

BORDERLESS KENYA — Payaza × HACKHOUSE NAIROBI HACKATHON

PAZAONE

One balance. Africa without borders.

*“Today, money still knows where the border is.
We want to make the user not care.”*

PROPOSED MVP

Nairobi → Kampala · KES → UGX

INFRASTRUCTURE

Payaza

TEAM

Deborah · Lucy · Amos

Hackathon Product Submission & Build Plan — a proposed MVP, honestly framed.

Money still knows where the border is

East African trade doesn't stop because traders don't know what to buy. **It slows down because moving money across borders** introduces currencies, payment rails, fees and settlement uncertainty.

Multiple currencies

A Nairobi trader paying a Kampala supplier must think in KES and UGX — plus FX — before the trade even starts.

Fragmented rails

Payment initiation, settlement status and confirmation are separate concerns on separate systems.

Friction at the worst moment

Complexity peaks exactly where the transaction should be simplest: paying the supplier.

The product question: Can a cross-border trade payment feel like one local transaction — while Payaza handles the underlying rails?

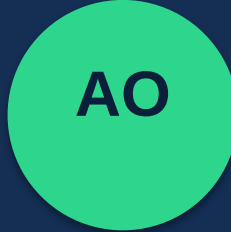
Three people, clearly accountable

**DM****DEBORAH MUTUKU****FRONTEND DEVELOPER**

Owns the PazaOne user interface, transaction flows and the trader experience across MVP screens.

**LO****LUCY OSWO****PRODUCT MANAGER**

Owns product strategy, user research, requirements, business model and product direction.

**AO****AMOS OMWANDO****BACKEND DEVELOPER**

Owns backend architecture, Payaza integration, transaction processing, database and APIs.

Every workstream has a single owner — no ambiguity about who delivers what at the demo.

PazaOne: a Payaza-powered cross-border transaction layer

We propose **PazaOne** — designed to make EAC trade feel like a local payment. A trader funds an account, picks a destination market, selects a supplier, enters an amount, sees the destination-currency equivalent and fee, attaches a trade or invoice reference — and confirms.

Payaza — INFRASTRUCTURE LAYER

Would provide the underlying payment and settlement infrastructure: acceptance, checkout, payouts, settlement, webhooks.

PAZAONE — EXPERIENCE LAYER

Would provide the trader-facing experience: unified balance, transaction workflow, trade records and receipts.

Core principle: PazaOne is not the money. It's a product and accounting abstraction over supported fiat currencies — not a cryptocurrency.

MVP scope: start with one focused corridor — Nairobi → Kampala, KES → UGX. Once the core flow is validated, the same architecture could extend to Tanzania and Rwanda.

Proposed MVP user flow — one honest transaction journey

1

Create account

Trader signs up and completes basic onboarding.

2

Fund via Payaza

Account topped up through the Payaza-supported collection flow.

3

Send to Supplier

Trader selects Uganda and a Kampala supplier.

4

Enter amount

KES amount entered; trade / invoice reference attached.

5

Review quote

Expected UGX amount and applicable fee shown before confirmation.

6

Confirm

Backend creates the payment intent / transaction.

7

Payaza in action

Proposed architecture routes collection, payout and settlement through Payaza rails.

8

Status & receipt

Webhook-driven status updates; receipt and history entries for both sides.

Ambitious, but honest: the prototype demonstrates the proposed Payaza transaction and status flow — nothing claimed that isn't built.

Core product — what the trader sees

Unified balance

One user-facing available balance with currency context shown before every transaction.

Top up

Collect funds through Payaza-supported payment methods.

Send cross-border

Choose Uganda, Tanzania or Rwanda; enter amount; pick recipient.

Quote first

Source amount, destination amount, fees and transaction reference — always before confirmation.

Settlement

Route the transaction through the appropriate Payaza payout / settlement rail.

Trade records

Attach an invoice or trade reference to every payment.

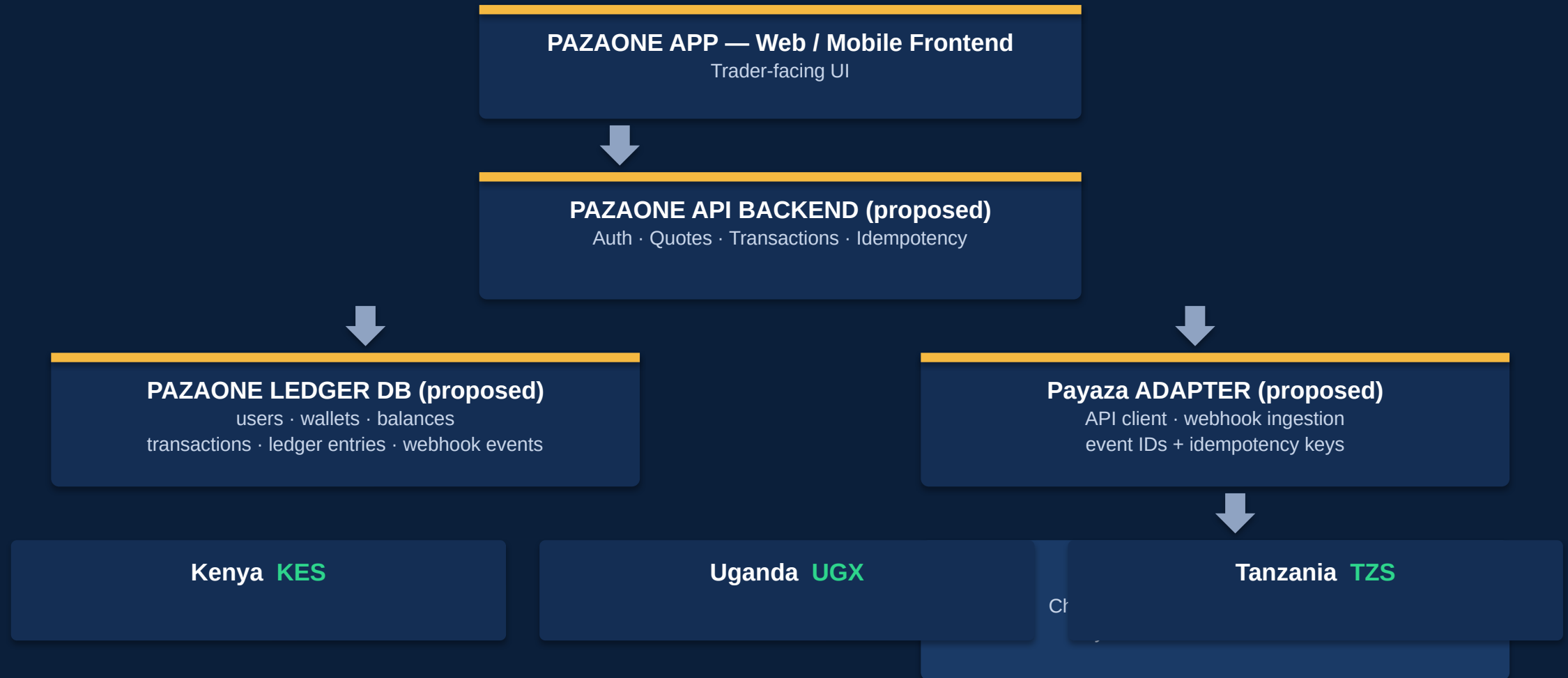
Transaction history

Original amount, destination currency, status and settlement outcome.

Later

Merchant QR / checkout, recurring supplier payments, collections export, more markets.

Payaza underneath, PazaOne on top



Exact endpoints, request schemas and supported corridor capabilities will be confirmed against Payaza's developer documentation and the team's credentials — not hard-coded from assumptions.

Database structure & MVP screens — what the team intends to build

PROPOSED DATA MODEL

users — identity, country, KYC status

wallets / balances — unified display + per-currency balances

transactions — source/dest amounts, fee, corridor, status, Payaza reference, idempotency key

recipients — supplier payout identifiers

trade_orders — invoice / trade references linked to transactions

webhook_events — provider events, dedupe, payloads

ledger_entries — double-entry record of every movement

10 MVP SCREENS

01 Landing / Login — value proposition, sign in, create account

02 Onboarding — name, phone/email, country, verification

03 Home — PazaOne balance + quick actions: Top Up, Send, Pay, Receive

04 Top Up — amount + currency → Payaza checkout → success/failure

05 Send — country → recipient → amount → quote

06 Payment Review — amounts, fee, recipient, trade reference

07 Processing — initiated → processing → settlement → completed

08 Receipt — amounts, reference, timestamp, settlement status

09 Transactions — filters: sent, received, pending, completed

10 Trade Detail — invoice reference, supplier, country, settlement

Monetize the transaction, not the trader's patience

Primary — per-transaction fee

Transparent service fee on cross-border payments, aligned with Payaza's underlying costs and commercial terms.

SME tier — monthly plan

For frequent traders: supplier management, invoice references, transaction history and support.

Future — merchant surfaces

Merchant checkout, exporter collections, merchant APIs and embedded flows.

Important: we will not promise a fixed FX spread or fee until Payaza's live commercial and API terms are confirmed.

Why Payaza — and why this is buildable this weekend

WHY Payaza

- ▶ **Deliberately built on Payaza**, not around a replacement rail.
- ▶ **Documented capabilities:** cards, bank transfers, mobile money, wallets, web checkout, payouts, transfers, sub-accounts, webhooks.
- ▶ **Hackathon calls out** checkout, settlement and multi-currency as the infrastructure to build on.
- ▶ **Fit:** Payaza provides the rails; PazaOne provides the borderless user experience.

FEASIBILITY & PILOT

- ▶ **One corridor:** Nairobi → Kampala, KES → UGX. Done extremely well.
- ▶ **Sandbox-first:** use Payaza test capabilities where available.
- ▶ **Own the ledger** and transaction state machine.
- ▶ **Webhooks** make transaction status real-time and trustworthy.
- ▶ **Pilot** with a small group of Kenyan traders and their existing suppliers; expand to Tanzania and Rwanda after validation.

Risks & mitigations

RISK

MITIGATION

FX / settlement availability differs by corridor



Only expose corridors and currencies confirmed by Payaza — never fabricate rates.

Duplicate callbacks / retries



Provider event IDs + idempotency keys on every transaction.

User confusion about what PazaOne is



Clear messaging: a unified transaction balance abstraction — not a cryptocurrency.

Regulatory complexity



Position the MVP as a Payaza-powered product layer operating within Payaza's supported rails and compliance requirements.

Overbuilding



Demo one end-to-end corridor extremely well.

Two minutes, honestly told

OPENING

“Today, money still knows where the border is. We want to make the user not care.”

PROBLEM

A Kenyan trader buying from Kampala doesn't want to think about rails, FX, settlement networks and status. They want to pay their supplier and know the money arrived.

SOLUTION

We propose PazaOne — one simple balance and one simple experience to top up, send, receive and pay across supported markets. Payaza handles the underlying payment and settlement infrastructure.

DEMO

Fund a PazaOne account via Payaza. Select Uganda, choose the supplier, enter KES, attach the trade reference. PazaOne shows the exact UGX the supplier receives and the KES fee. Confirm — the status flows back through webhooks, and the receipt is in history.

WHY Payaza

We're not building another rail. Payaza already provides acceptance, mobile money, checkout, payouts, transfers, settlement and webhooks. We build the product layer on top.

EXPANSION

Start Nairobi → Kampala. The same architecture extends to Tanzania and Rwanda — then exporters, merchants and other cross-border users.

CLOSE

PazaOne isn't another currency. It removes currency and payment complexity from the user's experience.

ONE-SENTENCE SUBMISSION

PazaOne is a Payaza-powered unified transaction layer that lets East African traders top up, pay and settle cross-border transactions through one simple balance and experience — starting with Nairobi → Kampala.

BUILD PRIORITIES 1. Real Payaza integration 2. One complete transaction 3. Excellent UX 4. Clear trade use case 5. Corridor expansion — only after the core flow works

Payaza moves the money. PazaOne makes the border disappear.

Thank you — Deborah Mutuku · Lucy Oswo · Amos Omwando