

O-Lab: a reproducible online laboratory for musical prediction and pleasure

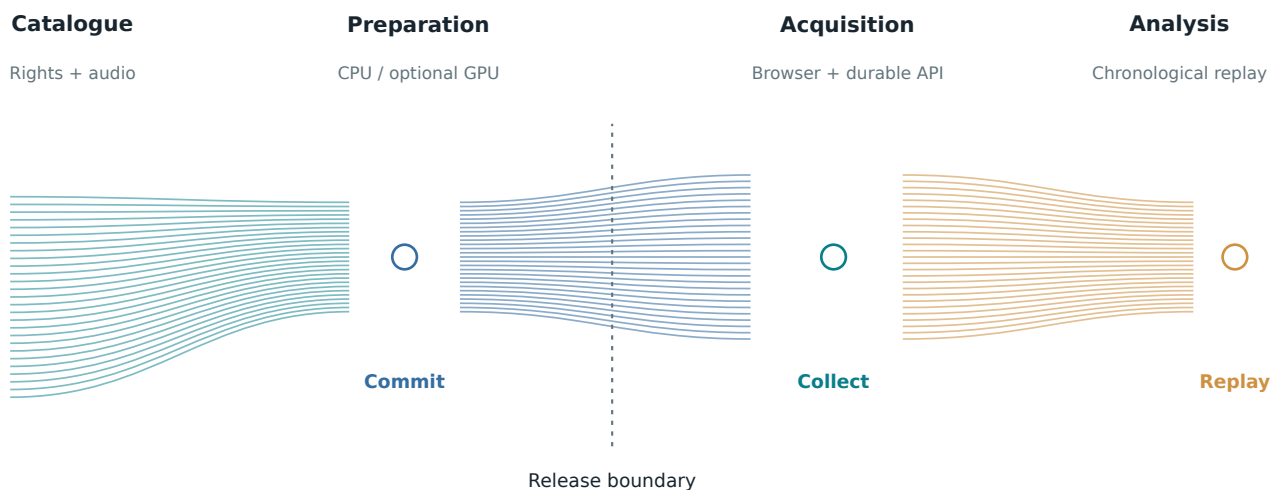
Amaç Erdem

Independent researcher · Correspondence: amace@bu.edu · ORCID 0009-0000-1711-0036

An online music experiment must preserve more than audio and ratings. To interpret a response, the investigator must recover what the listener already knew, which computational representation selected the stimulus, what was predicted before the response and which version of the experiment produced it. O-Lab is a proposed infrastructure for this chain of evidence. It connects a personal profile built from 20 familiar classical works to precomputed Musical Intelligence (MI) representations, immutable stimulus releases and time-resolved behavioural collection. The initial programme targets 1,000 registrations and combines controlled expectation tasks with cross-work and within-work listening comparisons. A managed relational service stores consent, versioned profiles and observations; isolated batch workers prepare shared and personal artifacts; the live session presents prepared audio and durably records responses. Chronological replay supports learning analyses without requiring online model inference. The design specifies identity, permissions, retry semantics, three timing domains, transactional email, CPU/GPU boundaries and recovery tests. Its methodological contribution is a common provenance structure for multidimensional music analysis and individual learning, allowing competing models to be evaluated against the same observations. No deployed service, measured throughput or participant dataset is claimed.

Architecture and implementation specification · Revision 3. Services, capacity assumptions and acceptance criteria are proposed. This document accompanies the scientific protocol, not a production deployment.

a Preparation and acquisition have different computational responsibilities



Shared audio artifacts → personal release → observed responses → reproducible analysis

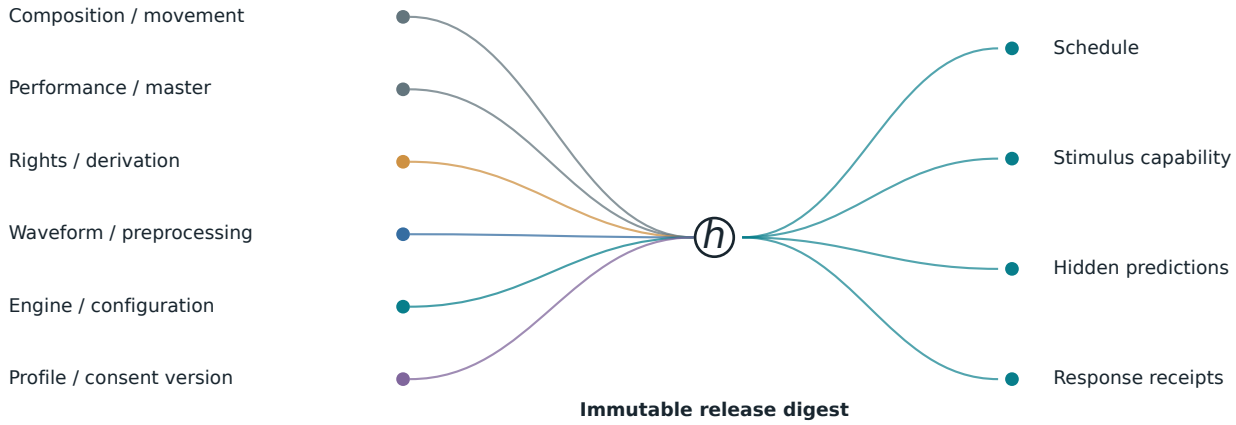
Fig. 1 | A release boundary separates preparation from observation. Shared catalogue processing and personal preparation run before a participant is admitted to listening. Acquisition then retrieves approved artifacts and appends responses. Analysis can update models through chronological replay, while retaining the original schedule and predictions. Curves depict artifact lineage rather than measured traffic. Heavy computation is absent from the listening path.

The system is part of the measurement

The central scientific question is whether a personal model learned from selected music helps explain and predict subsequent pleasure. Answering it requires linking acoustic structure, temporal context, expectation, uncertainty and responses across many listeners. A web form can collect a rating; a scientific platform must also establish which information was available before that rating. O-Lab treats this temporal boundary as a first-class data object.

The companion protocol[1] separates listener learning from model learning and policy value from prediction accuracy. The infrastructure preserves those distinctions in storage and execution. An updated model cannot overwrite a committed prediction, a repeated upload cannot become a second observation, and an edited personal profile cannot silently change a session already released. These properties determine whether a later statistical comparison is interpretable.

a An observation is interpretable only with its preparation history



b A profile edit creates a new version; it never rewrites the previous release



Fig. 2 | Provenance joins musical identity to computational and personal context. **a**, A release digest depends on audio identity, rights, preprocessing, engine configuration and a versioned personal baseline. Hidden predictions and participant-facing schedules derive from that release. Response receipts refer back to it. The digest verifies content identity; it does not anonymize personal data. **b**, A profile edit creates a new version. Existing exposures retain their original lineage.

Four domains with one evidence chain

The public research site remains a static React/Vite application. It presents the manuscripts, recruitment information and an entry to the study. The private laboratory adds authenticated registration, controlled catalogue playback and a session runner. The proposed data layer uses Supabase Auth, Postgres, private object storage and a durable queue. A specific European region, initially Frankfurt, is a planning preference that must be confirmed with the responsible institution and all subprocessors before deployment[2].

The preparation domain uses isolated container workers. A shared job decodes and verifies each admitted audio asset, computes the frozen MI representation and stores the resulting feature artifact. A personal job consumes the versioned 20-work profile, initializes predictions, assigns the experimental arm and generates the complete schedule. The acquisition domain accepts only released sessions. The analysis domain exports pseudonymous, versioned observations and replays candidate learners under explicit information cutoffs.

These domains share stable identifiers but not unrestricted credentials. The browser receives the minimum session manifest required to run the task, short-lived access to its audio and write access through validated endpoints. It never receives other profiles, server keys or hidden model predictions. Analysts receive a de-identified research view without contact details. A preparation worker can read its assigned baseline and audio artifacts through scoped service operations; it has no reason to access the contact database.

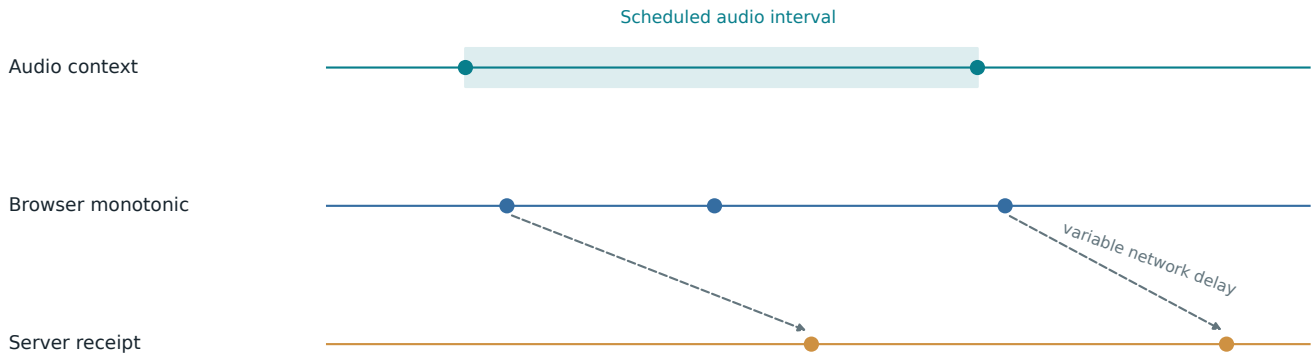
What a personal release contains

A release is an immutable agreement between stimulus preparation and later observation. It records the protocol version, consent version, baseline digest, catalogue snapshot, engine revision, preprocessing recipe, model coefficients, seeds, selection policy assignments, schedule and fixed predictions. It also records whether a quantity is a shared acoustic feature, a personal calibration output, a future analysis prediction or an observed response. This typing prevents a model-labelled “surprise” channel from being confused with a listener’s surprise report.

The catalogue distinguishes composition, movement, arrangement, performance, recording master and excerpt. Content hashes detect duplicate encodings and accidental changes; musical identity still requires curated metadata. A rights record belongs to the exact source and permitted derivative use. The corpus may contain one eligible reference for many families while only a subset supports three matched performances. Requiring three versions for every transfer item would unnecessarily constrain the broader scientific design.

O-Lab’s contribution is therefore not a novel database engine. It is the specification of how personal learning, multidimensional audio and temporally ordered observations remain comparable. Standard managed services can implement that specification if their boundaries are tested. The design intentionally avoids adding an orchestration cluster, vector database or message broker without a measured need.

a Three clocks; only one describes scheduled audio playback



b At-least-once delivery; one logical observation



A repeated upload preserves one event. Repeated listening creates another exposure.

Fig. 3 | Reliable measurement requires separate clocks and explicit delivery semantics. **a**, Scheduled audio time, browser monotonic time and server receipt time answer different questions. Network delay cannot be used as a proxy for listening time. Marker locations are illustrative. **b**, Delivery can occur more than once; database uniqueness and payload verification preserve one logical event. Listening to the same excerpt again creates a new exposure even if its audio hash is unchanged.

The browser is an instrument, not the archive

The session runner uses a pinned version of jsPsych or equivalently audited task components around Web Audio[3]. It loads the next approved stimulus before playback and records the audio-context start, browser monotonic timestamps, visibility changes, underruns and user responses. An absolute server timestamp accompanies receipt. These clocks remain distinct. A timestamp’s resolution does not establish the accuracy of the physical sound or motor response; device-level timing is qualified in the pilot[4].

The primary pleasure endpoint is an immediate post-excerpt rating. Continuous pleasure is sampled at 5 Hz during 25-second transfer excerpts and batched every two seconds. Keeping the discrete endpoint primary reduces dependence on unvalidated response lag, while the trace supports secondary tests at pre-declared MI moments. Probability allocations, confidence, expected enjoyment, surprise and experienced pleasure are different event types, attached to explicit before- or after-outcome phases.

The browser maintains a bounded IndexedDB queue until the server acknowledges a committed batch. Each event has a generated identifier, monotonically increasing sequence within an exposure, a schema version and a payload hash. Re-sending an existing identifier with an identical payload returns the original receipt; a changed payload is a conflict. Acknowledgement occurs only after durable database commit. The user can see

a recoverable upload state without being shown computational claims about their listening.

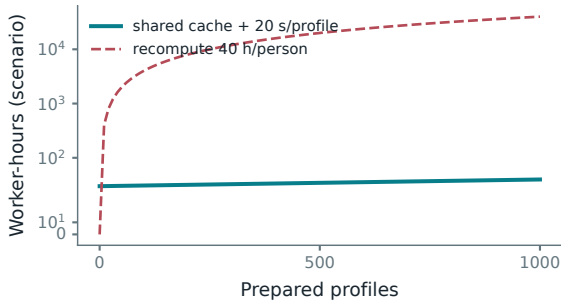
Preserve failure without manufacturing observations

A disconnected network does not necessarily invalidate local measurement. Playback and reports can continue only within the bounded, already authorized session and local storage budget. Expired authorization, exhausted queue capacity or missing audio pauses the session at a safe boundary. A reload reconstructs acknowledged progress; it does not guess whether an unacknowledged response was heard or delivered. The trial ledger resolves that ambiguity.

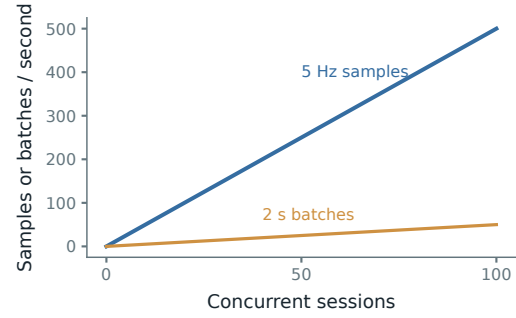
If an excerpt is replayed after interruption, the new exposure receives a new identifier linked to the failed attempt. The first-exposure analysis can then exclude or qualify it according to the scientific protocol. Upload duplication is a transport property; listening duplication changes the participant’s experience. Collapsing both into a single “retry” would corrupt learning analyses.

The same principle applies to probabilistic tasks. The intended distribution, sampled sequence, delivered audio and reported belief survive in separate fields. Sampling an outcome with probability 0.20 does not fix its subjective surprise at $-\log_2(0.20)$: that quantity depends on the participant’s prior report. This distinction makes the dataset useful for competing theories, without requiring them to adopt MI’s assumptions.

a Shared computation scales with unique audio



b Live acquisition scales with concurrent listeners



c A declared workload, not a vendor benchmark

600

recordings

200 families × 3 versions

40 h

shared audio

600 × 4 minutes

25.4 GB

source PCM

44.1 kHz · stereo · 16 bit

3,000

trace samples/person

24 × 25 s × 5 Hz

Fig. 4 | Capacity planning follows the workload, with assumptions exposed. **a**, Illustrative preparation cost assumes 40 hours of unique audio, one worker-hour per audio-hour and 20 seconds of personal preparation. These are scenario inputs, not benchmarks. Reusing a shared representation avoids recomputing the same corpus for each person. **b**, At 5 Hz, 100 concurrent listeners produce 500 trace samples per second; two-second batches produce about 50 batch writes per second before metadata and retries. **c**, The 600-recording corpus is a sizing scenario, not a verified inventory. PCM storage uses decimal gigabytes and excludes derivatives, features, replicas and backups.

Computation before the session

MI provides a multidimensional description of each stimulus, with acoustic coordinates, temporal operations and candidate listening-related calculations[5]. These shared features are keyed by waveform digest, engine digest and configuration. A cache hit is valid only when all three match. Personal initialization is a separate artifact keyed additionally by the baseline, model version and seed. Editing one person's reference ratings does not require reprocessing the catalogue.

CPU jobs perform decoding, identity checks, loudness analysis, most feature aggregation and personal readout fitting. AWS Batch on Fargate is one proposed execution environment for these bounded CPU tasks. GPU workloads, if a selected representation actually requires them, use a separate GPU-capable Batch/EC2 environment[6, 7]. GPU support is optional; no inference benchmark has established a need for a particular accelerator. Initial CPU candidates of 2 vCPU/4 GiB and 4 vCPU/8 GiB, and an optional 16–24 GB GPU class, are benchmark configurations rather than promised requirements.

Preparation throughput is measured on a representative catalogue subset before scaling. The benchmark records decode time, feature time, peak memory, artifact size, failures and cost per audio hour, using pinned hardware and software. A second benchmark measures profile preparation and queue delay. Registration receives a readiness notification only after successful release, not an invented estimate of GPU completion. Retries have finite budgets and quarantine persistent failures.

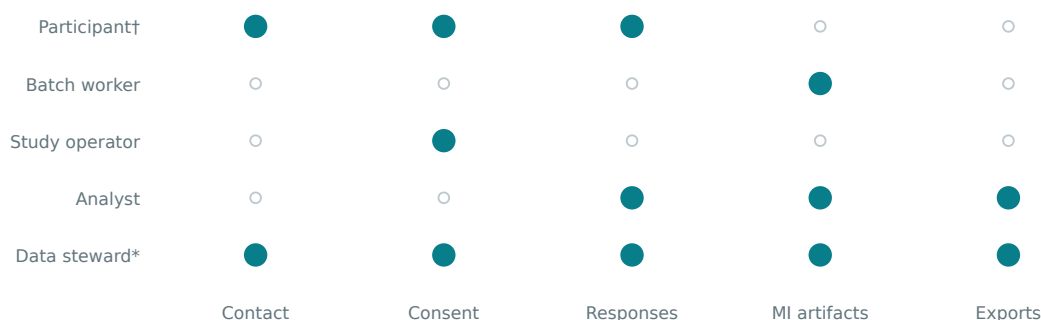
A dataset grows along several axes

A cap of 1,000 registrations is not equivalent to 1,000 concurrent sessions or independent observations. The proposed 24-excerpt transfer task produces 3,000 continuous samples per completer at 5 Hz; 1,000 completions would yield three million trace samples before mechanism tasks. Those samples remain nested within people, works, performances and exposures. Dense traces improve temporal description; they do not multiply the number of independent listeners.

The sizing example uses 200 composition families with three four-minute recordings each: 600 recordings, 40 audio hours and approximately 25.4 GB of stereo, 44.1-kHz, 16-bit PCM. It is a transparent engineering scenario, not a claim that the required rights-cleared catalogue already exists. Compressed playback derivatives, feature matrices, object versioning and recovery copies add separate storage terms. Feature dimensionality and compression must be measured rather than inferred from the number of modules.

For live collection, a practical initial acceptance test uses 50 concurrent sessions and a 100-session stress condition. At this scale, a managed relational database can be evaluated before more complex infrastructure is justified. Capacity is a release criterion demonstrated by a load test, not a property conferred by a vendor name.

a Permissions follow the task, not the person running the server



† Own allowed records only. * Exceptional, audited access; not a shared administrator login.

b Restore data without restoring withdrawn access



Fig. 5 | Access and recovery preserve the meaning of consent. **a**, Proposed task-scoped access. Filled markers indicate an allowed category, subject to row and operation restrictions. Participants access only their own permitted records; operators inspect consent and operational status, not raw responses. Preparation workers receive narrowly scoped inputs, and analysts use pseudonymous research views. Steward access is exceptional and audited. **b**, A restore must replay withdrawal and suppression records before reopening access. A restored backup cannot be treated as permission to contact a withdrawn participant.

Data location is a chain, not one region label

Postgres stores structured consent, profiles, releases and event metadata. Private object storage holds admitted audio, feature matrices, large trace exports and recovery artifacts. Contact details are confined to a restricted identity schema, linked through a pseudonymous participant identifier. Analysis exports omit email, authentication identifiers and unnecessary fine-grained demographic fields. Pseudonymization reduces routine exposure but does not make the data anonymous.

Row-level security and explicit grants enforce access to browser-visible records[8]. Server credentials can bypass ordinary row policies and therefore remain outside client code; they require constrained endpoints, secret rotation and audit. Object access is governed separately from database rows[9]. A short-lived signed URL is a bearer capability: whoever possesses it can use it until its access conditions expire. It is not digital rights management or proof of immediate revocation.

The proposed European primary region does not by itself establish that every processor, support action, email or optional language model remains in that region. The deployment record must enumerate the frontend host, database, object storage, worker environment, email provider, telemetry and any model provider. Controller identity, processor terms, approved transfers and retention are resolved before recruitment; they are not inferred from a check box in a cloud console.

Email and language models have bounded roles

Authentication uses a custom SMTP configuration rather than Supabase's development-oriented default sender[10]. Resend is the proposed SMTP and transactional-mail provider. Auth handles verification and sign-in messages; study messages use an application outbox committed in the same transaction as the triggering state transition. Readiness mail cannot precede a committed ready release. Idempotency keys suppress duplicate provider sends within their supported window, while a permanent application key protects the study's own message identity[11].

A language model is unnecessary for inference or data collection. If later introduced, its default scope is an approved explanation assistant or curator aid. It cannot generate consent decisions, familiarity reports, primary stimuli, personal scientific predictions or outcome labels. Any metadata suggestion requires human verification and provenance. No participant response is sent to an LLM in the baseline design. An optional regional provider must undergo a separate data-flow review; cross-region model routing is not assumed to preserve a single-region processing guarantee[12].

This division protects the experimental question. A helpful assistant must not coach a participant toward expected answers or reveal a selection policy. The operational fallback is static, approved text, which also permits the experiment to continue when optional assistance is unavailable.

System specification

Current implementation boundary

The existing project supplies a research website, manuscript records and an executable MI architecture. This manuscript proposes the laboratory service around them. It does not establish that Supabase projects, DNS records, mail domains, production queues, consent forms or GPU workers have been provisioned. The existing interest-registration affordance must not be described as a completed participant-management system. Launch requires the operational evidence defined below.

The study contract accompanies the manuscript as a versioned JSON document. It fixes the 1,000 registration cap, proposed 60-registration pilot, 20 distinct familiar composition families, 1:1 mechanistic arm assignment, 24 transfer excerpts, four selection policies and four paired-performance trials. Arm U has 24 probe outcomes; Arm L has 32. A release always refers to one contract digest. Changing these values creates an amendment rather than silently changing an ongoing experiment.

Registration and participant state

An unauthenticated visitor first receives plain-language study information. Email verification establishes account control, not consent or scientific eligibility. After a minimal age/language/device screen, a verified eligible account claims one registration slot through a transaction with an atomic cap check. A unique account constraint and an idempotent request key prevent retries from consuming additional slots. Operational identity checks may flag suspected duplicate people for review, but the system does not claim that email uniqueness proves person uniqueness.

The state sequence is verified registration, consented, baseline in progress, baseline locked, preparation queued, preparation running, ready, session active and completed. Ineligible, preparation failed, expired and withdrawn are explicit branches. Only server-authorized operations change state. Consent requires a version identifier and affirmative action; baseline locking validates distinct composition families, prior-recognition responses and usable calibration ratings. A person may save an incomplete baseline without being counted as ready.

Once locked, the baseline is immutable. A permitted correction creates a new baseline version, invalidates any unstarted release and queues new personal preparation. A worker finishing the previous version cannot mark the new participant state ready. This check uses both the baseline digest and a compare-and-swap state version. After a session starts, corrections are recorded as annotations or future versions; they do not rewrite the history that generated already heard stimuli.

Relational model and object layout

The core relational entities are participant, consent event, baseline version, baseline item, catalogue work, performance, recording asset, excerpt, rights record, feature artifact, preparation job, experiment release, scheduled trial, exposure, response event, batch receipt, email outbox and withdrawal event. Each has an immutable identifier and creation time. Natural identifiers such as email or work title are attributes, not relational keys.

A baseline item links one participant version to one composition family and a specific reference excerpt. Its fields preserve prior recognition, graded familiarity, performance recognition, pleasure and missingness reason. A scheduled trial links the release to its intended excerpt, policy, task context and position.

An exposure records what actually played and when. Response events reference the exposure and the measurement phase. This structure permits intended, delivered and analysed sequences to be compared without editing one to resemble another.

Large objects use names derived from controlled identifiers and content digests rather than personal names. Audio masters, playback derivatives, features and exports occupy separate private prefixes with separate policies. Object metadata records byte size, media type, creation recipe and source hash. A content digest is useful for integrity; it is not an access-control or anonymization mechanism. Deletion and retention operate across database and object inventories together.

Minimal API and validation rules

The registration endpoint accepts a verified identity token and idempotency key, returning a stable participant record or an explicit cap/eligibility outcome. The baseline endpoint validates schema, family uniqueness and permitted catalogue references. The lock endpoint creates the versioned preparation input transactionally. A ready-session endpoint returns the participant-facing schedule and renewable playback capabilities only if consent, state and release checks pass.

The event-batch endpoint checks authentication, session ownership, schema version, release membership, exposure identifier, event sequence and payload size before attempting insertion. It rejects invalid phase transitions, impossible rating ranges and attempts to submit hidden prediction fields. A unique constraint on event identifier plus participant scope detects duplicate delivery; the server compares payload hashes before returning the original receipt. Conflicting duplicates are preserved in an audit record and rejected. Clients cannot write raw model outputs, ready status or assignment labels directly.

The completion endpoint verifies required receipts and classifies the session as complete or partial without discarding usable data. Withdrawal is a distinct authenticated operation and remains available after partial completion. Every state-changing endpoint records actor, operation, version and result in a restricted audit stream without copying sensitive payloads into general logs.

Preparation queue and release transaction

The database transaction that locks a baseline also creates an outbox/queue entry, eliminating the gap between “saved” and “scheduled”. A Postgres-backed queue is the proposed durable mechanism[13]. Workers acquire bounded leases and renew them while processing. At-least-once delivery is expected, so the job identity is a deterministic digest of job type and input versions. A successfully completed artifact is reused when its digest and validation status match.

Shared extraction and personal preparation are separate job classes. Shared extraction performs decode checks, preprocessing and MI computation. Personal preparation validates the 20-work baseline, fits the constrained personal readout, draws the assignment and candidate partition, ranks policy selections and serializes the schedule. All hidden predictions are saved before release. Optional GPU jobs run only when a dependency explicitly requires them; they never block the live response endpoint’s resource pool.

Release is one transactional decision after all required artifacts exist and their digests verify. It rechecks consent, withdrawal, current baseline, catalogue permissions, complete policy quotas and the locked model/configuration version. An obsolete or

withdrawn job may finish computation but cannot authorize a session. Failures use bounded exponential backoff with jitter, explicit retryable versus terminal codes and a quarantine queue after the limit. A study operator resolves terminal failures without editing completed scientific artifacts.

Randomization and hidden predictions

Assignment seeds are generated server-side and recorded with the release. The scientific design uses 1:1 assignment to U/L after readiness, stratified by broad musical training. Arm L additionally randomizes changed versus stable distributions. Outcome identities, policy slot order and paired-performance order are counterbalanced independently. All schedules are computed before the first experimental outcome.

Candidate pools for the four selection policies are randomized partitions of the same eligible catalogue for each participant. Their assignments, inclusion rules and selection probabilities are saved. The resulting policy labels and scores remain hidden from the participant; only the playback order is exposed. The API returns no future pleasure prediction in a browser payload, analytics event or downloadable filename. Revealing the model's preference could itself alter expectation and would create another experimental manipulation.

Chronological learning and multidimensional exports

Every shared feature vector has a namespace, units or scaling, temporal support, engine version and semantic status. Acoustic coordinates, temporal summaries and cognitive-labelled proxies remain distinguishable. Raw probability reports and behavioural outcomes are never overwritten by transformed values. This allows alternative definitions of uncertainty or surprise to be computed from the same observations while preserving the declared primary analysis.

The replay engine accepts a release, an ordered exposure ledger and a frozen learner specification. Before trial t , it exposes only admissible intake and earlier responses, updates the state, writes a prediction commitment, and then reveals the current target to score it. A replay record includes the model digest, training cut-off, state digest, target exposure and prediction. Replaying with a new learner creates another analysis version. It cannot replace the fixed predictions made before collection.

Exports retain participant, work, performance and exposure grouping keys so that dense traces are not mistaken for independent samples. A common response table supports simultaneous modelling of expectation, surprise, confidence and pleasure; explicit missingness identifies measures not collected in a task. Multivariate analysis therefore draws strength from shared provenance and overlapping observations, rather than pretending that every listener completed every possible assay. Ablations can remove personal calibration, temporal features or mechanism outputs while scoring the same held-out responses.

Timing, buffering and session recovery

Audio is preloaded and decoded before a trial begins. Playback uses a recorded Web Audio start time; interface responses use the browser's monotonic clock. Their mapping is sampled at session and trial boundaries, with offset and drift retained. Server receipt is used for delivery audit, not for reconstructing millisecond event onset. Wall-clock changes and timezone do not alter trial order.

The local queue stores immutable event records and acknowledged batch receipts. The planned 5-Hz trace produces ten sam-

ples per two-second batch, plus metadata. A queue limit is defined in both bytes and elapsed unacknowledged time. On reaching it, the next trial is withheld rather than overwriting older samples. The system reports when data are saved locally, uploading or acknowledged without promising completion before server commit.

Visibility loss, suspended audio, device changes and playback failures generate explicit quality events. A recovery controller reconstructs the last acknowledged trial and reconciles local pending events. Resuming within a partially heard excerpt is not equivalent to a fresh exposure; the scientific protocol determines whether to restart and exclude it from the first-exposure endpoint. Repeated playback remains observable in all exports.

Mail reliability and privacy

Supabase Auth delegates verification and sign-in delivery to the approved SMTP service. Study messages originate from the application outbox with a unique key combining participant, message purpose and release or consent version. The worker saves the provider identifier and delivery state. Resend's provider-level idempotency window is 24 hours[11]; the outbox uniqueness persists for the study's retention period, so a delayed retry cannot create a new message merely because that window expired.

Webhooks require signature verification and replay protection. Delivery events are deduplicated and reconciled with outbox state; absence of an acknowledgement is not proof that no mail was sent. Operational states distinguish queued, accepted, delivered, bounced and suppressed. Delivery is not interpreted as participant comprehension or reading. Open/click tracking is disabled in the baseline design. Verified sending domains, SPF, DKIM and DMARC are launch tasks; this manuscript makes no DNS changes.

Emails use neutral subject lines and contain no musical preferences, response scores or sensitive study inferences. Readiness links require authentication and reveal no raw storage URL. Withdrawal and bounce suppression are checked at send time, including after retries. A failed notification does not reverse a valid scientific release; the participant may also see readiness after signing in.

Security and role tests

Database tables with browser access enable row-level security and explicit least-privilege grants. Policies are tested with anonymous, participant, operator, worker and analyst identities. Negative tests are as important as permitted reads: participant A must not access B's baseline, a worker must not query contact details, an analyst must not retrieve identity links, and a browser must not set its own readiness or selection policy. Server credentials are kept in managed secrets, never frontend environment bundles[14].

Short-lived audio access is renewed through authorized session endpoints. Playback URLs can be copied while valid, so rights clearance must permit the actual delivery model. Revocation prevents new capabilities; already issued capabilities are limited by their short lifetime. Rate limits, bounded request sizes, dependency pinning and sanitized telemetry protect service reliability without changing task content. A security incident pauses new releases and preserves the audit trail needed to assess affected exposures.

Retention, withdrawal and recovery

The initial retention proposal is seven days for abandoned unverified registrations, 90 days after study closeout for operational contact details, five years for approved research records, 30 days for sanitized operational logs and 35 days for rolling recovery copies. These are proposed policy values requiring institutional approval, not statements of a statutory requirement. Consent and any required compliance records may need a separately approved retention schedule. Public, irreversible aggregates cannot always be retracted; consent must explain that boundary.

A withdrawal event immediately suppresses invitations and future preparation or sessions. It creates an auditable deletion/restriction request across identity, research and object stores according to the consented policy. A restricted suppression ledger persists long enough to prevent restored backups from reactivating withdrawn access. Worker and email operations recheck this ledger at their final commit or send boundary.

Database recovery does not automatically recover object payloads. Supabase documents database backups separately from Storage objects[15]. O-Lab therefore requires independent, versioned object backups and a manifest linking each database snapshot to recoverable object digests. Restore tests must recover a complete release, its stimuli, feature artifacts and response receipts, then reapply withdrawal and suppression events before reopening service. Proposed recovery objectives are RPO of five minutes for committed observations and RTO of four hours. Achieving them requires suitable point-in-time recovery, protected event receipts and measured drills; a daily database backup alone does not satisfy the proposal.

Cost model and procurement

Cost is calculated from measured workload rather than a static product-price table. The monthly estimate includes database base/compute fees, object bytes and replicas, audio egress, worker CPU/GPU hours, queue/API requests, authentication and study email, monitoring and recovery retention. If N_a unique audio hours cost c_a worker-hours per audio-hour, and N_p profiles require t_p seconds, the preparation demand is

$$T_{\text{prep}} = N_a c_a + N_p t_p / 3600. \quad (1)$$

The two terms are measured separately. Reprocessing after an engine change adds another shared term and creates a new artifact generation; it must not overwrite the previous generation.

The corpus example has $N_a = 40$ and assumes $c_a = 1$ only to illustrate scaling. At $N_p = 1,000$ and $t_p = 20$ seconds, the scenario gives 45.56 worker-hours. This is exact arithmetic under declared assumptions, not an observed MI runtime or price quote. Transfer playback at 192 kbit/s for 24 excerpts of 25 seconds uses approximately 14.4 MB per completed listener, before intake, mechanism tasks, paired performances, retries and protocol overhead. Procurement must use current quotes after benchmarking and approved data-location choices.

Acceptance experiments

Before human pilot use, synthetic accounts exercise the entire sequence from registration to export. A race test attempts more than 1,000 eligible slot claims, with repeated idempotency keys and simultaneous requests; exactly 1,000 unique successful claims may exist. Baseline fixtures include duplicate movements, remasters, missing recognition and insufficient usable ratings. None may produce an incorrectly ready release.

A worker test injects duplicate jobs, lease expiry, changed base-lines, removed rights and withdrawal while preparation runs. It must produce at most one release per exact version, with obsolete jobs unable to authorize listening. An acquisition test drops acknowledgements, reorders batches, repeats payloads and sends conflicting identifiers. Reconstructed logical events must equal the source fixture, with conflicts explicit and no silent duplication.

The load test runs 50 simulated active sessions for 30 minutes, then 100 for 15 minutes, using the real batch sizes and authorization path. Proposed targets are 95th-percentile committed batch acknowledgement below one second, no lost acknowledged event and error rates below 0.1% excluding deliberately invalid requests. These are acceptance thresholds to measure, not achieved service levels. The browser timing assay separately checks audio onset and response timing on representative devices; network performance cannot substitute for it.

A network trace must demonstrate zero MI, GPU or LLM inference calls during a prepared session. A restore drill must recover one complete synthetic study subset with matching object and event hashes, including a withdrawal introduced after the snapshot. An export/replay fixture must reproduce predictions and scores from the same earlier-only information. Permission tests, delivery reconciliation and source/asset licence review complete the release evidence. Failure blocks the affected capability rather than being hidden in an overall “passed” badge.

Pilot and operational release

The proposed 60-registration pilot is included within the total cap and excluded from confirmatory analyses. It measures catalogue coverage, intake effort, task comprehension, timing, missingness, preparation throughput and attrition. It also determines the compact model basis and hyperparameters under the scientific protocol. After pilot review, the team freezes the protocol amendment, catalogue, learner, environment, consent and analysis plan together. Confirmatory participants never enter a moving target of silently updated models or stimulus rules.

Operational dashboards display counts and failures by state, with no need to reveal hidden predictions or personal musical responses. Recruitment is staged against observed readiness throughput and completion burden. Planned pauses and support procedures are part of the participant information. The design does not require a 24-hour availability claim; it requires that interrupted work remain interpretable and recoverable.

Discussion

O-Lab makes a particular scientific ambition operational: learning from a person’s chosen reference music, placing that profile within a multidimensional representation and asking what improves prediction of later responses. The same infrastructure supports uncertainty-surprise tests, changing expectations and controlled performance comparisons because every observation retains its context. The advantage of scale is not merely more rows. It is the ability to examine whether relationships persist across listeners, compositions, performances and model variants while respecting their dependence.

The proposal remains limited by catalogue rights and coverage, browser measurement quality, self-selection into classical listening, and the identifiability of personal models from a small intake. Cloud services do not resolve these scientific limitations. They provide a manageable environment in which the limitations can be measured, declared and reproduced. The sys-

tems contribution is the evidence chain that lets a negative result remain useful: whether a failed prediction arose from an unsupported mechanism, weak personalization, a deficient stimulus pool or a broken measurement can be investigated without rewriting the original experiment.

Availability and authorship

Editable manuscript source, original vector figures, a design contract, analytic capacity calculations and visual QA accompany this edition. Infrastructure code, deployment credentials

References

- [1] Erdem, A. Learning to predict musical pleasure. Prospective scientific protocol, revision 5, 10 October 2026. Companion manuscript in this edition.
- [2] Supabase. Available regions. [Official documentation](#) (accessed 10 October 2026).
- [3] jsPsych. Version 8 documentation: running experiments and plugin API. [Experiment execution](#); [core API](#) (accessed 10 October 2026).
- [4] Bridges, D., Pitiot, A., MacAskill, M. R. & Peirce, J. W. The timing mega-study: comparing a range of experiment generators, both lab-based and online. *PeerJ* **8**, e9414 (2020). [doi:10.7717/peerj.9414](#).
- [5] Erdem, A. Musical Intelligence: A Glass-Box Architecture for Computational Models of Music Listening. Author manuscript, 8 October 2026. [Architecture record](#).
- [6] Amazon Web Services. Amazon ECS task definition differences for Fargate. [Official documentation](#) (accessed 10 October 2026).
- [7] Amazon Web Services. Run GPU jobs in AWS Batch. [Official documentation](#) (accessed 10 October 2026).
- [8] Supabase. Row Level Security. [Official documentation](#) (accessed 10 October 2026).
- [9] Supabase. Storage access control. [Official documentation](#) (accessed 10 October 2026).
- [10] Supabase. Send emails with custom SMTP. [Official documentation](#) (accessed 10 October 2026).
- [11] Resend. Idempotency keys; webhooks. [Idempotency](#); [delivery events](#) (accessed 10 October 2026).
- [12] Amazon Web Services. Cross-Region inference in Amazon Bedrock. [Official documentation](#) (accessed 10 October 2026).
- [13] Supabase. Supabase Queues. [Official documentation](#) (accessed 10 October 2026).
- [14] Supabase. Understanding API keys. [Official documentation](#) (accessed 10 October 2026).
- [15] Supabase. Database Backups. [Backup scope and recovery](#) (accessed 10 October 2026).
- [16] Wilkinson, M. D. et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* **3**, 160018 (2016). [doi:10.1038/sdata.2016.18](#).

and participant data are not part of the package because the laboratory is not yet deployed. Public artifacts will use versioned manifests; restricted data will require governed access. FAIR principles guide metadata and reproducibility without requiring open release of personal records or licensed audio[16].

Amaç Erdem developed the study architecture and manuscript. Responsible institution, ethics approval, funding, operational roles and any conflicts must be finalized in the recruitment record. This author manuscript reports a proposed method and no production performance result.