

What ships next?

Choose six policies, assign seven release-readiness packages, and require live receipts before widening traffic. The demo is a starting point—not a launch approval.

Seven investigations · /max synthesis · Source [fab0259a](#)

Run a bounded readiness pass, not a production launch vote. Resolve payment, access and identity risks before widening traffic. Team analytics and Stripe catalog consolidation need not hold that work hostage.

This brief reviews [badass-courses/egghead at fab0259a](#), not legacy egghead-next. Every package below names an owner role, a decision or blocker, and an acceptance receipt. **All work remains proposed.**

Approve evidence work. Not a production switch.

Decisions to make in the meeting

These are **recommended defaults**, not existing approvals.

D1 — Membership catalog

RECOMMENDED DEFAULT

Retain separately mapped monthly/yearly memberships initially. Consolidate to one Stripe product/two prices only if cleaner catalog administration warrants adapter, catalog, checkout and webhook compatibility work.

D2 — Cancellation and payment grace

RECOMMENDED DEFAULT

Default cancellation to period-end, honoring paid-through access. Define bounded failed-payment grace and recovery separately. Grace reduces accidental lockouts but permits some unpaid access; `past_due` alone is not a complete policy.

D3 — Seats and invitations

RECOMMENDED DEFAULT

Keep explicit owner seat claiming and allocation on acceptance. Recommend expiring, revocable, email-scoped invitations. Keep self-service shrink disabled until reconciled; support can unassign before reducing quantity. Safer capacity management costs administrative steps.

D4 — Public progress privacy

RECOMMENDED DEFAULT

Recommend opt-in public activity, or defer public sharing without blocking private progress. Always-public profiles simplify sharing but expose learning activity without a consent step.

D5 — Identity/import continuity

RECOMMENDED DEFAULT

Prove active/paid customer continuity through the existing migration owner; preserve existing Stripe subscriptions. Defer dormant-user and historical-progress imports. Smaller cutover scope means support-led long-tail recovery.

D6 — Connection, writes and traffic ownership

RECOMMENDED DEFAULT

Approve beta connection and each write family separately. Preserve production and beta-commerce restrictions until separately reviewed changes. Traffic/read authority, billing writers, migration writes and search indexing are distinct approvals—not one “go live” switch.

What the Loom proves—and does not

The [Loom](#) demonstrates a localhost/Stripe-sandbox journey: email-link sign-in, profile editing, completion/progress views, team purchase activation, invitation controls, owner seat claiming, paid playback and a cancellation notice.

It does **not** establish recipient invitation acceptance, renewal, failed-payment recovery, access after cancellation becomes effective, concurrent seat allocation, imported-customer continuity or deployed readiness. GitHub integration is discussed; complete account-linking safety is not demonstrated. Production email delivery is the narrator’s report, not independently verified evidence.

Implemented: the corresponding auth, profile, progress, checkout, team and billing paths exist. They are not missing features. **Requires verification:** their deployed behavior and failure handling. **Externally blocked:** migration receipts and permission to change runtime/write ownership. The source’s production restrictions are deliberate controls, not cleanup to remove casually.

Ordered release backlog

Agree **EH-R1** first. Catalog, identity and access investigation can then proceed locally in parallel. **EH-C2** needs the catalog choice and test identities; **EH-L1** needs identity proof. **EH-R2** collects the receipts before broader exposure.

File labels below are relative to `apps/web/src` unless otherwise stated.

Runtime and permissions

EH-R1 — Establish an enforceable permission matrix

Owner role: Runtime/DB, with Release approval

Decision/blocker: D6; target credentials and permitted write families are unspecified.

Correction to Runtime/Learning/Operations: beta is **not necessarily read-only**.

Approved beta permits normal MySQL connections to PlanetScale; credentials determine database privileges. Auth, profile and progress mutations do not use the commerce assertion. Even `ensureLocalResourceProgressTable()` lacks a local-only assertion: its returned `localDockerOnly: true` is merely a label. Conversely, app-owned checkout/team/Inngest writers explicitly remain local-only. [Runtime policy, progress writers](#).

Outcome: extend existing assertions where the agreed policy requires them; audit `getCourseBuilderAdapter`, the CourseBuilder handler boundary and `manageMembership` —the latter can open Stripe Portal without a commerce assertion. Align README and `app/api/route.ts` diagnostics with enforceable behavior.

Receipt: an operation-by-runtime matrix covering approved-beta commerce denial, authorized account/progress writes, blocked DDL and Portal/webhook boundaries.

`SELECT 1` alone cannot prove read-only operation.

Commerce

EH-C1 — Freeze the supported catalog mapping

Owner role: Commerce/CourseBuilder integration

Decision/blocker: D1; actual catalog rows and package internals are unavailable.

Correction to Catalog and earlier advice: one Stripe product/two prices is **not established as data-only cleanup**. `getActiveMembershipProducts()` selects

CourseBuilder products with one interval and active merchant mappings; the checkout writer resolves a CourseBuilder product through the Stripe **product identifier**. Generic Stripe modeling guidance does not establish compatibility. The investigator's exact uniqueness/cardinality claims require package inspection. [Catalog](#), [checkout writer](#).

Outcome: record Product → MerchantProduct → MerchantPrice → Stripe identifiers and account mode. If consolidating, make interval/price selection and webhook resolution explicit through the existing adapter/provider seams.

Receipt: monthly/yearly checkout selects the intended price, amount and quantity; inactive mappings disappear; quantities 0/101 fail; sandbox/live identifiers cannot be mixed unnoticed.

EH-C2 — Make subscription and seat state converge

Owner role: Subscriptions/Teams

Decision/blocker: D2/D3, EH-C1 and test identities.

Source-supported risks—not reproduced bugs:

- Personal checkout writes `stripe_ent_*`; every complete entitlement found by `stripeCustomerSubscriptionUpdated` is subsequently synchronized with `teamSeat: true`, targeting `stripe_seat_*`. The original personal grant can remain unchanged until its existing expiry.
- Updates select only non-deleted entitlements. Recovery after status-based revocation can therefore update subscription status without restoring seats. Blindly restoring deleted rows would instead resurrect deliberately removed members.
- Ordering protection is per entitlement; seat fields and the no-entitlement status branch do not share that protection.

These follow from `syncStripeSubscriptionEntitlement` and the [update handler](#).

Outcome: preserve or deliberately migrate grant identity; distinguish seat removal from billing revocation; synchronize status, quantity and entitlements under a shared subscription-ordering rule. Cover terminal deletion and chosen grace/recovery

transitions through the existing provider/Inngest path. Only checkout-completed and subscription-updated functions are registered here; package forwarding remains unverified.

Correction to Lifecycle/README: no add-seat action appears in supplied team code. Portal quantity editing is unverified configuration, not an implemented app feature. Prefer a tested Portal/support path over requiring new UI. Shrink currently does not prune excess seats. [Team helpers](#), [team actions](#).

Receipt: personal/team renewal, scheduled/immediate cancellation, failure/recovery, duplicate/out-of-order events, seat races, removal and quantity transitions produce agreed access. Concurrent/replaced checkout attempts—including session-expiry failure—must not create two payable subscriptions.

Access, identity and learner trust

EH-A1 — Reconcile authorization predicates and paid-content classification

Owner role: Access/Migration

Decision/blocker: D5, EH-C2; representative legacy projections are externally unavailable.

Source-supported risks—not production findings:

- **Legacy status:** access SQL reads `metadata.status`, but granting logic ignores it. `rails_account_subscription` bypasses ordinary expiry. A non-deleted granting row with canceled metadata could therefore still grant; whether such rows exist depends on the importer.
- **Organization inheritance:** flagged `hasAccountMemberRole` membership can select another user's entitlement in the same organization. New Stripe seats grant directly through `userId`; missing that flag does **not** mean new seats are broken.
- **Subscription classification:** `getCurrentSubscriptionForUser()` accepts any non-team subscription in a member's organization without requiring ownership. Its seat

lookup also omits expiry. This can affect checkout blocking/account presentation; it does not establish unauthorized billing access.

- **Support explanations:** support's auxiliary query uses different expiry/membership predicates, although its final access boolean calls the shared evaluator.

Sources: [access evaluator](#), [subscription status](#), [support readback](#).

Paid-gating qualification: standalone lessons are deliberately ungated. Missing parent linkage/slug can classify content as standalone; this is a migration risk, not proof that every standalone lesson needs a paywall. Denied placeholders also expose Mux-derived posters; whether those identifiers enable playback requires media-policy verification. [Classification](#), [gate/rendering](#).

Outcome/receipt: align these existing helpers around an approved cohort matrix: personal subscriber, unseated owner, seated/removed member, organization-only member, canceled legacy subscriber, lifetime, playlist/country-restricted purchase and quarantine-only user. Preserve legitimate grants and bare-**pro** denial. Known paid fixtures must not become free through missing linkage or expose usable denied playback through page/embed responses.

EH-I1 — Prove identity continuity and token boundaries

Owner role: Auth/Security

Decision/blocker: D3/D5; provider/package behavior and import mapping require receipts.

GitHub's **allowDangerousEmailAccountLinking: true** is explicit, but verified-email behavior and exploitability are not established. Magic-link expiry/replay protection lives in absent package internals; UI wording is not proof. Auth and team invitation signing both have a development-secret fallback. Invitations contain subscription ID and optional email, but no per-invite expiry/revocation state. [Auth configuration](#), [invitation token](#).

Outcome: require a real secret outside local development; extend existing invitation helpers with agreed expiry/revocation. Recommend explicit authenticated GitHub

connection until verified same-email linking is demonstrated—safer linking trades away some convenience.

Receipt: real email/OAuth callbacks retain the intended database user and access; expired/replayed magic links fail; mismatched emails cannot claim scoped invitations; revoked invitations cannot reclaim removed seats; disconnect preserves another usable sign-in method.

Migration correction: public profiles use the database user ID; only `/api/current-user` hashes its exposed ID. Neither is automatically a Rails identity mapping. [Profile identity](#), [API identity](#).

EH-L1 — Verify progress persistence and settle public exposure

Owner role: Learning/Product

Decision/blocker: D4, EH-I1 and permitted account/progress writes.

Learning report qualifications: public history loads at most 100 completions and filters that set in the server component—not a full-history query. Totals use broader predicates. “Empty courses never complete” is overstated:

`syncCourseProgressForUser()` leaves an existing completion row unchanged when there are no lesson IDs. [Profile reads](#), [course synchronization](#).

Outcome/receipt: complete/uncomplete persists across sessions; last-lesson/course state agrees; partial failure and empty-course cases are explicit. Private reads reject another actor. Public page, metadata and OpenGraph follow D4 and omit email/provider/purchase details. If sharing is deferred, gate public reads—not merely the copy-link button. Full public-history pagination is optional.

Proof and future cutover

EH-R2 — Collect staged receipts and rehearse rollback

Owner role: Release/Operations, with Migration and Support

Decision/blocker: D6; preceding acceptance receipts and external approvals.

- 1. Local proof.** Use the existing sparse checkout and a prepared local-Docker/dependency environment or existing runner—not a giant clone/install campaign. Extend/run `beta:runtime-guard`, `stripe:subscription-contract`, `routing:v2-contract`, `profile:contract` and `email:contract`; obtain typecheck/build receipts. These are largely function/source/fixture checks, **not end-to-end proof**. Separately trace sandbox Checkout → webhook → Inngest → rows → browser access. `package.json`.
- 2. Approved beta proof.** Record deployment URL, SHA, explicit runtime, database target/write permissions and the selected Vercel project root/build configuration. Two Vercel files are a configuration question, not proof of misdeployment. Reuse `/api`, its advertised `/.well-known/coursebuilder-app`, `/api/health/db` and `/api/current-user`; no parallel agent API. Obtain real auth/progress receipts where authorized. **Operations' proposed beta checkout replay cannot work unchanged:** commerce remains blocked. Sandbox beta commerce needs separate approval and reviewed guard changes first. Confirm primary routes/search and whether results use Typesense or SQL fallback. Indexing remains separately approved. `builder:smoke` may skip its beta probe; its imported builder files are omitted here. Request a `--require-beta` receipt only where that dependency matters. [Builder smoke wrapper](#).
- 3. Future controlled cutover.** Obtain the external `migrate-egghead` identity/entitlement reconciliation and existing standalone-check receipt. Production runtime/auth restrictions require reviewed changes, not an environment-only flip. Compare cohorts against legacy promises, recording intentional differences. Preserve customer/subscription IDs; importing access must not create new subscriptions. Establish one authoritative checkout/billing writer per cohort and reconcile in-flight sessions/events before traffic handoff. Rehearse restoring traffic and the previous web deployment while stopping unintended writers and reconciling queued events. Vercel rollback does **not** undo Stripe charges, entitlement changes or learner writes. Monitor existing HTTP diagnostics, Stripe webhook failures and Inngest runs. [Migration/release boundaries](#).

Optional/post-launch

PR #20 remains optional. Supplied PR evidence describes an open `/team-preview` using fictional members/topics/activity, without live team-data wiring. It cannot be presented as production analytics. Current seat purchasing does not depend on it.

Likewise, do not automatically require an add-seat UI, historical-progress import, new support API or new observability platform.

Copyable meeting checklist

 `meeting-decision-packet.txt`

```
[ ] D1 catalog mapping chosen
[ ] D2 cancellation, grace boundary and recovery agreed
[ ] D3 allocation, shrink and invitation policy agreed
[ ] D4 public-progress privacy decided
[ ] D5 identity/import owner role and external receipts recorded
[ ] D6 connection/write/traffic approvals separated
[ ] Each EH package has owner role, disposition and acceptance receipt
[ ] Local checks distinguished from live beta proof; skips recorded
[ ] No-double-charge and rollback scenarios demonstrated
[ ] PR #20 explicitly deferred or labeled fictional preview
```

Claim boundary: this is static synthesis, not verification. No tests, database queries, sign-ins or transactions were executed. Live rows, service configuration, published-package internals and external migration artifacts remain material missing context. No suspected defect above is claimed as a reproduced production bug.

USE THIS IN THE MEETING

Human review: choose D1–D6, put a person against each owner role, and approve or defer each package. Accept observable receipts—not a green source check presented as a successful payment.

Agent handoff: read [the versioned discussion record \(brief.json\)](#), select an exact task ID, and compare its source revision with the checkout. Stop on revision mismatch, missing external evidence, or an unauthorized write. This artifact grants no implementation or deployment permission.

Team copy: [download the discussion guide as a PDF \(egghead-team-discussion.pdf\)](#).

Record of analysis: [download the complete /max response \(full-analysis.md?raw=1\)](#). The rendered brief changes presentation, not its release recommendation.

Research-only discussion artifact. No application changes performed. Section illustrations are unavailable because the required Eggo character guide could not be read; no substitute mascot artwork was generated.