NDYRA Amendment — Signals + Aftermath Interaction Layer

Audience: Aelric Architect (Lead Dev)
Owner: William Davis Moore
Status: Approved (treat as law)

1. Purpose

This amendment defines two NDYRA interaction modules that add familiarity and daily retention without allowing platform drift:
• Signals: NDYRA's disciplined, ephemeral, media-first layer.
• Aftermath: an optional biometric performance reveal attached to a post/Signal and shown via tap-and-hold.

Aelric: you are empowered to choose the cleanest implementation that fits the current codebase and database, but you must preserve existing structural rules (RLS-first, can_view_post gating, no parallel stacks).

2. Non-Negotiables

- Audio is muted by default; user taps to hear (do not autoplay audio).
- Signals limits are enforced at the database layer: 2 active Signals per user; 10 active Signals per gym; 10 active Signals per club.
- Signals expire at 24 hours by default. Users can Pin (convert to Highlights) to keep them on profile.
- Aftermath is optional and privacy-controlled. It must never leak via related tables/media/metadata.
- No new permissive RLS policies. No 'select true' policies for post-adjacent data.
- All post-adjacent reads (media, reactions, comments, stats, Aftermath payload) must be gated through can_view_post(post_id).
- No parallel media pipeline. Reuse existing upload/Storage strategy and moderation primitives.
- AI inspiration voiceover is PAUSED. Do not implement voice generation or voice UI in the current build. Leave the architecture extension-friendly for future reconsideration.
- Smoothing filter is allowed, but it must be limited to skin smoothing (Off/Low/Medium). No facial reshaping.

3. Interaction Model (Gesture Law)

Posts and Signals share the same gesture language to remain familiar and consistent:

- Single tap: open full-screen viewer for media OR open Post Detail for text-only. In full-screen, single tap toggles play/pause for video and toggles UI chrome.
- Double tap: quick react (default reaction is fire).
- Tap-and-hold: reveal Aftermath overlay (only if Aftermath exists for the post). Release to dismiss.

Gesture conflict rules (must implement):

- Tap-and-hold threshold: ~300ms.
- If finger moves beyond a small threshold, cancel Aftermath reveal.
- If user is interacting with reaction tray, do not trigger Aftermath.
- If Aftermath is not available, tap-and-hold does nothing (no error toast).

4. Signals Module

4.1 What a Signal is

A Signal is an ephemeral 24-hour media moment designed for training identity (not vanity). Signals support photo, short video, workout card, optional music, and curated stickers/GIFs.

4.2 Signal Content Types

- Photo (portrait 9:16 preferred; accept 4:5 and 1:1 with safe cropping).
- Video (max 45 seconds for MVP; enforce file size and duration caps client-side and server-side).
- Workout card (auto-generated from timer/class/imported workout).
- Optional overlays: curated text (fonts) + curated stickers/GIFs + optional music.

4.3 Limits and Expiration

Limits (enforced in DB, not UI):

- User: max 2 active Signals.
- Gym: max 10 active Signals (staff-only by default).
- Club: max 10 active Signals (club-admin by default; membership posting can be enabled later).
- Expiry: signals default to expires_at = created_at + interval '24 hours'.

4.4 Editing Controls (Disciplined Expression)

Allow expression, enforce guardrails, preserve NDYRA look:

- Fonts: curated set of 4 NDYRA fonts only (Sans, Serif, Condensed, Mono).
- Overlays cap: max 3 overlays total (text + sticker + GIF count together).

- Font cap: max 2 distinct font styles per Signal.
- GIFs: curated NDYRA pack only; max 1 GIF per Signal.
- Stickers: NDYRA pack only; keep monochrome/white with red accent variants.
- Music: curated clip library (MVP). Avoid open catalogs until licensing is solved.

4.5 UI Surfaces

- FYP: SignalStrip at top (circles with ring state). Priority order: self -> gyms -> clubs -> following.
- Full-screen Signal Viewer: swipe left/right; tap toggles UI; tap to unmute; react; report; mute account; reply later (optional).
- Create Signal: camera/import + overlays + music + stickers; show remaining slot indicator (e.g., 1/2).
- Profile: Signals row + Highlights row under performance strip.
- Gym profile: Gym Signals row (max 10). Visibility respects follower/member rules.
- Club page: Club Signals row (max 10) + members grid with streak-first stats.

4.6 Required States (UI)

- Empty: user has 0 active Signals -> show '+ Add Signal'. Gyms/clubs with 0 -> hide row or show placeholder tile.
- Loading: show skeleton circles in SignalStrip and skeleton full-screen card.
- Error: show retry action (do not throw global toast loops).
- Expired: should disappear automatically; pinned items move to Highlights.

5. Aftermath Module

5.1 What Aftermath is

Aftermath is a compact biometric performance overlay captured after training and optionally attached to a post or Signal. It is revealed via tap-and-hold (not a permanent dashboard). It reinforces the NDYRA identity loop.

5.2 Aftermath Data (MVP Fields)

- Workout type / label
- Duration
- Average HR
- Max HR
- HR zone time summary (optional if available)
- Calories (if available)
- Streak delta (optional)
- Recovery score (optional, if available)

5.3 Optional Context Toggles

After workout completion, the user can choose to include source context in Aftermath:

- Include Gym (if workout was completed at/through a gym tenant).
- Include Timer Template (if workout came from a saved timer template).

Defaults:
• If workout originated from a gym class/check-in: Include Gym = ON by default.
• If workout originated from a personal template: Include Template = ON by default.
User can deselect either before posting.

5.4 Aftermath Overlay UI Spec (Minimal but Precise)

- Triggered by tap-and-hold on post in feed OR in full-screen viewer.
- Overlay behavior: dim background (80-90%), fade in a centered card, subtle red accent line/glow.
- Dismiss: release finger; fade out within ~100ms.
- Card layout: title 'Aftermath' + primary metrics (2 columns) + optional tags (Gym, Template) if toggled.
- If viewer lacks permission to gym/template, hide that row (do not show locked label).

5.5 Privacy Rules (Hard)

Aftermath must obey the strictest visibility rule between (a) the post/Signal visibility and (b) any gym/template visibility. Do not display gym or template references that the viewer cannot access. Aftermath payload must be filtered server-side (or stored already filtered) so clients cannot infer hidden data.

6. Implementation Guidance (No Parallel Systems)

Preferred: implement Signals as a post subtype to avoid a second social system. Aftermath is stored as structured JSON attached to the post/workout reference. If the existing schema already has posts and post_media, extend them instead of creating standalone signals tables.

6.1 Data Model Option A (Preferred): Extend posts

Add/extend these columns on posts (names can be adjusted to match existing conventions):

- kind: 'standard' | 'signal' (enum or text)
- expires_at: timestamptz NULL (required when kind='signal')
- pinned_to_profile: boolean default false
- aftermath: jsonb default '{}'::jsonb (optional; only present when attached)

Recommended indexes (partial where possible):

- posts(kind, expires_at) WHERE kind='signal'
- posts(author_user_id, kind, expires_at) WHERE kind='signal'
- posts(author_tenant_id, kind, expires_at) WHERE kind='signal'
- posts(club_id, kind, expires_at) WHERE kind='signal'

6.2 Enforcing the 2/10/10 Limits

Limits must be enforced in DB via trigger or RPC, not in frontend. Definition of active: kind='signal' AND expires_at > now() AND pinned_to_profile=false AND is_deleted=false.

Pseudo logic (illustrative):

```
IF NEW.kind = 'signal' THEN
  ASSERT NEW.expires_at <= now() + interval '24 hours';
  IF NEW.author_user_id IS NOT NULL THEN
    ASSERT (SELECT count(*) FROM posts WHERE author_user_id=NEW.author_user_id
AND kind='signal' AND active) < 2;
  ELSIF NEW.author_tenant_id IS NOT NULL THEN
    ASSERT (SELECT count(*) FROM posts WHERE
author_tenant_id=NEW.author_tenant_id AND kind='signal' AND active) < 10;
  ELSIF NEW.club_id IS NOT NULL THEN
    ASSERT (SELECT count(*) FROM posts WHERE club_id=NEW.club_id AND
kind='signal' AND active) < 10;
  END IF;
END IF;
```

6.3 Expiry Cleanup

Run a scheduled cleanup job (hourly or daily):

• For expired Signals: set is_deleted=true OR move to an archive state.

• If pinned_to_profile=true: convert to Highlight item and clear expires_at (or keep as pinned highlight).

6.4 Highlights (Pinned Signals)

If Highlights already exist, reuse them. If not, implement as collections referencing post ids:

• highlights(id, owner_user_id, title, cover_media_path)

• highlight_items(highlight_id, post_id, position)

Pin action creates/updates highlight, inserts item, and marks the post pinned_to_profile=true.

6.5 Storage + Media Rules

Do not invent a second upload system. Use existing post_media strategy.

- Bucket: post-media (or existing). Use a clear naming convention (e.g., p/{post_id}/{uuid}.jpg).
- Signal edits (stickers/text/music reference) should be stored as JSON metadata, not baked into image/video for MVP.
- Audio tracks (music) should be separate media entries (media_type='audio') or stored in a dedicated bucket if the current system requires separation.

6.6 Query Blueprint (Supabase)

Do not create a second feed system. SignalStrip is a scoped query over posts where kind='signal' and active.

Example (conceptual):

```
-- SignalStrip: self + followed gyms + followed users (active only)
select id, created_at, author_user_id, author_tenant_id, club_id, expires_at
```

```
from posts
where kind='signal'
  and expires_at > now()
  and is_deleted=false
  and (
    author_user_id = auth.uid()
    or author_user_id in (select followee_id from follows_users where follower_id =
auth.uid())
    or author_tenant_id in (select tenant_id from follows_tenants where follower_id
= auth.uid())
    or club_id in (select club_id from club_members where user_id = auth.uid())
  )
order by created_at desc
limit 50;
```

7. 2026 Best Practice Notes (Short, Practical)

- Muted-by-default audio avoids user annoyance and is consistent with modern premium UX.
- Avoid open GIF libraries early; curated packs drastically reduce moderation overhead.
- Skin smoothing: implement primarily in native capture flow (Capacitor/React Native) using device GPU. Expose Off/Low/Medium only.
- Do not transcode video server-side in MVP unless required. Enforce upload constraints and generate thumbnails; transcode later if needed.

8. Analytics (Must Instrument)

Signals and Aftermath are retention levers; instrument them from day one:

- signal_created (type=photo|video|workout_card, has_music, has_gif, has_stickers)
- signal_viewed (viewer_id, signal_id, completion_percent)
- signal_audio_enabled (music)
- signal_reacted (reaction_type)
- aftermath_attached (include_gym, include_timer)
- aftermath_viewed (hold_duration_ms)

9. Moderation + Safety

- Signals are content: reporting, blocking, and takedown rules apply exactly like posts.
- Rate-limit Signal creation for new accounts.
- Auto-hold content if user is in slow-mode/strike state (use existing moderation primitives).
- Never expose reporter identity to reported users.

10. Anti-Drift Gate (Before Merge)

Aelric: treat these as merge blockers for Signals/Aftermath changes:

- Run AntiDrift audit script; zero new permissive policies; RLS enabled on every new/changed table.

- Run RLS regression tests to confirm no private leaks (post_media, reactions, comments, stats, Aftermath payload).
- Confirm can_view_post() gates every post-adjacent select policy.
- Confirm limits are enforced in DB (attempting to exceed 2/10/10 via API or direct insert must fail).
- Confirm no new parallel folder/framework patterns were introduced without updating blueprint governance rules.