

Specification – Finalize `liveSession` for normal Prompt sessions

Context

Slice 1 corresponds to the functional subsystem introduced at commit

`f65adc8e5172cc1e4f9781bf54ffa6dd089d3d33` . The present reinspection was redone from scratch on the current clean repository at commit `c3d66083b46b178e24d4ce497cc3cd5d914a4170` (`c3d66083`). Commits `b42c734c4178d0725ce98933c085a763288eb697` , `613acafc9ac510e432986f56456d63f185d77347` , `b16c5bdde3b4a27cec2ee3f44c9a471b44465ac6` and `0b9f52cc402ce6d0ae19372fa62a432851c248b0` are not direct ancestors of the current HEAD; their relevant common base with HEAD is `ed3bd94687cc34f6603730689bfc775818881bae` . The diff between `f65adc8e5172cc1e4f9781bf54ffa6dd089d3d33` and `ed3bd94687cc34f6603730689bfc775818881bae` is empty. The differences between `b42c734c` and HEAD concern the specification tooling; the additional differences since `ed3bd946` concern that same tooling and personal documentation out of scope. No source tree targeted by this slice has changed.

The current foundation already provides:

- a process-local registry of exact runs;
- a complete ordered buffer of `AiStreamedChunk` ;
- exact viewer leases;
- the `sourceSettled` and `handoffReady` barriers;
- live-safe opening without Orch memory reset;
- backlog + tail replay;
- anti-ABA protection via locator epoch;
- the exact terminal and the distinction between `TOOL_USE` / true end of turn;
- generic Slimbk and message bus transports;
- a generic frontend hook reusing Queue A, Queue B, Prompt locks and sanitizers;
- an EasyAgents adapter preserving historical product behaviour.

The user problem is that a Prompt renderer can turn grey, notably after Electron memory saturation. Today, closing the panel explicitly executes `Abort` → `AwaitStop` → `ClearMemory` , destroys the generation, and forces the last user message to be relaunched. The final slice must allow a normal

Prompt generation to continue within the extension, be visible in Activity, be reopened on its exact source, and reconstruct the history then the stream without loss or duplication.

The slice is atomic: source capture, closing, Activity projection, reopening, replay, Stop, sound, frontend finalization, terminal persistence, cleanup and destructive protections must all ship together. No incomplete intermediate state may be introduced.

The conservation invariants remain mandatory:

- no algorithmic change in Queue A, Queue B, `PromptStreamLockStore` or the sanitizers; only their inputs, outputs and mutations are guarded by attempt identity;
- no duplicate Activity timer or sound during a replay;
- no second Stop button in Prompt;
- no "invisible" panel: a panel is `open` or `closed` ;
- no `*.generated.*` file modified manually;
- no forbidden personal documentation file loaded or modified;
- code, tests, comments and identifiers added or modified in English;
- no `any` , `eslint-disable` , `TODO`, unnecessary cast, production dynamic import, or unrequested implicit compatibility.

Objectives described by the user

Initial need expressed by the user

- Deliver the entire final slice in a single coherent implementation.
- Allow a normal Prompt generation to continue when its panel is closed.
- Keep the item in Activity when the panel is open, or closed with a source still active.
- Remove the item from Activity when the panel is closed and the source has finished.
- Keep the item Idle if the panel is open when the source finishes, including if the `liveSession` record is removed for lack of baseline or true terminal.
- Add a third Activity line with a Stop button only for `Running/source/closed` .
- Use the existing Stop icon, `size="small"` , `variant="danger"` , label `Stop` , with immediate disabling after click.
- Make the detached Stop line disappear as soon as the panel reopens or the source finishes.
- Reuse the composer's normal Stop in the reconnected panel.
- Restore normal Prompt as early as possible: after installing the baseline and arming replay, allow input of the next draft and preparation of attachments, while keeping Send locked.
- Extend SidebarStore to represent session activities and not only open windows.
- Keep the complete raw-live buffer, with no compression or added limit.
- Test all created, moved or modified code, notably the closing/reopening/Stop/transport retry/successive sources/delete/EasyAgents Retry races.

- Update the generic programming docs for `liveSession` , EasyAgents, isomorphism, thinking, Queue B and fallback persist.
- Respect the modular architecture, strict typing and all project quality rules.

Implicit clarifications added by AI Code

- `LiveSessionRunRef` becomes `{ locator, turnId, sourceId }`. `turnId` remains the persistent logical correlation; `sourceId` identifies a precise backend invocation.
- `streamInstanceId` is frontend-only, changes for each stream actually accepted, and replaces `streamKey`. `replayOpenAttemptId` is also frontend-only, changes for each initial Open sequence or Retry, and is never fabricated to represent a stream that never started.
- The private live descriptor comprises exactly `openMode` , `liveSessionTurnId` , `liveSessionViewerLeaseId` , `liveSessionClosePolicy` , `liveSessionSourceId` , `liveSessionAdapter` and `easyAgentsBatchLineNumber` . Any unknown key prefixed `liveSession` , any partial presence, and any critical duplicate make the query invalid.
- The shared contract exposes a strict union of complete descriptors: `easyAgentsBatch + stopOnLastViewerClose + easyAgentsBatchLineNumber + sourceId===turnId` , or `normalPrompt + continueOnPanelClose` with no line number. Extension and frontend use this same union after duplicate detection.
- A valid explicit live query always keeps priority. An invalid explicit live query is rejected before the low-level opener and never degrades into a classic opening. Frontend validation remains defense in depth.
- `Prompt.tsx` is the sole owner of the frozen raw query snapshot. The dispatcher is pure, uses `URLSearchParams.getAll(...)` , never reads the global router, and returns the classification along with a strict locator when it is unambiguous.
- An external React boundary handles invalid queries before mounting the Prompt runtime. No sentinel locator or conditional hook is allowed.
- Any explicit live query, valid or invalid, disables classic hydration and any auto-run. The `normalPrompt` injection also strips all `autoRun*` parameters.
- A stream's origin context is extension-only, non-serialized data: the message bus passes the handler the `HtmlPage` and the exact webview that emitted the request. The start guard compares this webview to the one currently registered for the locator. A simple locator test is insufficient, because an old queued message must never start on a new panel reopened with the same locator.
- `useAiStream.start()` returns synchronously `{ accepted: false } | { accepted: true; runRef; streamInstanceId }` . No buffer, lock, queue, scaffold, placeholder, `uiPending` or continuation ref is mutated on refusal.
- Send, SPECIF, Retry, Edit, Compact context, Reply and Continue all apply pure preparation, call `start()` , then commit exactly one scaffold only after acceptance.

- Prompt's clear/focus barriers are correlated to `{ turnId, streamInstanceId }`. If `fallback_persisted` arrives for this attempt, the composer is cleared as today. If the attempt finalizes without this ack, the transactional mask is removed but the text, attachments, `nextTurnId`, `pendingEditTurnId` and maximized mode remain intact.
- The `onData`, `onComplete`, `onError` callbacks, their continuations after `await`, and the `runMerged` already emitted from Queue A are guarded by `sourceId + streamInstanceId`.
- `completed` or `transportError` is published, and the acceptance lock released, only after Queue A idle, Queue B idle and the last commit barrier.
- If real terminal data has already been consumed and a transport/auxiliary error then arrives, the terminal remains unique; no second `meta.error` or suffix is injected and the outcome becomes `transportError`.
- Direct normalPrompt handoff requires exact terminal data, transport normally completed, Queue A/Queue B drained, and matching `finalizedStreamInstanceId`. It moves directly to `normal` with no second Open and no second composer hydration.
- Every Open response is guarded by `replayOpenAttemptId`. A `terminal` branch allocates no `streamInstanceId`.
- normalPrompt retries use a `preserveLocal` hydration that does not read, validate, normalize or record `promptDraft`, `promptDraftAttachments` or `meta.nextTurnId`. Tests also use trapped getters to prove the real absence of reads.
- The normalPrompt wrapper idempotently persists any `meta.error` received from Orch before appending it to the registry or exposing it. This barrier notably covers the auto-accept path where Orch may have built a terminal while its first durable patch attempt failed.
- Any synthetic terminal wrapper after throw/EOF is created only when there is no prior terminal, durably patched with a canonical suffix, then appended. If persistence fails, no terminal is appended and the transport fails.
- The private V1 protocol only concerns the first source `meta.error` chunk. After a `200_OK` header, a `meta.error` remains a normal body chunk.
- The current Slimbk framework serializes an exception occurring after the `200_OK` header as a non-200 `SlimbkResult` in the body. The generated client currently passes it to the `onChunk` callback. Extension handlers must recognize this envelope before any `AiStreamedChunk` logic, turn it into a transport error, and never buffer it, emit it, or attempt to decode it as the source V1 protocol.
- A source reservation claims the active pointer and advances the epoch before any `await`. Its promotion keeps the same epoch. Cancellation, handoff or retirement remove the pointer, advance the epoch, then record the outcome with the post-retirement epoch.
- A duplicated exact reservation is refused. The handler that did not obtain the reservation may neither call the source endpoint, nor cancel, promote or finalize the first handler's reservation.

- Post-forwarding finalization is reserved to `ownership='source'` and returns exactly `reservationCancelled | handoffReady | retiredMissingBaseline | retiredMissingTerminal | stale`.
- A record with no baseline or no real terminal is removed exactly; no `sourceSettled` record that cannot be handed off remains stuck. If both are missing, `retiredMissingBaseline` takes priority.
- Structured retirements, used for fallback/idempotence, remain distinct from exact process-lifetime tombstones that prevent the resurrection of an observed run key.
- After a complete mutation, the registry emits two deeply immutable wake-ups: observer become inactive, and destructive claim released. Listeners are isolated and these events carry no business decision.
- Durable deletions and EasyAgents Retry use an opaque destructive locator claim. Acquisition and release advance a destructive generation distinct from the run epoch. Public eviction requires the exact handle and never releases the claim.
- A dispose under claim keeps the detached mapping and defers `Abort/Await/ClearMemory`, orphan cleanup, sound and trivial cleanup until the release wake-up, then revalidation.
- Accepting a source Stop also synchronously updates `stopRequested` in the Activity runtime before the abort. Thus, if the panel closes right after the composer's Stop, the detached Stop already appears disabled.
- EasyAgents keeps `sourceId=turnId`, its historical viewer elapsed time, `last viewer → kill`, its business settlement/handoff, and its read-only UX.

Overview of the chosen method

Frontend identity and transactional start

1. Each command builds a pure plan with no visible mutation.
2. `useAiStream.start()` synchronously acquires the single-flight; on refusal it returns immediately with no mutation.
3. On acceptance, it generates `streamInstanceId` and, for a Prompt source, `sourceId`, builds the exact runRef, and returns the identity.
4. The caller then commits the scaffold, the thinking/assistant refs, `uiPending`, scroll and local barriers.
5. The transport effect only starts on the next render; the barriers are therefore armed in the same tick before any network callback.
6. Extension handlers receive an exact, non-serialized origin context. They verify that the emitting webview is still the one registered for the locator before the first mutation.

Backend cycle of a normal Prompt source

1. Exact panel guard by origin webview.
2. Exact source reservation as the first synchronous mutation; claiming the active pointer, epoch bump, and capture of `sourceStartedAtMs`.
3. Synchronous allocation of the Activity runtime/timer and of `activityRunId`; publication is started separately and remains auxiliary.
4. Arming of the exact sound and retrieval of `soundAttemptId`.
5. Optional persistence of the SPECIF Code preference, catalog loading, then capture without `await` of the continuation baseline immediately before `askStream(...)`.
6. Lazy priming of the `Readable`, reservation → record promotion via `registerRun`, application of an early abort.
7. `continueAssistant` immediately publishes its message baseline; `replaceTurnAssistants` waits for `fallback_persisted`.
8. Each ordinary chunk is parsed, possibly persisted if it is a `meta.error`, appended to the registry, then exposed.
9. A throw/EOF with no terminal attempts to persist a canonical terminal before appending; a throw after terminal remains transport.
10. The source service never finalizes the registry.
11. The endpoint encodes only a first `meta.error` in the V1 header. After `200_OK`, chunks remain in the body; a post-header exception becomes a Slimbk error envelope in the body.
12. The extension handler maintains a single FIFO for ordinary chunks and the terminal decoded from the first header. A post-header Slimbk error envelope is intercepted before the FIFO and classified as transport.
13. Auxiliary errors are recorded in order without blocking the chunk's remaining steps or subsequent chunks.
14. After the Slimbk return, FIFO draining, atomic registry finalization, stopping the exact sound, then delegation to the coordinator for Activity/cleanup.
15. The frontend publishes the outcome only after Queue A, Queue B and commit; the Prompt controller then publishes `finalizedStreamInstanceId` after its own UI finalization.

Replay and handoff

- `replaceTurnAssistants` reloads the recent persisted envelope and removes only the exact turn's assistants.
- `continueAssistant` overlays the immutable message baseline on the most recent persisted envelope.
- normalPrompt `not_ready` is retried with a single in-flight Open request and a cancellable delay.
- `not_found`, transportError, completion with no terminal, or eviction with no terminal lead to `liveSessionReplayFailed` for normalPrompt.

- normalPrompt handoff is direct on the replayed state; EasyAgents and already-terminal bootstraps use the terminal snapshot.
- A normalPrompt retry replaces messages and hydration stores but keeps the local composer fully intact, even if Open responds directly with `terminal`.

Registry, retirements, tombstones and claims

The registry distinguishes:

- active source reservation;
- active record;
- structured retirement with outcome + post-retirement epoch;
- exact process-lifetime tombstone;
- opaque destructive locator claim;
- run epoch and destructive generation, independent of each other.

Opens capture and revalidate both generations. An acquire/release cycle only invalidates the in-flight Open. A new source makes any late repeat of an old finalization `stale`. Eviction under claim removes reservations, records, leases and replayable retirements, keeps tombstones, and does not release the claim.

Panels, Activity, sound and Stop

- The public `openCentralPanel(...)` becomes async. It delegates Prompt to the coordinator and keeps the low-level opener private for other pages.
- The coordinator is an injected class that never imports the low-level opener.
- Open, dispose, sourceSettled and wake-ups are serialized by locator in queues removed at idle.
- Low-level dispose synchronously removes the panel from the registry before calling the coordinator.
- The coordinator is the sole owner of a normal source's terminal Activity transition.
- `SidebarActivityQueryItem` is a strict union: Idle/open; Running/observer/open; Running/source/open|closed.
- The Activity runtime keeps runRef, ownership, runId, timestamps, timer, usage and `stopRequested`; transitions are exact by runRef + runId.
- The detached Stop exists only for Running/source/closed. The subtree is keyed by `sourceId`.
- Sound uses the exact runRef for deduplication and `soundAttemptId` to isolate successive viewer attempts.

Destructive deletion and Retry

1. Historical UI/Git prechecks.

2. Synchronous acquisition of the destructive claim right before closing.
3. Panel closing; dispose notices the claim and defers all concurrent cleanup.
4. Durable deletion: eviction only after disk success.
5. EasyAgents Retry: historical pre-delete eviction under the same claim.
6. Release in `finally`, generation bump, and coordinator wake-up.

Exact list of files with implementation details

Shared contracts

1)

`AI Code/src/sharedlib/src/types/liveSessionTypes/LiveSessionReplay.type.ts` – created

- Define `ZLiveSessionReplayMode / LiveSessionReplayMode` :
`replaceTurnAssistants | continueAssistant`.
- Document the frontend reconstruction of each mode.

2)

`AI Code/src/sharedlib/src/types/liveSessionTypes/LiveSessionRun.type.ts` – modified

- Add strict, trimmed, non-empty `LiveSessionSourceId` and make `sourceId` mandatory on the `runRef`.
- Include `sourceId` in the run key, without changing the locator key.
- Document the difference between the logical `turnId` and the executional `sourceId`.

3)

`AI Code/src/sharedlib/src/types/liveSessionTypes/LiveSessionPanel.type.ts` – modified

- Add `LIVE_SESSION_SOURCE_ID_PARAM`, `LIVE_SESSION_ADAPTER_PARAM`, and centralize `EASY_AGENTS_BATCH_LINE_NUMBER_PARAM`.
- Define `ZLiveSessionAdapter : easyAgentsBatch | normalPrompt`.
- Define the canonical readonly list of private parameters.

- Define a strict union `ZLiveSessionPanelDescriptor` : EasyAgents requires `stopOnLastViewerClose` , a positive line, and `sourceId===turnId` ; normalPrompt requires `continueOnPanelClose` and forbids the EasyAgents line.
- Reuse the runRef/sourceld/lease schemas; no extension/frontend parser redefines these pairs.

4)

`AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionOpenResult.type.ts` – modified

- Add `sourceId` to the live/terminal branches and `replayMode` only to live.
- Extend the transport and `superRefine` , including properties present as `undefined` .
- Keep the complete canonical schemas without making `ZChatSessionFile` deep-strict.
- Add `ZLiveSessionOpenResultEnvelope` with `session: unknown` , strict on discriminant and identity, reserved for policy-aware frontend parsing.

5)

`AICode/src/sharedlib/src/types/liveSessionTypes/__test__/LiveSessionContracts.test.ts` – modified

- Cover run keys, sourceld, replay modes, parameters, private list, adapters, complete pairs, EasyAgents source/turn equality, required/forbidden line, and `undefined` properties.
- Compare the complete Open schema and the selective envelope, with no accidental historical tightening.

6)

`AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts` – modified

- Add canonical `SYSTEM_ERROR/INTERRUPTED` factories.
- Define the exact V1 encoder/decoder: `500_SERVER_ERROR` , code/prefix `AICODE_LIVE_SESSION_SOURCE_ERROR_V1` , strict payload `{ version: 1, value }` .
- The decoder accepts a string, an Error, and a structural object; it strictly respects `status` and the proper presence of `errorCode` .
- Extract a useful stack line and reconstruct `AiErrorMetaValue` exactly.

7)

`AICode/src/sharedlib/src/types/aiTypes/__test__/aiStackOpSuffix.util.test.ts` – **modified**

- Test reversibility, statuses/codes, property present as `undefined`, stacks, multi-line details, invalid extras, and canonical chunks.

Orch terminal persistence

8)

`AICode/src/extension/src/aiTools/orch/context/memory/core/msg/OrchMessagesMemoryAssistantPatch.ts` – **modified**

- Add the idempotent canonical suffix for `SYSTEM_ERROR` as for `INTERRUPTED`.
- Preserve blocks, metadata, usage and sink; never duplicate an existing suffix.

9)

`AICode/src/extension/src/aiTools/orch/context/memory/core/msg/__test__/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts` – **created**

- Cover `SYSTEM_ERROR`, `INTERRUPTED`, text absent/present, existing suffix, sink, and isomorphic hydration.

Sidebar contracts

10)

`AICode/src/sharedlib/src/types/busTypes/ui/SidebarStore.ts` – **modified**

- Replace `SidebarActivityQueryWindowItem` with the strict union `SidebarActivityQueryItem`.
- Branches: `Idle/open`; `Running/observer/open`; `Running/source/open|closed`.

- Keep title, elapsed, usage, `updatedAt` , `openedAt` ; include exact `runRef` and `stopRequested` only for source.
- Rename `queryState.windows` to `queryState.items` .

11)

`AICode/src/sharedlib/src/types/busTypes/ui/SidebarUiMessages.ts` – modified

- Add `sidebar->ext:openQueryActivity(locator)` and `sidebar->ext:stopDetachedQuery(runRef) -> { accepted }` .

12)

`AICode/src/sharedlib/src/types/busTypes/global/MessageRegistry.ts` – modified

- Add `sourceId` to the normal and `liveSession` streams.
- Never transport `streamInstanceId` , `replayOpenAttemptId` , or the extension-only `webView` context.

Orch `liveSession` backend

13)

`AICode/src/extension/src/aiTools/orch/liveSession/liveSession.programming-doc-readme.md` – modified

- Document identity, reservations, epochs, replay modes, first-header and post-header-body error protocols, the persistence barrier for any `meta.error` , post-forwarding finalization, retirements/tombstones, claims, wake-ups, exact panel origin, Activity, sound, and frontend-only identities.

14)

`AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRecord.type.ts` – modified

- Add `baselineOnly` , `replayMode`, baseline messages, timestamp, `abortRequested`, source/observer ownership, reservations, and tombstones.

- Define `LiveSessionPostForwardingFinalizationResult` and the opaque handle `LiveSessionLocatorDestructionClaim`.
- Add destructive generation to Open permits.
- Define immutable observer-inactive and claim-released events.
- Define active/reserved snapshots without exposing a mutable.

15)

`AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRegistry.service.ts` – modified

- Implement exact source reservation, cancellation, atomic promotion via `registerRun`, refusal of duplication, and application of early abort.
- The reservation claims the pointer and bumps the epoch; promotion does not rebump it.
- Implement listings/snapshots, exact source/viewer abort claims, and removal of `canAbortSource`.
- Implement destructive claims, destructive generation, eviction requiring the handle, and blocking of reserve/register/Open.
- Finalize the source after forwarding; remove any record with no baseline/terminal, missingBaseline priority, idempotence within the epoch then stale.
- Keep tombstones after eviction and emit both wake-ups after a complete mutation with isolated listeners.

16)

`AICode/src/extension/src/aiTools/orch/liveSession/registry/__test__/LiveSessionRegistry.service.test.ts` – modified

- Cover reservations, duplicate owner, promotion, abort, baseline modes/messages, finalization with five outcomes, stale, retirements/tombstones, successive sources, claims, generations, events, eviction under handle, and a waiter woken with no terminal.

17)

`AICode/src/extension/src/aiTools/orch/liveSession/core/LiveSessionService.ts` – modified

- Expose all new operations/snapshots/claims/subscriptions/finalization.
- Remove `canAbortSource` and make destructive eviction accessible only by exact claim.

18)

`AIcode/src/extension/src/aiTools/orch/liveSession/core/__test__/LiveSessionService.test.ts` – **modified**

- Cover complete forwarding, reservations, successive sources, stale, missing terminal, claims, generations, signals, and tombstones.

19)

`AIcode/src/extension/src/aiTools/orch/liveSession/open/LiveSessionOpen.service.ts` – **modified**

- Return `not_ready` for an exact reservation and refuse under claim.
- Revalidate run epoch + destructive generation around any read.
- Apply `replaceTurnAssistants` or `continueAssistant` on the recent envelope.
- Control terminal fallback via retirement + post-retirement epoch and return `sourceld/replayMode`.

20)

`AIcode/src/extension/src/aiTools/orch/liveSession/open/__test__/LiveSessionOpen.service.test.ts` – **modified**

- Cover reservations, baselines, continuation, `sourceld`, retirements, successive source, claim, invalidated in-flight Open, and a record removed with no terminal.

21)

`AIcode/src/extension/src/aiTools/orch/liveSession/replay/__test__/LiveSessionReplay.service.test.ts` – **modified**

- Add `sourceld`, subscription isolation, waiting for finalization, eviction with no terminal, and public eviction under claim.

22)

`AIcode/src/extension/src/aiTools/orch/liveSession/source/LiveSessionSourceLifecycle.service.ts` – **created**

- Extract the generic OOP lazy `Readable` lifecycle: priming, first next, sync start callback, consumption, return/destroy, cleanup, and deterministic aggregation.
- Allow the caller to forbid thread-level abort on post-return failure.

23)

`AICode/src/extension/src/aiTools/orch/liveSession/source/__test__/LiveSessionSourceLifecycle.service.test.ts` – created

- Cover priming, cleanup, single settlement, combined errors, and no-third-party-abort mode.

24)

`AICode/src/extension/src/aiTools/orch/liveSession/source/NormalPromptLiveSessionSource.service.ts` – created

- Map `AiQuery`→`replayMode` and `baselineOnly` readiness.
- Capture the continuation baseline without `await` before `askStream`, validate the assistant, prime then promote the reservation.
- Publish continuation baseline immediately or replace baseline on `fallback_persisted`.
- Before appending any Orch `meta.error`, idempotently call `patchLastAssistantMeta` with reason/message/details and suffix; append/yield only after success.
- This barrier must fix the case where Orch emitted an auto-accept terminal despite a first durable patch attempt failing.
- Append-before-yield for all chunks; follow the real terminal.
- Synthesize `SYSTEM_ERROR/INTERRUPTED` only with no terminal and after exact persistence; otherwise propagate the transport.
- Never finalize the registry.

25)

`AICode/src/extension/src/aiTools/orch/liveSession/source/__test__/NormalPromptLiveSessionSource.service.test.ts` – created

- Cover all query types, continuation, baseline, order, abort, first error/header, late error, incoming `meta.error` already persisted or requiring a second patch, auto-accept persistence failure, throw/EOF, throw after terminal, and no premature settlement.

EasyAgents adaptation

26)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/EasyAgentsBatchLiveSessionAdapter.service.ts` – **modified**

- `replayMode='replaceTurnAssistants'` , `sourceId=turnId` , timestamp.
- Replace `canAbortSource` with exact claims.
- Expose source/reservation authority, destructive claim, and eviction under handle, without depending on the coordinator.

27)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/__test__/EasyAgentsBatchLiveSessionAdapter.service.test.ts` – **modified**

- Test identity, replayMode, timestamp, claims, eviction handle, and generic wake-up.

28)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunExecution.service.ts` – **modified**

- Build all runRefs with `sourceId=liveTurnId` , without changing Batch transitions.

29)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.live-activity-and-gating.service.test.ts` – **modified**

- Extend identity/replayMode assertions while preserving Watchable/Handoff.

30)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run`

`n/__test__/EasyAgentsBatchRunExecution.commit-and-recovery-failures.service.test.ts` – **modified**

- Add sourceId to handoff assertions without changing the commit/cleanup/publication order.

31)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.post-commit-persistence.service.test.ts` – **modified**

- Add sourceId and preserve the parent patch → publication → handoff order.

32)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunChildExecution.service.ts` – **modified**

- Replace the local lifecycle with the injected `LiveSessionSourceLifecycleService` ; keep parsing/classification/callbacks.

33)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunChildExecution.service.test.ts` – **modified**

- Keep the Batch tests and move the pure lifecycle tests.

34)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunManager.service.ts` – **modified**

- Instantiate/inject the shared lifecycle.
- Migrate `items` and complete the Retry precheck with registry authority.
- Pass destructive claims and eviction under handle to Retry.

35)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.testHarness.ts` – **modified**

- Inject lifecycle and sourceId.

36)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunLifecycle.service.tests.ts` – **modified**

- Adapt construction to the shared lifecycle.

37)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchLastViewerStop.service.test.ts` – **modified**

- Add sourceId and keep the last-viewer/old-new run races.

38)

`AIcode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md` – **modified**

- Document sourceId=turnId, shared lifecycle, observer/open, historical elapsed time, last viewer kill, Retry claim, and observer/claim wake-ups.

Endpoints and message bus

39) `AIcode/src/extension/src/slimb-local/endpoints/ai/orch/liveSession/open.endpoint.ts` – **modified**

- Require sourceId and transport sourceId/replayMode.

40) AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/open.endpoint.test.ts – **modified**

- Cover fields, branches, and mismatches.

41) AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/stream.endpoint.ts – **modified**

- Require sourceId and relay the complete runRef.

42) AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/stream.endpoint.test.ts – **modified**

- Cover identity, terminal, eviction with no terminal, waiting for finalization, and post-header body error.

43) AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abort.endpoint.ts – **modified**

- Require sourceId and use the atomic viewer claim with an active lease.

44) AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/abort.endpoint.test.ts – **modified**

- Cover viewer, stale, double Stop, replacement, and idempotence.

45) AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abortSource.endpo

`int.ts` – created

- Exact Prompt source Stop by locator+turn+source, reservation or record, claim then abort with no intermediate `await`.
- Refuse observer, settled, stale, double claim, and absent.
- After the claim is accepted, synchronously notify the Activity runtime `sourceStopRequested` before the abort; Sidebar publication remains auxiliary.

46) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/abortSource.endpoint.test.ts` – created

- Cover reservation, record, stale, double click, observer, claim → `stopRequested` → abort order, and immediate panel closing after Stop.

47) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/stream.endpoint.ts` – modified

- Add `sourceId`, preserve the SPECIF/catalog order, and delegate to the source service.
- Encode only the first `meta.error` under the V1 header.
- After `200_OK`, send `meta.error` as normal body chunks.
- Never attempt a second application header. A post-header exception is left to the Slimbk framework, whose body envelope will be filtered by extension handlers.
- Do not finalize the registry.

48) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/__test__/stream.endpoint.test.ts` – modified

- Test delegation, first V1 header, late `meta.error` in a 200 body, prior persistence, absence of synthesis after terminal, absence of finalization, and a post-200 exception serialized as a Slimbk body envelope.

49) `AICode/src/extension/src/messageBus/handlers/ia/live`

SessionStreamHandlers.ts – modified

- Relay sourceId and receive the extension-only origin context.
- Refuse a request whose origin webview is no longer the exact registered panel, even if a new panel with the same locator exists.
- Source ownership: no extra timer/sound. Observer: resources only for that exact panel.
- Filter any non-200 post-header Slimbk envelope before the FIFO and classify it as transport.
- Use runRef + soundAttemptId, synchronous Activity allocation, and exact cleanup.

50)

AICode/src/extension/src/messageBus/handlers/ia/__test__/liveSessionStreamHandlers.test.ts – modified

- Cover sourceId, ownership, exact context, replaced old panel, closed observer, failed Activity publication, stale sound, post-header body-error, terminal, and eviction.

51)

AICode/src/extension/src/messageBus/handlers/ia/iaStreamHandlers.ts – modified

- Receive sourceId and the exact origin context.
- Verify webview + locator before reservation; an old message from a disposed panel must not use a panel reopened with the same locator.
- Reserve, allocate Activity and sound synchronously before any business `await`.
- Start Activity publication as an auxiliary task without blocking the endpoint.
- Maintain a single FIFO; `onChunk` first filters the post-header Slimbk envelope, otherwise queues the real chunk.
- `onError` first decodes the first-header V1 protocol, otherwise records transport with no throw; idempotent callback.
- Each FIFO task attempts Activity, sound and emit separately, continues with the chunks, and aggregates errors in order.
- After call + FIFO: determine completed/transport, finalize only if the reservation is owned, stop the sound per attempt, and call `notifyPromptSourceSettled(...)` exactly once.

52)

AICode/src/extension/src/messageBus/handlers/ia/__test__/iaStreamHandlers.queryDoneSound.test.ts – modified

- Cover exact origin, replaced panel, sourceId/soundAttemptId, auxiliary Activity, FIFO, first source-error header, meta.error body, post-header Slimbk body error, terminal then transportError, non-short-circuiting auxiliary errors, idempotent onError, and reservation conflict.

53)

`AIcode/src/extension/src/messageBus/handlers/ia/__test__/iaStreamHandlers.runIdPropagation.test.ts` – **modified**

- Verify the reservation → allocation → endpoint/FIFO → finalization → sound → coordinator order, terminal header within the FIFO, and transport with no terminal removing the run.

54)

`AIcode/src/extension/src/extension/panels/prompt/PromptQueryDoneSound.manager.ts` – **modified**

- Correlate by exact runRef; `onAiStreamStart` returns a monotonic `soundAttemptId`.
- Require runRef+attempt for chunk/stop, ignore stale, never implicitly create state.
- Deduplicate terminal by runRef without rearming on reconnection.
- Keep state as long as source/observer is active; cleanup only via the coordinator.

55)

`AIcode/src/extension/src/extension/panels/prompt/__test__/PromptQueryDoneSound.manager.test.ts` – **modified**

- Cover successive sources, attempts of the same run, stale, dedup, closing, and source preservation.

Extension Activity runtime

56)

`AIcode/src/extension/src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts` – **modified**

- Rename the internal class `SidebarQueryActivityRuntimeService`.

- Store runRef, ownership, runId, startMs/sourceStartedAtMs, timer, usage and stopRequested per locator.
- Expose a synchronous start allocation { runId, publication }.
- Expose onSourceStopRequested(runRef) : synchronous internal/store mutation, auxiliary broadcast, runRef+runId guard.
- Expose a readonly snapshot for rehydrate and a source terminal transition called only by the coordinator.
- Source elapsed time from sourceStartedAtMs; observer from the viewer attempt.

57)

`AICode/src/extension/src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.race.test.ts` – **modified**

- Cover successive sources, stale, open/closed, allocation before broadcast, failed broadcast, exact stopRequested, and closing after Stop with no button reactivation.

58)

`AICode/src/extension/src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.usage.test.ts` – **modified**

- Cover usage, Idle/removal, missing terminal, failed publication, and exact timer stop.

Prompt panel coordination

59)

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelQuery.util.ts` – **created**

- Parse raw occurrences, detect any explicit query, unknown `liveSession*` keys, locator/live duplicates, and validate with the shared `ZLiveSessionPanelDescriptor`.
- Preserve any valid explicit query; reject invalid ones.
- Build normalPrompt with a fresh lease, exact sourceId/adaptor/policy, and complete removal of `autoRun*`.

60)

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelQuery.util.ts` – **created**

eSession/__test__/PromptLiveSessionPanelQuery.util.test.ts – created

- Cover shared pairs, EasyAgents source/turn equality, required/forbidden line, duplicates, unknowns, order, lease, invalid-before-opener, and autoRun.

61)

AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelCoordinator.service.ts – created

- Injected class with no low-level import, FIFO per locator.
- `openPromptPanel`, `onPanelDisposed`, `onSourceSettled`, observer/claim wake-ups.
- Preserve valid explicit, reject invalid, inject normalPrompt only for an active/reserved source.
- Adapter-specific releases; normalPrompt only releases the lease.
- Sole owner of the Activity terminal transition and revalidated cleanups.
- Under claim, keep the detached mapping and defer all cleanups.

62)

AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionPanelCoordinator.service.test.ts – created

- Cover query priorities, open/close/settle races, missing terminal, Idle open, observer wake-up, claim wake-up, reopening/new source, and single cleanup.

63)

AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionDetachedStop.service.ts – created

- Synchronously validate the exact source/closed projection, an active/reserved source, and an available abort claim.
- Share the Activity runtime's `onSourceStopRequested` transition, then abort with no intermediate await.
- Never abort a different source from a stale projection.

64)

`AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionDetachedStop.service.test.ts` – **created**

- Cover accept/refuse, reopened, replacement, double click, reservation, stale store, destructive claim, and stopRequested/abort order.

65)

`AICode/src/extension/src/extension/panels/prompt/PromptThreadCleanup.manager.ts` – **modified**

- Index by full locator; separate register, detach, historical cleanup, terminal cleanup, and orphan cleanup.
- Never abort/clear an active source or observer.
- Keep the detached mapping during deferred cleanups and under claim; revalidate after each await.

66)

`AICode/src/extension/src/extension/panels/prompt/__test__/PromptThreadCleanup.manager.attachmentsCleanup.test.ts` – **modified**

- Cover scopes, detach, terminal, new source, observer wake-up, claim wake-up, attachments, and idempotence.

67)

`AICode/src/extension/src/extension/panels/core/openPanel.util.ts` – **modified**

- Keep the public name `openCentralPanel(...)` but make it `async`.
- Extract a private synchronous low-level opener; build the module-scoped coordinator with this injected port.
- Prompt goes through the coordinator; other pages go directly through the low-level opener.
- Export `notifyPromptSourceSettled(...)` for the source handler.

- Capture the initial descriptor; on dispose, synchronously remove the panel from the registry before any delegation.
- Adapt titles/Activity to `items/status` , inject subscriptions and runtime, preserve absence of cycles.

68)

`AICode/src/extension/src/extension/panels/core/__test__/openPanel.util.test.ts` – **modified**

- Await facade, wiring with no cycle, exact descriptor, reveal, dispose order, sourceSettled, subscriptions, claim, and non-Prompt pages.

69)

`AICode/src/extension/src/messageBus/handlers/ui/uiGenericHandlers.ts` – **modified**

- Await `openCentralPanel` ; Prompt is coordinated, other pages unchanged.

70)

`AICode/src/extension/src/messageBus/handlers/ui/__test__/uiGenericHandlers.openExternalUrl.test.ts` – **modified**

- Mock the async facade and preserve the URL.

71)

`AICode/src/extension/src/extension/commands/NewChat.command.ts` – **modified**

- Await the coordinated opening.

72)

`AICode/src/extension/src/extension/commands/__test__/NewChat.command.test.ts` – **modified**

- Verify await and error propagation.

73)

`AICode/src/extension/src/extension/uriHandlers/registerDeepLinkUriHandler.ts` – **modified**

- Deep-link chat via the coordinated facade, fire-and-forget with an explicit catch.

74)

`AICode/src/extension/src/extension/uriHandlers/__test__/registerDeepLinkUriHandler.scope.test.ts` – **modified**

- Cover scopes, facade, and guards.

SidebarStore and Activity

75)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.listeners.manager.ts` – **modified**

- Atomic transitions `panelOpened`, `sourceStarted`, `sourceStopRequested`, `sourceSettled`.
- `SourceStarted` receives an already-allocated run/runtime; publication is separate.
- Non-stale final mapping: `open`→`idle`, `closed`→`remove`; stale/different source is a no-op.

76)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.mutations.manager.ts` – **modified**

- `windows`→`items`, `panelClosed`, `forceRemove`, exact elapsed and `stopRequested`.
- Keep only `Running`/source closed; on `panelClosed`, derive `stopRequested` from the exact registry/runtime snapshot.

77)

`AICode/src/extension/src/extension/panels/sidebar/Si`

`debarStore.core.manager.ts` – modified

- Initialize items and call `rehydrateQueryActivities`.

78)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.rehydrate.manager.ts` – modified

- Merge panels, active observers, and sources/reservations; deduplicate.
- Forbid observer/closed and Idle/closed.
- Keep an unfinalized buffered terminal in Running; a panel open with no record in Idle.
- Reconstruct title/openedAt via runtime/store/panel/persistence/default in that order, without making SidebarStore the authority on timers.

79)

`AICode/src/extension/src/extension/panels/sidebar/__tests__/SidebarStore.rehydrate.manager.test.ts` – modified

- Cover all projections, missing publication, reservation, closed observer, pre-forwarding terminal, missing terminal, and stopRequested.

80)

`AICode/src/extension/src/extension/panels/sidebar/__tests__/SidebarStore.manager.queryRunningUpsert.test.ts` – modified

- Test union, transitions, auxiliary publication, strict outcomes, and sourceStopRequested.

81)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.chatTitles.manager.ts` – modified

- Read items and preserve the fallback title for a closed/running source.

82)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.handlers.manager.ts` – **modified**

- Register open activity and detached Stop; delegate to coordinator/service.

83)

`AICode/src/extension/src/extension/panels/sidebar/__test__/SidebarStore.handlers.manager.test.ts` – **created**

- Verify payloads, accepted, and delegation.

84) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/delete.endpoint.ts` – **modified**

- Read items and liveSession authority.
- Acquire a claim before closing; close/remove the projection under claim.
- Durable delete with no concurrent cleanup; eviction only after success, with the exact handle.
- Release in finally; no preventive eviction on disk failure.

85) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/delete.endpoint.liveSession.test.ts` – **created**

- Cover stale store, reservation, unavailable claim, refused concurrent start, exact handle, success/failure, tombstones, and deferred cleanup.

86) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/list.endpoint.withLock.eperm.test.ts` – **modified**

- Adapt items fixtures.

87) AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/diskAliases.lastWins.endpoints.test.ts – modified

- Adapt items and preserve disk delete.

88) AICode/src/extension/src/aiTools/orch/context/persistence/__test__/ChatPersistenceWriter.windowsFallback.test.ts – modified

- Adapt items fixture.

89) AICode/src/extension/src/aiTools/orch/context/memory/core/mem/__test__/OrchMemory.naive_title.user_prompt_specif_multiline.e2e.repro.test.ts – modified

- Adapt items fixture.

90) AICode/src/frontend/src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx – modified

- Consume items/union, open via the new RPC, and show the Stop third line only for source/closed.
- `SquareStop`, small, danger, label Stop, disableAfterClick, enabled `!stopRequested`, stable testIds.
- Callbacks via `useCallback`, objects via `useMemo`, no hook inside map; Stop sub-component keyed by sourceId.

91) AICode/src/frontend/src/app/sidebar/sidebar-activity/__test__/SidebarActivityQuery.test.tsx – modified

- Cover projections, open RPC, exact Stop, disabling, disappearance, and A→B source remount with no fragile labels.

Frontend `liveSession` and Prompt

92)

`AICode/src/frontend/src/app/prompt/liveSession/adapters/NormalPromptLiveSession.adapter.ts` – **created**

- Bind the shared normalPrompt descriptor and not_ready/replayFailed/direct handoff/interactive composer policies.
- No banner Stop; single composer Stop.
- Initial baseline = full hydration; retries = preserveLocal.

93)

`AICode/src/frontend/src/app/prompt/liveSession/adapters/__test__/NormalPromptLiveSession.adapter.test.ts` – **created**

- Cover parsing/binding, policies, surfaces, banner, handoff, and preserveLocal.

94)

`AICode/src/frontend/src/app/prompt/liveSession/adapters/PromptLiveSessionBootstrap.adapter.ts` – **created**

- Pure dispatcher receiving the raw snapshot; getAll on critical parameters.
- Import the shared list/union; detect partials, unknowns, duplicates, and ambiguous locator.
- Return classic/valid/invalid, an optional strict locator, and an explicit bool; no global read.

95)

`AICode/src/frontend/src/app/prompt/liveSession/adapters/__test__/PromptLiveSessionBootstrap.adapter.test.ts` – **created**

- Cover dispatch, identical/different duplicates, unknowns, adapter, EasyAgents equality, invalid with no locator, and purity.

96)

`AIcode/src/frontend/src/app/prompt/liveSession/core/PromptLiveSession.types.ts` – modified

- Add `liveSessionReplayFailed`, `sourcelid`, and explicit policies: `composer`, `Send`, `history`, `cleanup`, `Stop`, `handoff`, `snapshot policy`, `banner`.
- Separate `PromptLiveSessionBannerViewModel` from the `composer` surface.
- Represent `directAfterReplay` vs `terminalSnapshot` and `replayOpenAttemptId` vs `streamInstancelid`.

97)

`AIcode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.tsx` – modified

- Consume the dedicated banner model, compatible with the `EasyAgents` viewer and the `normalPrompt` banner above the `composer`.
- Keep `status/details/Stop/Retry`, stable callbacks, and `testIds`.

98)

`AIcode/src/frontend/src/app/prompt/liveSession/components/__test__/PromptLiveSessionBanner.test.tsx` – modified

- Adapt fixtures and add `normalPrompt` `Retry` with no `Stop`.

99)

`AIcode/src/frontend/src/app/prompt/liveSession/core/usePromptLiveSession.hook.ts` – modified

- Maintain `replayOpenAttemptId` for `bootstrap/retry/timers/unmount`.
- Parse the envelope first; full parse for `hydrateFromSnapshot`, opaque session for `preserveLocal`.
- Retry `normalPrompt` `not_ready`; `not_found/transport/completion` with no `terminal/eviction` → `replayFailed`.
- Start the observer only after `start` is accepted and keep the exact `streamInstancelid`.
- Direct `handoff` with no second `Open` after `terminal data + completed + queues + exact finalization`.
- `Terminal` `retry` with no fictitious stream; `preserveLocal` for `messages/stores` only.

- Handle transportStopAvailable and explicit removal on terminal/not_found/reset/Stop.

100)

AI Code/src/frontend/src/app/prompt/liveSession/core/
__test__/usePromptLiveSession.hydration.test.tsx –
modified

- Cover adapter policies, full vs preserveLocal, malformed draft/unread getters, and stale Open/timer/unmount.

101)

AI Code/src/frontend/src/app/prompt/liveSession/core/
__test__/usePromptLiveSession.handoff.test.tsx –
modified

- Cover EasyAgents terminal, direct normal, Queue B, stale finalization, terminal data + transportError, source error data, and absence of a second Open.

102)

AI Code/src/frontend/src/app/prompt/liveSession/core/
__test__/usePromptLiveSession.normalPrompt.test.tsx –
created

- Cover not_ready/not_found, surface, draft/attachments/nextTurnId, Stop, retry, direct terminal, terminal retry with no stream, opaque draft envelope, abortable replayFailed, and stale attempts.

103)

AI Code/src/frontend/src/app/prompt/liveSession/core/
__test__/usePromptLiveSession.normalPromptIsomorphism
.test.tsx – created

- Compare stream/replay/hydration for all query types, thinking, tools, usage, edits, errors, EOF, and post-terminal transport.

104)

`AIcode/src/frontend/src/app/prompt/liveSession/adapters/EasyAgentsBatchLiveSession.adapter.ts` – **modified**

- Import the shared constant, add `sourceId=turnId/adapter`, validate the shared union, and preserve `UX/read-only/elapsed`.

105)

`AIcode/src/frontend/src/app/prompt/liveSession/adapters/__test__/EasyAgentsBatchLiveSession.adapter.test.ts` – **modified**

- Adapt `query/sourceId/dispatcher/constant` and the full-hydration policy.

106)

`AIcode/src/frontend/src/messageBus/AiStreamStartArgs.type.ts` – **modified**

- Define the internal `runRef+streamInstanceId+outcome` state, the start return union, and the `AiStreamFallbackPersistedAck { turnId; streamInstanceId }` type.
- The observer transports the full `runRef`, `lease`, `replayMode`, and `Stop` policy.
- No frontend-only identifier reaches the backend beyond the `runRef`'s `sourceId`.

107)

`AIcode/src/frontend/src/messageBus/useAiStream.hook.ts` – **modified**

- Synchronous single-flight ref; IDs generated only after acceptance.
- Return an `accept/refuse` union and mutate nothing on refusal.
- Use `streamInstanceId` as the listener key; expose `runRef`, IDs, `outcome`, terminal data, `transportStopAvailable`, `Stop` state, and the fallback ack correlated to `streamInstanceId`.
- Keep `callbacks/runMerged` before/after awaits; invalidate before reset.
- Wait for Queue A, Queue B, then commit before `outcome/release`.
- `onError` with no terminal creates `SYSTEM_ERROR`; after terminal keeps a unique terminal and a `transportError` outcome.

- Stop source via abortSource, observer via lease abort; availability consumed before the first await.

108)

`AICode/src/frontend/src/messageBus/__test__/useAiStream.hook.liveSessionObserver.test.ts` – **modified**

- Cover IDs, routes, stale callbacks/runMerged, Queue B, outcomes, terminal+error with no duplicate, transportStopAvailable, and correlated ack.

109)

`AICode/src/frontend/src/messageBus/__test__/useAiStream.hook.promptSourceIdentity.test.ts` – **created**

- Cover ID generation after acceptance, double start in the same tick, refusal with no mutation, observer sourceId provided, Stop result, and stale callbacks.

110)

`AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.hook.ts` – **modified**

- Propagate start results, runRef/streamInstance/outcome/terminal/transportStopAvailable/correlated fallback ack.
- StartLiveObserver prepares/commits placeholders only after acceptance.
- Bind the thinking base to the accepted streamInstanceId.
- Expose `finalizedStreamInstanceId`, clear transport availability, and full reset.

111)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunController.liveSessionObserver.test.tsx` – **modified**

- Cover accept/refuse, no placeholder on refusal, IDs, thinking base, post-transport Stop, and finalization ack.

112)

`AICode/src/frontend/src/app/prompt/Prompt/usePromptCancelController.hook.ts` – **modified**

- Separate `isStreaming` and `transportStopAvailable`.
- UI-only cancel only if both are false; otherwise a single exact Stop.
- For `replayFailed` outside transport, finish the handshake after the Stop response and keep `availability` false.

113)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptCancelController.stopCore.test.tsx` – **modified**

- Add the `isStreaming=false + transportStopAvailable=true` case, double click, and absence of a UI-only source mutation.

Continuation and thinking

114)

`AICode/src/frontend/src/app/prompt/Prompt/preparePromptContinuationStreaming.util.ts` – **created**

- Pure preparation: exact non-thinking assistant, assistant/thinking bases, reused UID, unique thinking placeholder, separator, and planned resets with no mutation.

115)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/preparePromptContinuationStreaming.util.test.ts` – **created**

- Cover bases, missing assistant, separator, successive sources, and no duplication/mutation.

116)

`AICode/src/frontend/src/app/prompt/Prompt/preparePromptStreamingTurn.util.ts` – **modified**

- Separate the pure plan and the commit; reset the thinking base/separator for replace.

117)

`AIcode/src/frontend/src/app/prompt/Prompt/__test__/preparePromptStreamingTurn.util.test.ts` – **modified**

- Verify the plan is not committed on refusal, and replace resets.

118)

`AIcode/src/frontend/src/app/prompt/Prompt/usePromptThinkingLifecycle.hook.ts` – **modified**

- Initialize the thinking base per streamInstanceId, unique append/separator, explicit reset, and cleanup only when the base is empty.

119)

`AIcode/src/frontend/src/app/prompt/Prompt/__test__/usePromptThinkingLifecycle.continuation.test.tsx` – **created**

- Cover base, append, tool/text before thinking, reinsertion, reset, and retry of the same source/new attempt.

120)

`AIcode/src/frontend/src/app/prompt/Prompt/usePromptToolUse.hook.ts` – **modified**

- Only remove the bubble if baseThinking is empty; avoid double separators.

121)

`AIcode/src/frontend/src/app/prompt/Prompt/usePromptRunController.finalization.hook.ts` – **modified**

- Keep all mutations by streamInstanceId, reset bases, and publish finalizedStreamInstanceId after complete finalization.

122)

`AIcode/src/frontend/src/app/prompt/Prompt/usePromptRunCommands.hook.ts` – **modified**

- Pure plans + synchronous start + commit after acceptance for Send/SPECIF/Retry/Edit/Compact/Reply/Continue.
- Return the strict result to the caller; remove artificial asyncs.
- On refusal, zero durable mutation and `uiPending=false` in the same tick.

123)

`AIcode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.compactContextNow.test.tsx` – **modified**

- Adapt acceptance and confirm replace with no inherited base.

124)

`AIcode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.retryQuery.attachmentOnly.test.tsx` – **modified**

- Adapt acceptance and preserve attachment-only.

125)

`AIcode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.retryQuery.compactContextNow.test.tsx` – **modified**

- Adapt acceptance and preserve the compact mapping.

126)

`AIcode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.continueResponse.test.tsx` – **created**

- Verify continuation, unique thinking, same turn/new attempt, and refusal with no mutation.

Prompt integration and documentation

127)

`AICode/src/frontend/src/app/prompt/Prompt/Prompt.tsx`

– modified

- Separate the external bootstrap boundary and the resolved internal runtime.
- Read the raw snapshot only once; no live decision via scalar hooks.
- Invalid with no locator: dialog/close with no controller, hydration, or autoRun.
- Separately derive composer, Send, Stop, history, cleanup, and banner.
- Allow draft/attachments during normalPrompt replay according to policy.
- Use transportStopAvailable in locks/hotkeys/cancel.
- After Send/Edit is accepted, store { `turnId`, `streamInstanceId` } in the fallback barrier and only then arm submitTxn/focus.
- Single effect: matching ack → clear history; matching finalization with no ack → unmask/restore focus without erasing text, attachments, nextTurnId, pendingEditTurnId, or maximized.
- Any explicit live query disables classic hydration and autoRun.

128)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/P`

`rompt.liveSessionBootstrap.test.tsx` – modified

- Cover raw snapshot, duplicates, invalid with no runtime, neutralized autoRun, surface/replay failure, and start refusal with no barriers.

129)

`AICode/src/frontend/src/app/prompt/PromptInput/Prompt`

`tInput.tsx` – modified

- Support composer alone, viewer banner replacing the composer, or composer + normalPrompt banner.
- normalPrompt banner above the textarea, with no second Stop; VscTextarea remains the single Stop.
- Separate showComposer, edit, Send lock, and banner model.

130)

`AICode/src/frontend/src/app/prompt/PromptInput/Prompt`

`tInput.module.css` – modified

- Add the normalPrompt banner's wrapper/spacing, VS Code variables only, no inline CSS.

131)

`AICode/src/frontend/src/app/prompt/PromptInput/__test__/PromptInput.batchLiveViewerMode.test.tsx` – modified

- Preserve the EasyAgents viewer; add normalPrompt composer + Retry banner with no Stop, attachments, and Send lock.

132) `AICode/src/frontend/src/app/easy-agents/easy-agents-batch/core/__test__/EasyAgentsBatch.liveWatch.test.tsx` – modified

- Adapt query source/adapter and the shared constant.

133) `AICode/src/frontend/src/app/easy-agents/easy-agents-item/core/batch/__test__/EasyAgentBatchSurface.test.tsx` – modified

- Adapt the exact query with no UX change.

134)

`AICode/src/frontend/src/app/prompt/hydration/isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-readme.md` – modified

- Document closing/reopening, continuation/thinking, finalization, first-header error, late meta.error, filtered post-header Slimbk envelope, persistence of any meta.error before appending, unique terminal+transportError, direct handoff, and opaque preserveLocal.

135) `AIcode/src/frontend/src/app/prompt/thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md` – modified

- Document thinking base continuation, unique separator, text/tool guards, and streamInstanceId.

Cross-cutting frontend contracts

136)

`AIcode/src/frontend/src/app/prompt/Prompt/applyHydratedPromptSessionState.util.ts` – modified

- Discriminated union `hydrateFromSnapshot | preserveLocal`.
- Full: current behaviour.
- Preserve: no composer setter, no reading/validation/normalization of draft/attachments/nextTurnId; build a sanitized FrontChatSession only from consumed fields.
- Trapped getters on ignored fields must not be executed.
- `void` return; doc message refs always recorded.

137)

`AIcode/src/frontend/src/app/prompt/Prompt/__test__/applyHydratedPromptSessionState.util.test.ts` – modified

- Adapt full and cover `preserveLocal`, unread getters, messages/stores/thread/reset/scroll, and void return.

138)

`AIcode/src/frontend/src/app/prompt/Prompt/usePromptSessionHydration.hook.ts` – modified

- Call `hydrateFromSnapshot` ; disable for any explicit live query.

139)

`AIcode/src/frontend/src/app/prompt/hydration/frontSessionHydration.type.ts` – modified

- Generalize `allowOrphanToolResults=true` to the EasyAgents and normalPrompt live-safe baselines, with no change to pairing.

140)

`AI Code/src/frontend/src/app/prompt/lib/usePromptSessionLocator.hook.ts` – **modified**

- Add `usePromptRouterQueryStringRawSnapshot()` via `useMemo(..., [])`.
- Keep existing locator parsers for other consumers; document that the snapshot is the Prompt bootstrap path.

141)

`AI Code/src/frontend/src/app/prompt/lib/__test__/usePromptSessionLocator.hook.test.tsx` – **modified**

- Add a harness proving the snapshot is frozen after a global mutation/rerender.

142)

`AI Code/src/frontend/src/app/prompt/Prompt/__test__/Prompt.autoRunPrompt.test.tsx` – **modified**

- Mock the classic raw snapshot and the acceptance union; preserve classic `autoRun`.

143)

`AI Code/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.noThumbTurnIdDrift.test.tsx` – **modified**

- Mock snapshot; synchronous Edit acceptance; preserve `turnId` thumbnails until ack.

144)

`AI Code/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.restoreAttachments.test.tsx` – **modified**

- Mock snapshot and preserve attachment restoration.

145)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.ensureNextTurnId.stability.test.tsx` – **modified**

- Mock snapshot/acceptance and preserve nextTurnId.

146)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearAttachments.test.tsx` – **modified**

- Verify clearing only on an exact ack, and preservation of attachments/nextTurnId/pending edit during a finalization with no ack.

147)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearInput.test.tsx` – **modified**

- Verify the mask is armed after acceptance, cleared on an exact ack, unmasked with no clear on a finalization with no ack, and focus restored.

148)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.promptDraft.hydration.test.tsx` – **modified**

- Mock snapshot/acceptance and preserve debounce/hydration.

149)

`AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.startAcceptance.test.tsx` – **created**

- Common harness for seven commands; refusal with no mutation, acceptance with exactly one scaffold, and an exact synchronous result.

150) `AIcode/src/frontend/src/messageBus/prompt-stream-dispatch-lock.programming-doc-readme.md` – modified

- Document streamInstance guards, Queue A+B idle, final commit, forbidden stale reset, and terminal+transportError with no duplicate.

151)

`AIcode/src/extension/src/aiTools/orch/ask/stream/core/fallback-persist-and-title.programming-doc-readme.md` – modified

- Document barriers armed after acceptance.
- Add the case of an accepted attempt finalizing with no fallback ack: unmask with no erasing of the draft/attachments/nextTurnId/edit state.

Destructive atomicity of EasyAgents Retry

152)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunRetry.service.ts` – modified

- After non-destructive guards, acquire the exact claim right before closing.
- Hold the claim during closing, pre-delete eviction, UI removal, and delete.
- Pass the exact handle to eviction; release in finally on every outcome.
- Preserve all Git/reconstruction/viewer rules.

153)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunRetry.testHarness.ts` – modified

- Add strict claim/release/authority/evict(handle) mocks, with no unnecessary casts.

154)

`AIcode/src/extension/src/aiTools/easyAgents/batch/run`

n/__test__/EasyAgentsBatchRunRetry.guards.service.test.ts – modified

- Cover unavailable claim, order, exact handle, release success/errors, refused concurrent source, and deferred cleanup.

Transport barriers and exact origin revealed by the reinspection

155)

AIcode/src/sharedlib/src/slimbk/framework/parseSlimbkStreamBodyError.util.ts – created

- Detect, before any business logic, a non-200 `SlimbkResult` envelope appearing in the body after a 200 header.
- Validate a known status, a string errorMessage, and an optional errorCode; any shape with an inconsistent top-level `status` field becomes a protocol error, never an `AiStreamedChunk`.
- Never interpret this path as the source V1 protocol, even if text/code coincide.

156)

AIcode/src/sharedlib/src/slimbk/framework/__test__/parseSlimbkStreamBodyError.util.test.ts – created

- Cover valid envelopes, ignored 200, malformed statuses/codes, ordinary objects/chunks, and a deliberate collision with the V1 prefix.

157)

AIcode/src/extension/src/messageBus/core/extTypedMessageBus.util.ts – modified

- Define an extension-only `StreamHandlerOriginContext` containing `htmlPage` and the exact webview.
- Extend `onStreamed`, `dispatchStreamedCall` and `getStreamHandler` to pass it as a third argument without touching the serialized contracts.

158)

AIcode/src/extension/src/messageBus/core/extMessageB

us.util.ts – modified

- Build the origin context in `initializeViewWithMessageBus` and pass it down the stream path.
- Require this context for a webview stream; non-stream RPC calls remain unchanged.

159)

AICode/src/extension/src/messageBus/core/__test__/extMessageBus.streamOriginContext.test.ts – created

- Prove that the handler receives the exact emitting webview, that two Prompt panels have distinct contexts, and that no data is added to the serialized payload.

160)

AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.module.css – modified

- Update the maintenance comment: the banner is no longer purely read-only, nor necessarily rendered in place of the composer.
- Keep the VS Code variables and existing style, except for adjustments strictly necessary for the reusable model.

Exact summary of file paths

Created

- `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionReplay.type.ts`
- `AICode/src/extension/src/aiTools/orch/context/memory/core/msg/__test__/_OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/source/LiveSessionSourceLifecycle.service.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/source/__test__/LiveSessionSourceLifecycle.service.test.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/source/NormalPromptLiveSessionSource.service.ts`

- `AICode/src/extension/src/aiTools/orch/liveSession/source/__test__/NormalPromptLiveSessionSource.service.test.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abortSource.endpoint.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/abortSource.endpoint.test.ts`
- `AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelQuery.util.ts`
- `AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionPanelQuery.util.test.ts`
- `AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelCoordinator.service.ts`
- `AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionPanelCoordinator.service.test.ts`
- `AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionDetachedStop.service.ts`
- `AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionDetachedStop.service.test.ts`
- `AICode/src/extension/src/extension/panels/sidebar/__test__/SidebarStore.handlers.manager.test.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/delete.endpoint.liveSession.test.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/adapters/NormalPromptLiveSession.adapter.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/adapters/__test__/NormalPromptLiveSession.adapter.test.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/adapters/PromptLiveSessionBootstrap.adapter.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/adapters/__test__/PromptLiveSessionBootstrap.adapter.test.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.normalPrompt.test.tsx`
- `AICode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.normalPromptIsomorphism.test.tsx`

- `AICode/src/frontend/src/messageBus/__test__/useAiStream.hook.promptSourceIdentity.test.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/preparePromptContinuationStreaming.util.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/preparePromptContinuationStreaming.util.test.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptThinkingLifecycle.continuation.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.continueResponse.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.startAcceptance.test.tsx`
- `AICode/src/sharedlib/src/slimbk/framework/parseSlimbkStreamBodyError.util.ts`
- `AICode/src/sharedlib/src/slimbk/framework/__test__/parseSlimbkStreamBodyError.util.test.ts`
- `AICode/src/extension/src/messageBus/core/__test__/extMessageBus.streamOriginContext.test.ts`

Modified

- `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionRun.types.ts`
- `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionPanel.type.ts`
- `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionOpenResult.type.ts`
- `AICode/src/sharedlib/src/types/liveSessionTypes/__test__/LiveSessionContracts.test.ts`
- `AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts`
- `AICode/src/sharedlib/src/types/aiTypes/__test__/aiStopSuffix.util.test.ts`
- `AICode/src/extension/src/aiTools/orch/context/memory/core/msg/OrchMessagesMemoryAssistantPatch.ts`
- `AICode/src/sharedlib/src/types/busTypes/ui/SidebarStore.ts`
- `AICode/src/sharedlib/src/types/busTypes/ui/SidebarUiMessages.ts`
- `AICode/src/sharedlib/src/types/busTypes/global/MessageRegistry.ts`

- `AICode/src/extension/src/aiTools/orch/liveSession/liveSession.programming-doc-readme.md`
- `AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRecord.type.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRegistry.service.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/registry/__test__/LiveSessionRegistry.service.test.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/core/LiveSessionService.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/core/__test__/LiveSessionService.test.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/open/LiveSessionOpen.service.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/open/__test__/LiveSessionOpen.service.test.ts`
- `AICode/src/extension/src/aiTools/orch/liveSession/replay/__test__/LiveSessionReplay.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/EasyAgentsBatchLiveSessionAdapter.service.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/__test__/EasyAgentsBatchLiveSessionAdapter.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunExecution.service.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.live-activity-and-gating.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.commit-and-recovery-failures.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.post-commit-persistence.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunChildExecution.service.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunChildExecution.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunManager.service.ts`

- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.testHarness.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunLifecycle.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchLastViewerStop.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunRetry.service.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunRetry.testHarness.ts`
- `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunRetry.guards.service.test.ts`
- `AICode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/open.endpoint.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/open.endpoint.test.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/stream.endpoint.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/stream.endpoint.test.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abort.endpoint.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/abort.endpoint.test.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/stream.endpoint.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/__test__/stream.endpoint.test.ts`
- `AICode/src/extension/src/messageBus/handlers/ia/liveSessionStreamHandlers.ts`
- `AICode/src/extension/src/messageBus/handlers/ia/__test__/liveSessionStreamHandlers.test.ts`
- `AICode/src/extension/src/messageBus/handlers/ia/iaStreamHandlers.ts`
- `AICode/src/extension/src/messageBus/handlers/ia/__test__/iaStreamHandlers.queryDoneSound.test.ts`

- `AICode/src/extension/src/messageBus/handlers/ia/__test__/iaStreamHandlers.runIdPropagation.test.ts`
- `AICode/src/extension/src/extension/panels/prompt/PromptQueryDoneSound.manager.ts`
- `AICode/src/extension/src/extension/panels/prompt/__test__/PromptQueryDoneSound.manager.test.ts`
- `AICode/src/extension/src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts`
- `AICode/src/extension/src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.race.test.ts`
- `AICode/src/extension/src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.usage.test.ts`
- `AICode/src/extension/src/extension/panels/prompt/PromptThreadCleanup.manager.ts`
- `AICode/src/extension/src/extension/panels/prompt/__test__/PromptThreadCleanup.manager.attachmentsCleanup.test.ts`
- `AICode/src/extension/src/extension/panels/core/openPanel.util.ts`
- `AICode/src/extension/src/extension/panels/core/__test__/openPanel.util.test.ts`
- `AICode/src/extension/src/messageBus/handlers/ui/uiGenericHandlers.ts`
- `AICode/src/extension/src/messageBus/handlers/ui/__test__/uiGenericHandlers.openExternalUrl.test.ts`
- `AICode/src/extension/src/extension/commands/NewChat.command.ts`
- `AICode/src/extension/src/extension/commands/__test__/NewChat.command.test.ts`
- `AICode/src/extension/src/extension/uriHandlers/registerDeepLinkUriHandler.ts`
- `AICode/src/extension/src/extension/uriHandlers/__test__/registerDeepLinkUriHandler.scope.test.ts`
- `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.listeners.manager.ts`
- `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.mutations.manager.ts`
- `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.core.manager.ts`
- `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.rehydrate.manager.ts`

- `AICode/src/extension/src/extension/panels/sidebar/__test__/SidebarStore.rehydrate.manager.test.ts`
- `AICode/src/extension/src/extension/panels/sidebar/__test__/SidebarStore.manager.queryRunningUpsert.test.ts`
- `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.chatTitles.manager.ts`
- `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.handlers.manager.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/delete.endpoint.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/list.endpoint.windowsLock.eperm.test.ts`
- `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/diskAliases.lastWins.endpoints.test.ts`
- `AICode/src/extension/src/aiTools/orch/context/persistence/__test__/ChatPersistenceWriter.windowsFallback.test.ts`
- `AICode/src/extension/src/aiTools/orch/context/memory/core/mem/__test__/OrchMemory.naive_title.user_prompt_specif_multiline.e2e.repro.test.ts`
- `AICode/src/frontend/src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx`
- `AICode/src/frontend/src/app/sidebar/sidebar-activity/__test__/SidebarActivityQuery.test.tsx`
- `AICode/src/frontend/src/app/prompt/liveSession/core/PromptLiveSession.types.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.tsx`
- `AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.module.css`
- `AICode/src/frontend/src/app/prompt/liveSession/components/__test__/PromptLiveSessionBanner.test.tsx`
- `AICode/src/frontend/src/app/prompt/liveSession/core/usePromptLiveSession.hook.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.hydratation.test.tsx`

- `AICode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.handoff.test.tsx`
- `AICode/src/frontend/src/app/prompt/liveSession/adapters/EasyAgentsBatchLiveSession.adapter.ts`
- `AICode/src/frontend/src/app/prompt/liveSession/adapters/__test__/EasyAgentsBatchLiveSession.adapter.test.ts`
- `AICode/src/frontend/src/messageBus/AiStreamStartArgs.type.ts`
- `AICode/src/frontend/src/messageBus/useAiStream.hook.ts`
- `AICode/src/frontend/src/messageBus/__test__/useAiStream.hook.liveSessionObserver.test.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.hook.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunController.liveSessionObserver.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/usePromptCancelController.hook.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptCancelController.stopCore.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/preparePromptStreamingTurn.util.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/preparePromptStreamingTurn.util.test.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/usePromptThinkingLifecycle.hook.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/usePromptToolUse.hook.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.finalization.hook.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/usePromptRunCommands.hook.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.compactContextNow.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.retryQuery.attachmentOnly.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.retryQuery.compactContextNow.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/Prompt.tsx`

- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.liveSessionBootstrap.test.tsx`
- `AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.tsx`
- `AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.module.css`
- `AICode/src/frontend/src/app/prompt/PromptInput/__test__/PromptInput.batchLiveViewerMode.test.tsx`
- `AICode/src/frontend/src/app/easy-agents/easy-agents-batch/core/__test__/EasyAgentsBatch.liveWatch.test.tsx`
- `AICode/src/frontend/src/app/easy-agents/easy-agents-item/core/batch/__test__/EasyAgentBatchSurface.test.tsx`
- `AICode/src/frontend/src/app/prompt/hydration/isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-readme.md`
- `AICode/src/frontend/src/app/prompt/thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md`
- `AICode/src/frontend/src/app/prompt/Prompt/applyHydratedPromptSessionState.util.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/applyHydratedPromptSessionState.util.test.ts`
- `AICode/src/frontend/src/app/prompt/Prompt/usePromptSessionHydration.hook.ts`
- `AICode/src/frontend/src/app/prompt/hydration/frontSessionHydration.type.ts`
- `AICode/src/frontend/src/app/prompt/lib/usePromptSessionLocator.hooks.ts`
- `AICode/src/frontend/src/app/prompt/lib/__test__/usePromptSessionLocator.hook.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.autoRunPrompt.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.noThumbTurnIdDrift.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.restoreAttachments.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.ensureNextTurnId.stability.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearAttachments.test.tsx`

- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearInput.test.tsx`
- `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.promptDraft.hydration.test.tsx`
- `AICode/src/frontend/src/messageBus/prompt-stream-dispatch-lock.programming-doc-readme.md`
- `AICode/src/extension/src/aiTools/orch/ask/stream/core/fallback-persist-and-title.programming-doc-readme.md`
- `AICode/src/extension/src/messageBus/core/extTypedMessageBus.util.ts`
- `AICode/src/extension/src/messageBus/core/extMessageBus.util.ts`

Moved

None.

Deleted

None.

Regression risk factor

1. **Calculated risk: 92%.** Severity 5 , coupling 5 , blast radius 5 , uncertainty 4 , low confidence 3 .
Base = $10 \times 5 + 3 \times 5 + 2 \times 5 + 2 \times 4 + 3 \times 3 = 92$; risk = $\text{round}(92 \times 5/5) = 92\%$.
2. **Complex zones:** transactional start, exact origin webview, successive sources, reservations/epochs, Queue A/B, stale callbacks, terminal then transportError, post-header Slimbk envelope, first-header V1 protocol, persistence of every meta.error before appending, post-forwarding finalization, retirements/tombstones, destructive claims, wake-ups, continuation/thinking, raw query parsing, React boundary, preserveLocal with no draft reading, single Activity transition, shared stopRequested, per-attempt sound, delete/Retry under claim, and unmasking the composer with no fallback ack.
3. **Justification:** the slice modifies the normal path of every Prompt request and several asynchronous coordinations. The fixes added by this reinspection close four particularly dangerous races: an old stream message can no longer start on a new panel with the same locator; a post-header Slimbk exception can no longer be mistaken for an `AiStreamedChunk` ; a `meta.error` only becomes handoffable after exact persistence, including the auto-accept case; and an accepted attempt that finalizes before `fallback_persisted` no longer leaves the composer masked indefinitely. Exact identities, finalizations, claims, wake-ups, and isomorphism tests remain essential to avoid loss, duplication, resurrection, or concurrent cleanup.

Summary

- Extend `liveSession` to normal Prompt sources without killing the source on close.
- Add an exact backend sourceId, a frontend streamInstanceId, and an Open-only replayOpenAttemptId.
- Make start synchronous/transactional and all commands mutation-free on refusal.
- Verify the exact origin webview, not just the locator, before any reservation.
- Reserve before await, buffer before forwarding, finalize only after the extension FIFO and frontend Queue A/B.
- Persist every meta.error before appending; encode only the first meta.error in the V1 header.
- Filter post-header Slimbk error envelopes before they contaminate the chunk pipeline.
- Keep a single terminal data when a transportError follows.
- Finalize into handoff only with a baseline + durable terminal; remove otherwise.
- Separate retirements, tombstones, run epoch, destructive claim, and destructive generation.
- Serialize Activity/cleanup by locator; Idle if the panel is open, removal if closed.
- Propagate stopRequested to Activity so the detached Stop stays disabled after closing.
- Reconnect Prompt with an interactive composer, locked Send, a single Stop, and a Retry banner with no Stop.
- Direct handoff with no second Open; preserveLocal retries with no reading of draft fields.
- Unmask with no erasing of the composer if an accepted attempt finalizes with no fallback ack.
- Preserve EasyAgents, timers, sound, queues, locks, sanitizers, and generated files.

Commit name

`feat(live-session): keep prompt runs alive across panel close and reopen`