

Group Synced Meditation — Preliminary Spec (v0)

Agreed constraints

- Max participants: 20 (hard cap enforced server-side on join)
- Participant video: OFF
- Leader media: video + audio broadcast
- Participant audio: floor-controlled, one speaker at a time only (raise-hand queue)
- Chat: excluded in v1 (voice + reactions only)
- Room access: invite link + join code only; no waiting room
- Avatar dynamics: discrete, low-frequency presence signals only
- joined / left (optional: reconnecting)
- speaking on/off (only for the current floor holder)
- hand raised on/off (queue)
- reaction (enum) with TTL auto-expire (e.g., 8-12s)
 - acknowledge / confused / agree / question (expandable later)

Core architecture

Two planes: 1) Control plane (state, presence, reactions, chat, moderation) - Supabase Realtime (preferred) or small WS service - Authoritative room_state with start_at_ms for sync - room_member_state snapshot for late joiners - room_events append-only for audit/moderation

2) Media plane (leader A/V, optional participant audio) - WebRTC via managed SFU (recommended) - Short-lived, server-minted tokens - Role-based publishing permissions (leader always; participant only when granted floor)

Sync model (breath timer)

- room_state carries:
- version (monotonic)
- phase: prestart | countdown | running | paused | ended
- start_at_ms (server time, includes countdown offset)
- breath_pattern { inhale, hold1, exhale, hold2 }
- duration_sec
- Clients estimate server offset and compute elapsed = now_est - start_at_ms
- Visual drift correction: ease-correct if drift exceeds threshold (no hard jumps)

Data model (suggested)

- rooms: id, title, created_by, leader_user_id, is_live, max_participants, locked, policy
- room_state: room_id, version, phase, start_at_ms, breath_pattern json, duration_sec, flags
- room_members: room_id, user_id, role (leader/cohost/mod/participant), joined_at, last_seen_at, muted, banned
- room_member_state: room_id, user_id, avatar_ref, hand_raised, reaction, speaking, floor_granted

- room_events: room_id, seq, type, payload json, created_at, actor_user_id
- chat_messages (optional): room_id, id, user_id, text, created_at, deleted_at/by

Event types (discrete)

- member_joined / member_left
- hand_raise {value}
- reaction {kind} (TTL)
- request_floor / grant_floor {value}
- speaking_start / speaking_stop (only valid for floor holder)
- mod actions: mute, kick, lock_room, delete_chat

Moderation (minimum)

- Join muted by default
- Raise-hand queue
- Grant/revoke floor
- Mute / kick / ban
- Lock room
- Chat slow-mode toggle (if chat enabled)
- Audit trail retained 7–30 days

UI overlay placement

- Leader tile (video + name + mic indicator) anchored at top, collapsible
- Participant grid/table below leader tile
- badges: hand raised + reaction icon
- speaking indicator (halo/pulse)
- leave/disconnect visual (fade + "leader disconnected" banner when applicable)
- Breathing circle / practice UI below the participant grid

Open decisions to resolve later

- 1) Room lifecycle (leader disconnect handling) - If leader disconnects unexpectedly (not an intentional "Close Room"): - Room stays open for 5 minutes. - UI banner: "Leader disconnected. Room closes in 5:00 unless leader returns." - If leader returns within 5 minutes, room continues. - If leader does not return, room closes. - If leader intentionally closes the room: close immediately.
- 2) Access model - Invite links and join codes only. - No waiting room.
- 3) Chat scope - Excluded for now; voice-only.
- 4) Avatar reference strategy - Avatar state is persistent and always reflects each user's current app/profile state (not frozen at join).

5) Provider choice for SFU - Selection criteria and shortlist to be decided; requires server-minted tokens and role-based publish permissions.

6) Observability - Minimal metrics and error reporting approach to be decided.

Phase plan (high level)

- Phase 1: Presence + avatar ring + reactions (no media)
- Phase 2: Leader A/V broadcast
- Phase 3: Floor-controlled participant audio + speaking indicators
- Phase 4: Hardening (access control, admin console, analytics)