Herman Kapange

Web: hermankapange.com

Email: info@hermankapange.com

Paperless Africa — Zambia Councils PWA

Paperless Africa | Ongoing

Zero app store dependency for civic payments

< 60 seconds	Multi-tenant	3
Payment Target	Architecture	Networks

The Challenge

Paperless Africa's native mobile app for Zambian local government councils created a critical operational bottleneck. App store approval delays of 2-4 weeks prevented councils from deploying urgent updates and fixes, frustrating administrators and citizens alike.

The existing app also lacked direct citizen-council communication, relied on multiple fragmented payment channels, and performed poorly on the budget Android devices (Tecno, Infinix, Samsung entry-level) and 3G networks that 70%+ of users depend on.

Our Approach

As an independent contractor, I authored the comprehensive product requirements and am actively building the PWA.

PWA Architecture: Installable via QR code with zero app store dependency. Cache-first strategy for the app shell, network-first with offline fallback for API data. Optimized for sub-60KB initial bundle to work on 3G connections.

Multi-Tenant Efficiency: Single codebase deployed to all councils via subdomain-based routing with runtime configuration per council — branding, API endpoints, contact information all configured at runtime, not build time.

Mobile Money Integration: Unified payment interface across all three Zambian networks (Airtel, MTN, Zamtel). Citizens can view invoices, check balances, make payments, and download receipts. Two-way chat transforms the platform from a payment tool into a civic engagement channel.

Outcomes

Progressive Web App architecture eliminating 2-4 week app store approval delays — councils can deploy updates instantly.

Multi-tenant design serving multiple councils from a single deployment, with per-council branding and configuration.

Mobile money payment integration across all three Zambian networks for universal citizen access.

Offline-capable with automatic sync — cached balances, transaction history, and downloaded documents available without connectivity. Currently in active development.

Technology Stack

React 18 | TypeScript | Vite | Tailwind CSS | Zustand | Service Workers