# Rasulo

Bank partnership and product proposal · 11 November 2026 · Controlled prototype

**Developed by Hash Developed Projects LTD.**
**Representative:** AHMED ADAM
**Contact:** eng.ahmed@hashdevpro.com · +211911085575
**Proposed launch market:** South Sudan
**Recipient bank:** To be selected

## Executive summary

Rasulo is a financial technology product in development for customers and merchants in South Sudan. We seek a bank partnership to evaluate an approved account, custody and payment model, obtain sandbox access, and define a controlled pilot. The proposed relationship combines Rasulo's customer software with partner-approved financial services and agreed operational responsibilities.

The current product is a working simulation: one bilingual Flutter application, a Spring Boot backend with exact per-currency accounting, a public product website and a restricted interactive prototype. SSP and USD are separate balances. No bank or card provider is connected, no real funds move, and no financial licence, partnership or regulatory approval is claimed.

## Problem and opportunity

The product hypothesis is that people and small merchants would benefit from a clear interface for identifying a recipient, reviewing a payment, understanding its currency and fees, and checking whether it completed. Merchants also need confirmation tied to an authoritative transaction rather than a customer's screen or an informal message.

These needs must be validated with prospective users and the partner bank. This proposal makes no claim about market size, customer adoption, transaction volumes or revenue. South Sudan is the initial focus; the product uses +211 sample phone numbers and English/Arabic interfaces. Juba Arabic campaign wording remains subject to local language and cultural validation.

## Proposed Rasulo solution

One app serves customers and merchants. Its priorities are readable balances, simple send-and-receive journeys, eligible funding sources, explicit fees, honest status, and a traceable receipt. The brand's working campaign is "Rasili bi Rasulo!" Financial screens keep clear terminology; "Rasulo… Khalas!" is reserved for confirmed successful Rasulo payment experiences.

In the prototype, a customer resolves a sample recipient, selects a same-currency source, enters an exact amount and optional note, reviews the server quote, and confirms using an illustrative demo PIN. A receipt appears only after the backend confirms success. An uncertain response is recovered using the original operation key rather than submitting another payment automatically.

## Implemented product and use cases

| Area | Current controlled prototype | Future integration dependency |
|---|---|---|
| Access and sign-in | Restricted reviewer access; fixed demo OTP and sample identity | Real customer identity, account onboarding, secure authentication and authorization |
| Wallets | Independent SSP and USD balances; exact minor-unit API representation | Customer-scoped durable accounts and approved custody model |
| Payments | Simulated Bank → Bank, Bank → Wallet and Wallet → Wallet | Partner-supported payment rails, limits, status and settlement |
| Add money | Simulated Card → Wallet and bank funding; source currency capabilities | Hosted/tokenized provider flow and bank consent/authentication |
| Requests | Actual versioned QR with currency and optional exact amount | Verified customer aliases, signed/expiring requests and approved deep links |
| Receipts and activity | Principal, fee, total, currency, recipient, source, reference and time; filtered history and CSV | Durable statements, support/disputes and reconciliation |
| Merchant confirmation | Scenario UI and sound/voice specification | Merchant isolation, verified events, durable deduplication and any hardware |
| Operations dashboard | Customer home dashboard only; no separate admin service exists | Authorized operational roles, auditability and partner-supported workflows |

Wallet → Bank is prohibited. There is no withdrawal shortcut, currency exchange, biller network, lending, cross-border service or live card entry. Existing legacy identifiers remain compatible; historical evidence and records retain their original meaning.

## Currency and financial integrity

SSP and USD each use two decimal places in this initial implementation. Amounts carry an explicit ISO code and exact minor units as integer strings in JSON. The backend uses exact fixed-scale arithmetic. Principal, fee and total debit are separate values, and fees are restricted to the payment currency. Current demonstration fees are zero.

Illustrative initial balances are SSP 245,000.00 and USD 420.00. They are independent samples, not exchange-rate equivalents. A payment of SSP 15,000.00 leaves SSP 230,000.00 while USD stays unchanged. A payment of USD 5.00 changes only USD. Insufficient SSP cannot use USD as a fallback.

Source currency, recipient capability, requested currency and available funds including fees are checked server-side. Debit, credit, fee postings and transaction status publish atomically and balance by currency. A repeated key and original payload return the original result; a changed payload conflicts. Legacy Sudan/SDG data must never be relabeled as SSP.

This demo ledger is in memory and resets on service restart. It demonstrates financial invariants; it is not a durable or distributed production ledger. New regional currencies will require explicit scale, limits and provider capabilities rather than changing a display label.

## Technical architecture

The existing architecture is retained: Flutter for Android, iOS and the controlled web app; Spring Boot for the simulation API; Nginx for HTTPS, the public website and restricted prototype access. The VPS deployment is isolated from Hash Developed Projects LTD.'s other products. API routes, canonical financial owners, app identifiers and saved-operation namespaces are preserved through the rebrand.

For production, the proposed architecture requires authenticated customer and merchant isolation, durable transactional storage, a ledger and idempotency/status records committed together, a reliable event/outbox mechanism, partner adapters, signed status verification, reconciliation and operational tooling. The final architecture and assurance requirements will be agreed with the bank. No unimplemented provider or webhook is represented as operating today.

## Security and operational approach

The controlled prototype uses HTTPS and reviewer access controls. Its backend port is restricted to loopback and its runtime is isolated. Reviewer access does not replace financial authentication: demo OTP, PIN and the shared identity remain unsuitable for customer funds. Locally retained unresolved operations exclude PIN; retries preserve the original financial intent.

Before a live pilot, implement strong account/session controls, actual transaction authorization, rate limits and fraud controls appropriate to the service, secure provider credentials, customer/merchant permissions, verified callbacks, durable audit records, incident handling, monitoring, tested backup recovery and independent security review. Define information retention, privacy, account deletion and dispute handling for the integrated product. Do not log credentials or sensitive financial request bodies.

## Proposed bank and Rasulo responsibilities

| Area | Proposed bank responsibility | Proposed Rasulo responsibility |
|---|---|---|
| Accounts and custody | Define the approved account/funds model, safeguarding and supported currencies | Present authorized account information and customer controls |
| Customer eligibility | Approve onboarding, identity checks, limits and consent requirements | Implement the agreed onboarding and consent experience |
| Payments | Execute authorized transactions and expose trusted status and identifiers | Integrate APIs, preserve intent/idempotency and show accurate status |
| Settlement and reconciliation | Supply executed financial records and reconciliation interfaces | Match app/provider records and escalate exceptions |
| Compliance and fraud | Agree responsibilities, applicable controls and escalation | Implement assigned controls and supply agreed evidence |
| Support and incidents | Resolve financial-account and settlement issues | Provide app support and coordinate agreed incident/dispute processes |

These are negotiation proposals. A partner bank's licence does not automatically establish Rasulo's status or authority. The parties must determine any separate authorization and applicable approvals before real funds are handled.

## Business model and revenue opportunities

Potential models include agreed transaction-fee sharing, merchant software subscriptions for additional tools, or future business workflow services. All require customer research, partner agreement, cost analysis and applicable review. No fees, pricing, revenue forecasts, minimum volumes or commercial commitments are assumed in this proposal.

Operating costs to assess include hosting and monitoring, identity and messaging services, provider fees, reconciliation/support, security assurance, compliance work and any approved merchant devices. Avoid committing to a hardware rollout before demand, integration and operational support are established.

## Merchant confirmation roadmap

A future soundbox would announce an incoming payment only after the merchant's backend confirms the correct account, currency, amount and successful credit. Example: "You have received five thousand South Sudanese pounds. Rasulo… Khalas!" Pending, failed, unverified or wrong-merchant events must not announce success.

Merchants need volume and amount-privacy controls. Event and transaction deduplication must persist across restarts. Delayed/offline confirmations require fresh authoritative status before playback. The current scenario prototype explores these rules using local fixtures and browser audio; it is not a connected merchant account or working physical device. Juba Arabic voice requires validated recordings and has not been invented for this prototype.

## Phases and decision gates

| Phase | Outcome | Gate before continuing |
|---|---|---|
| Controlled prototype | Coherent Rasulo brand, working simulated journeys, public explanation and restricted review | Verified builds/tests, accurate disclosures and isolated deployment |
| Partner feasibility | Named sponsor, supported APIs/currencies, custody model, technical/compliance contacts | Agreed sandbox access and responsibilities |
| Production foundation | Durable ledger and status, secure identity/authorization, customer isolation and operations | Financial/security assurance and reconciliation evidence |
| Provider sandbox | Real provider contract tested for success, failure, pending, duplicates and reversals | Joint acceptance and required approvals |
| Controlled live pilot | Agreed eligible participants, limits, support and monitoring | Bank/regulatory clearance and explicit go-live decision |
| Expansion assessment | Country-specific research, partnerships, currencies and operating model | Separate approval for each market and service |

Pilot participant counts, duration, limits and success thresholds remain to be agreed. Do not infer readiness for a live pilot from a successful simulation or website launch.

## Regional expansion

Uganda, Kenya and Tanzania are future evaluation markets, followed by other African markets only where the service is justified and supported. Each requires independent partner, licensing, currency, customer-research, language and operational assessment. No current coverage, cross-border corridor or financial authorization is claimed. Currency exchange remains a separate future quoted-and-confirmed product.

## Risks and mitigations

| Risk | Proposed mitigation |
| --- | --- |
| Uncertain custody/authorization model | Resolve the operating model with the bank and qualified local advisers before real funds |
| Ambiguous payment result or duplicate debit | Durable idempotency/status, trusted provider lookup and daily reconciliation |
| Mixed currencies or unintended conversion | Separate account partitions, exact amounts, capability checks and explicit exchange scope |
| Shared demo state mistaken for individual accounts | Restrict prototype access and disclose simulation; implement real customer/merchant isolation before pilot |
| Weak network connectivity | Clear pending/error states, original-operation recovery and measured network testing |
| Merchant false confirmation | Verified incoming events, account binding, durable dedupe and freshness checks |
| Brand/language uncertainty | Final artwork, trademark and local language review before public production |
| Infrastructure interference or failed release | Independent services, resource limits, health checks, staged artifacts and documented rollback |

## Partnership request and next meeting

We request a product walkthrough with the bank's partnerships, digital banking, operations and compliance teams. Discuss supported payment routes and currencies, account/custody model, onboarding and authorization, fees/limits, sandbox documentation, identifiers/callbacks, reversals, reconciliation, support and commercial terms.

The desired outcome is a named bank sponsor and an agreed feasibility/sandbox plan. Before external circulation, confirm the supplied company's incorporation details and address the proposal to the chosen bank. No outreach has been sent by this implementation.

## Reference and status notes

The Bank of South Sudan publishes its bank list and regulations. These are starting points for the parties' review, not a determination of Rasulo's licence eligibility or approval. No capital requirement, licence fee, approval timeline or regulatory endorsement is asserted here.

Confirmed implementation facts describe the repository and controlled demonstration. Proposed business benefits, revenue models, pilot arrangements, merchant devices, bank services and regional expansion remain assumptions or future work until validated and agreed.