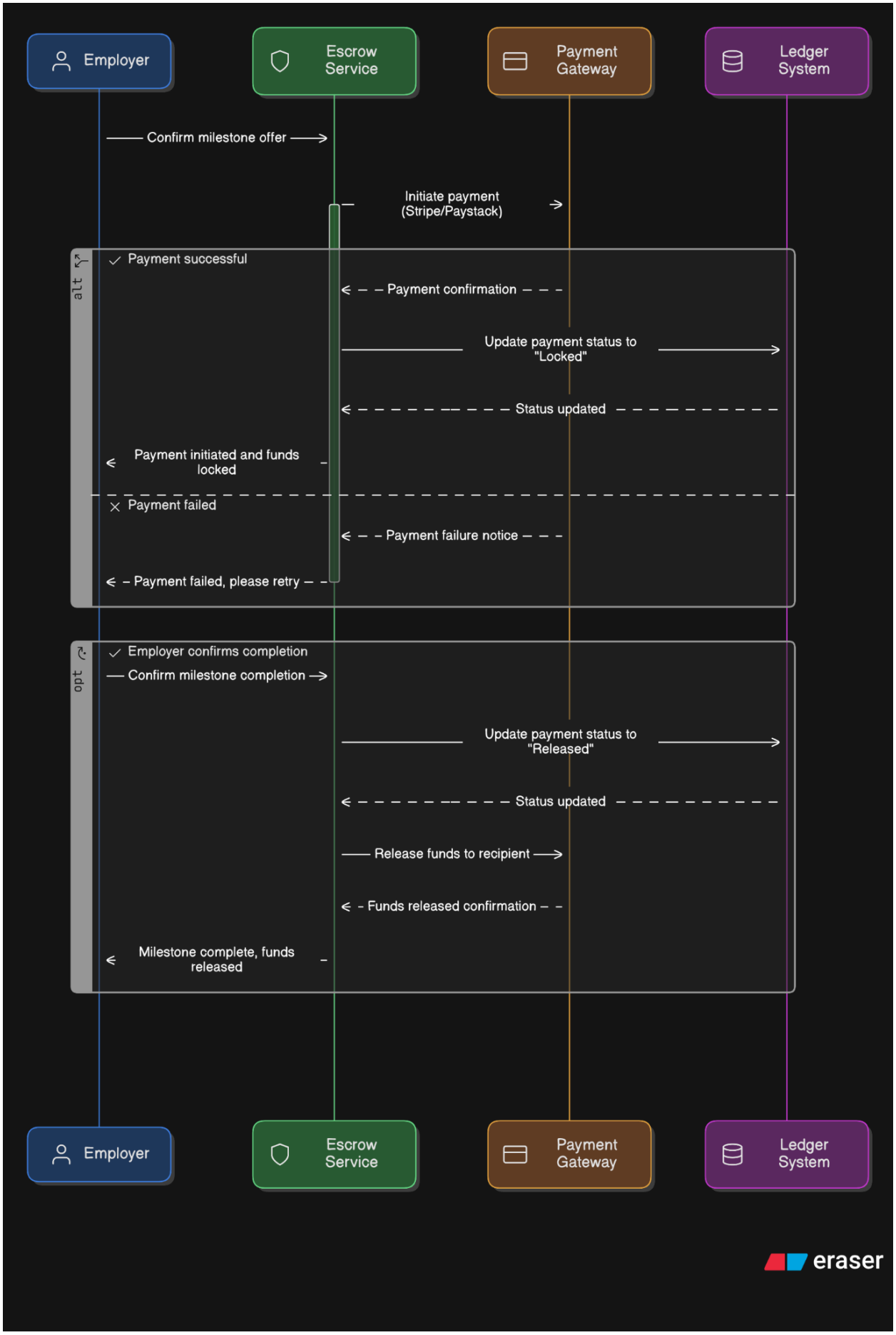


Diagram C: Escrow Transaction

1. Employer confirms milestone offer
2. Stripe/Paystack initiated via Escrow Service
3. Payment status updated in ledger
4. Funds locked until milestone confirmed


Employer
Escrow Service
Payment Gateway
Ledger System
Confirm milestone offer
Initiate payment (Stripe/Paystack)
alt
Payment successful
Payment confirmation
Update payment status to "Locked"
Status updated
Payment initiated and funds locked
Payment failed
Payment failure notice
Payment failed, please retry
opt
Employer confirms completion
Confirm milestone completion
Update payment status to "Released"
Status updated
Release funds to recipient
Funds released confirmation
Milestone complete, funds released
Employer
Escrow Service
Payment Gateway
Ledger System
eraser

5.5 Cloud Infrastructure (GCP-based)

Compute:

- Cloud Run for all container workloads
- Cloud Functions for event triggers (email, webhook retry)

Storage:

- AlloyDB (Postgres) for transactional data
- GCS for document and file storage
- Qdrant for embeddings and vector searches

Networking:

- HTTPS load balancer
- API Gateway with JWT middleware
- Cloud NAT for secure internet egress

Monitoring & Security:

- IAM per service account
- Stackdriver for logging and metrics
- Alerts on error rates, latency, and CPU load

Deployment:

- GitHub Actions triggers Terraform apply
- Blue/green deploys with rollback on error signals

5.6 Architecture Design Principles

- **Security-first:** All data access governed by encryption and RBAC
- **AI-augmented:** System assumes RAG pipelines in all flows
- **Composable:** Agents, workflows, and APIs pluggable without downtime
- **Observable:** All services traceable with distributed logs and metrics
- **Scalable:** Autoscaling built into compute and database layers
- **Resilient:** No single point of failure; failover ready

---

6. Data Design

The WKforce platform handles complex, multi-tenant data spanning user profiles, job roles, contracts, payments, AI embeddings, and delivery records. Its data design must balance security, extensibility, query efficiency, and real-time responsiveness.

This section outlines logical models, relational schemas, vector search structures, data normalization principles, and consent-focused audit layers.

6.1 Core Data Models and Relationships

A. User

- Fields: `user_id`, `email`, `role`, `display_name`, `kyc_status`, `created_at`, `updated_at`
- Enum `role`: Employer, Freelancer, Agency, AgentProvider
- Relationships:
    - May belong to a team (via `team_id`)
    - May submit job briefs or receive contracts

B. Team

- Fields: `team_id`, `team_name`, `parent_id`, `created_by`, `created_at`
- Supports hierarchical structure (parent-child teams)
- Used primarily by agencies and employer groups

C. Project

- Fields: `project_id`, `title`, `description`, `status`, `budget`, `currency`, `employer_id`, `created_at`
- Status enums: Draft, Open, InProgress, Complete, Cancelled

D. RoleSpec (Job Card)

- Fields: `role_id`, `project_id`, `title`, `description`, `skills[]`, `duration_est`, `embedding_vector[]`, `llm_summary`
- Linked to Qdrant vector for semantic search

E. Contract

- Fields: `contract_id`, `role_id`, `freelancer_id`, `status`, `signed_url`, `escrow_id`, `milestones[]`
- Stores agreement terms, schedule, and payment breakdown

F. Escrow Transaction

- Fields: `escrow_id`, `contract_id`, `amount`, `currency`, `provider`, `status`, `created_at`
- Status enums: Pending, Funded, Released, Disputed, Refunded
- Logs held by payment integration service

G. Ledger Entry

- Fields: `entry_id`, `escrow_id`, `event`, `timestamp`, `amount`, `debit_credit_flag`, `actor_id`
- Immutable transaction logs for audit trail

H. Chat / Comment

- Fields: `message_id`, `thread_id`, `sender_id`, `text`, `timestamp`, `visibility`
- Supports system messages (e.g. "Agent uploaded file")

I. AgentSocket

- Fields: `agent_id`, `project_id`, `webhook_url`, `last_status`, `throttle_limit`, `usage_counter`
- Connects to LLMs, RPA agents, drone APIs

6.2 Schema Architecture and Best Practices

- **Multi-Tenancy:** Implemented via tenant-scoped schemas with Row-Level Security (RLS)
- **Indexing:** Indexes on `user_id`, `role_id`, `project_id`, `contract_id`, `created_at`
- **Partitioning:** Time-based partitioning for logs and chats to support archiving
- **Normalization:** Up to 3NF except for `embedding_vector[]` and `milestones[]`, which use JSONB

6.3 Vector Search and Semantic Matching

The matching engine leverages dense vector embeddings for job descriptions and candidate profiles. Vectors are stored in **Qdrant**, a high-speed similarity search engine.

Process Flow:

1. Role brief is parsed into a structured `RoleSpec`

2. Skills and description are converted to embeddings
3. Candidate profiles are pre-indexed in the same vector space
4. Qdrant performs cosine similarity lookup
5. Top N candidates returned based on semantic match + rules

Example Vector Structure (Qdrant):

```JSON
{
  "id": "role_id_123",
  "vector": [0.318, 0.847, 0.227, ...],
  "payload": {
    "role": "Data Scientist",
    "duration": "2 weeks",
    "skills": ["Python", "ML", "NLP"]
  }
}
```

Match Tuning:

- Cosine similarity threshold: 0.75+
- Filters applied: skill tags, availability, location, price range

6.4 File & Document Storage

- **Provider:** Google Cloud Storage (GCS) with signed URLs
- **Sensitive Docs:** Encrypted using customer-specific keys
- **Access Control:** Tokenized access with expiration timers
- **Audit Metadata:** All uploads generate a ledger log

Document Types:

- KYC documents
- Signed contracts
- Deliverable files
- System-generated summaries (PDF, HTML)

6.5 Data Privacy and Consent Management

Consent Logging:

- Every PII action (upload, view, edit) generates a consent record
- Fields: `user_id`, `action`, `timestamp`, `ip`, `doc_id`, `purpose`

User Controls:

- GDPR/NDPA-compliant download/delete requests
- Consent screen during onboarding with TTL (time-to-live) policies
- Anonymization pipeline for account closures

Audit Layer:

- Immutable logs (signed, timestamped)
- Redundant backup of ledger to cold storage
- Admin-only access to full logs (RBAC-enforced)

6.6 Backup, Retention, and Archiving

- **Backup Interval:** Every 15 minutes for primary SQL + Qdrant
- **Retention Policies:**
    - User data: 7 years minimum

    - Logs and chats: 5 years active, 2 years archived
    - KYC: 10 years (per NDPA standard)
- **Storage Tiers:**
    - Hot: AlloyDB, Redis
    - Cold: GCS Nearline / Archive tier

Archiving Strategy:

- Daily ETL to analytics DB (BigQuery or Snowflake)
- Weekly roll-up of chat threads to archive blob
- Audit snapshots stored encrypted + checksummed

---

7. External Interfaces

The WKforce platform integrates with a wide range of third-party systems and APIs to deliver end-to-end automation, payment flows, document signing, and AI capabilities. This section outlines each interface in terms of its function, architecture, integration methods, error handling, and operational controls.

7.1 Payments and Treasury Integrations

A. Stripe (US market)

- **Use Cases:** Escrow funding, milestone-based releases, refunds
- **API Type:** REST with secure webhooks
- **Auth:** Bearer token + customer secrets stored in Vault
- **Key Events Handled:**

    - `payment_intent.succeeded`
    - `payout.failed`
    - `dispute.created`

Error Handling:

- Retries via exponential backoff
- Manual override via Admin Console for disputes

Logging: All transactions logged with status, actor, and reference IDs

B. Paystack (Nigeria)

- **Use Cases:** Localized payments in NGN, compliance with CBN
- **Webhook Events:** `charge.success`, `transfer.failed`
- **Currency Handling:** Conversion to USD for escrow if needed

Special Logic:

- NGN credits may be restricted for NG-local freelancers only
- Admin flow to override cross-border disbursement rules

C. Adyen (EU Market)

- **Use Cases:** Eurozone payments and multi-currency treasury
- **Features:**
    - FX conversion
    - SEPA payments
    - IBAN validation

Integration Roadmap: Phase 2 (Post-MVP), tied to EU market expansion

7.2 AI and Language Model Interfaces

A. OpenAI / Anthropic APIs

- **Use Cases:**
    - Job spec generation (text → structured roles)
    - AI chat with hybrid escalation
    - Project summary generation

Mode: Prompt/Completion + Retrieval-Augmented Generation (RAG)

Failure Handling:

- Latency spikes trigger retries and fallback model (e.g., Claude if GPT fails)
- Max retries: 2
- Prompt audit trail stored in `llm_logs`

Sample Prompt:

```JSON
{
  "input": "I need a growth marketer with experience in B2B SaaS",
  "model": "gpt-4",
  "context_docs": ["Marketing briefs", "Budget constraints"],
  "return_format": "role_card"
}
```

7.3 E-Signature Platform

A. DocuStub

- **Use Cases:**
    - Generating and signing contracts between employers and freelancers
    - Archiving signed statements of work (SoW)

Workflow:

1. Contract drafted by WKforce
2. Uploaded to DocuStub API
3. Signature request sent to both parties
4. Final document hash returned to `contracts` table

Document Tracking:

- Signed URL expires after 30 days
- Doc hash logged for verification
- Admin view of signature status: Draft → Sent → Signed → Expired

7.4 Communication & Productivity Tools

WKforce integrates with popular tools for collaboration and project tracking, allowing talent and employers to sync communications or manage workflows externally.

A. Slack & Telegram

- Real-time notification bots
- Invite freelancers to shared channels
- Escalation alerts for overdue milestones

Security: Message content encrypted before transit

B. ClickUp, Jira, Trello (API Phase 2)

- Task import/export
- Status synchronization between Delivery Hub and client tool
- Uses OAuth 2.0 token to act as delegated user

C. Google Meet & Calendar

- Scheduling embedded within Delivery Hub
- 1-click generation of meetings
- Link tokens expire after 60 mins (for access control)

7.5 Developer & Partner API (Private Beta)

A REST-based developer API is planned for agency partners and enterprise clients to:

- Automate project brief submissions
- Query talent availability
- Pull ledger snapshots for audits

Authentication:

- API keys + OAuth2 with scoped tokens

Rate Limits:

- 100 RPS default
- Higher tiers available for whitelisted clients

Monitoring:

- Usage analytics via partner dashboard
- Alerts for error rate > 2% sustained over 5 mins

8. Compliance & Regulatory Matrix

WKforce operates in highly regulated regions (Nigeria, US, EU) and is subject to a mix of data protection, financial, labor, and digital service laws. This section outlines the legal landscape and embedded compliance protocols.

8.1 Regional Regulatory Summary

Region	Frameworks & Laws	Requirements
Nigeria	NDPA, CBN PSSP	Consent logging, KYC archival, Naira payments compliance
EU (Netherlands)	GDPR, WAADI, PSD2	Data portability, agent registration, treasury licensing
USA (Delaware)	IRS, DE Licenses	Worker classification, escrow tax logging, contractor forms (1099)

8.2 Data Protection and Privacy Laws

General Data Protection Regulation (GDPR)

- Enforced across EU clients
- Features:
    - Right to access, correct, delete personal data
    - Consent on PII collection
    - Breach notification in <72 hours

Nigeria Data Protection Act (NDPA)

- Covers all Nigerian users
- Required:
    - KYC documents retained for 10 years

    - Consent screen with timestamp
    - Report of data sharing with external processors

Data Subject Rights

- Request portal integrated into user dashboard
- Admin tooling for compliance officers to validate & export reports

8.3 Escrow and Payment Law Compliance

PSD2 (EU)

- Partner PSP (Adyen) used to avoid direct license
- Multi-factor authentication enforced
- Real-time fraud detection services integrated

CBN Payment Service Provider (PSSP) Rules

- Paystack partnership fulfills local requirement
- Local currency ledger separation enforced
- Central Bank filing supported monthly

IRS Filings and US Tax

- US contractors prompted to complete W9/1099 intake
- Income report generated each January
- Escrow ledger linked to employer tax profile

8.4 Employment & Labor Compliance

WAADI (NL)

- Mandates registration of employment platforms
- WKforce initiates declaration for agent providers
- Auditable agent contracts and hourly tracking retained

Worker Classification (IRS, EU Labor Law)

- Role types (freelancer, agent, team) reviewed against local tests
- Automated classification wizard for employer briefs
- Legal team approval queue for edge cases

8.5 Auditability & Legal Artefacts

System Artefacts Generated:

- Signed contracts with e-sign hash
- Consent logs per PII record
- Tax reports (IRS format + EU XML schema)
- Platform fee receipts

Audit Infrastructure:

- Immutable logs (WORM storage)
- Redundant backup + cryptographic checksums
- Access logs per admin session

---

9. Dependencies & Integrations

This section outlines the critical third-party platforms, systems, and services that WKforce depends on to deliver core features, scalability, and regulatory compliance. Each dependency is categorized based on its functional role in the platform.

9.1 Payment Infrastructure

A. Stripe (US Market)

- Used for escrow funding and disbursement to US-based freelancers
- Integrated via REST API with OAuth2 key rotation
- Escrow wallet creation, transaction history, refund APIs

B. Paystack (Nigeria)

- Used for Naira-based funding, CBN-compliant disbursement
- Ensures FX compliance and local partner validation

C. Adyen (EU Markets)

- Supports SEPA/IBAN transactions and multi-currency wallets
- FX APIs help convert between USD, EUR, GBP (pilot only)

Integration Strategy:

- Abstracted payment interface for switching providers per region
- Fallback mechanisms if PSP experiences downtime

9.2 AI & LLM Dependencies

A. OpenAI / Anthropic APIs

- For role card generation, chat assistant, and project summaries
- Integrated via secure API proxy
- Cost monitored to maintain budget ceiling (<20% of platform COGS)

B. Qdrant (Vector DB)

- Handles job and profile embeddings for semantic search
- Self-hosted to optimize for compliance and control
- Synchronous API used in match engine microservice

9.3 Cloud Infrastructure

A. Google Cloud Platform (GCP)

- Cloud Run: Containerized deployment
- GCS: File/document storage with signed URLs
- Pub/Sub: Internal service-to-service communication

B. GitHub

- Code repository, GitHub Actions used for CI/CD pipelines
- Pull request compliance, branch protection enforced

9.4 External Workflow & Comms Tools

A. Slack, Telegram

- Notifications and job updates for talent and managers

B. ClickUp, Jira, Trello (Planned Phase 2)

- API integration to sync boards and task states
- OAuth2 scoped access for client-linked accounts

9.5 Identity & Compliance Services

A. Veriff or Ondato (TBD)

- KYC fallback for high-risk registrations

- Document scanning and facial verification workflows

B. DocuStub

- Contract signature lifecycle tracking (draft → signed → archived)

---

10. Quality Assurance & Testing

This section outlines the comprehensive quality assurance (QA) strategy for WKforce, including testing levels, coverage targets, QA tooling, test automation plans, and defect tracking methodology. Ensuring a robust QA process is essential given WKforce's reliance on AI workflows, payments, compliance, and external integrations.

10.1 Testing Strategy Overview

WKforce will follow a multi-layered QA approach:

- **Unit Testing:** Validate core business logic in isolation
- **Integration Testing:** Ensure interaction between components (e.g., API ↔ DB, agent ↔ vector DB)
- **End-to-End (E2E) Testing:** Simulate user journeys across UI, API, and third-party services
- **Performance Testing:** Validate responsiveness under load (API RPS, UI latency)
- **Security Testing:** Test for vulnerabilities (XSS, CSRF, SQLi, RBAC bypass)
- **Accessibility Testing:** Ensure WCAG 2.1 AA compliance for users with disabilities

10.2 Test Automation Framework

Tools:

- **Unit:** PyTest, Jest (React), Go Test
- **Integration:** Postman/Newman, PyTest + Docker Compose

- **E2E:** Playwright + BrowserStack (cross-browser)
- **Performance:** K6, Locust
- **Security:** OWASP ZAP, Snyk, GitHub Advanced Security

CI/CD Integration:

- GitHub Actions runs tests on PR creation and merge
- Code coverage reports stored per branch
- Linting, static analysis (SonarQube) as pre-commit hooks

10.3 Test Coverage Goals

Layer	Minimum Coverage Target
Unit Tests	85%+
Integration Tests	80%
E2E Tests	70%
Security Scan Pass	100% critical issues
Accessibility Audit	100% for critical paths

10.4 Sample Test Cases

A. Unit Test (Job Matching)

- Input: Role vector + skill filters
- Expected: List of top 10 ranked profiles with cosine > 0.75
- Result: Assert correct sort order + filter compliance

B. Integration Test (Contract Signing Flow)

- Flow: Brief → Offer → DocuStub sign → Ledger entry

- Expected: Ledger entry created after DocuStub webhook confirms signature

C. E2E Test (User Registration)

- Flow: User registers, uploads KYC, waits for approval
- Expected: Notification sent + access to dashboard on approval

D. Performance Test (Escrow API)

- Load: 200 concurrent escrow POST requests
- Expected: p95 latency < 300ms, error rate < 0.5%

10.5 Defect Tracking & Release Criteria

Bug Tracking:

- Jira integrated with GitHub commits
- Severity labels (Blocker, Critical, Major, Minor, Trivial)
- SLA-based triage (Blockers fixed <24h, Criticals <48h)

Release Readiness Checklist:

- All critical tests passing
- Regression suite green
- No open Blocker or Critical bugs
- Deployment approved via staging sign-off

Beta Testing Program:

- Pilot clients (DCarbon, Fundcirkle, WINNIIO) to run scripted test plans
- Feedback captured via Intercom + Notion QA tracker

10.6 Accessibility and Usability Testing

- WCAG 2.1 AA conformance tools (axe-core, Lighthouse)
- Manual keyboard navigation testing
- Screen reader tests (NVDA, VoiceOver)

Inclusive Design Audits:

- High-contrast UI mode
- Font scaling for low vision
- ARIA landmarks and labels across UI

---

11. Risk Management

This section outlines the key risks to the successful delivery, performance, and operation of the WKforce platform. It includes risk identification, impact assessment, likelihood analysis, and mitigation strategies. A structured approach to risk management is essential given the platform's regulatory exposure, AI integration, and global transaction model.

11.1 Risk Management Framework

WKforce follows a continuous risk management cycle:

1. **Identification** – Through sprint reviews, compliance audits, and incident reports
2. **Assessment** – Scored using an Impact × Likelihood matrix
3. **Mitigation Planning** – Owned by engineering, security, or compliance leads
4. **Monitoring** – Dashboard alerts, regular audits, and service metrics
5. **Response** – SOPs for response, rollback, communication, and recovery

11.2 Risk Register

ID	Risk Description	Severity	Likelihood	Mitigation Strategy
R1	AI hallucination in role spec generation	High	High	Retrieval-augmented generation + human approval loop
R2	Escrow/payment disputes	Medium	Medium	Clear contract terms, 24h support SLA, automated ledger trail
R3	Robot/API agent downtime	High	Medium	Fallback agents, health checks, circuit breakers
R4	Infrastructure outage (Cloud Run)	High	Medium	Multi-zone deployment, auto-scaling + rollback strategy
R5	Data breach or unauthorized access	Critical	Low	Encryption at rest, RBAC, pen testing, continuous monitoring
R6	Compliance violation (e.g., GDPR, NDPA)	High	Low	Auto-consent logging, legal review of workflows, data exports
R7	Misclassification of worker status	Medium	Medium	Legal templates, automated classification wizard
R8	AI cost overruns due to API usage spikes	Medium	Medium	Usage monitoring, credit caps, fallback models
R9	Integration failure with external APIs	Medium	Medium	Retry logic, manual override, status alerts
R10	SLA breach during scale events	High	Medium	Load testing, scaling thresholds, service degradation protocols

11.3 Risk Scoring Criteria

Severity Levels:

- **Critical:** Irreparable brand/legal impact or data loss
- **High:** Major functional downtime or regulatory breach
- **Medium:** Limited service interruption or internal-only issue
- **Low:** Minor annoyance or cosmetic issue

Likelihood Levels:

- **High:** Occurs >1/month
- **Medium:** Occurs ~1/quarter
- **Low:** Rare or no occurrence, but theoretically possible

11.4 Mitigation Strategies

Security:

- Periodic pen-testing with third-party vendors
- Zero-trust architecture and secrets rotation
- GCP IAM audits and anomaly detection

AI Safety:

- Prompt engineering to constrain completions
- Human-in-the-loop review required for legal/contract outputs
- Confidence scoring and embedded disclaimers

Operational Resilience:

- Health checks and restart-on-failure containers
- Real-time alerting via Prometheus/Grafana
- Multi-cloud backup plan under review

Compliance:

- Annual GDPR and NDPA compliance audits
- Change logs on every PII data field
- Legal monitoring of evolving WAADI/PSD2 rules

11.5 Incident Response Planning

Response Team Roles:

- **Incident Commander:** Owns resolution and communication
- **Scribe:** Documents events and logs during incident
- **Resolver:** Fixes issue at service or code level
- **Communicator:** Informs stakeholders, clients, regulators

Escalation Matrix:

- Tiered support levels (L1 → L3)
- Escalation via Slack, PagerDuty, and email (failover channels)
- Incident review (post-mortem) held within 48 hours

Recovery Timelines:

- **P0 (Critical):** RTO < 1 hour, RPO < 15 minutes
- **P1 (High):** RTO < 4 hours, RPO < 1 hour
- **P2 (Medium):** RTO < 12 hours, RPO < 4 hours

---

12. Implementation Roadmap

This section outlines the phased rollout plan for the WKforce platform. The implementation roadmap breaks development into manageable, time-bound sprints, ensuring that each milestone delivers tangible value. The roadmap also aligns product delivery with pilot client feedback and regulatory compliance deadlines.

12.1 Roadmap Structure

The roadmap is divided into:

- **Pre-MVP Planning & Foundation (June–July 2025)**
- **MVP Feature Sprints (August–November 2025)**
- **Post-MVP Expansion & Scale-Up (December 2025–Q2 2026)**

Each sprint has specific exit criteria, dependency mapping, and deployment deliverables.

12.2 Sprint Breakdown

Sprint 0: Infrastructure & Foundations

- **Duration:** June 2025
- **Goals:** Set up GCP environment, CI/CD, secrets manager, skeleton UI
- **Key Deliverables:**
    - GitHub project + workflow templates
    - Terraform IaC for Cloud Run, GCS, and AlloyDB
    - Base React frontend scaffolding

Sprint 1: Worker On-Ramp (KYC Engine)

- **Duration:** August 2025
- **Goals:** Register users and teams, upload KYC, admin approval flow
- **Exit Criteria:**
    - 95% KYC auto-approval within 5 minutes
    - Admin panel can override, approve, or reject

Sprint 2: Job Post + Match

- **Duration:** September 2025
- **Goals:** Let employers post briefs and return matched talent/agents

- **Exit Criteria:**
    - LLM-generated role specs render on UI
    - Top-20 match list appears <150ms
    - Embedded prompt/response stored

Sprint 3: Contract + Escrow POC

- **Duration:** October 2025
- **Goals:** Let employers send offers, sign contracts, fund escrow
- **Exit Criteria:**
    - DocuStub signature confirmed
    - Stripe/Paystack test-mode transaction logs created
    - Ledger reflects escrow split

Sprint 4: Delivery Hub Alpha

- **Duration:** November 2025
- **Goals:** Deliverables, file vault, chat, and agent webhooks
- **Exit Criteria:**
    - Kanban board renders 3+ workflows
    - File vault supports 100MB upload
    - AI agent posts status to board

12.3 Pilot Client Milestones

Client	Feature Focus	Feedback Date

DCarbon	Role brief → delivery tracking	Aug 2025
Fundcirkle	Contract & compliance testing	Sep 2025
WINNIIO	AI assistant feedback loop	Oct 2025

12.4 Engineering Dependencies

Dependency	Needed By	Mitigation Plan
Qdrant Deployment	Sprint 2	Self-hosted test cluster in staging
Stripe Integration	Sprint 3	Use test-mode with real ledger logic
Agent Webhooks	Sprint 4	Use stubbed endpoints with auth tokens

12.5 Release Criteria

- All sprint exit criteria met
- Security and performance tests passed
- Regulatory sign-off from compliance team
- Pilot client greenlight from at least two testers

12.6 Post-MVP Expansion (Q1 2026+)

Key upcoming initiatives include:

- Multi-currency wallet (EUR/GBP via Adyen)
- RPA integrations (UiPath, Zapier bots)
- ML-based match scoring replacing rule-based filters
- Robotic workforce beta (Phase 3 experimental)
- API for enterprise clients and team sourcing

13. Maintenance & Support

This section details the policies, procedures, and infrastructure that support ongoing system uptime, user assistance, regulatory compliance, and incident management.

13.1 Service Level Agreements (SLAs)

Service Tier	Uptime SLA	Response Time	Resolution Time
Core API	99.5%	< 2 hours	< 8 hours
Escrow Engine	99.9%	< 1 hour	< 4 hours
AI Chat	99.0%	< 4 hours	< 12 hours

13.2 Support Channels

- **Tier 1:** Self-service Knowledge Base
- **Tier 2:** Live Chat via Intercom (business hours)
- **Tier 3:** Email ticketing with SLA enforcement
- **Tier 4:** Dedicated account managers for enterprise clients

Emergency Escalation:

- PagerDuty integration for P0 incidents

- Support contact made available in platform footer

13.3 Monitoring & Incident Reporting

- Real-time monitoring via GCP Cloud Monitoring, Prometheus, and Grafana
- Alerting thresholds for API latency, CPU load, disk IO, and failed requests
- Daily health checks and synthetic user journeys for major workflows

Incident Log Lifecycle:

- Critical incidents logged in Notion + Jira
- Root cause analysis (RCA) within 48 hours
- Transparent updates posted to client status page

13.4 Logging & Audit Trail

- Centralized structured logging per microservice
- Correlation IDs for tracing across APIs, chat, agent, and payment services

Audit logs include:

- KYC/PII access events
- Escrow transactions
- Contract signature status
- Admin overrides or deletions

Storage & Retention:

- 30-day hot logs
- 1-year searchable log archive

- 7-year compliance archive (immutable)

13.5 Compliance Maintenance

- Quarterly reviews of NDPA, GDPR, and PSD2 implications
- Auto-expiry policy checks (e.g., consent TTL)
- Compliance dashboard for legal team
- Annual 3rd-party audits (security + legal)

13.6 System Upgrades & Patch Policy

- Monthly scheduled maintenance window
- Emergency patches applied within 24 hours of CVE disclosure
- Versioning tracked via GitHub releases
- Blue/green deployment with rollback option

13.7 Business Continuity

- Nightly database backups to cold storage (3 regional copies)
- Disaster recovery plan tested bi-annually
- Legal escrow of source code
- Secondary contact registry for client-side continuity

---

14. Appendices

14.1 Glossary of Terms

- **Agent Socket:** Interface for bots/agents to interact with workflows
- **Brief:** Natural-language description of project need
- **Contract Ledger:** Immutable record of payment events
- **KYC:** Know Your Customer process
- **RAG:** Retrieval-Augmented Generation (AI strategy)
- **Role Card:** Structured project role specification
- **Spec Wizard:** Interface for generating specs from prompts

14.2 UI Mockups (Descriptions)

- Employer Wizard → step-by-step brief intake
- Matching Dashboard → ranked cards + filters
- Admin Console → approve KYC, review contracts
- Delivery Hub → Kanban + file vault + chat
- AI Chat Assistant → sidebar interface

14.3 API Reference (Outline)

- `POST /auth/login`\n- `GET /projects/{id}`\n- `POST /contracts`\n- `GET /match/{role_id}`\n- `POST /escrow/fund`

14.4 Database Schema

- Tables: `users`, `teams`, `projects`, `contracts`, `escrows`, `roles`, `ledger_entries`, `messages`, `files`
- JSONB fields for flexible roles and milestones
- Vector data indexed in Qdrant

14.5 Legal Templates

- SoW Template
- Freelance Agreement
- NDPA/GDPR Consent Language
- IRS 1099 Contractor File
- EU Platform Terms Addendum

14.6 Credit Pricing Bands

Credits	Unit Cost	Bonus	Validity
100	$1.00	0%	12 mo
500	$0.95	5%	12 mo
1000	$0.90	10%	18 mo
5000	$0.85	15%	24 mo

14.7 Pilot Feedback Log (Assumptions/Examples)

Client	Feedback Summary	Date
DCarbon	Wanted better dashboard filters	Aug 2025
Fundcirkle	Escrow release logic felt too complex	Sep 2025
WINNIIO	Wanted AI chat responses rated by humans	Oct 2025