SDG BLOCKCHAIN ACCELERATOR

Technical Architecture Document

1. Project Information

- **Project Name:** *AFRIKABAL*
- **Challenge & UNDP Office:** *Malaysia*
- **Document Version:** 1.0.0

2. Overview

This document describes the production-grade technical design of **AFRIKABAL**, a supply-chain coordination platform built to run on the Cardano blockchain. It replaces all placeholders with final text and embeds the diagrams required for review by UNDP and Cardano assessors.

2.1 Executive Overview

AFRIKABAL digitizes purchase orders, lots, handovers, and proof-of-delivery across buyers, producers, cooperatives, logistics providers, and partners. The platform uses Cardano's eUTxO model and Aiken smart contracts to achieve auditable state transitions with minimal on-chain data (hash pointers to evidence). Off-chain services handle identity, policy, orchestration, and reporting. Farmers can interact using **USSD**; buyers and farmers have a **mobile app**; all roles have a **web app**.

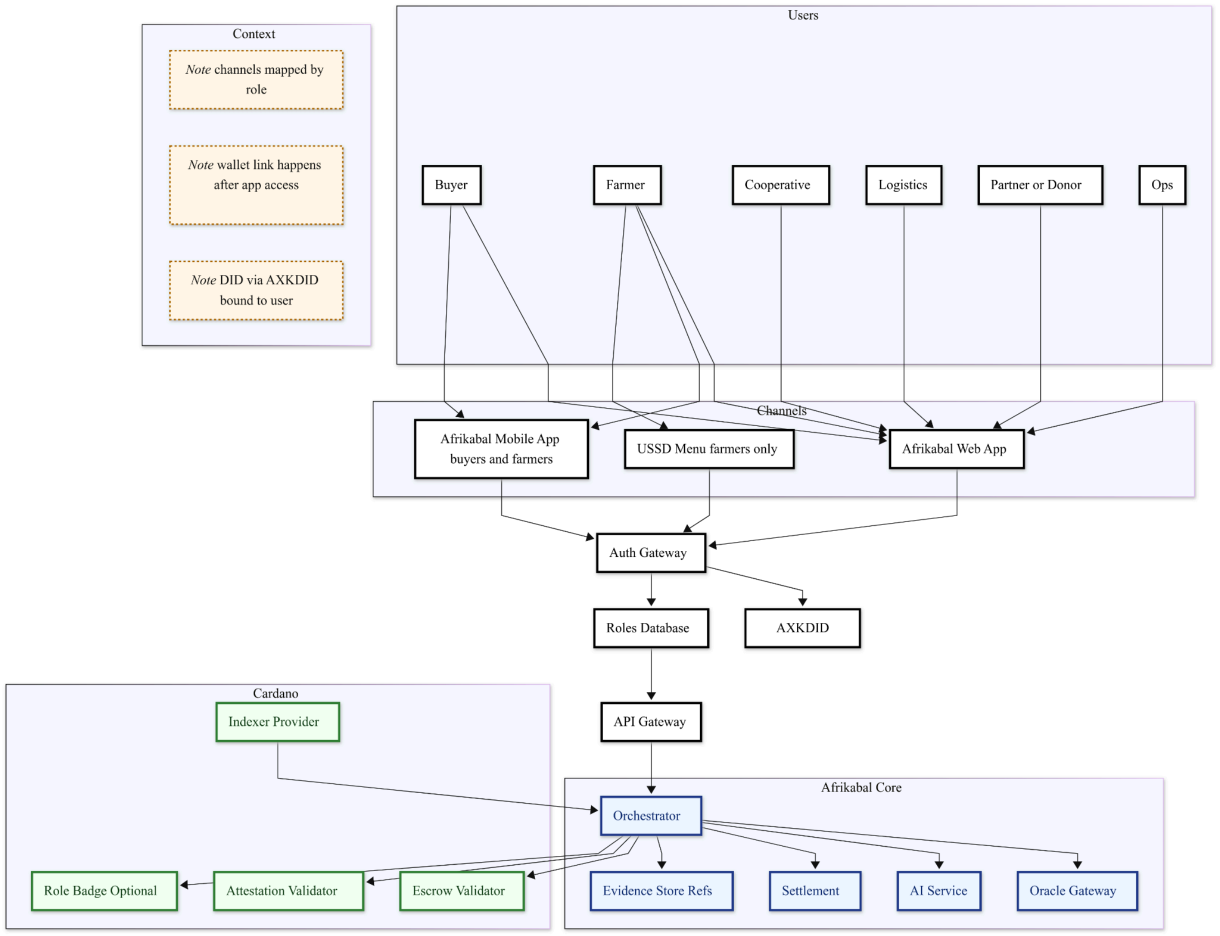
- Business value: faster, more transparent trade with proof-backed settlements and SDG-aligned impact metrics.
- Security posture: least-privilege identities, signed transactions, strict validator checks, and encrypted evidence with IPFS pinning.
- Deployment: Preview/Preprod for PoC and soak tests; controlled mainnet rollout after audit and operational readiness checks.

2.2 Users & Channels

- Buyers — web & mobile
- Producers (farmers) — web, mobile & USSD
- Cooperatives — web
- Logistics providers — web
- Partners/Donors — web
- Ops (support/administration) — web

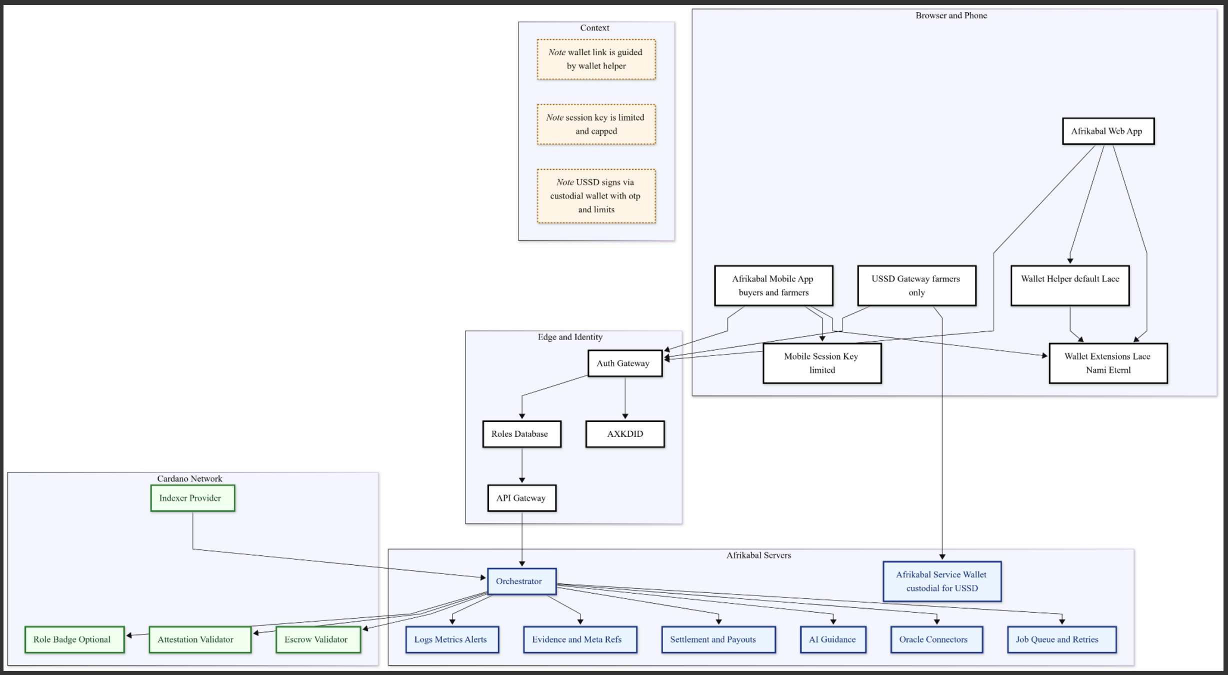
3. System Architecture Diagram

```
Figure 1: System Context: Users, channels, identity edge, orchestrator, and Cardano network. (Who talks to what)
```



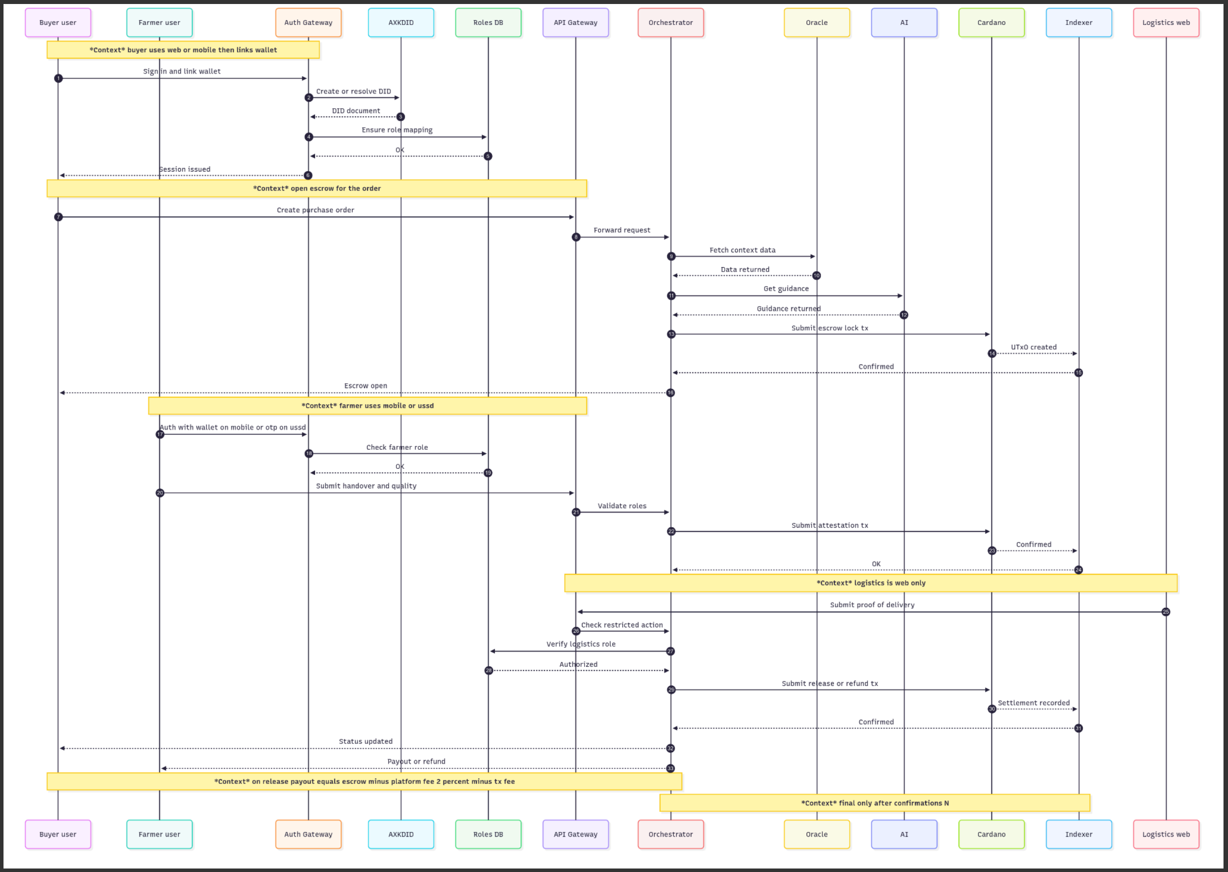
Multi-channel system context with role-based access, identity, core services, and Cardano validators. ***Buyers use Web or Mobile. Farmers use USSD and Mobile; others use Web. Identity and roles are enforced before core services write to Cardano.***

Figure 2: Runtime & Deployment: What runs and where (browser/phone, edge & identity, Afrikabal core, Cardano).


Context
Note wallet link is guided by wallet helper
Note session key is limited and capped
Note USSD signs via custodial wallet with otp and limits
Browser and Phone
Afrikabal Web App
Afrikabal Mobile App buyers and farmers
USSD Gateway farmers only
Wallet Helper default Lace
Mobile Session Key limited
Wallet Extensions Lace Nami Eternl
Edge and Identity
Auth Gateway
Roles Database
AXKDID
API Gateway
Cardano Network
Indexer Provider
Role Badge Optional
Attestation Validator
Escrow Validator
Afrikabal Servers
Orchestrator
Afrikabal Service Wallet custodial for USSD
Logs Metrics Alerts
Evidence and Meta Refs
Settlement and Payouts
AI Guidance
Oracle Connectors
Job Queue and Retries

Container view showing Web, Mobile, and farmer-only USSD, identity layer, Afrikabal servers, and Cardano validators with indexer.
Web and Mobile link a wallet after app access; Mobile can use a limited session key for small actions. USSD is farmers only and signs via a custodial service wallet with OTP, limits, and audit.

Figure 3: Order→Settlement sequence: wallet/DID link, escrow open, attestations, release/refund with confirmations N. (Flow Summary)



Multi-channel sequence showing the order journey with context notes for each step.
Buyer starts on Web or Mobile. Farmer acts via Mobile or USSD. Logistics is Web only. Same validators for everyone, with a 2 percent platform fee on release and confirmations N before final.

4. Blockchain Design

4.1 Validators (Aiken)

- Escrow Validator — locks buyer funds; releases to producer after attested proof-of-delivery or refunds on timeout/dispute.
- Attestation Validator — records signed supply-chain events (handover, quality checks, PoD) with evidence CIDs & checksums.
- Role Badge — native token/badge required for restricted actions (e.g., logistics PoD); ‘present-and-return’ enforced.

4.2 Datum & Redeemer

- **EscrowDatum**: order_id; buyer_pkh; seller_pkh; amount_lovelace; fee_bps=200 (2%); deadline_slot; required_roles; evidence_cids.
- **EscrowRedeemer**: Open | Release | Refund | Dispute | Close.
- **AttestationDatum**: att_id; order_id; actor_role; evidence_cid; checksum; ts.
- **AttestationRedeemer:** Record | Amend | Void.

4.3 UTxO Model Usage

Each escrow state is a single UTxO; every transition consumes the previous UTxO and creates the next. Reference inputs supply oracle facts or role badges. Minimal data is stored on-chain; evidence lives off-chain with hash/CID references for auditability.

4.4 Native Assets

PoC: **AFRIKABALPROOF** test asset was minted and burned on *Preview* to validate the minting policy and UTxO handling.

Production: the native token is **AXKCoin** for platform utility/fees and pilot settlement experiments; per-role badges remain optional (single-issuer, present-and-return). The **AXKCoin** policy is a controlled-mint multisig with time-locks; no arbitrary inflation.

4.5 Security in Validators

- **Multi-sig as needed;** strict role checks; time-locks for refunds; present-and-return semantics for badges.
- **Datums** carry only identifiers and hashes; no plaintext PII or large blobs on-chain.

5. Data Flow & Transaction Lifecycle

1. Create order: Web/Mobile → API → Orchestrator; fetch oracles; AI guidance; build escrow lock tx; buyer signs; submit; wait **N** confirmations.

2. Handover & quality: Farmer via Mobile/USSD; evidence SHA-256 hashed and pinned to IPFS; Attestation tx references CID.

3. Proof of delivery: Logistics via Web; role verified; Attestation tx submitted.

4. Settlement: Release or Refund. Payout = escrow − **2% platform fee** − tx fee. Confirmations **N** observed before finalizing status.

5. Reporting: indexer streams into dashboards; evidence CIDs are fully auditable.

6. Off-chain Components

- Orchestrator (TypeScript + Lucid) — builds/signs/ submits transactions; enforces policy & sequencing; retries via job queue.

- Identity: AXKDID service for DID creation/binding; Auth Gateway for OAuth/JWT; Roles DB for fine-grained authorization.

- Channels: Web & Mobile are non-custodial (wallet extensions); USSD is custodial with OTP and per-role/amount limits.

- Oracles: weather, reference prices, compliance lists; referenced in txs via reference inputs.

- Evidence Store: object storage + IPFS pinning; only CIDs/hashes on-chain; sensitive files encrypted before pinning.

- Indexer/Provider: Blockfrost for Preview/Preprod; Ogmios/Kupo added for HA in production.

- Observability: structured logs, metrics (p95 latency, tx success), full audit trail of identity and tx intents.

7. Sandbox/Testnet Results

Timestamp (UTC)	Operation	Network	Tx Hash	Notes
2025-08-26T00:37:53	Submit simple tx	Preview	32dd4dfc4e3c44417d212e73b135f23b2265e38fd38600eae41229f9ad733ab	Connectivity via Blockfrost
2025-08-26T00:40:37	Mint AFRIKABALPROOF	Preview	20b44b056f1537bfad20b4857a5f9ce05c679a3eed54c82d0ca6459de6cc17ba	Policy via key-hash
2025-08-26T00:44:22	Mint AFRIKABALPROOF	Preview	feed2ef46e9dc926d7556d81ef981eb454ed2b1fb1dabea2e67ccc95686215a1	Second mint multi-UTxO
2025-09-03T14:27:17	Burn AFRIKABALPROOF	Preview	c8038ef332543dd72c5309cbe444ef288a6edb3d8bdc778a8137c9c6871c7178	Fix: collect token UTxO before burn

Primary address observed during tests:
addr_test1vrlazhaxp3tqmddw6wg0rn593zgl5z6vfda7982evs2qwdg6mvhm2

8. Tools and Environments Used

- Aiken v1.1.17 (stdlib v2.2.0) — validators
- Node.js 24, TypeScript 5, Yarn 4, Lucid 0.10.x — off-chain services and scripts
- Providers: Blockfrost (Preview/Preprod); Ogmios/Kupo to be added for HA in prod

- CI/CD: Cloud Build; provenance (SLSA); automated tests and dry-runs
- Quality: aiken check/build; unit/property tests; fault injection for retries

9. Security & Compliance

- Identity & auth: DID + OAuth/JWT; per-role limits; mobile session keys with tight TTL and scopes.
- Keys: service wallet keys in KMS/HSM; rotation procedures; PAW for ops.
- Data: encrypt sensitive evidence; only hashes/CIDs on-chain; DSR workflows for redaction via re-pinning.
- Network: WAF/Cloud Armor; private subnets & NAT; least privilege IAM; audit logging.
- Validators: formal review and external audit prior to mainnet.

10. Cloud Infrastructure (GCP) & IPFS

10.1 GCP

VM-based Infrastructure Overview

- VMs; one public edge, private core. Zero-downtime via Nginx upstream drains and PM2 rolling restarts.
- Only required ports exposed; everything else denied.

Inventory

VM Name	Role	vCPU	RAM	Disk	OS	Open Ports	Notes
axk-glass-1	Public edge: Nginx + static web	4	8 GiB	100 GiB NVMe SSD	Ubuntu 22.04 LTS	80, 443	TLS (LE), reverse proxy to core
axk-core-1	API/Orchestrator (TS+Lucid)	8	16 GiB	200 GiB NVMe SSD	Ubuntu 22.04 LTS	3000 (internal)	PM2 cluster; /healthz & /readyz
axk-core-2	API/Orchestrator (HA)	8	16 GiB	200 GiB NVMe SSD	Ubuntu 22.04 LTS	3000 (internal)	Blue/green with glass drain
axk-queue-1	Redis	4	8 GiB	100 GiB SSD	Ubuntu 22.04 LTS	6379 (internal)	Idempotency, rate limits, jobs
axk-db-1	MariaDB 10.11 Primary (InnoDB, GTID, semi-sync)	8	32 GiB	500 GiB SSD (provisioned IOPS)	Ubuntu 22.04 LTS	3306 (internal)	Nightly dump + binlog; buffer pool ~24 GiB; 7d retention
axk-observe-1	Monitoring (Grafana+Prometheus+Loki)	4	8 GiB	200 GiB SSD	Ubuntu 22.04 LTS	3000,9090,3100 (internal)	Node exporter on all VMs; long-retention logs

axk-ipfs-1	IPFS pin node/gateway	4	16 GiB	1 TB SSD	Ubuntu 22.04 LTS	5001,8080 (internal)	CID pinning; gateway behind glass if needed
axk-db-2	MariaDB 10.11 Replica (InnoDB)	8	32 GiB	500 GiB SSD (provisioned IOPS)	Ubuntu 22.04 LTS	3306 (internal)	Semi-sync replica; read traffic; HA candidate
axk-db-3	MariaDB 10.11 Replica (InnoDB)	8	32 GiB	500 GiB SSD (provisioned IOPS)	Ubuntu 22.04 LTS	3306 (internal)	Semi-sync replica; HA candidate
axk-proxy-1	ProxySQL (read/write split)	4	8 GiB	50 GiB SSD	Ubuntu 22.04 LTS	6032,6033 (internal)	Writes→primary, reads→replicas; admin 6032

Security & Access Controls

- Private subnet for all core VMs; only axk-glass-1 has a public IP.
- UFW default deny; allowlist: 80/443 on axk-glass-1; 22 SSH restricted to admin IPs; no SSH on core/db from the internet.
- SSH: key-only, no root login, fail2ban; sudo for named ops users; auditd enabled.
- TLS: Let's Encrypt on axk-glass-1; upstream to core over private IP.
- Secrets: .env files readable only by app user; service wallet keys encrypted at rest; monthly rotation.

Deployment & Zero-Downtime

- APIs: PM2 cluster mode with rolling `pm2 reload` (no dropped connections).

- Blue/Green: remove axk-core-1 from Nginx upstream (drain), deploy & warm, re-add; repeat for axk-core-2.
- Frontend: atomic symlink swap on axk-glass-1 (`/var/www/afrikabal/current`), cache-busted assets.
- DB: schema migrations via controlled steps in maintenance window; backward-compatible changes first.

Health Checks & Monitoring

- L7: `/healthz` (process alive) and `/readyz` (DB + provider check) on core; Nginx only routes to ready backends.
- System: node_exporter + Grafana; alerts CPU>80% (5m), RAM>85%, disk>85%, 5xx spikes, tx submit failures.
- Uptime: HTTPS checks on 443; cert expiry alarms; IPFS pin lag alerts (pins >2m).

Backups & Restore

- MariaDB nightly dumps + binary log (binlog) archiving; 7d retention; monthly restore test in staging.
- Weekly VM snapshots: axk-core-*, axk-db-1; logs archived to object storage (90d lifecycle).
- Config-as-code (nginx, systemd, PM2) in git; VM rebuild from base + scripts < 60 minutes.

Stress & Load Testing

- API: k6 scenarios (Order→Attestation→Settlement); 100–300 VUs; targets p95 < 300ms (API path), error < 0.5%.**High Availability (Auto‑Heal) & DB Replication**

- Tx Orchestrator: burst submit respecting provider limits; backoff verified; queue drains without duplicates.**Application tier — auto‑healing replicas**

- Frontend: autocannon 300–500 rps static; Nginx CPU < 60%; caching headers validated.axk-core-1 and axk-core-2 run behind axk-glass-1 (Nginx).

- **Simple Runbook (High Level)**Instance Group pattern: instance template + health check; if a core VM is unhealthy, a new one is auto‑created from the template.

- Core down → glass removes unhealthy backend; ssh; `pm2 logs` then restart; re-add when `/readyz` OK.Rolling deploys: drain from Nginx upstream → deploy to one core → warm‑up until /readyz OK → re‑add → repeat on the other core. Zero‑downtime.

- Provider outage → switch to fallback; increase tx fee if mempool congestion; watch confirmations N.State is stateless on cores; session/rate/idempotency lives in Redis (axk‑queue‑1) and data in MariaDB, so instances can be replaced at any time.

- DB incident → restore latest dump + binlog; run read-only until catch-up done.**MariaDB primary‑replica (semi‑sync) with automatic failover**

10.2 IPFS

- Topology: 1 primary (axk-db-1) + 2 replicas (axk-db-2, axk-db-3). InnoDB, GTID enabled.

- Evidence stored off-chain; encrypted if sensitive; pinned to IPFS; datum keeps CID + SHA-256 checksum.Replication: semi-synchronous; primary waits for at least one replica ACK → reduces data loss on failover (target RPO < 15s).

- Access via signed URLs and rate limits; provider + self-hosted gateway for reliability.Router: ProxySQL on axk-core-* (or dedicated axk-proxy-1) directs writes to primary and reads to replicas; stickiness by session where needed. (proxy 6033, admin 6032).

10.3 SLOs

- Failover: Orchestrator (or MariaDB Replication Manager) promotes the most up-to-date replica; ProxySQL updates backends automatically. Target RTO < 2 min.

- Time-to-First-Tx ≤ 1 business dayBackups: nightly logical dumps + continuous binlog; point-in-time recovery using GTID + binlog sequence.

- Tx submit p95 ≤ 3s; first confirmation ≤ 30s on testnets**Health checks & failure drills**

- Evidence pin success ≥ 99.5% within 2 minutesCores: `/readyz` checks DB, Redis, and provider; Nginx only routes to ready backends.

- API uptime ≥ 99.9%DB: replication lag alarms (seconds_behind_master); semi-sync status; automatic failover dry-runs monthly.

11. Roadmap & Next Steps

- Chaos tests: kill one core VM → auto-heal; stop primary DB → orchestrated promotion; verify RTO/RPO targets.
- Complete property-based tests and fuzzing for validators; finalize formal review.

Targets

- Preprod soak testing with realistic load; provider failover drills (Blockfrost ↔ Ogmios/Kupo).Uptime: API 99.9% monthly;

- Privacy and legal review for evidence retention and DSR processes.RTO < 2 minutes for DB primary failover; RPO < 15 seconds under semi-sync;

- Mainnet pilot with selected partners after audit; phased rollout by role and region.Zero-downtime app deploys (no dropped connections).

Instance Sizing & Headroom

- Targets: keep steady-state CPU ~50–60%, RAM < 70%; scale vertically or add a core node if sustained load grows.
- Storage: fast SSD/NVMe; DB volumes on provisioned-IOPS; snapshots weekly; binlog retained for PITR.
- Network: ≥1 Gbps NICs; Nginx tuned for keep-alive and HTTP/2; OS TCP backlog and conntrack sized for bursts.
- Capacity tests: k6 and autocannon scripts validate headroom at 2× expected pilot traffic.

12. Glossary

- **AXKDID** — Afrikabal DID service (creates/binds decentralized identifiers to users/wallets).
- **CID** — Content Identifier for IPFS-pinned evidence.
- **Present-and-return** — restricted action requires badge UTxO present and returned unchanged.
- **Confirmations N** — number of blocks after inclusion before we mark a transaction final in the app.
- **AXKCoin** — Afrikabal native token for utility/fees; controlled-mint policy (multisig + time-locks).