

# Test Plan Document (Revised)

Project: seda.fm

Document Type: TEST PLAN

Version: 2

Created by: TEST\_LEAD

Created: Sat Sep 06 2025

---

Executive summary As Principal Software Test Lead, I present the revised, comprehensive QA/testing strategy and implementation plan for seda.fm's Recommendation Engine and UX surfaces. This version integrates Q&A clarifications into the original plan to ensure testability of residency guarantees, deletion/purge proofs, EU-runner behavior, admin override governance, sandbox tooling, and cohort privacy defaults. The plan preserves the original architecture and objectives, and adds concrete, actionable test cases, tooling expectations, acceptance criteria, telemetry assertions, and sandbox timelines that QA needs to deliver auditable, regulatory grade verification for production readiness.

Scope & priorities Core scope (MVP + compliance-critical items)

- Feed APIs (candidate gen + re-ranker), EmbeddingsService, vector store (pgvector), EU routing enforcement, admin/override flows, provider integrations (Spotify, Apple), real time (Socket.IO) presence/toasts, privacy export/delete, KMS audit hooks, testing sandboxes, and observability. High-risk / high-priority test areas
1. EU residency enforcement (authoritative users.data\_residency routing EU runner + EU Supabase writes; no cross-region writes without two approver override or per user opt in).
  2. Deletion/purge compliance and audit proof for the 30 day physical purge requirement (Production canonical approach: Option A — backup/PITR retention ≤30 days; Option B crypto erase supported as alternate in sandbox/enterprise).
  3. Embeddings spend-cap enforcement (simulate 80% & 100%) and admin override lifecycle.
  4. Provider integrations: Spotify token persistence & unlink delete behavior; Apple session-only imports.
  5. Real time presence: coalescing, optimistic join semantics, toast caps, privacy-based avatar suppression.
  6. Privacy export/delete with reauth + typed confirmation, privacy\_job lifecycle, and auditable proof artifacts (KMS logs for Option B where applicable).

## 7. Performance/ANN/pgvector SLOs and EU runner capacity/latency.

### Test objectives

- Validate correctness of business rules (residency precedence, pre-write validators, provider token policies, push permission flows).
- Validate robustness under provider and infra failures (provider 429s/5xxs, EU-runner outages).
- Validate cost-control automation and two approver override governance.
- Validate privacy guarantees and auditable deletion/export proofs aligned to production backup strategy.
- Validate real time UX constraints (coalescing, toast caps, optimistic join).
- Validate performance scalability and ANN latency targets.
- Validate security: secret handling, RLS, KMS key lifecycle, RBAC and two approver enforcement.

### Environments & sandbox tooling (deliverables, timelines, quotas) Environments

- Local dev, QA, sandbox-us, sandbox-eu, production parity. Platform deliverables (what QA will rely on)
- sandbox-us and sandbox-eu Supabase projects (pgvector enabled).
- EU self hosted SentenceTransformers runner (deployed in EU region) — canonical EU path for QA/MVP.
- Mock Provider Service (OAuth + failure/latency knobs) and Mock EmbeddingsService (deterministic vectors + fail/latency knobs).
- Event Replayer (server-side injection + headless Socket.IO clients).
- Admin/test APIs:
  - POST /admin/test/embeddings/simulate-spend
  - POST /admin/test/accelerate-purge
  - POST /admin/override/request, /admin/override/approve, /admin/override/execute
  - KMS test endpoints for sandbox (simulate key destroy)
- Pre-seeded persona and role accounts (qa users, SRE/approver/executor accounts), JWT helper, seed\_db scripts, and runbooks. Sandbox quotas & profiles (default)
- Smoke: ~1k users, 500 artists, 1k playlists, 200 rooms, 10k vectors. Target rec API 10–50 QPS; presence events 100–200 e/s.
- Medium: ~25k users, 50k artists, 50k playlists, 2k rooms, 150–300k vectors. Target rec API 50–200 QPS; presence events 1k–5k e/s (bursts to 10k).

- Large (scheduled): ~250k users, 250k artists, 300k playlists, 5–10k rooms, 500k–1M vectors. Target rec API 200–500 QPS; presence events sustained 5k–10k e/s (bursts to 20k on request). Sandbox SLAs & timelines
- Smoke/Medium: delivery within 3 business days.
- sandbox-eu (includes EU runner & KMS keys): 5–7 business days.
- Large or bespoke resource requests: 7–14 business days (advance booking required).
- On demand capacity increases: small increases within 24 hours; moderate increases require 48 hours; large runs require 3–7 business days lead time. Booking process documented in runbook.

## Test strategy & approach Testing pyramid

- Unit tests: business logic (residency precedence, pre-write validators, embedding\_meta handling).
- Integration tests: provider adapters, EmbeddingsService adapters, KMS envelope workflows, DB RLS behaviors, job workers.
- Contract tests: OpenAPI / endpoint contracts (BFF clients), provider adapter contracts.
- End to End tests: onboarding link providers feed join link/unlink privacy export/delete end-to-end (including sandbox accelerate hooks).
- Performance & Load tests: k6/Artillery (ANN, rec API), Event Replayer for real-time bursts.
- Security & Compliance tests: pen testing, RLS verification, KMS proofs.
- Chaos & resiliency tests: Mock Provider and Mock Embeddings failure injections, EU-runner outages, DB replication/backups tests. Automation & CI
- PR: unit + static + security checks (GitHub Actions).
- Merge: run integration + contract tests against sandbox (smoke).
- Nightly: medium performance tests + residency enforcement smoke tests (sandbox-eu).
- Release: canary with automated operational + product gates. Tooling
- Unit/Integration: Jest, Supertest.
- API contract: Pact/OpenAPI validators, Postman/Newman.
- E2E: Playwright (web + mobile flows).
- Load: k6 for HTTP/ANN; Event Replayer + headless clients for Socket.IO.
- Security: Snyk/OWASP Dependency Check, OWASP ZAP/Burp.
- Observability: OpenTelemetry traces, Prometheus/Grafana, Sentry.

## Test data plan Seed templates per profile (as above). Key test accounts:

- EU-resident (users.data\_residency='eu'), Non-EU, Enterprise (billing EU), Teen (dob 13–17), Privacy-conscious.

- Linked accounts: spotify\_linked (bg\_sync false), spotify\_linked\_bg\_on, apple\_session\_imported. Embeddings vectors
- Deterministic vectors from Mock Embeddings to enable reproducible nearest-neighbor results.
- embedding\_meta stored with provider, model, dim, region, provenance, override\_id where applicable.

Detailed test cases (representative, actionable) Each test case includes prerequisites, steps, expected outcomes, telemetry/assert checks, and cleanup.

#### A. Residency routing & cross-region enforcement

- TC R1: Residency precedence matrix
  - Prereq: user records with combinations of billing\_country, user\_declare, KYC, IP.
  - Steps: call residency resolution API; assert canonical users.data\_residency set by precedence. Check residency\_changes audit entry.
  - Assert: embedding jobs for users.data\_residency='eu' route to eu-runner; embedding\_meta.region == 'eu'.
- TC R2: Embedding routing & storage enforcement
  - Prereq: EU test user seeded in sandbox-eu.
  - Steps: trigger embedding compute; assert eu-runner invoked, vectors stored in sandbox-eu Supabase only, embedding\_meta.provider == 'eu-runner'.
  - Fail-case: attempt forced write to non-EU project -> ensure pre-write validator rejects and logs error.
- TC R3: Residency change workflow
  - Steps: user changes residency flag (self-declared); reauth + typed confirmation required; audit log created; background propagation job queued to re-route embeddings.
  - Validate: subsequent embeddings use new region and caches invalidated.

#### B. EmbeddingsService & spend cap enforcement

- TC E1: Normal embedding compute
  - Steps: compute user vector; assert embedding\_meta fields (provider, model, dim, region), job recorded, embedding\_audit entry created.
- TC E2: Simulate 80% spend (sandbox)

- Steps: POST /admin/test/embeddings/simulate-spend {percent:80}
  - Expect: enforcement engine triggers soft throttles, fallback to cheaper model, admin alert emitted, UI banner visible for admins; embeddings\_actions\_log entries.
- TC E3: Simulate 100% spend (sandbox)
  - Expect: non-essential jobs paused; only critical on demand cheap model allowed or queued; override required to resume; verify audit logs + override lifecycle.
- TC E4: Shadow-write constraints
  - Validate shadow writes limited to configured cohort; costs attributed; EU users not shadowed to non EU providers.

#### C. Admin override two approver flow (sandbox + prod acceptance)

- TC O1: Happy path
  - Steps: qa-requester creates override; two distinct approver accounts (technical + business/legal) approve; sre-operator executes.
  - Assert: override\_request record with approvals, execution log, actions performed, auto-revert scheduled; audit immutability.
- TC O2: Single approver execution attempt fails
  - Expect: execution rejected with 403, audit entry shows insufficient approvals.
- TC O3: Approver==executor negative test
  - Expect: attempt rejected.
- TC O4: Residency override requires Legal/DPO approver when scope affects EU routing

#### D. Provider integrations (Spotify & Apple)

- TC P1: Spotify link with background\_sync OFF
  - Steps: OAuth link; verify refresh\_token stored encrypted only after background sync consent; background\_sync\_enabled=false.
- TC P2: Background sync and token expiry
  - Simulate 429/invalid\_grant; assert sync\_status -> 'needs\_reauth' and user notification triggered.
- TC P3: Spotify unlink

- Expect: refresh\_token\_encrypted deleted; immediate\_exclusion true; privacy\_job queued; vectors removed after purge; embedding\_audit.deleted\_at set.
- TC P4: Apple session-only import
  - Expect: transient musicUserToken not persisted; imported summary triggers embedding compute; job logged.

E. Privacy/export/delete flows (Option A production path + Option B sandbox) Production canonical: Option A (backups/PITR retention ≤30 days). Sandbox supports both Option A (accelerated expiry) and Option B (KMS test destroy).

- TC D1: Export data
  - Steps: POST /privacy/export-data; assert export ZIP contains JSON + CSV manifest with embedding\_meta.provenance but no raw vectors or tokens.
  - Assert signed URL returned; audit event rec\_export\_queued.
- TC D2: Delete imports (reauth + typed confirmation)
  - Steps: reauth flow, typed confirmation, POST delete; assert immediate\_exclusion true, privacy\_job queued, caches invalidated, embedding\_meta.deleted\_at recorded.
- TC D3: Accelerated purge (sandbox)
  - Steps: POST /admin/test/accelerate-purge {privacy\_job\_id} ; assert privacy\_job completes; vectors removed from DB.
- TC D4: Crypto erase test (sandbox Option B)
  - Steps: delete + POST /admin/test/kmstest/destroy-key {dek\_id}; assert KMS deletion log entry; attempt decrypt of stored ciphertext fails; include KMS deletion request\_id in audit bundle.

Audit artifacts (required for DSARs)

- privacy\_job record, embedding\_audit entries with deleted\_at, immediate\_exclusion evidence, KMS deletion logs (Option B), backup snapshot metadata showing retention/expiry (Option A), purge completion proofs, and exportable compliance package (ZIP) with README mapping items to logs.

F. Real-time presence & toasts

- TC RT1: Presence coalescing & aggregation

- Steps: Event Replayer send join/leave events in 250–500ms windows; server aggregates and broadcasts per spec; client renders coalesced updates (500–1000ms).
- TC RT2: Avatar caps & privacy
  - Validate only allowed avatars shown; if a friend hides presence show count-only copy.
- TC RT3: Optimistic join idempotency
  - Use idempotency\_key across retries; ensure no duplicate joins; server reconciliation ack; UI rollback on hard failure.
- TC RT4: Toast caps and cooldown
  - Ensure max 3 toasts per session; 10-minute cooldown; verify server-side enforcement.
- TC RT5: Presence defaults & cohort behavior
  - Confirm presence default settings per cohort (see "Presence defaults" later) and immediate enforcement when toggled (SLO: visible within 2s, fully reconciled within 6s).

## G. Feed & ranking correctness

- TC F1: Candidate generation & re-rank workflow
  - Validate ANN -> top K -> re-rank weighting (embedding\_similarity, social boost, recency); enforce diversity constraint (>=30% novel content).
- TC F2: Dismiss semantics & undo
  - Dismiss (hide for 24h), Not interested (30 days), Never show (persist); ensure rec\_dismiss events influence re-ranker and persisted suppression table.

## H. Performance & Scalability

- Load tests (k6 + Event Replayer)
  - Rec API targets: base 50–200 QPS (Medium), spike to 500 QPS. SLOs: rec P95 <250ms (alert at >350ms).
  - pgvector ANN queries: P50 <50ms, P95 <120ms.
  - Real-time (Socket): sustained 1k–5k e/s; bursts to 10k; P99 end-to-end ≤500ms (MVP).
  - EU runner throughput targets (design targets):
    - Baseline sustained: 1,000 items/min.

- Provisioned initial capacity: 2,000 items/min; scale to 5,000 items/min; burst up to 10,000 items/min.
- Latency SLOs: P50 ≤100 ms, P95 ≤250 ms, P99 ≤1000 ms.
- Stress tests
  - Live concert scenario: event replayer 10k e/s burst for 60–120s; observe degradations; verify counts-only fallback and aggregation.
- Embedding pipeline throughput:
  - Background batch throughput target ≥1k items/min sustained; verify batch job behavior under throttling.

## I. Security & RLS testing

- TC S1: RLS policies
  - Tests for users\_select\_own, recommendations\_select\_user, service\_role-only access; clients cannot read others' rows.
- TC S2: Token & secret handling
  - Verify envelope encryption for refresh\_token\_encrypted, no plaintext logs, KMS-based unwrapping by least privileged workers.
- TC S3: Admin API authorization & RBAC
  - Ensure only authorized roles can call override/execute endpoints; require MFA and produce audit logs.
- TC S4: Pen test & dependency scans
  - OWASP ASVS checks, dependency scanning, Burp/ZAP against BFF and admin endpoints.

Observability verification & test telemetry Telemetry events to assert in tests:

- fact\_events: rec\_impression, rec\_click, join\_room, listen\_30s, rec\_dismiss, embedding\_use (with model/provider/region/override\_id), spend\_cap\_event, privacy\_job\_\* events, override.requested/approved/executed/reverted.
- Prometheus metrics: embedding\_cost\_runrate, rec\_api\_latency\_p95, socket\_fanout\_latency\_p99, queue\_depths, embedding\_provider\_error\_rate.
- Traces: correlation\_id across API embedding job vector write with job\_id. Audit artifacts retention and access

- KMS audit logs (cloud audit trail) retained per policy and attached to privacy\_job bundles for Option B tests in sandbox.
- Ensure QA can export an audit ZIP for DSAR validation.

## Automation & CI/CD integration

- GitHub Actions:
  - PR: run unit + static + security scans.
  - Merge main: run integration + contract tests against sandbox (smoke), accessibility (axe), quick E2E (Playwright).
  - Nightly: medium perf tests, residency enforcement smoke across sandbox-eu/us.
  - Release: canary with automated gates. Test automation patterns & helpers
- Helper libraries (node): seed\_user\_with\_imports(user), request\_delete\_and\_wait(user), accelerate\_purge(job\_id), simulate\_spend(percent), get\_override\_token override\_id).
- Idempotency & deterministic seeds: deterministic embeddings via Mock Embeddings for reproducible ANN tests.

## Acceptance criteria & exit gates (must pass) Functional

- Residency routing enforced: EU users embeddings compute & storage occur in EU Supabase and EU runner only, unless two approver override or per user opt in.
- Embeddings compute and embedding\_meta fields recorded for Medium seed profile.
- Provider link/unlink: Spotify link/unlink, Apple session import validated. Security & compliance
- RLS policies enforced, secrets KMS-wrapped, pen-test critical findings resolved or mitigated. Performance
- Rec API: P95 <250ms (Medium load); ANN P95 <120ms; socket P99 <500ms under expected load. Embedding cost enforcement
- Simulated 80%/100% spend triggers correct enforcement actions; admin override lifecycle validated end-to-end (sandbox). Privacy & purge
- Production canonical: Option A validated — backups/PITR ≤30 days; immediate exclusion + purge job flows validated.
- Sandbox: both Option A (accelerated expiry) and Option B (KMS destroy simulation) validated with proof artifacts. Operational
- EU-runner throughput & latency targets met for baseline load; emergency fallback only via two-approver override or per-user opt-in.

## Risk analysis & mitigations (top items)

## 1. Embedding cost overrun

- Risks: runaway model costs.
- Tests: simulate spend scenarios (80%/100%). Validate throttles, fallback model switch, admin notification, shadow write caps.
- Mitigations: automated enforcement, shadow-budget caps, two approver override with audit trail.

## 2. EU non-compliance / cross-region writes

- Risks: accidental non-EU processing of EU user data.
- Tests: negative tests that attempt cross-region writes; pre-write validators; audit checks.
- Mitigations: EU project isolation, pre-write validator, two approver governance, per-user opt-in; production default backup retention ≤30 days.

## 3. Real-time scale thrash

- Risks: fanout storm causing latency, increased infra costs.
- Tests: Event Replayer bursts; validate aggregation/coalescing and counts-only fallback.
- Mitigations: server aggregation, client coalescing, counts-only fallback, toast caps.

## 4. Provider outages

- Risks: missing provider data, token issues.
- Tests: Mock Provider failure simulations; validate circuit-breakers, retries, caching.
- Mitigations: caching, fallback models, reauth workflows, admin override only with approvals.

## 5. Privacy & DSAR evidentiary gaps

- Risks: inability to prove physical unrecoverability.
- Tests: audit bundle generation, Option A backup metadata retrieval, Option B KMS deletion proof simulation.
- Mitigations: predefined audit bundle fields, automated export, sandbox test hooks for KMS simulation.

### Presence defaults (cohort rules)

- Adult non-EU: friend-only presence = ON (default); public presence = OFF.
- Adult EU: friend-only presence = OFF (default); public presence = OFF.
- Teens (13-17): friend-only presence = OFF (default); public presence = OFF. Note: EU and privacy-conscious cohorts default to conservative settings. Tests must validate default API

responses and mutual visibility enforcement.

#### Pre-permission prompt gating (server authoritative) Definition of session

- Session begins on foreground/authenticated open; session\_end on background/close or after 30 minutes idle. Client emits session\_start, session\_heartbeat (every 5 minutes), session\_end. Meaningful engagement events (server-evaluated)
- join\_room, follow\_artist, save\_playlist (add), create\_playlist, import\_playlist, listen\_30s. Gating rule (server authoritative)
- show\_pre\_permission = true if (session\_count >= 2) OR (meaningful\_engagement\_flag == true)
- Teens/EU cohorts: stricter gating — e.g., require >=2 sessions AND a meaningful engagement for teens. Client behavior
- Client may preemptively display pre-permission UX, but only invoke OS push prompts after receiving server confirmation token. Dismiss/backoff rules: do not re-prompt for 7 days or until thresholds met.

#### Provenance UI rules (EU badge placement)

- Hybrid approach:
  - Show EU provenance badge in discovery module headers (once per module) with info tooltip linking to Settings Processing Details.
  - Per-card provenance shown only on demand (card detail modal / info icon) or for hero/promoted cards (max 1-2 per viewport).
  - Emergency non EU fallback: one-time concise in-app notification linking to Settings with override summary. Tests
- Assert module header badge presence for EU users; per-card badge only in detail modal; rec\_impression events include provenance metadata; ensure accessible and localized.

Observability & audit artifacts required for deletion/purge compliance For each privacy deletion/purge action deliver:

- privacy\_job record with job\_id, user\_id, reauth evidence, typed confirmation text, timestamps, job steps, final\_status.
- immediate\_exclusion evidence: API responses and feed snapshots pre/post deletion.
- embedding\_audit entries marked deleted\_at with job\_id.
- Backup snapshot metadata listing snapshot IDs containing the artifact and scheduled expiry (for Option A).
- DEK destruction proof (KMS audit log entry) when Option B used (sandbox simulated).

- purge completion record and reconciliation manifest.
- Exportable compliance package (ZIP) with README mapping artifacts for auditors.

#### EU self-hosted runner & operations (provisioning & SLA)

- Canonical EU path for QA/MVP: Platform/Infra provisions EU self hosted SentenceTransformers runner and EU Supabase project (no cross-region replication).
- Responsibility:
  - Platform/Infra: provisioning, KMS, networking, cluster lifecycle.
  - Backend SRE: operation and on call.
  - Backend Engineering: integration and model releases.
  - Data Science: model builds. SLA & capacity targets (for QA tests)
- Baseline cluster capacity: provision for 2,000 items/min sustained; scale to 5,000 items/min; burst 10,000 items/min.
- Latency SLOs: P50 ≤100 ms, P95 ≤250 ms, P99 ≤1000 ms.
- HA: multi-AZ in EU region, N+1 redundancy, autoscaling.

#### Test harness & admin/test APIs (sandbox expectations)

- POST /admin/test/embeddings/simulate-spend — simulate spend levels.
- POST /admin/test/accelerate-purge — accelerate purge pipeline for sandbox verification.
- POST /admin/test/kmstest/destroy-key — simulate DEK destruction and generate KMS deletion proof in sandbox.
- Override lifecycle APIs with two approver enforcement.
- Event Replayer templates for concert spikes and presence surges.

#### Detailed observability assertions for tests

- rec\_impression includes embedding\_meta.provenance.region.
- embedding\_use records with model/provider/region/override\_id.
- spend\_cap\_event emitted on simulated spend thresholds.
- privacy\_job lifecycle emits queued/in\_progress/completed events with correlation IDs.
- KMS audit logs (sandbox) included in audit package for Option B tests.

#### Test tooling, runbooks & pre-run checklist

- Provide seed scripts, jwt\_helper, and Event Replayer CLI templates.
- QA pre-run checklist:

- Confirm sandbox (us/eu) provisioned with chosen profile.
- Confirm test accounts and roles created.
- Run smoke scenarios: onboarding link Spotify feed join room.
- Validate admin/test endpoints accessible (simulate-spend, accelerate-purge).
- Book large runs with Platform if required.

Quality metrics & reporting Release report includes:

- Test pass % per suite, critical defects, performance SLO compliance, security findings, spend projections, residency enforcement checklist, privacy\_job test results. Weekly QA dashboards:
- Onboarding success, time-to-first-feed, rec\_ctr, listen\_30s%, skip\_rate, rec\_dismiss%, embedding\_cost\_runrate, override events.

Operational & test runbook snippets

- Residency enforcement quick check:
  - Create EU user in sandbox-eu; trigger embedding; verify embedding\_meta.region == 'eu' and vector present only in sandbox-eu Supabase. Negative test: attempt cross-region write -> pre-write validator rejects.
- 30-day purge quick check (sandbox acceleration):
  - Create test user + embeddings; call delete + accelerate-purge; verify vectors removed, embedding\_audit.deleted\_at exists, and for Option B simulate KMS destroy and attempt decrypt (must fail).

Open questions & actions required (remaining items for product/infra)

1. Production approver identities: QA needs canonical role mappings and named contacts (SRE Lead, Product Owner, Finance Lead, Legal/DPO) to map RBAC in prod. Provide role-to-principal mapping for RBAC tests in production readiness.
2. Backup/PITR production confirmations: QA requires confirmation that production projects storing user derived artifacts will have backup/PITR retention ≤30 days (Option A). If exceptions exist (enterprise customers), which projects are Option B-enabled?
3. KMS mode in production for Option B: if Option B used for specific tenants, define the KMS operational policy (creation, rotation, destruction window) and access to signed deletion records for audits.
4. Sandbox access timelines & quotas: Platform will confirm scheduling process and emergency contact for booked large runs (Platform has provided typical SLAs above; QA needs final approval channels).

5. Approver role lists for sandbox already provided; QA needs production equivalents or acceptance to use role mappings (SRE/Eng/Product/Legal/Finance) for prod RBAC validation.
6. UI provenance policy: confirm whether approver names (first name/initial allowed) are permitted in Settings Processing Details, or whether only roles should be shown. QA currently assumes roles + initials allowed if policy permits.

## Appendix — Representative test checklists (short)

- Residency smoke:
  - Create EU user compute embedding assert embedding\_meta.region=='eu' && vector only in EU DB.
- Delete + Option A purge:
  - Request delete immediate\_exclusion true accelerate-purge (sandbox) embeddings removed backup metadata captured.
- Delete + Option B (sandbox):
  - Request delete simulate KMS destroy attempt decrypt backup ciphertext must fail collect KMS deletion log.
- TwoApprover override flow:
  - Request -> Approve via two distinct approvers -> Execute by distinct executor -> assert override\_id present on affected jobs -> auto-revert at expiry.
- Presence default check:
  - Create adult non-eu / adult eu / teen accounts assert presence defaults via GET user/privacy.
- Pre-permission gating:
  - Simulate sessions & events to validate show\_pre\_permission server flag.
- Performance:
  - ANN P95 <120ms on Medium; rec API P95 <250ms; socket P99 <500ms; EU-runner P95 ≤250ms.

Closing / next steps I will produce the prioritized test implementation backlog, sample automated test scripts (GitHub Actions snippets and k6 scenarios), and acceptance checklists

for the first three sprints on confirmation of sandbox provisioning timelines and production backup policy (Option A confirmation). Please provide production approver role mappings and final confirmation of the backup/PITR policy so I can finalize RBAC tests and DSAR audit automation.

---

Generated by Feature Forge • 9/10/2025, 1:15:38 PM