Design v1.1 — Pipeline-Aware Mapping

Project: AI-Enhanced Music Mashup Studio
Evolved from: Design v1 + Requirements v1.3
Principle: *If the pipeline cannot see it, it cannot evaluate it.*

---

1. Requirements-to-Components Mapping

Req ID	Requirement Summary	Component(s)	AI Involved?	Trust Risk
F1	Display BPM & key for selected tracks	TrackHeader, TrackMetadata, SpotifyService	No	Low
F2	Scrub forward/backward during playback	WaveformDisplay, AudioEngine	No	Low
F3	Simultaneous playback of ≥2 tracks	AudioEngine, TrackCard	No	Med
F4	Pause/resume/restart tracks independently	AudioEngine, TrackCard	No	Low
F5	EQ/volume changes isolated to selected track	BasicControls, CollapsibleEQ, AudioEngine	No	Med
F6	Runs entirely in browser, no hardware	App.js, AudioEngine (Web Audio API)	No	Low
F7	Save/load full project state	SaveButton, LoadButton, FirebaseService	No	Med
F8	Persist config without audio files	FirebaseService, Firestore schema	No	Med
F9	Custom name/label for saved mashups	PlaylistNameEditor, FirebaseService	No	Low
F10	Recommend tracks by BPM + key	AIRecommendationEngine, SpotifyService	**Yes**	**High**
F11	Show reasoning behind AI recommendations	SuggestionCard, AIContext, AIRecommendationEngine	**Yes**	**High**

F12	AI must not auto-replace user tracks	AIPanel, AddButton, App.js	**Yes**	**High**
F13	Full manual function if AI fails	AIPanel (graceful degradation), MainWorkspace	**Yes**	**High**
F14	Allow disabling AI pitch/tempo adjustments	PitchControl, SpeedControl, AudioAdjustments	**Yes**	Med
F15	Near-real-time audio update on control change	AudioEngine, BasicControls, AudioAdjustments	No	Med
F16	Fetch Spotify metadata within 5s	SpotifyService, TrackSearchModal	No	Med
F17	Limit recommendations to 5–10	SuggestionList, AIRecommendationEngine	**Yes**	Med
F18	Visual feedback for audio changes	WaveformDisplay, VolumeSlider	No	Low
F19	Disclose AI bias to users	AIContext, SuggestionCard	**Yes**	Med
NF1	AI suggestions are optional, not authoritative	AIPanel, AddButton	**Yes**	**High**
NF2	Warn on extreme pitch/tempo shifts	PitchControl, SpeedControl	No	Med
NF3	No third-party data sharing beyond Spotify	FirebaseService, SpotifyService	No	**High**
NF4	Prioritize responsiveness over audio fidelity	AudioEngine	No	Med
NF5	Notify user on API/network failures	SpotifyService, FirebaseService, error UI	No	Med
NF6	No copyright compliance guarantee	N/A — disclosure only	No	Med

⚠ **Requirements that cannot be satisfied without AI:** F10, F11, F12 (AI guard), F13 (AI fallback path), F17, F19, NF1

⚠ **Requirements likely to fail silently:** F12 (no UI block = silent bypass), F11 (explanation could be hardcoded), NF3 (data flow invisible to user), F13 (AI failure may not surface)

2. AI Entry Points

Component	AI Activity	Observable Signal	Risk
AIRecommendation Engine	Scores BPM/key/genre compatibility; ranks tracks	Compatibility score (0–100%) logged per suggestion	Wrong weights → misleading scores silently
SpotifyService.getRecommendations()	Calls Spotify's recommendation endpoint (external ML)	HTTP response code + returned track IDs logged	Spotify model bias; no introspection available
SuggestionCard	Renders AI-generated explanation text	Explanation string visible in UI + logged payload	Hardcoded/template text could misrepresent actual scoring
AIContext	Adapts context message to playlist state	Context string rendered in UI	Static fallback indistinguishable from dynamic reasoning
AIPanel	Triggers full recommendation pipeline	API call timestamp + result count logged	Silent empty result if AI fails and fallback not shown
AIRecommendation Engine.explainRecommendation()	Generates human-readable reason for match	Explanation string logged alongside score factors	Explanation may not match actual scoring factors used

3. Components-to-Pipeline Stages Mapping

Component	Pipeline Stage(s)	Why This Stage	Observable Behavior
App.js, AuthProvider	Build, Deploy	Root bundle; auth wiring must be present at deploy	App loads without crash; auth redirect works

SpotifyService	Build, Test, Monitoring	API calls must be validated; rate limits observable at runtime	HTTP response codes, latency metrics
FirebaseService	Build, Test, Deploy, Monitoring	CRUD must be correct; data schema validated; runtime writes observed	Read/write success rates in Firestore logs
AudioEngine	Build, Test, Integration	Web Audio API compatibility; audio chain correctness	No crash on segment play; output audible
AIRecommendationEngine	Build, Test, Integration, Monitoring	Scoring logic testable in isolation; runtime outputs need observation	Score range [0–100], recommendation count, explanation presence
TrackCard, TrackSettings	Build, Test	UI rendering; control state correctness	Settings persist after reload; no state corruption
WaveformDisplay	Build, Test	Waveform renders from Spotify audio analysis data	Waveform visible within 5s of track add
AISidebar / AIPanel	Build, Test, Integration, Monitoring	AI display + trigger logic; end-to-end recommendation flow	Suggestions appear; fallback shown if AI fails
SuggestionCard	Build, Test	Renders score + explanation correctly	Score displayed; explanation non-empty
TrackSearchModal	Test, Integration	Search + add flow; Spotify API dependency	Track added to stack within 5s
SaveButton / LoadButton	Test, Integration, Deploy	Persistence round-trip correctness	Saved state fully restores on load

Header	Build, Deploy	Navigation always present	Header visible; no z-index/layout breaks

---

4. Candidate Trust Gates

Gate G1 — AI Score Sanity Check

- **Location:** Test stage (unit) + Monitoring (runtime)
- **Checks:** analyzeCompatibility() returns a number in [0, 100]; never NaN or null; BPM sub-score and key sub-score are each present
- **Evidence Collected:** Score value, sub-scores (BPM, key, genre), input track IDs, timestamp
- **Initial Threshold:** Score ∈ [0, 100]; sub-scores sum to ≈100% weight [PLACEHOLDER — needs calibration]
- **Failure Action:** Log warning; substitute score = 0; do not surface recommendation
- **Linked to:** F10, F11, NF1; AIRecommendationEngine; AI Entry Point row 1

Gate G2 — Recommendation Count Boundary

- **Location:** Integration stage + Monitoring
- **Checks:** generateRecommendations() returns between 3 and 10 suggestions; empty result triggers fallback UI
- **Evidence Collected:** Result count, Spotify API response code, engine input (seed tracks)
- **Initial Threshold:** count ∈ [3, 10] [ARBITRARY — 3 minimum assumed for UX; adjust after user testing]
- **Failure Action:** If count = 0 → show "No suggestions available" UI (not silent); if count > 10 → truncate + log
- **Linked to:** F13, F17; AIPanel, SuggestionList; AI Entry Point rows 4–5

Gate G3 — Explanation Presence Check

- **Location:** Test stage (unit) + Monitoring
- **Checks:** explainRecommendation() returns non-empty string; string references at least one of: BPM, key, genre
- **Evidence Collected:** Explanation string, triggering score factors, presence of required keywords
- **Initial Threshold:** String length > 10 chars; contains at least one of ["BPM", "key", "genre"] [WEAK CHECK — improve in v1.2 with semantic validation]
- **Failure Action:** Log warning; display generic fallback text flagged as "generic" in UI
- **Linked to:** F11, F19; SuggestionCard, AIContext; AI Entry Point row 3

Gate G4 — AI Non-Replacement Guard

- **Location:** Integration stage + Monitoring
- **Checks:** Track stack state does not change unless user explicitly clicks "+ Add to Mix"; no automatic track insertion/reorder by AI code paths
- **Evidence Collected:** Track state before/after AI panel interaction; action origin (user click vs. code)
- **Initial Threshold:** Track array unchanged after recommendation fetch unless add action logged [PLACEHOLDER — requires event-sourced audit log]
- **Failure Action: Block pipeline** if AI code path can write to track state without user action; flag for immediate human review
- **Linked to:** F12, NF1; AIPanel, App.js; AI Entry Point row 4

Gate G5 — Spotify API Failure Visibility

- **Location:** Monitoring (runtime)
- **Checks:** Any Spotify API call returning non-2xx triggers visible user notification; silent fallback without notification is a failure
- **Evidence Collected:** HTTP status code, endpoint called, error message, user-facing notification fired (boolean)

- **Initial Threshold:** 100% of non-2xx responses must produce a user-visible error OR explicit silent-failure log entry [ARBITRARY — 100% may be relaxed for transient 429s]
- **Failure Action:** Log; show user notification; do not silently degrade
- **Linked to:** NF5; SpotifyService, AIPanel, FirebaseService; AI Entry Point row 2

Gate G6 — Audio Degradation Warning Gate

- **Location:** Test stage + Runtime (event trigger)
- **Checks:** When pitch > ±3 semitones OR speed outside [0.85, 1.15], warning UI fires before audio applies
- **Evidence Collected:** Pitch/speed value at time of warning; whether user acknowledged or dismissed
- **Initial Threshold:** Thresholds as stated in NF2 [PLACEHOLDER — ±15% tempo = ±0.15 multiplier, mapped to [0.85, 1.15]]
- **Failure Action:** Log if warning not shown despite threshold exceeded; human review flag
- **Linked to:** F14, NF2; PitchControl, SpeedControl; no direct AI entry point (human-triggered)

---

5. Failure Visibility Analysis

Component	Silent Failure Scenario	Pipeline Detects?	Instrumentation Needed
AIRecommendationEngine	scoreByKey() returns 0 for all tracks due to key format mismatch (e.g., "Am" vs. "A minor"); all scores artificially low; user sees bad suggestions with no error	❌ No — scores look valid	Log key format received vs. expected; assert normalization in test; alert if >80% of scores = 0
AIPanel	Spotify getRecommendations()	❌ No — empty UI	Assert result count > 0 or trigger fallback UI

	returns 200 but empty array; SuggestionList renders nothing; no error shown	with no feedback	path; log empty-result events
SuggestionCard	explainRecommendation() returns template string not tied to actual score factors; user shown misleading reasoning	❌ No — looks correct	Log explanation + actual score sub-factors together; Gate G3 partial mitigation
FirebaseService	savePlaylist() resolves successfully but writes partial data (segment array truncated by Firestore limit); load returns incomplete state	❌ No — save shows success	Checksum or count field on save; verify count on load; integration test round-trip
SpotifyService	Token refresh fails silently; subsequent API calls return 401 but error swallowed in catch block; app appears to work but fetches stale/no data	❌ No	Wrap all API calls with response code logging; surface 401 as auth-expired notification; Gate G5

---

6. Gap Identification

Components with no clear pipeline stage:

- DragHandle.js — pure UI, no test coverage specified
- AddSegmentButton.js — no integration test for segment-add → Firestore write path

AI entry points with no trust gate:

- SpotifyService.getRecommendations() (external ML model) — no gate on *what Spotify returns*, only on downstream count (G2)
- AIContext.js context message adaptation — no gate on whether context message is accurate to playlist state

Trust gates with no evidence source:

- G4 (Non-Replacement Guard) requires event-sourced audit log that does not currently exist in design

Requirements with no component mapping:

- NF6 (copyright disclaimer) — disclosure only; no component surfaces this in the UI; needs a DisclaimerBanner or equivalent
- F19 (bias disclosure) — partially covered by AIContext but no dedicated component or logged event

Pipeline stages with no trust gates:

- **Deployment stage** — no gate before pushing to staging/production (e.g., smoke test, auth sanity check)
- **Build stage** — no gate on bundle size or Web Audio API compatibility check

---

7. Red Flags Assessment

🚩 **RF1 — AI explanation text is cosmetic, not derived**
explainRecommendation() could return template strings ("Great harmonic match!") disconnected from actual scoring. Gate G3 is a weak mitigaton.
Remediation: Log score sub-factors alongside explanation string; add test asserting explanation content correlates with top-weighted factor.

🚩 **RF2 — No audit trail for AI non-replacement (F12/NF1)**
There is no event log proving that track state changes originate from user actions, not AI code paths. Gate G4 cannot function without this.
Remediation: Implement an action-origin field on all track state mutations (values: "user" | "ai_suggestion" | "system") before v1.2.

🚩 **RF3 — Spotify external ML is a black box with no trust gate**
getRecommendations() delegates to Spotify's own model. The design has no gate on what that model returns — biased, irrelevant, or low-quality results pass through silently.
Remediation: Add post-fetch validation gate that re-scores Spotify results using AIRecommendationEngine and filters out suggestions below a minimum threshold before display.

🚩 **RF4 — Firebase save "success" is not a correctness signal**
savePlaylist() resolving does not guarantee data integrity (partial writes, schema drift). This is a silent failure risk for F7/F8.
Remediation: Write a round-trip integration test (save → load → diff) that runs in the integration pipeline stage.

🚩 **RF5 — Deployment stage has no trust gate**

There is no defined smoke test or go/no-go check before production deployment. Any broken auth, blank AI panel, or missing Spotify credentials would be caught only by users. *Remediation: Add a minimal deployment gate: auth flow reachable, Spotify token exchange returns 200, AI panel renders at least one suggestion in staging environment.*

Visual Trace (Key AI Path)

F10/F11/NF1 → AIRecommendationEngine → Integration Stage → Gate G1 (Score Sanity)

→ Gate G2 (Count Boundary)

→ Gate G3 (Explanation Presence)

→ SpotifyService.getRecommendations() → Gate G5 (API Failure Visibility)

→ SuggestionCard (display)

→ [User clicks "+ Add to Mix"] → Gate G4 (Non-Replacement Guard)

→ App.js track state update

F7/F8 → FirebaseService.savePlaylist() → Integration Stage → Round-trip test [GAP: no gate]

→ FirebaseService.loadPlaylist() → Monitoring → RF4 flag