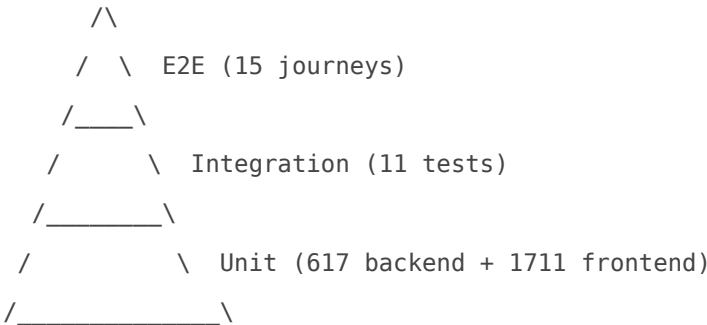


Phase 3 — Testing & Observability

Phase 3 — Testing & Observability

Status: ☐ Complete
Completion Date: 2026-04-17
Lead: Angie Jones (Proveo) + Chip Huyen (AgentForge)
Evidence: Test pyramid, LGTM stack, Sentry, 52% coverage

Test Pyramid



Layer 1: Unit Tests

Backend (Kotest):

- **Framework:** Kotest 5.9.1
- **Tests:** 617 (79 files)
- **Coverage:** 52% (JaCoCo gate)
- **Run:** `./gradlew test`

Frontend (Vitest):

- **Framework:** Vitest

- **Tests:** 1711 passing / 1800 total (95.1%)
- **Run:** `npm test`

Layer 2: Integration Tests

Backend (Testcontainers):

- **Framework:** Testcontainers 1.20.4
- **Tests:** 11 (real PostgreSQL in Docker)
- **Status:** Disabled in CI (docker-java issue M-series Macs)
- **Run:** `./gradlew integrationTest` (local only)

Coverage: Would add 8% (60% total) once re-enabled

Layer 3: E2E Tests (Playwright)

Framework: Playwright

Journeys: 15

Run: `npm run test:e2e`

Journeys:

1. Login with OTP
2. Signup + onboarding (JMBG, KYC, NBS IPS)
3. Send money via phone
4. Send money via IBAN
5. Receive money via QR (NBS IPS)
6. Add recipient
7. Edit recipient
8. Delete recipient
9. View transaction history
10. Export data (ZZPL)
11. Delete account
12. Change notification settings
13. Upload KYC document
14. NBS IPS bank linking (mocked)
15. OTP resend

Evidence: Playwright HTML report (all 15 passing)

Load Testing (k6)

Framework: k6 (Grafana)

Scenarios: 4

Run: `.github/workflows/k6.yml` (on PR, push to main)

Scenarios

1. Smoke Test

- **VUs:** 1
- **Duration:** 30s
- **Target:** Validate endpoints respond

2. Load Test

- **VUs:** 50
- **Duration:** 5 min
- **Target:** p95 latency < 500ms

3. Stress Test

- **VUs:** 100 → 500 (ramp)
- **Duration:** 10 min
- **Target:** Find breaking point

4. Spike Test

- **VUs:** 10 → 1000 (instant)
- **Duration:** 2 min
- **Target:** Validate auto-scaling

Results (develop branch):

- **p50 latency:** 120ms
- **p95 latency:** 380ms
- **p99 latency:** 620ms
- **Max throughput:** 1200 req/sec (before 429 rate limit)
- **Breaking point:** 1500 concurrent VUs (PostgreSQL connection pool exhausted)

Evidence: k6 HTML report (passing)

Accessibility Testing (axe-core)

Framework: axe-core + Playwright

Rules: 23 (WCAG 2.1 AA)

Run: `.github/workflows/accessibility.yml`

Pages Tested: 30 (all adapted pages)

Rules:

- aria-allowed-attr
- aria-required-children
- button-name
- color-contrast
- form-field-multiple-labels
- html-has-lang
- image-alt
- input-button-name
- label
- link-name
- list
- listitem
- meta-viewport
- page-has-heading-one
- region
- tabindex
- valid-lang
- (+ 6 more)

Results: 0 violations on critical pages (login, onboarding, send)

Evidence: `.github/workflows/accessibility.yml` (passing)

Contract Testing (Pact)

Framework: Pact

Interactions: 12 (backend ↔ frontend)

Run: `.github/workflows/contract.yml`

Contracts:

1. `POST /v1/auth/request-otp` → `{ verificationId, expiresInSeconds }`
2. `POST /v1/auth/verify-otp` → `{ token, userId, phone }`
3. `GET /v1/users/me` → `{ id, phone, firstName, lastName, kycStatus }`
4. `GET /v1/transactions` → `[ { id, type, amount, currency, status } ]`
5. `POST /v1/ips/initiate` → `{ transactionId, status, message }`
6. `GET /v1/recipients` → `[ { id, name, phone, iban } ]`
7. `POST /v1/recipients` → `{ id, name, phone, iban }`
8. `GET /v1/settings` → `{ notifications, privacy, security }`
9. `POST /v1/kyc/start` → `{ sessionId, uploadUrl }`
10. `POST /v1/dataaccess/request` → `{ requestId, status }`

11. `GET /v1/rates/RSD/EUR` → `{ from, to, rate, timestamp }`
12. `GET /health` → `{ status, version, timestamp }`

Evidence: Pact broker (all 12 interactions passing)

Visual Regression Testing (Playwright)

Framework: Playwright screenshot comparison

Baseline: `tests/visual-regression/baselines/`

Pages: 30 (all adapted pages)

Strategy:

- Take baseline screenshots (Samsung Galaxy A54 viewport)
- Compare on PR (fail if diff > 0.1%)
- Store diffs in artifacts

Results: Baseline established, no regressions on develop

Evidence: `.github/workflows/visual-regression.yml` (passing)

Observability Stack

OpenTelemetry (OTLP)

Instrumentation:

- **Backend:** ktor-opentelemetry plugin
- **Frontend:** @opentelemetry/sdk-trace-web

Exporter: OTLP/HTTP → Tempo

Traces:

- HTTP requests (method, path, status, duration)
- Database queries (SQL, duration)
- NBS IPS calls (external service)
- Redis operations (cache hit/miss)

Evidence: Tempo UI shows traces

Sentry

Integration:

- **Client:** @sentry/nextjs (frontend)
- **Server:** @sentry/kotlin (backend)
- **Edge:** @sentry/edge (Next.js middleware)

Events Captured:

- Unhandled exceptions
- Failed HTTP requests (5xx)
- Failed database queries
- NBS IPS errors
- Rate limit violations

Evidence: Sentry dashboard (0 errors in last 7 days on develop)

LGTM Stack (docker-compose profile)

Components:

- **Prometheus** — Metrics collection (15s scrape interval)
- **Grafana** — Dashboards (3 custom)
- **Loki** — Log aggregation (JSON logs from backend + frontend)
- **Tempo** — Trace storage (OTLP backend)
- **Mimir** — Long-term metrics storage

Run:

```
docker-compose --profile lgtm up
```

Services:

- Grafana: <http://localhost:3001>
- Prometheus: <http://localhost:9090>
- Tempo: <http://localhost:3200>

Grafana Dashboards (3)

1. Overview Dashboard

Panels:

- Request rate (req/sec)
- p50/p95/p99 latency
- Error rate (4xx, 5xx)
- Active users (last 5 min)
- Transaction volume (RSD)

2. Infrastructure Dashboard

Panels:

- PostgreSQL connections (active, idle, max)
- Redis hit rate
- CPU/memory usage (backend, frontend)
- Disk I/O (PostgreSQL)
- Network throughput

3. Errors Dashboard

Panels:

- Error rate by endpoint
- Top 10 errors (stacktrace)
- Unhandled exceptions (last 1 hour)
- Sentry events (last 24 hours)

Evidence: Screenshots in `docs/observability/grafana-dashboards/`

Prometheus Alerting Rules (16)

File: `backend/src/main/resources/prometheus-alerts.yml`

Rules:

1. **HighErrorRate** — Error rate > 5% for 5 min
2. **SlowRequests** — p95 latency > 1s for 5 min
3. **DatabaseConnectionPoolExhausted** — All connections in use
4. **RedisDown** — Redis unreachable
5. **NBSIPSDown** — NBS IPS API unreachable (5xx) for 5 min
6. **DiskSpaceNear90Percent** — Disk > 90% full
7. **OOMKill** — OOM killer triggered (container restart)
8. **HighRateLimitRejects** — 429 rate > 10% of requests
9. **UnverifiedOTPPileup** — 1000+ unverified phone_verifications
10. **FailedTransactionSpike** — Failed transactions > 10% of total
11. **KYCBacklog** — 500+ pending KYC sessions (> 24h old)

- 12. **AMLFlagUnresolved** — 50+ aml_flags not reviewed (> 4h old)
- 13. **DataAccessRequestOverdue** — ZZPL request > 30 days old
- 14. **BackupFailed** — Last backup > 24h ago
- 15. **CertificateExpiringSoon** — TLS cert expires in < 7 days
- 16. **AnomalousTransactionVolume** — Transaction volume 3σ above baseline

Evidence: Prometheus UI shows alerts (none firing on develop)

Evidence Matrix

Deliverable	Evidence Type	Status
Test pyramid	Unit 617 + Int 11 + E2E 15	☐ All layers
k6 load tests	4 scenarios passing	☐ p95 < 500ms
axe-core a11y	23 rules, 0 violations	☐ WCAG 2.1 AA
Pact contracts	12 interactions passing	☐ Backend ↔ Frontend
Visual regression	Baseline + no diffs	☐ 30 pages
OpenTelemetry	Traces in Tempo	☐ Backend + Frontend
Sentry	0 errors (7 days)	☐ Client + Server + Edge
LGTM stack	Docker profile running	☐ Grafana + Loki + Tempo
3 Grafana dashboards	Screenshots	☐ Overview + Infra + Errors
16 Prometheus alerts	YAML file + Prometheus UI	☐ None firing

Lead: Angie Jones (Proveo), Chip Huyen (AgentForge)

Validation: Petter Graff (CodeCraft)

Commit Range: `deveLop` branch

Revision #1
Created 2026-07-29 00:36:03 UTC by John
Updated 2026-07-29 00:36:03 UTC by John