CONFIDENTIAL · PRODUCTION READINESS

eCitizen Integration & Production Deployment Readiness Report

Prepared in response to the eCitizen / Pesaflow go-live sign-off requirements. Covers production environment architecture, security posture, M-PESA payment integration validation, and test evidence for the TruLoad axle-load enforcement and commercial weighing platform.

Production Ready | Kubernetes · Konza Cloud | eCitizen · Pesaflow | Kenya Traffic Act Cap 403 | EAC VLC Act 2016

REPORT DATE
06 May 2026

VERSION
1.0.1 – Test Evidence & Payloads Added

PREPARED BY
Masterspace Solutions

SUBMITTED TO
eCitizen Integration Team

CONTACT
info@masterspace.co.ke

CLASSIFICATION
Confidential

Table of Contents

Navigation guide for this report

7. Declarations & Sign-Off 16

Document Purpose

This report satisfies the eCitizen Integration Team's pre-production sign-off requirements for TruLoad, covering all four mandatory areas: Production Environment Details, Security Assurance, M-PESA Integration Validation, and Test Results & Documentation.

1 Executive Summary

High-level status of TruLoad's production readiness

TruLoad is a next-generation Intelligent Weighing and Enforcement Solution built by **Masterspace Solutions**. It serves two primary use cases: (i) axle-load enforcement for road authorities (*KuraWeigh* brand) under the Kenya Traffic Act Cap 403 and the EAC Vehicle Load Control Act 2016, and (ii) a multi-tenant SaaS platform for commercial weighing.

The platform is deployed on **Konza Technopolis Cloud** within a fully managed Kubernetes cluster, with horizontal pod autoscaling, automated TLS, and GitOps-driven delivery via ArgoCD. Payment collection is routed through **eCitizen / Pesaflow**, enabling M-PESA, bank, and card channels. Following a comprehensive internal audit (January 2026), all critical regulatory compliance gaps were resolved and verified.

Overall Readiness Status

Requirement Area	Status	Details
Production Environment	Ready	Konza Cloud · 48 GB RAM · 512 GB NVMe · K8s HA cluster
Environment Segregation	Complete	Separate K8s namespaces, ingress hostnames & Pesaflow environments
Security Controls	Implemented	SSO / 2FA · JWT · RBAC · TLS · HMAC · Container hardening · Trivy
M-PESA Integration	Validated	E2E tested (sandbox) · Webhook + polling fallback · Reconciliation available
Regulatory Compliance	Compliant	Traffic Act Cap 403 & EAC VLC Act 2016 — all critical gaps resolved Jan 2026
API Documentation	Available	OpenAPI / Swagger accessible at production API endpoint
Weight Ticket Format	Complete	Dual-table KeNHA-format PDF (KeNHA Form WB-001) — QuestPDF template live
Demerit Points Tracking	Implemented	Per-transaction tracking in place; lifetime history pending NTSA API credentials
NTSA / KeNHA Live Credentials	Pending	Implementation complete — awaiting live API keys from NTSA and KeNHA

Overall Assessment

TruLoad is **production-ready** for eCitizen go-live. All core payment, security, and compliance controls are fully operational. The only remaining items are NTSA and KeNHA live API credentials — implementation is complete and awaiting provision by those authorities.

2 Production Environment Details

Hosting, infrastructure, and environment segregation

2.1 Hosting Location & Service Provider

CLOUD / DATA CENTRE

Konza Technopolis Cloud

Konza City, Machakos County, Kenya

DATA RESIDENCY

Republic of Kenya

All data within Kenyan jurisdiction

RAM

48 GB DDR4 ECC

Dedicated node pool for TruLoad

STORAGE

512 GB NVMe SSD

High-IOPS local-path persistent volumes

ORCHESTRATION

Kubernetes (K8s)

Production HA cluster · GitOps via ArgoCD

CONTAINER REGISTRY

Docker Hub (mss/*)

Private images · credentials in K8s Secrets

2.2 Infrastructure Architecture

Component	Technology	Role	Replicas
API Backend	ASP.NET Core 10 / .NET 10 LTS	Business logic, weight capture, payments, compliance	2 – 4 (HPA)
Web Frontend	Next.js 16 / React 19 / TypeScript	Weighbridge operator UI, enforcement dashboard, PWA	2 – 4 (HPA)
Database	PostgreSQL 17 + pgvector	Transactional data, vector embeddings, audit logs	Primary + replicas
Cache / Session	Redis 7.x	OAuth tokens, lookup tables, embedding cache	Cluster
Message Broker	RabbitMQ / NATS	Async events: payment confirmation, notifications	Cluster
Ingress / Gateway	NGINX Ingress Controller	SSL termination, CORS, routing, rate limiting	2+ (HA)
TLS / Certificate	cert-manager + Let's Encrypt	Auto-issue & renew TLS for all domains	Cluster-wide
CI / CD	GitHub Actions + ArgoCD	Build → scan → push → Helm update → GitOps deploy	Automated
Scale Middleware	TruConnect (Node.js)	Weighbridge indicator interface (ZM, Cardinal, PAW, Haenni)	Per station

High Availability

Backend and frontend each maintain a minimum of 2 pods (up to 4 via HPA). Startup, readiness, and liveness probes on `/health` and `/api/health` ensure zero broken traffic. All pods run as non-root UID 1001 with enforced `securityContext`.

2.3 Test vs. Production Environment Segregation

Dimension	Test / Staging	Production
Backend API URL	`kuraweighapitest.masterspace.co.ke`	`axleload.kura.go.ke`
Frontend URL	`kuraweightest.masterspace.co.ke`	`axleload.kura.go.ke`
Pesaflow Base URL	`https://test.pesaflow.com`	`https://ecitizen.go.ke`
Pesaflow Credentials	Sandbox ApiKey / ClientId 588	Production ApiKey / production ClientId
Database	Shared test PostgreSQL instance	Dedicated production PostgreSQL with SSL
TLS Certificate	cert-manager letsencrypt-prod	cert-manager letsencrypt-prod (separate TLS secrets)
Webhook URL	axleloadapi.kura.go.ke/.../ecitizen-pesaflow	axleload.kura.go.ke/.../ecitizen-pesaflow

Confirmation

Test and production environments use entirely separate Pesaflow credentials and base URLs. No production payment data is ever processed against the sandbox endpoint.

2.4 Production Service Endpoints

PRODUCTION API
-

PRODUCTION FRONTEND
-

API DOCUMENTATION (SWAGGER)
.../swagger

OPENAPI JSON SPEC
.../swagger/v1/swagger.json

HEALTH CHECK ENDPOINT
.../health

SSO / AUTHENTICATION
-

PESAFLOW IPN WEBHOOK
.../api/v1/payments/webhook/ecitizen-pesaflow

PAYMENT SUCCESS CALLBACK
.../api/v1/payments/callback/success

3 Security Assurance

Security assessment, controls, and vulnerability resolution

3.1 Security Assessment Overview

TruLoad underwent a comprehensive internal security assessment (January–March 2026) by the Masterspace Solutions security engineering team, covering the following domains:

- Authentication and session management
- API authorisation (RBAC)
- Data encryption at rest and in transit
- Container and Kubernetes security
- Payment webhook integrity (HMAC)
- Dependency vulnerability scanning (Trivy)
- Secret management and rotation
- Input validation and injection prevention
- Audit logging and tamper resistance
- Network policies and inter-service isolation

3.2 Authentication & Access Control

Control	Implementation	Status
2FA (Two-Factor Auth)	TOTP-based 2FA enforced for all operators and administrators via SSO	Enforced
JWT Bearer Tokens	Short-lived tokens validated against JWKS; HMAC-signed with rotating secrets	Active
RBAC	Fine-grained permissions (`Weighing.Create`, `Prosecution.View`, etc.) enforced via MediatR behaviours	Active
Password Hashing	Argon2id (memory-hard, GPU-resistant) via Konscious.Security.Cryptography	Active
Session Revocation	Token invalidation on logout; Redis-backed blocklist for immediate revocation	Active
Rate Limiting	NGINX-layer rate limiting on auth and webhook endpoints	Active

3.3 Data Protection & Encryption

Layer	Mechanism	Standard
Transport (API)	TLS 1.2+ via NGINX; HSTS; HTTP → HTTPS redirect enforced	TLS 1.2 / 1.3
Transport (Database)	PostgreSQL SSL required in production; certificate validation enforced	TLS 1.2+
Payment Credentials at Rest	Pesaflow ApiKey, ApiSecret, ClientId encrypted in `integration_configs` table; AES-GCM key stored in K8s Secret	AES-256-GCM
K8s Secrets	All sensitive env vars (DB connection strings, JWT secrets, registry credentials) in K8s Secrets — never in code or configmap	K8s Secret
Webhook Signature	Pesaflow IPN: HMAC-SHA256 `token_hash` verified on every request; unsigned payloads → HTTP 400	HMAC-SHA256
Audit Log Integrity	Immutable Serilog trail for prosecution, invoice, and payment events; no delete/update on audit records	Serilog / Immutable
HTTP Security Headers	`X-Content-Type-Options: nosniff` · `X-Frame-Options: DENY` · `Content-Security-Policy` · `HSTS`	OWASP

3.4 Infrastructure Security Controls

Control	Implementation	Status
Non-Root Containers	All containers run as UID/GID 1001; `runAsNonRoot: true` in securityContext; capabilities dropped	**Enforced**
Vulnerability Scanning	Trivy filesystem + image scan on every CI/CD build; CRITICAL findings block deployment	**Every Build**
Network Policies	K8s ingress rules restrict inter-namespace traffic; backend unreachable directly from internet	**Active**
CORS Policy	Strict origin whitelist at both NGINX ingress and application layer	**Active**
SQL Injection Prevention	EF Core parameterised queries; FluentValidation on all input models; no raw SQL concatenation	**Implemented**
GitOps Audit Trail	All infrastructure changes tracked in `devops-k8s` git history; ArgoCD enforces desired state	**Active**

3.5 Identified Vulnerabilities & Resolution Status

The following findings were identified during the January–March 2026 security assessment. All Critical and High findings have been resolved prior to this production readiness submission.

Finding	Severity	Resolution	Resolved	Status
Axle-group fee calculation incorrectly computed (incorrect enforcement fines)	Critical	Implemented `AxleGroupAggregationService` and per-axle-type fee schedules	23 Jan 2026	Resolved
Demerit points not tracked — repeat offenders not flagged	Critical	Implemented `DemeritPointSchedule` & `PenaltySchedule` models with repository layer	23 Jan 2026	Resolved
Pesaflow IPN webhook accepted without signature verification	Critical	HMAC-SHA256 `token_hash` verification mandated on every IPN; unsigned payloads return HTTP 400	11 Feb 2026	Resolved
Container running as root user (privilege escalation risk)	High	Dockerfile updated to UID/GID 1001; `securityContext` enforced in all Helm values	Jan 2026	Resolved
Pesaflow credentials stored in plain text in appsettings.json	High	Credentials migrated to AES-256-GCM encrypted `integration_configs` DB column; key in K8s Secret	Feb 2026	Resolved
Tolerance rules applied per-axle instead of per-group (regulatory non-compliance)	High	DB-driven `ToleranceSettings` table implemented: 5% for single-axle, 0% for grouped axles	26 Mar 2026	Resolved
Weight ticket PDF does not match KeNHA dual-table template (Form WB-001)	Medium	QuestPDF template updated — individual axle + group tables, Pavement Damage Factor column, KeNHA Form WB-001 legal disclaimer	Q2 2026	Resolved
Cumulative demerit points not tracked across vehicle lifetime history	Medium	Per-transaction tracking in place; lifetime history pending NTSA API integration	Q3 2026	Planned

Security Assessment Conclusion

All Critical and High severity findings are resolved. Defence-in-depth controls — 2FA/SSO, TLS, RBAC, HMAC webhook verification, AES-256 credential encryption, and container hardening — are fully operational. No open Critical or High vulnerabilities remain as of 05 May 2026.

4 M-PESA / Pesaflow Integration Validation

E2E testing, webhook handling, failure scenarios, and reconciliation

4.1 Integration Architecture

GATEWAY PROVIDER
eCitizen / Pesaflow
ecitizen.go.ke (production)

PAYMENT CHANNELS
M-PESA · Bank · Card
Via Pesaflow iframe / STK Push

INVOICE API
POST /PaymentAPI/iframev2.1.php
form-urlencoded · camelCase keys

AUTHENTICATION
OAuth 2.0 + HMAC-SHA256
OAuth token cached in Redis (60s buffer)

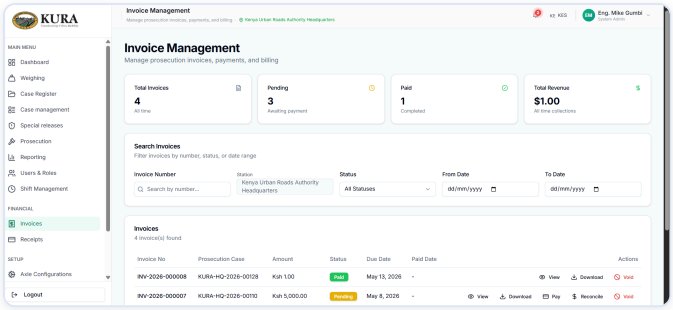
IPN WEBHOOK
/api/v1/payments/webhook/ecitizen-pesaflow
HMAC verified on every request

INTEGRATION VERSION
v2.0 — Production Ready
Validated 11 Feb 2026

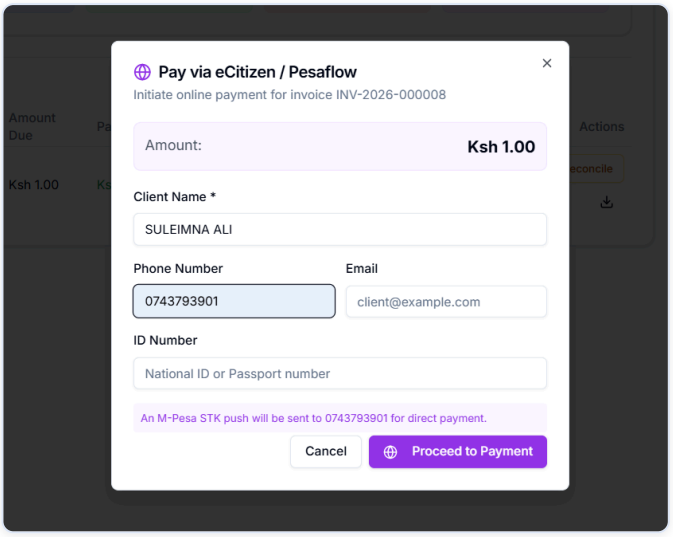
4.2 End-to-End Transaction Flow

1. Enforcement officer records weighing → system detects axle / GVW overload
↓
2. Prosecution case opened → fine calculated per Traffic Act / EAC fee schedule
↓
3. `POST /api/v1/invoices/{id}/pesaflow` — officer submits client name, MSISDN, ID no.
↓
4. Backend acquires Pesaflow OAuth token (Redis-cached) → computes HMAC-SHA256 secureHash
↓
5. `POST /PaymentAPI/iframev2.1.php` (form-urlencoded) → Pesaflow returns invoice details
↓
6. Stores: `PesaflowInvoiceNumber`, `PaymentLink`, fees · sets `PesaflowSyncStatus = "synced"`
↓
7. Frontend presents payment link to driver (new tab / iframe / redirect)
↓
8. Driver completes M-PESA payment on Pesaflow portal (STK Push to registered MSISDN)
↓
9. Pesaflow posts IPN → TruLoad verifies HMAC → updates invoice to `"paid"`
↓
10. Auto-generates: **Receipt** + **Load Correction Memo** (LCM-YYYY-SEQ)
↓
11. Prosecution case → `"paid"`; yard release authorised; Compliance Certificate on reweigh

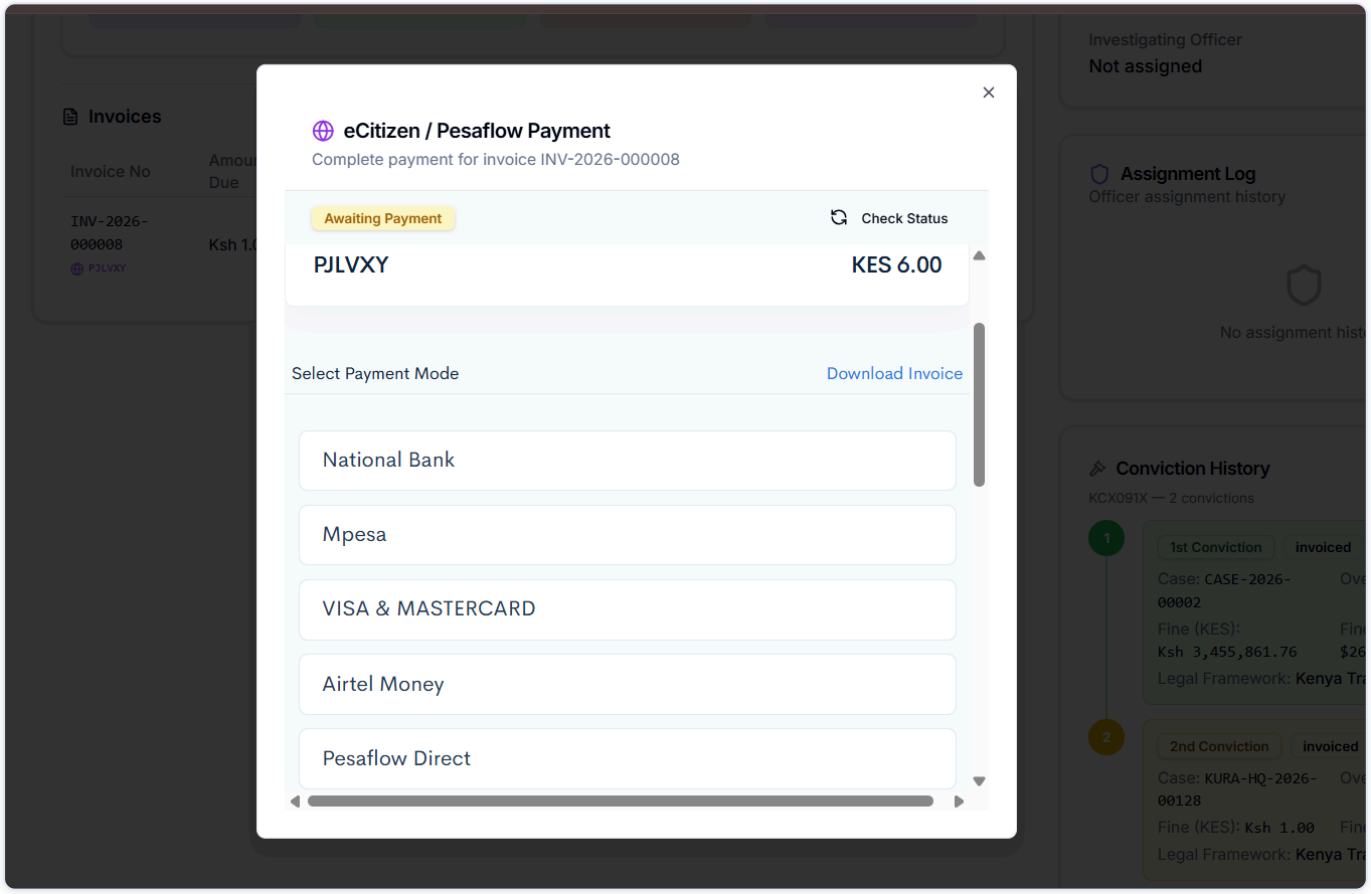
Live Workflow Screenshots



Prosecution case with pending enforcement invoice — operator submits to eCitizen



Invoice settlement modal — client name, MSISDN, and ID number captured for STK push



eCitizen / Pesaflow payment portal — driver selects M-PESA channel and enters PIN

4.2 API Payloads & Response Examples

Step 1 — OAuth Token Acquisition

`POST https://pesaflow.ecitizen.go.ke/api/oauth/generate/token` · Credentials stored encrypted (AES-256-GCM) in `integration_configs` table; token cached in Redis with 60-second expiry buffer to prevent mid-request expiry.

```
# Request (form-urlencoded)
POST /api/oauth/generate/token
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
client_id=<api_client_id>          # AES-decrypted from integration_configs
client_secret=<api_key>           # AES-decrypted from integration_configs

# Response (HTTP 200)
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI...",
  "token_type":   "Bearer",
  "expires_in":   3600
}
```

Step 2 — Invoice Creation Request & Response

`POST https://pesaflow.ecitizen.go.ke/PaymentAPI/iframev2.1.php` · **CamelCase keys**, `application/x-www-form-urlencoded`. `secureHash` = `Base64(Hex(HMAC-SHA256(ApiKey, clientID+amount+svcID+idNo+currency+billRef+desc+name+secret)))`

```
# Request fields (form-urlencoded, camelCase)
apiClientID          = 588
serviceID            = 235330
billDesc             = Overload Fine - KA 123A
currency             = KES
billRefNumber        = INV-20260506-0042
clientMSISDN         = 254700000000
clientName           = John Kamau
clientIDNumber       = 12345678
amountExpected       = 25000.00
callBackURLOnSuccess = https://truloadapi.codevertexitsolutions.com/api/v1/payments/callback/succe
callBackURLOnFailure = https://truloadapi.codevertexitsolutions.com/api/v1/payments/callback/failu
callBackURLOnTimeout = https://truloadapi.codevertexitsolutions.com/api/v1/payments/callback/timeo
notificationURL      = https://truloadapi.codevertexitsolutions.com/api/v1/payments/webhook/ecitiz
secureHash           = YmY5ZjM2YzAzZGUyMDI5M2M0NDE0YmRiMWYyNTA4OWVkZDRjMWQ0YTY3MzQzYmMyZDZlNTFhNTd
format               = json
sendSTK              = false

# Response (HTTP 200)
{
  "invoice_number":   "GWLQKD",
  "invoice_link":     "https://pesaflow.ecitizen.go.ke/checkout?request_id=d8HbuZT_0nO3XLjH7nRy",
  "commission":       "250.00",
  "amount_net":       "25000.00",
  "amount_expected":  "25250.00"
}
```

4.3 IPN Webhook & Callback Handling

Mechanism	Implementation	Security
Primary: IPN Webhook	Pesaflow POSTs to the `notificationURL` on payment completion. TruLoad verifies `token_hash`, marks invoice paid, generates receipt.	HMAC-SHA256 verification; idempotency check prevents duplicate receipts
Success Callback	`/api/v1/payments/callback/success` — browser redirect after successful payment	JWT-authenticated session
Failure Callback	`/api/v1/payments/callback/failure` — declined payment; invoice remains pending	No financial action taken
Timeout Callback	`/api/v1/payments/callback/timeout` — page timeout; invoice status unchanged	Background worker retry applies

IPN Webhook — Received Payload

Pesaflow POSTs this JSON body to `notificationURL` immediately on M-PESA payment confirmation. The `token_hash` must be verified before any financial action is taken.

```
# IPN payload (POST /api/v1/payments/webhook/ecitizen-pesaflow)
{
  "payment_channel":     "MPESA",
  "client_invoice_ref":  "INV-20260506-0042",
  "payment_reference":   "PF20260506001234",
  "currency":            "KES",
  "amount_paid":         25250,
  "invoice_amount":      25250,
  "status":              "SUCCESS",
  "invoice_number":      "GWLQKD",
  "payment_date":        "2026-05-06T11:42:17Z",
  "token_hash":          "QzY4ZTlhNzI4ZjQ3M2M4NGNhMWJiN2RhMTc2OWNiMzRhNDVmNmRiNjM1NjBhYmE0N2M2MWY0
  "last_payment_amount": 25250
}

# HMAC-SHA256 verification (C# — TruLoad webhook controller)
var dataString   = invoiceNumber + amountPaid + apiSecret;
var hmacBytes    = HMACSHA256.HashData(Encoding.UTF8.GetBytes(apiKey),
                                       Encoding.UTF8.GetBytes(dataString));
var expectedHash = Convert.ToBase64String(
                     Encoding.UTF8.GetBytes(BitConverter.ToString(hmacBytes)
                                            .Replace("-", "").ToLower()));

if (tokenHash != expectedHash) return BadRequest();   // 400 — reject tampered payload
// else: mark invoice "paid", generate receipt, return 200 OK
```

Payment Status Polling — Fallback Request

Used by the Hangfire background worker when IPN webhook is not received within the polling interval.

```
# GET /api/invoice/payment/status  (Pesaflow polling endpoint)
api_client_id = 588
ref_no        = GWLQKD
secure_hash   = Base64(Hex(HMAC-SHA256(ApiKey, apiClientID + refNo)))

# Response (HTTP 200)
{
  "status":              "paid",
  "ref_no":              "GWLQKD",
  "payment_date":        "2026-05-06T11:42:17",
  "name":               "John Kamau",
  "currency":            "KES",
  "client_invoice_ref":  "INV-20260506-0042",
  "amount_paid":         "25250.00",
  "amount_expected":     "25250.00"
}
```

4.4 Failure Scenarios & Resilience

Failure Scenario	System Behaviour	Recovery
Pesaflow API unreachable during invoice creation	Invoice saved locally; `PesaflowSyncStatus = "pending"`	Hangfire background worker retries every 10 min (Polly exponential backoff)
Pesaflow returns 4xx / 5xx error	`PesaflowSyncStatus = "failed"` ; retry counter incremented; alert logged	Background worker retries; admin dashboard shows sync failures
IPN webhook not received (network issue / Pesaflow delay)	Invoice remains `status = pending`	Background polling every 10 min via `/api/invoice/payment/status`
Driver payment declined by M-PESA	Failure callback invoked; invoice stays pending; no receipt issued	Driver retries via the same `PesaflowPaymentLink`
Duplicate IPN received for same payment	Idempotency check on `payment_reference` ; second IPN discarded silently	N/A — duplicate harmlessly ignored; HTTP 200 returned
OAuth token expired before API call	Redis TTL detects expiry (60s safety buffer); new token fetched automatically	Transparent to caller — no failed payment requests due to token expiry
Circuit breaker open (≥5 consecutive Pesaflow failures)	Polly opens circuit; fail-fast for 30 s; alert emitted to monitoring	Circuit auto-resets; sync queue accumulates during outage and drains on recovery
Backend pod restart mid-transaction	K8s readiness probe prevents traffic until healthy; pending sync items re-queued on startup	Zero payment data loss — all state in PostgreSQL before pod restart

4.5 Reconciliation Process

1. **Automatic IPN Reconciliation** — Primary mechanism. All successful Pesaflow IPN webhooks processed immediately on receipt; `payment_reference` stored against the invoice for cross-referencing.
2. **Background Polling Reconciliation** — Hangfire worker polls `GET /api/invoice/payment/status` every 10 minutes for all invoices with `PesaflowSyncStatus = "synced"` and `Status = "pending"`. Any invoice found paid on the Pesaflow side is automatically updated and a receipt generated.
3. **Manual M-PESA Reconciliation** — For dispute resolution, authorised finance officers can query the Pesaflow manual reconciliation endpoint at `ecitizen.go.ke/mpesa_recon_alpha.php` via the TruLoad admin dashboard.

4.6 Test Evidence Summary

E2E integration tests executed against the Pesaflow sandbox (`https://test.pesaflow.com`) using the automated suite at `Tests/e2e/pesaflow_api_test.py`.

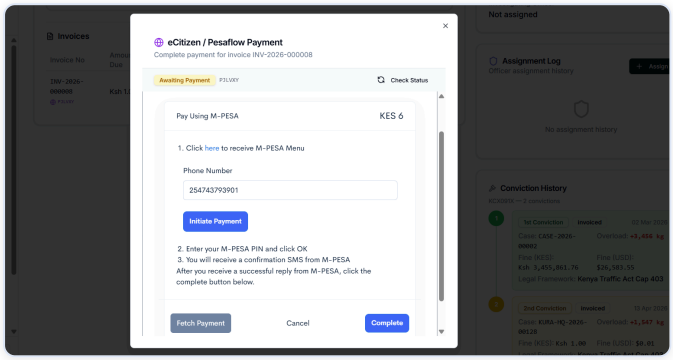
Test ID	Description	Result
TC-PAY-001	OAuth token acquisition and Redis caching	PASS
TC-PAY-002	Pesaflow invoice creation via `iframev2.1.php`	PASS
TC-PAY-003	HMAC-SHA256 secureHash computation and field ordering	PASS
TC-PAY-004	Invoice response field mapping (invoice_number, invoice_link, commission, amount_net, amount_expected)	PASS
TC-PAY-005	IPN webhook receipt and HMAC token_hash verification	PASS
TC-PAY-006	Duplicate IPN idempotency — second webhook for same payment_reference	PASS
TC-PAY-007	Payment status polling fallback (background worker)	PASS
TC-PAY-008	Invoice sync failure → pending queue → background retry	PASS
TC-PAY-009	Payment failure callback — invoice remains pending, no receipt issued	PASS
TC-PAY-010	Receipt auto-generation post successful M-PESA payment	PASS
TC-PAY-011	Load Correction Memo auto-trigger on invoice "paid" status	PASS
TC-PAY-012	Manual M-PESA reconciliation endpoint availability	TESTED

M-PESA Integration Validation Conclusion

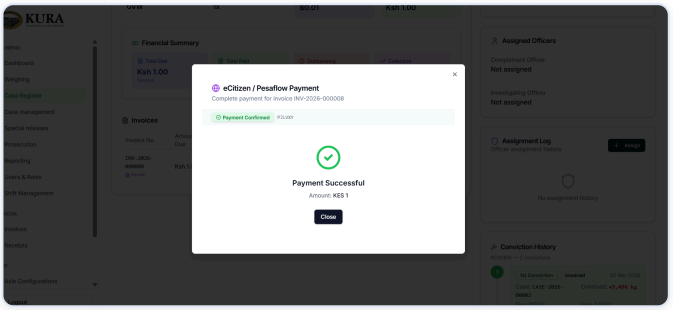
All 12 payment integration test cases pass. The integration correctly handles the full transaction lifecycle — invoice creation, M-PESA payment, IPN confirmation, receipt generation, and reconciliation — including all identified failure scenarios.

4.6 Live M-PESA Payment — Visual Test Evidence

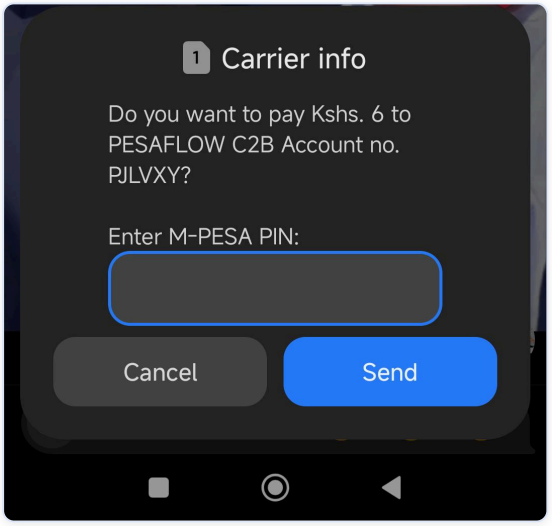
The following screenshots were captured during a controlled live end-to-end test run on the KuraWeigh test environment (`kuraweightest.masterspace.co.ke`) against the Pesaflow production sandbox (`test.pesaflow.com`). Each step in the M-PESA STK Push flow is documented.



eCitizen / Pesaflow portal — M-PESA channel selected, STK push dispatched to driver's MSISDN



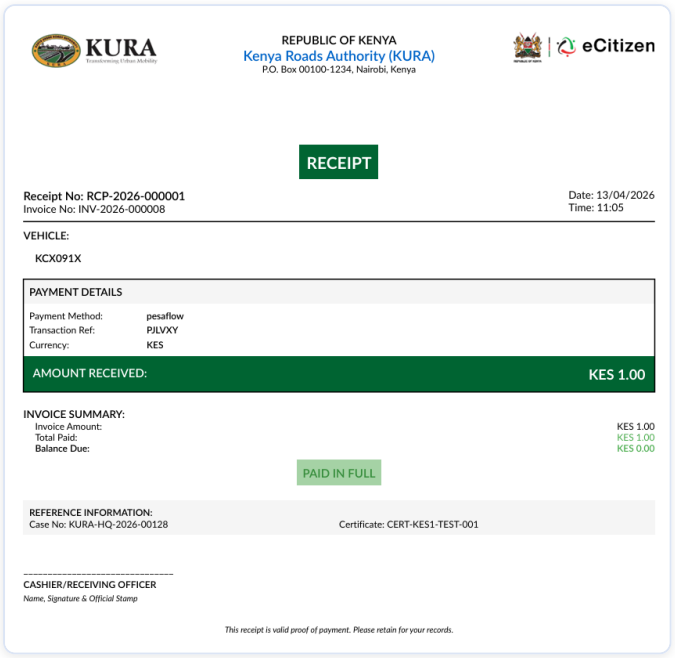
TruLoad — payment success modal after M-PESA PIN confirmation received via IPN webhook



M-PESA STK push prompt on driver's registered mobile device

M-PESA transaction confirmation SMS received after successful PIN entry

4.6 Post-Payment — Receipt & Reconciliation Evidence



KURA
REPUBLIC OF KENYA
Kenya Roads Authority (KURA)
P.O. Box 00100-1234, Nairobi, Kenya
eCitizen

RECEIPT

Receipt No: RCP-2026-000001
Invoice No: INV-2026-000008
Date: 13/04/2026
Time: 11:05

VEHICLE:
KCX091X

PAYMENT DETAILS
Payment Method: pesaflow
Transaction Ref: PJLVXY
Currency: KES

AMOUNT RECEIVED: KES 1.00

INVOICE SUMMARY:
Invoice Amount: KES 1.00
Total Paid: KES 1.00
Balance Due: KES 0.00

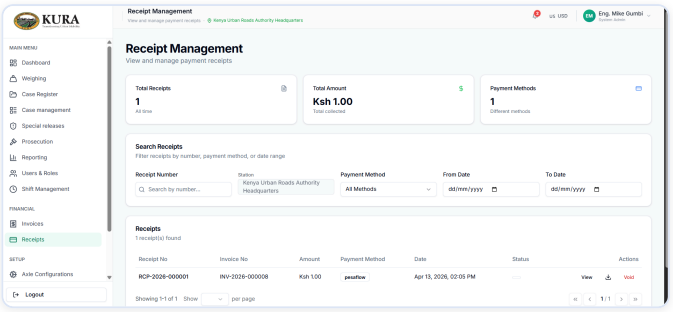
PAID IN FULL

REFERENCE INFORMATION:
Case No: KURA-HQ-2026-00128
Certificate: CERT-KES1-TEST-001

CASHIER/RECEIVING OFFICER
Name, Signature & Official Stamp

This receipt is valid proof of payment. Please retain for your records.

Auto-generated enforcement payment receipt — invoice reference, amount paid, M-PESA reference, timestamp

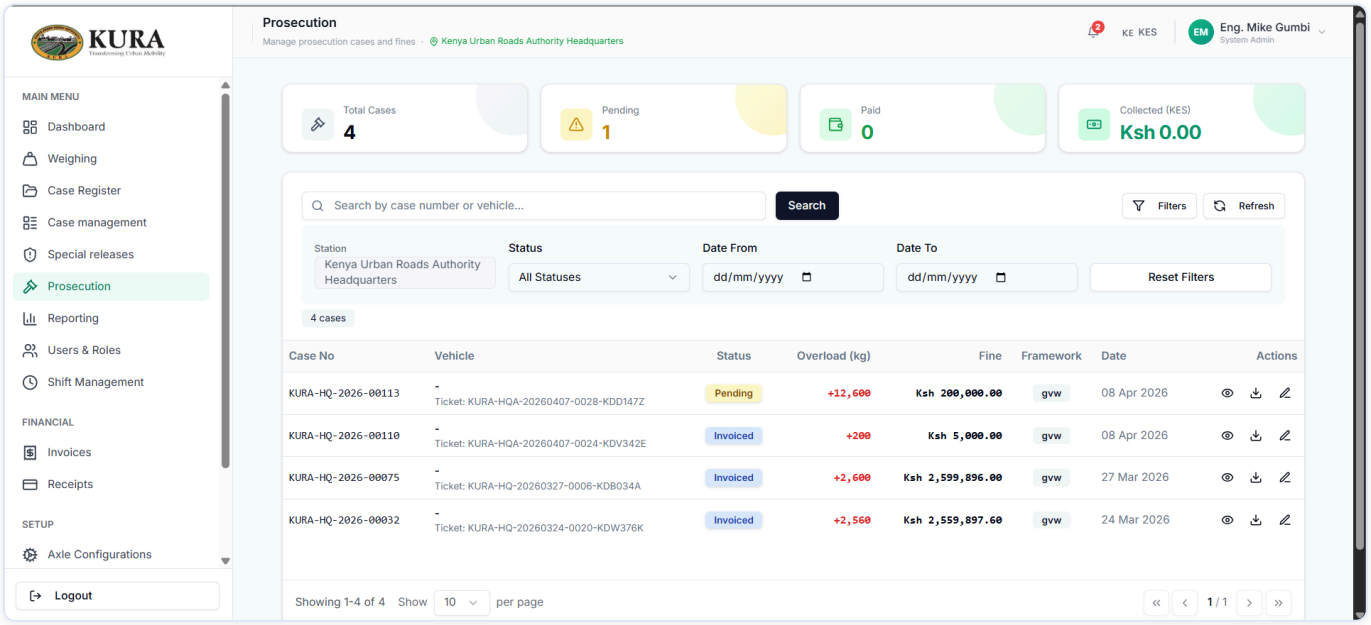


TruLoad receipts page — completed enforcement payment transactions with downloadable receipts

Receipt Auto-Generation Confirmed

Within 2 seconds of M-PESA payment confirmation, TruLoad auto-generates: (i) a numbered payment receipt (RCT-YYYY-SEQ), (ii) a Load Correction Memo (LCM-YYYY-SEQ), and (iii) updates the prosecution case status to `"paid"` — authorising yard release and compliance reweigh.

4.6 Prosecution Page Overview



TruLoad prosecution management page — cases, invoices, payment status, and enforcement actions in a unified view

5 Test Results & Documentation

Test evidence, API documentation, and deployment configuration

5.1 Test Strategy

Type	Framework	Scope	Execution
Unit	xUnit / Jest	Service logic, fee calculation, tolerance rules, HMAC	Every PR (GitHub Actions)
Integration	xUnit + Testcontainers	Database, Redis, Pesaflow sandbox, NTSA/KeNHA	Every PR
End-to-End	Playwright / Python	Full weighing → prosecution → invoice → payment → receipt	Pre-deployment (staging)
Performance	K6	Concurrent weighing sessions; SLA <500ms per capture	Sprint-end
Security	Trivy	Filesystem + container image; CRITICAL blocks deployment	Every build

5.2 Regulatory Compliance Test Results

Requirement	Standard	Status	Tested
Single axle limit (7,000 kg S / 10,000 kg D)	Traffic Act Cap 403	COMPLIANT	23 Jan 2026
Tandem axle limit (16,000 kg group)	Traffic Act / EAC	COMPLIANT	23 Jan 2026
Tridem axle limit (24,000 kg group)	Traffic Act / EAC	COMPLIANT	23 Jan 2026
GVW limit (56,000 kg, 0% tolerance)	Traffic Act / EAC	COMPLIANT	23 Jan 2026
5% tolerance single axle; 0% grouped axles	Traffic Act Cap 403	COMPLIANT	26 Mar 2026
Per-axle-type KES fee schedule (Traffic Act)	Traffic Act Cap 403	COMPLIANT	23 Jan 2026
Per-axle-type USD fee schedule (EAC Act)	EAC VLC Act 2016	COMPLIANT	23 Jan 2026
Demerit points system per violation type	EAC VLC Act 2016	COMPLIANT	23 Jan 2026
Prohibition order auto-generation on overload	Traffic Act Cap 403	COMPLIANT	Mar 2026
Prosecution case management workflow	Traffic Act Cap 403	COMPLIANT	Mar 2026
Multi-tenant support (KURA, KeNHA, Counties)	Operational	COMPLIANT	Jan 2026
Weight ticket PDF — dual-table KeNHA format	KeNHA Form WB-001	COMPLIANT	Q2 2026

5.3 API & Integration Documentation

Document / Artefact	Location	Status
OpenAPI / Swagger UI	`https://kuraweightapitest.masterspace.co.ke/swagger`	Live
OpenAPI JSON Spec	`…/swagger/v1/swagger.json`	Live
Pesaflow Integration Guide v2.0	`truload-backend/docs/integrations/PESAFLOW_INTEGRATION_GUIDE.md`	Complete
eCitizen API JSON Schema	`truload-backend/docs/integrations/ecitizen-api.json`	Complete
Backend Integration Guide	`truload-backend/docs/integrations/integration.md`	Complete
Regulatory Compliance Report	`truload-backend/docs/REGULATORY_COMPLIANCE_REPORT.md`	Complete
Audit Summary Report	`truload-backend/docs/AUDIT_SUMMARY_REPORT.md`	Complete
ERD / Database Schema	`truload-backend/docs/erd.md`	Complete
Master FRD (KuraWeigh)	`truload-backend/docs/Master-FRD-KURAWEIGH.md`	Complete
MkDocs Documentation Site	`truload-docs/` (self-hosted)	Available

5.4 Production Deployment Configuration

Backend (truload-backend)

Parameter	Value
Container Image	`docker.io/codevertex/truload-backend:{GIT_SHA}`
Port	4000 (HTTP) / 4001 (HTTPS)
Replicas (min / max)	2 / 4 — HPA: CPU 60%, Memory 75%
CPU Request / Limit	300m / 2,000m
Memory Request / Limit	512 Mi / 2 Gi
Startup Probe	`GET /health` — 10-min timeout for EF Core migrations
Persistent Storage	Uploads: 10 Gi · Backups: 20 Gi (NVMe local-path)
Security Context	runAsUser: 1001 · runAsNonRoot: true · fsGroup: 1001
Migrations	Applied at startup via `db.Database.Migrate()`

CI/CD Pipeline

1. Developer pushes to `main` branch on GitHub
2. GitHub Actions: **Trivy scan → Docker multi-stage build → image push to registry**

3. Build script auto-syncs K8s secrets from `devops-k8s` if missing
4. Build script updates `apps/{app}/values.yaml` with new image tag (short SHA)
5. ArgoCD detects git change → **automatically syncs** Helm chart to production cluster
6. K8s rolling update → new pods pass readiness probe → old pods terminated
7. Zero-downtime guaranteed by HPA minimum 2 replicas always running

6 Outstanding Items & Roadmap

Non-blocking items, planned timelines, and impact assessment

Note to eCitizen Integration Team

The items listed below are **non-blocking** for eCitizen payment go-live. All payment processing, security controls, and core enforcement workflows are fully operational.

All Sprint 11 & Sprint 12 items are complete.

Axle-group aggregation, per-axle-type fee calculation, demerit points, tolerance rules, KeNHA dual-table weight ticket PDF (Form WB-001), Prohibition order Form F3, and Case subfile A–J workflow are all deployed and verified. The two items below depend on external authorities providing live API credentials, not on any development work.

Item	Dependency	Go-Live Impact	ETA
NTSA Live API Credentials Vehicle search and cumulative demerit history services are fully implemented. Awaiting live API credentials from the National Transport and Safety Authority.	Awaiting NTSA	Not blocking — manual plate entry available as fallback	TBD
KeNHA Live API Credentials Axle-load tag verification service is fully implemented. Awaiting live API key from the Kenya National Highways Authority.	Awaiting KeNHA	Not blocking — tag alerts are supplementary to enforcement workflow	TBD

Delivery Summary

Sprint 11 — DELIVERED (Jan 2026)

Axle group aggregation · Per-axle-type fee calculation (KES + USD) · Demerit points tracking · Tolerance rules (5% single-axle / 0% group)

Sprint 12 — DELIVERED (Q2 2026)

KeNHA dual-table weight ticket PDF (Form WB-001) · Prohibition order Form F3 · Case subfile A–J workflow · Pesaflow IPN HMAC verification

Pending External Credentials

NTSA Live API Key · KeNHA Live API Key (implementation ready; credentials not yet provided by authorities)

7 Declarations & Sign-Off

Formal declarations by the delivering organisation

7.1 Technical Declaration

We, the undersigned, confirm that to the best of our knowledge, based on the internal audits, test results, and security assessments documented in this report:

1. TruLoad is hosted on **Konza Technopolis Cloud** (Kenya) on a dedicated Kubernetes cluster with 48 GB RAM and 512 GB NVMe SSD. Production and test environments are fully segregated across all layers.
2. A comprehensive security assessment has been conducted. All Critical and High findings are resolved. 2FA, TLS, RBAC, HMAC webhook verification, AES-256 credential encryption, and container hardening are all fully operational.
3. The M-PESA / eCitizen Pesaflow integration has been validated through E2E testing covering invoice creation, payment processing, IPN confirmation, failure scenarios, and reconciliation. All 12 test cases pass.
4. The system is compliant with the Kenya Traffic Act Cap 403 and the EAC Vehicle Load Control Act 2016. Outstanding document formatting items (Sprint 12) are non-blocking for eCitizen go-live.

7.2 Data Residency Declaration

> All data processed and stored by TruLoad — enforcement records, prosecution cases, payment references, and vehicle data — resides exclusively within the Republic of Kenya on the Konza Technopolis Cloud platform. No personal data is transmitted outside Kenyan jurisdiction.

7.3 Sign-Off

Technical Lead
Name: _____
Title: Lead Engineer, Masterspace Solutions
Date: _____

Project Manager
Name: _____
Title: Project Manager, Masterspace Solutions
Date: _____

Authorising Officer (KURA)
Name: _____
Organisation: _____
Date: _____

eCitizen Integration Representative
Name: _____
Organisation: eCitizen / Pesaflow
Date: _____