

UI/UX Specification Document (Revised)

Project: seda.fm

Document Type: UI UX SPECIFICATION

Version: 2

Created by: UI_UX_DESIGNER

Created: Sat Sep 06 2025

Prepared by: Principal UI/UX Designer

Purpose Deliver a comprehensive UI/UX specification for seda.fm's Recommendation Engine surfaces (Rooms, Playlists, Artists) and associated flows (onboarding, linking, feed, real time presence, notifications, privacy/export/delete). This spec translates the Architecture Document, PRD, tech stack and Q&A constraints into actionable UX deliverables developers and PMs can implement.

Executive summary (TL;DR)

- Primary UX goals: fast first meaningful feed (<60s), low-friction onboarding (1–3 screens), clear privacy & consent, high social signal salience (friend-driven cards/toasts), accessible & inclusive design (WCAG 2.1 AA), robust undo/dismiss mechanics, and predictable notification guardrails (in-app first, FCM opt-in).
- Key constraints to honor (updated):
 - Provider linking is optional (Spotify persisted tokens; Apple session-only).
 - EU-only processing is a hard requirement for EU-resident users: store EU users' vectors and derived artifacts in a dedicated EU Supabase project/DB (no cross-region replication). Route embeddings to provider EU endpoints only if contractual guarantees exist; otherwise use an infra-provisioned EU self-hosted embedding runner.
 - Embeddings spend cap: automated enforcement at 80% and 100% thresholds (throttle background jobs + switch to low-cost model at 80%; hard stop of non-essential jobs at 100%). Overrides require two party approval and SRE execution; shadow-write budget limited to 5% of cap (\$75/mo).
 - Real-time latency and batching (coalesce updates 500–1000ms).
 - Push caps (1/day), toast caps (3/session, 10m cooldown).

- Deliverables here: personas, information architecture, detailed user flows, visual wireframes (described), UI components & patterns, microinteraction specs, accessibility requirements, instrumentation & testing plan, developer integration notes, risks & mitigations, and clarifying questions.
- New/critical UX additions from Q&A:
 - Explicit EU residency UI surfaces, provenance metadata, emergency-fallback messaging, and admin controls for overrides and audit.
 - Clear user-facing messaging about embedding-cost fallbacks and degraded-quality tradeoffs.
 - Feature-flag governance recommendations (LaunchDarkly).
 - Sandbox QA resources and test hooks (EU sandbox, mock providers, embedding runner, event replayer).
 - Default privacy/presence defaults adjusted for teens and EU cohorts.

1. Personas (brief)

- Fan Fiona (primary): 22–35, heavy listener, wants fast discovery and social cues. Will link Spotify after seeing value.
- Newbie Noah: 18–25, time poor, needs help finding rooms quickly. Must see a useful feed within 60s.
- Artist/Host Aria: 25–40, creates rooms and playlists; needs discoverability and control over presence/privacy.
- Privacy conscious Pat: 30–50, cares about data portability/deletion; requires transparent flows.
- EU Resident Eva: 25–45, expects data processed & stored within the EU; needs explicit residency indicators and guarantees.
- Teen Theo: 13–17, needs stricter privacy defaults and clear parental/age gating as required.

2. Information architecture (high level) Top-level app areas

- Home / Discovery Feed (Recommended Rooms, Playlists, Artists)
- Search / Explore
- Room (live experience, chat, now playing)
- Library / Profile (Linked accounts summary, preferences)
- Notifications Center (history of toasts & feed updates)
- Settings Privacy & Linked Accounts (includes Data Residency)
- Onboarding flows (account, genre selection, progressive linking prompts)

- Admin / Status (internal): override controls, spend dashboard, job audit for ops (not visible to end users)

3. Core UX principles & constraints

- Minimal mandatory friction: only account creation + genre selection required.
- Immediate perceived personalization: seed feed from genres + cached top-N per genre; enrich asynchronously with provider imports or embeddings.
- Progressive disclosure: provider linking, invites, and deep personalization are optional CTAs, not blocking steps.
- Social signals prioritized visually but respect privacy (show avatars only when allowed by both parties and cohort defaults).
- Predictable dismiss/undo and settings to export/delete data (30 day physical purge guarantee).
- Accessibility-first: WCAG 2.1 AA, keyboard & screen reader support, reduced-motion options.
- EU Residency: all EU-flagged users are routed to EU-only embedding/storage. UI must show provenance and handle EU-fallbacks gracefully.
- Embeddings Spend Cap: UI and user-facing copy must surface when cheaper-model fallbacks or throttles are in effect and explain expected quality tradeoffs.

4. User flows (detailed)

4.1 Onboarding — minimal critical path (1–3 screens) Goal: Account + genre selection first meaningful feed in <60s.

Screens:

1. Welcome / Account creation

- Elements: headline, benefits copy, “Sign up” options (email, OAuth), DOB gate (13+).
- Accessibility: large clear CTAs, text size $\geq 16\text{px}$.
- Age gating: enforce 13+; special stricter defaults for 13–17 (presence OFF by default).

2. Genre selection (required)

- UI: grid of curated genre cards with 100+ choices, search, quick-select “Popular” pills.
- Requirement: select at least 3; show progress indicator.
- Backend: persist user.preferences, seed taste_embedding derived from genres (no external API call).

3. Immediate feed screen (persisted)

- Show “Recommended for you” module populated with cached top rooms/playlists/artists for selected genres. CTA: Join / Follow / Open.
- Soft CTA: “Link Spotify for even better recs” with “Maybe later” option.
- Timing: First paint with feed skeletons <1s; enriched personalization update via Socket.IO within 60s if available.
- Push pre-permission: do not request OS-level push permission here. Defer per Notifications policy (see 4.5).

4.2 Provider linking (optional, progressive)

- Entry points: Settings → Linked Accounts, soft CTA post-onboarding, contextual prompts after 2–3 sessions.
- Spotify flow: OAuth popup; on success, persist refresh_token_encrypted; enqueue background import job; show progress (in-app toast + import status on profile).
- Apple Music flow: client obtains musicUserToken via MusicKit → POST to backend import endpoint; do not persist user token. Show ephemeral progress; if import long, return 202 and show progress via socket. UX rules:
 - Consent screen lists scopes and exact data types being imported.
 - Allow unlinking with clear explanation (imports deleted within 30 days).
 - Spotify: background sync is allowed (persist refresh token) and toggled OFF by default. If enabled, show last_synced_at and allow “Refresh now.”
 - Apple: session-only import only; present as “One-time import” with clear copy that tokens are not stored.

4.3 Home / Discovery feed and cards Module structure:

- Module header: “Recommended Rooms for You” + info icon (explain why recommendations), filter (All / Rooms / Playlists / Artists).
- Card layout: horizontal list or grid (mobile: 1-column, large cards; tablet/desktop: 2–3 columns). Card components:
 - Cover art (thumbnail), title, subtitle (top artists / short description), metadata chips (genre, member count), friend avatars group (up to 4), primary CTA (“Join” for rooms; “Follow” for artists/playlists).
 - Avatar group behavior: show up to 4 avatars (32px), overflow shown as +N circle; if avatars are hidden by friends’ privacy, show count-only copy.
 - Badges: “LIVE” (if room live), “New” for novel items.
 - EU provenance chip: where applicable (EU processed), provide a subtle region badge in Settings and optionally in module header (not required on every card). Interaction:
 - 1 tap Join from card. On tap: optimistic local join UI, Socket.IO join event, server ack. If fail, rollback + error toast with Retry.

- Tap title or preview opens Room detail modal with more metadata and “Join” button.
- Dismiss (X) on card: show Snackbar with Undo for 10s. Dismiss semantics:
- Quick dismiss = hide for 24 hours.
- Not interested = downweight for 30 days.
- Never show = persistent block until user reverses in Settings. Instrumentation: emit rec_dismiss with reason where provided.

4.4 Real time toasts and presence Toast rules:

- Delivered via Socket.IO for friend-driven triggers (first friend join, threshold-crossing).
- Cap: max 3 per session; 10 min cooldown per user between toasts; do not count against push cap.
- Toast content: small cover art, short copy (“Alex is here — join them” or “3 of your friends are in #indie”), CTA “Join”.
- Avatars: up to 3 in toast (40px). Update cadence:
- Client coalesces events and re-renders at up to 1 update/sec (500–1000ms) to avoid thrash. Accessibility:
- Use aria-live polite for toasts; visually focusable; allow keyboard dismissal. Presence defaults (updated):
- New adult users: Friend-only presence sharing is ON by default (opt-out available). Public presence OFF by default.
- Teens (13–17): both friend-only and public presence OFF by default (opt-in required).
- EU/high privacy cohorts: default OFF for friend-only presence (opt-in).
- Avatars shown only when both viewer and friend allow relevant visibility; otherwise show count-only copy.

4.5 Notifications (FCM) & notification center Push rules & opt-in timing (updated)

- Push is opt-in; do not request OS-level push permission during the mandatory onboarding path.
- Pre-permission trigger: show in-app pre-permission after user completes 2 sessions OR after 1 meaningful engagement (join room, follow artist, save playlist), or at latest by session 3.
- After user consents in-app, invoke the OS prompt. If user grants OS permission, keep in-app Push toggle OFF by default until user explicitly enables in-app Push (prevents accidental opt-in).
- Push sending conditions: user opted-in + score \geq threshold + friend_presence_count \geq 2 OR other special conditions.
- Daily cap: 1 strong rec push/day max. Notification center:

- In-app list of historical toasts & feed events (persisted for 7 days).
- Controls: “Pause pushes”, frequency preferences, Quiet Hours (default follows device DND; app default 22:00–07:00 local). Provide options: Follow DND (default), App default, Custom; Priority exceptions off by default.

4.6 Room detail and live UX Room card opens modal or page:

- Header: room name, cover art, genre, current participants, friend avatars + +N.
- Now playing preview: artist, track name, progress bar.
- Controls: Join, Follow host, Room rules. Live presence:
- Show live participant strip with up to 8 avatars (with overflow +N).
- Presence updates: server aggregates per 250–500ms; client re-renders at up to 1 update/sec. Optimistic join:
- Add local avatar placeholder; on server ack, replace; on fail rollback.

4.7 Settings: Linked accounts, Privacy & Data Residency Settings main:

- Sections: Account, Linked Accounts, Privacy & Data Residency, Notifications. Linked Accounts:
- Row per provider: provider icon, linked status, last_synced_at, actions (Refresh, Unlink, Export imported data).
- Spotify: persist refresh_token_encrypted; background sync toggle present and OFF by default (opt-in). “Refresh now” button available.
- Apple: session import only; show label “Session import only” and “Refresh now” triggers session flow. Privacy & Data Residency:
- Data Residency row: indicates processing region for user (“EU—stored” or region of processing). For EU users show “EU stored” badge and an expandable explanation.
- Export my data (POST /privacy/export-data): show progress indicator, expected time, and email notification once ready; signed URL valid 7 days.
- Delete imported data (POST /privacy/delete-imports): Re-auth required; confirmation modal requiring typed confirmation; show immediate exclusion from ranking and 30-day physical purge timeline; show job progress status with audit metadata.
- Hidden recommendations list: UI to view and unhide items; “Never show” management.
- Accessibility & COPPA:
- DOB verification in account settings.
- Users under 13 blocked; users 13–17 get stricter presence defaults.

5. Wireframes (descriptive) No visuals included; textual wireframe descriptions (developers/designers use these to create prototypes).

5.1 Onboarding genre selection (mobile)

- Top: progress bar (1/2)
- Center: grid of pill buttons with icons (genre name)
- Bottom: "Selected: 3" counter; primary CTA "Get Recommendations" (disabled until ≥ 3)

5.2 Feed — Recommended Rooms (mobile)

- Top: header with avatar, search icon, filter pill
- Carousel/vertical list of cards:
 - Each card: left: rounded cover art (72x72), overlay LIVE badge. Right: Title (semibold), subtitle (small), metadata chips row, avatar group inline, primary CTA (Join) as rightmost.
- Dismiss X at top-right of card.
- EU users: optional subtle provenance badge in header (Settings) and import modal; avoid noisy badges on every card.

5.3 Real-time toast (ephemeral)

- Small floating bar bottom-left: left avatar cluster, center text (2 lines max), right CTA "Join" and tiny X to dismiss.

5.4 Linked Accounts row (Settings)

- Provider icon + name on left, status (linked/unlinked), secondary text (last synced), right actions (Refresh, Unlink). Tapping Unlink opens confirm modal with re-auth and typed confirmation CTA for Delete imports.

5.5 Privacy delete/export modal

- Modal content listing what will be deleted/what will remain, legal text, re-auth input, typed confirmation, confirm & cancel buttons.
- After submit: show job tile with job_id, progress, ETA, and socket updates for status.

6. Design system & components (specs)

Core tokens:

- Typography: body 16px, h1 28px, h2 22px. Scalable up to 200%.
- Colors: high contrast primary, accessible semantic colors for success/error/neutral.
- EU & privacy badges: small shield icon + label with accessible tooltip. Components:
- Card: 8px radius, shadow elevation 2, accessible label & role.
- AvatarGroup: up to 4 avatars + overflow circle, alt text attributes, keyboard focus.

- Buttons: primary, secondary, ghost; full-width on mobile.
- Toasts/Snackbars: persistent for 10s for undo; include aria-live.
- Modals: trap focus, close on ESC, show accessible headline.
- Skeleton loaders: for initial feed and previews. Accessibility attributes:
- All images have alt text.
- Use aria-live regions for toasts and presence count changes; polite for toasts, assertive only for critical errors.
- Keyboard navigation for feed cards (tab to card, Enter to open, Space to activate Join).
- Reduced motion: respect prefers-reduced-motion CSS.

7. Microinteractions & timing

- Avatar update coalesce: 500–1000ms.
- Toast display: appear with fade (unless reduced-motion), stay visible for 5–8s; manual dismiss allowed.
- Optimistic Join: immediate UI change; show pending spinner up to 6s; on failure revert and show error with Retry.
- Card dismiss: remove UI immediately and show Snackbar with Undo for 10s.
- Provider import progress: show progress bar with estimated time; if >10s, convert to async (202 + on completion notify via Socket.IO).
- Embedding spend-cap UI:
 - When automatic 80% actions trigger, show non-blocking banner: “We’re temporarily using an optimized model to manage costs — personalization is still active but may be slightly less tuned.” Link to System Status.
 - At 100% hard stop, show contextual modal/notice on affected flows: “Personalization delayed — background enrichment paused to prevent platform-wide cost impact. Showing cached recommendations for now.” Offer “Notify me” CTA.

8. Accessibility & inclusivity checklist

- WCAG 2.1 AA contrast & scaling.
- Keyboard navigable components; aria labels.
- Screen reader flows for feed, toasts, and notifications.
- Reduced motion preference support.
- Clear, plain-language consent screens and deletion explanations.
- Font size minimum 16px body; touch targets $\geq 44\text{px}$.

9. Instrumentation and analytics (what to emit from UI) Event mapping (client → fact_events)

- rec_impression (include rec_id, rank, surface, embedding_meta.provenance.region)
- rec_click (rec_id, position, surface)
- join_room (room_id, source)
- listen_30s (track_id, rec_origin)
- track_skip (track_id, elapsed_time)
- rec_dismiss (rec_id, reason)
- provider_linked, provider_unlinked, export_requested, delete_imports_requested
- embedding_use (model, provider, region, cost_tier) — emit when embedding used so cost/quality can be correlated with UX signals
- spend_cap_event (threshold, action_taken) Performance metrics:
- time_to_first_feed, time_to_enrich, toast_received_time, join_ack_latency, embedding_latency (P50/P95/P99), socket_fanout_latency.

10. Testing & validation plan Usability:

- Moderated usability test with 12 users per persona; tasks: complete onboarding, find and join a room, link Spotify, unlink & delete imported data. Accessibility:
- Automated axe checks + 5 manual screen reader sessions (VoiceOver, NVDA). Performance validation:
- Measure first feed paint and enrichment times under simulated network variability.
- Real time SLO verification against Socket.IO targets (see Real-time section). A/B experiments:
- Candidate weight variations (content vs social) gated via LaunchDarkly; sample size $\geq 5k$ impressions/variant; $p < 0.05$, 80% power. QA sandbox:
- Seeded staging with Mock Provider Service and Mock EmbeddingsService; EU sandbox with self-hosted runner; Event Replayer and load profiles (Smoke/Medium/Large).

11. Developer integration notes / API UX contract highlights

- Endpoints (UX-critical):
 - GET /feed?user_id= — returns cached top-50 + freshness flag + embedding_meta.provenance fields.
 - POST /join-room {room_id, client_id, idempotency_key} – returns ack with event_id.
 - POST /privacy/export-data — enqueues job; returns job id.
 - POST /privacy/delete-imports — requires reauth token (POST /auth/reauth) and typed confirmation; returns job id & immediate_exclusion == true.
 - POST /integrations spotify/link — OAuth handshake; store refresh_token_encrypted.

- POST /integrations/apple/import — expects transient musicUserToken (discard after job).
- Real-time:
 - Socket.IO channels per user and per room. Presence events include reasonable payload size; friend-driven toasts sent to friends-of-actor only.
 - Client-side constraints: coalesce socket updates and re-render at 500–1000ms.
- Embeddings Service:
 - Must support per-user region routing, embedding_meta (provider, model, version, region), failover to EU runner, and cost-tier metadata for telemetry.
 - Spend-cap enforcement: EmbeddingsService must be able to switch to low-cost model on 80% threshold and stop non-essential jobs at 100%; expose admin signals so UI can surface status and provenance.
- Admin endpoints (internal):
 - GET /admin/spend-status
 - POST /admin/override-embeddings (two approver flow; SRE execution)
 - GET /admin/embedding-audit?user_id=
- Client-side constraints:
 - Respect server headers: x-degraded (served stale), x-retry-after.
 - Use embedding_meta to show provenance in export manifests and to drive UI indicators.

12. Acceptance criteria (UX)

- Onboarding: ≥90% of test participants can complete onboarding and see feed within 60s.
- Accessibility: WCAG 2.1 AA automated pass rate + manual screen reader acceptance.
- Toast/Push caps: enforce per-session toasts and 1 push/day server-side; UI reflects disabled push status.
- Privacy flows: export and delete jobs visible in Settings with progress and completion notifications.
- EU compliance: for EU-flagged users, embedding_meta.region == 'eu' for all embedding jobs; vectors stored in EU project; no cross-region writes in tests.
- Embeddings spend enforcement: at 80% and 100% automated actions occur and relevant UI banners/notifications are surfaced.

13. Risks and mitigations

- Risk: Slow provider imports delay personalization Mitigation: seed feed from genres + show progress; use cached templates and notify when enriched.
- Risk: Real time thrash with high join rates Mitigation: server-side aggregation, client coalescing, avatar count fallback.
- Risk: Users mistrust data linking Mitigation: explicit consent screens, visible export/delete flows with clear timelines.
- Risk: Over-notification perceived spam Mitigation: strict caps, score thresholds, cooldowns, per-user quiet hours.
- Risk: Accessibility regressions Mitigation: include accessibility checks in CI and manual testing in QA.
- Risk: EU non compliance / accidental cross-region writes Mitigation: dedicated EU Supabase project, embedding_meta provenance, pre-write hooks rejecting non-EU writes for EU users, EU sandbox for QA, and explicit emergency fallback governance + audit logs.
- Risk: Embedding spend overrun Mitigation: automated throttles & model-switching at 80%, hard stop at 100%, override governance with two-party approval and timeboxed overrides, user-facing messaging for transparency.

14. Open questions (for PM / Legal / Infra)

- Brand voice confirmation: use Warmly confident, music-savvy, inclusive (as specified). Confirm tone acceptance for privacy & cost-fallback messaging.
- Push quiet-hours defaults: confirm default 22:00–07:00 local is acceptable vs strictly respecting device DND only.
- EU fallback policy: confirm whether any enterprise/paid contractual exceptions should exist beyond the emergency override process defined here.
- Localization rollout timeline: initial English-only; confirm priority markets & i18n timeline for expansion.
- Admin access model: confirm which SRE / Product / Finance roles will have override approval authority and who will be the designated executor.

15. Design ops & next steps

- Deliver high-fidelity prototypes (Figma) for:
 - Onboarding feed (mobile + desktop)
 - Provider linking flows (Spotify + Apple)
 - Feed cards with presence avatars and EU provenance chip
 - Real-time toast interactions and notification pre-permission modal
 - Settings Linked Accounts, Privacy & Data Residency, Push controls and Quiet Hours
 - Export/Delete flow modal and job progress UI (with re-auth and typed confirmation)

- Provide:
 - Accessibility test scripts and checklist
 - 2-week usability test plan & recruitment criteria based on personas
 - UI copy deck aligned with voice & tone guidelines (discovery energetic; privacy calm/formal)
- Collaborate with engineering to wire mock EmbeddingsService and provider mocks into sandbox for E2E testing.

Appendices (operational & QA references)

A. EU Residency — UX guarantees & verification (summary)

- Default: All EU-resident users are EU-routed.
- UI surfaces:
 - Consent dialogs + Settings show “Processed & stored in the EU” badge.
 - Import modal shows “Processing in: EU” during enrichment.
 - Export manifest includes embedding_meta.region == 'eu'.
 - If emergency fallback used, UI shows explicit notice: override_id, expiry, and purge ETA.
- QA verification steps:
 - Confirm embedding_meta.region == 'eu' in audit logs.
 - Query EU Supabase project to assert vectors exist only there.
 - Use admin console toggles to simulate outage and test emergency override path (timeboxed).

B. Embedding spend-cap enforcement — UX & admin contract

- Automated responses:
 - 80%: throttle background jobs + switch on demand to low-cost model; surface non-blocking banner.
 - 100%: hard stop non essential jobs; queue or degrade on demand to cheaper model; show contextual modal on impacted flows.
- Override governance:
 - Two-party approval required (SRE/Engineering Lead + Product Manager or Finance). SRE executes override; must be timeboxed (default max 4 hours).

- Every override logged with justification and cost projection.

C. Shadow-write policy (embeddings)

- Allowed up to 1–5% traffic, default ramp 1–2%, monthly budget capped at 5% of embedding cap (\$75/mo).
- EU users must never be routed to non-EU secondary providers during shadow tests.
- Shadow writes invisible to users; require audit metadata and cost integration.

D. Real-time scale targets (Socket.IO + Redis adapter)

- Baseline (MVP): sustained 1k–5k events/sec; bursts to 10k; concurrent sockets 5k–10k.
- P99 end-to-end notification SLA: product target \leq 200 ms; MVP acceptable \leq 500 ms. If P95 $>$ 200ms or P99 $>$ 500ms for $>$ 2 minutes, begin degradations and consider broker.
- Triggers to add broker (Kafka/Redpanda): sustained $>$ 5k events/sec for $>$ 5 minutes, bursts $>$ 10k $>$ 60s, P99 latency violations, queue backlogs $>$ 1000 per worker.

E. Push & Toast policy (concise)

- OS push prompt: defer until after 2 sessions or 1 meaningful engagement, show pre-permission explainer first.
- Default Push toggle: OFF in-app; user must enable after OS permission.
- Toasts: in-app ephemeral, default ON, 3/session cap, 10m cooldown.
- Quiet hours: follow device DND by default; app default 22:00–07:00 local; custom hours optional at launch.

F. Provider tokens & sync rules (concise)

- Spotify: persist refresh_token_encrypted; background sync allowed (OFF by default); re-auth on failure; export & delete semantics applied.
- Apple Music: session-only imports; DO NOT persist musicUserTokens; manual refresh only.
- On unlink/delete: immediate exclusion from ranking; physical purge of embedded artifacts + tokens within 30 days. Show job progress & confirmations.

G. Export & Delete UX contract (concise)

- Re-auth required (POST /auth/reauth returning short-lived reauth_token).
- Typed confirmation required for destructive delete.
- Exports delivered as ZIP: JSON + CSV + README manifest (includes embedding_meta provenance; raw vectors excluded).

- Job endpoints & real-time socket events (privacy:job_update, privacy:job_completed) for UI progress.
- Staging acceleration: test-mode hooks in sandbox to validate purge behavior (sandbox-only admin endpoints).

H. Feature flags & experimentation

- Use LaunchDarkly for production experimentation; Supabase flags for dev/local.
- Governance: Feature Flag Council (PM, Engineering, SRE, DS, UX); Platform provisioned LaunchDarkly org.
- Canary ramp: 1% initial (15–30min op window), 5% (4–24h), 25% (48–72h), 50% if needed; automatic operational & product gates (latency, error rates, CTR, listen_30s, rec_dismiss).

I. QA sandbox & tooling (deliverables)

- Sandbox projects: sandbox-us + sandbox-eu (pgvector), seeded size profiles (Smoke, Medium, Large).
- Mock Provider Service: OAuth + API + failure knobs.
- Mock EmbeddingsService + EU self-hosted runner: deterministic vectors, batching, failure injection.
- Event Replayer: server-side injection + headless clients; scenario templates for presence spikes and long-running loads.
- Admin console for QA: region toggles, outage simulation, emergency override simulation (sandbox-only), embedding_audit viewer.

Acceptance sign-off

- UX sign-off requires prototypes + accessibility checks + QA validation on EU residency, export/delete flows, and embeddings cap behaviors in sandbox.

If useful, next immediate deliverable:

- High-fidelity prototypes (Figma) for Onboarding Feed Join (mobile + desktop), plus the Settings Privacy & Linked Accounts flows and the Export/Delete UI with job progress.

End of specification.