

Permalink URI:

vscode://joelabenhaim.aicode-ja/chat?sessionId=709dde10b25c4c4396ad73608f964246-RS-Affiner-finalisat

<USER>

[Refine][SPECIF FINALISER LIVESESSION POUR LES SESSIONS PROMPT NORMALES 20260712 175152](#)

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧  scanning git: 🌀 ["État Git courant"](#), 🌀 ["Commit HEAD courant"](#), 🌀 ["Base commune tranche"](#), 🌀 ["Diff base fonctionnelle"](#)

🔧  reading project: 📁 t ["Architecture liveSession"](#), 📁 t ["Frontend liveSession"](#), 📁 t ["Coordination Prompt"](#), 📁 g ["Docs programmation liées"](#), 📄 [liveSession.programming-doc-readme.md](#), 📄 [easyAgents.programming-doc-readme.md](#), 📄 [isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-readme.md](#), 📄 [thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md](#), 📄 [prompt-stream-dispatch-lock.programming-doc-readme.md](#), 📄 [fallback-persist-and-title.programming-doc-readme.md](#), 🌀 e ["Usages LiveSessionRunRef"](#), 🌀 e ["Usages streamKey"](#), 🌀 e ["État fenêtres activité"](#), 🌀 e ["Hydrateur Prompt"](#), 🌀 e ["Son démarrage stream"](#), 🌀 h ["Registre liveSession"](#), 🌀 h ["Flux Prompt source"](#), 🌀 h ["Bootstrap live frontend"](#),

📄 [LiveSessionRun.type.ts](#),
📄 [LiveSessionPanel.type.ts](#),
📄 [LiveSessionOpenResult.type.ts](#),
📄 [LiveSessionStream.util.ts](#),
📄 [SidebarStore.ts](#),
📄 [SidebarUiMessages.ts](#),
📄 [MessageRegistry.ts](#),
📄 [LiveSessionRecord.type.ts](#),
📄 [LiveSessionService.ts](#),
📄 [LiveSessionReplay.service.ts](#),
📄 [LiveSessionRegistry.service.ts](#),
📄 [LiveSessionOpen.service.ts](#),
📄 [EasyAgentsBatchLiveSessionAdapter.service.ts](#),
📄 [EasyAgentsBatchRunChildExecution.service.ts](#),
📄 [EasyAgentsBatchRunRetry.service.ts](#),
📄 [EasyAgentsBatchRunManager.service.ts](#),
📄 [EasyAgentsBatchViewerPresence.service.ts](#), 📄 [EasyAgentsBatchLastViewerStop.service.ts](#), 🌀 e ["Observer sink liveSession"](#), 🌀 e ["liveTurnId runRef"](#), 🌀 h ["Suppression panneau Prompt"](#), 🌀 h ["Ouverture panneau Prompt"](#), 🌀 h ["Suppression chat"](#), 📄 [EasyAgentsBatchRunExecution.service.ts](#) [L280-L430], 📄 [EasyAgentsBatchRunManager.types.ts](#) [L150-L240],

📄 [PromptThreadCleanup.manager.ts](#),
📄 [OpenedPanelsRegistry.util.ts](#),
📄 [closePromptPanel.util.ts](#),
📄 [delete.endpoint.ts](#),
📄 [openPanel.util.ts](#),
📄 [iaStreamHandlers.ts](#),
📄 [liveSessionStreamHandlers.ts](#),
📄 [stream.endpoint.ts](#),
📄 [open.endpoint.ts](#),
📄 [stream.endpoint.ts](#),
📄 [abort.endpoint.ts](#), 📄 [SidebarQueryRunningBridge.ts](#), 📄 [PromptQueryDoneSound.manager.ts](#), 🌀 e ["Type SlimbkCallError"](#), 🌀 e ["Callback onError stream"](#), 🌀 h ["Client stream Slimbk"](#), 🌀 e ["errorCode Slimbk"](#),
🌀 e ["patchLastAssistantMeta"](#),
📄 [aiStopSuffix.util.ts](#), 📄 [OrchMessagesMemoryAssistantPatch.ts](#), 📄 [slimbkBuildHeader.ts](#) [L1-L120], 📄 [slimbkBuildHeader.ts](#) [L217-L430], 📄 [SlimbkResult.type.ts](#), 📄 [Slimbk.ts](#) [L110-L180], 📄 [Slimbk.ts](#) [L220-L360], 📄 [callLocalApi.generated.ts](#) [L40-

L110],

[useAiStream.hook.ts](#),
[AiStreamStartArgs.type.ts](#),
[useStreamListener.hook.ts](#),
[useAiStreamQueue.hook.ts](#),
[usePromptStreamDispatchGateQueue.hook.ts](#),
[frontMessageBus.util.ts](#),
[usePromptRunController.hook.ts](#),
[usePromptCancelController.hook.ts](#),
[preparePromptStreamingTurn.util.ts](#),
[usePromptRunController.finalization.hook.ts](#), [usePromptThinkingLifecycle.hook.ts](#), [usePromptToolUse.hook.ts](#),
[usePromptRunCommands.hook.ts](#),
[Prompt.tsx](#),
[PromptLiveSession.types.ts](#),
[usePromptLiveSession.hook.ts](#),
[EasyAgentsBatchLiveSession.adapter.ts](#),
[PromptLiveSessionBanner.tsx](#),
[PromptInput.tsx](#),
[applyHydratedPromptSessionState.util.ts](#),
[usePromptSessionHydration.hook.ts](#),
[usePromptSessionLocator.hook.ts](#),
[SidebarStore.queryWindow.listeners.manager.ts](#),
[SidebarStore.queryWindow.mutations.manager.ts](#),
[SidebarStore.rehydrate.manager.ts](#),
[SidebarStore.core.manager.ts](#),
[SidebarStore.handlers.manager.ts](#),
[SidebarActivityQuery.tsx](#), [SidebarStore.chatTitles.manager.ts](#), [VscButton.tsx](#), ["disableAfterClick usage"](#), ["SquareStop icon usage"](#),
[GenericUiMessages.ts](#),
[uiGenericHandlers.ts](#),
["openCentralPanel"](#),
["closePromptPanel"](#),
["cleanupOnPromptDispose"](#),
["sidebarQueryRunningBridge"](#),
["applyHydratedPromptSessionState"](#),
["usePromptSessionLocator"](#), [NewChat.command.ts](#), [registerDeepLinkUriHandler.ts](#) [L100-L170], [openPromptInNewSession.util.ts](#), ["Open panel bus Prompt"](#), [loadPanelFromHtml.util.ts](#), [useQueryString.hook.ts](#), ["Routeur query globale"](#), ["autoRun paramètres"](#), ["Snapshot messages Orch"](#), ["getThreadMessagesSnapshot"](#), [OrchAsk.ts](#) [L1-L180], [OrchMemory.ts](#) [L600-L720], [OrchMemory.ts](#) [L430-L490], ["Getter memory Orch stream"](#), ["readonly memory Orch stream"](#), [OrchAskStream.ts](#) [L1-L140], ["onStreamed extension"](#), ["stream done channel"](#),
[extTypedMessageBus.util.ts](#),
[extMessageBus.util.ts](#),
[AiStreamedChunk.type.ts](#), [AiStopErrorReason.type.ts](#), [AiStopReason.type.ts](#), ["SYSTEM_ERROR chunk creation"](#), ["INTERRUPTED chunk creation"](#), [OrchAskStreamTurnErrorHandler.ts](#) [L120-L230], [OrchAskStreamErrors.ts](#) [L430-L530], [OrchAskStreamTurnRuntime.ts](#) [L250-L360], [OrchAskStreamTurnRuntime.ts](#) [L820-L900], [OrchAskStreamTurnRuntime.ts](#) [L900-L936], ["AutoAccept terminal call"](#), [OrchAskStreamTurnRuntime.ts](#) [L730-L825], ["AutoAccept persistance échoue"](#), ["Tests autoAccept erreur"](#), ["autoAcceptFileEdits usages"](#), ["Appels endpoint ask stream"](#), ["Type activité fenêtre"](#), ["Champ windows activité"](#), ["queryState windows frontend"](#), ["Windows activité fixtures"](#), ["Fixtures windows vides"](#), ["Paramètres live constants"](#), ["Constante ligne Batch"](#), ["Politique fermeture live"](#), ["Mode liveSession query"](#),

 r ["useAiStream"](#),
 r ["AiStreamStartArgs"](#),
 r ["usePromptRunController"](#),
 r ["usePromptRunCommands"](#),
 r ["PromptLiveSessionViewModel"](#),  [PromptLiveSessionBanner.module.css](#),  [PromptInput.module.css](#) [L1-L180],  [PromptInput.module.css](#) [L180-L360],  [usePromptIndexPending.hook.tsx](#),  [usePromptRunController.uiState.hook.ts](#),  e ["Persistence promptDraft"](#),  h ["Écriture snapshot session"](#),  [ChatPersistenceWriter.service.ts](#),  [savePromptDraft.endpoint.ts](#),  h ["Erreur stream après header"](#),  e ["Stream ended before header tests"](#),  e ["Internal error stream initialized"](#),  e ["Son config message"](#),  e ["queryDoneSound message"](#),  [usePromptQueryDoneSoundSync.hook.ts](#),  e ["Mocks adaptateur liveSession"](#),  e ["Barrière clear fallback"](#),  e ["Littéraux runRef"](#),  r ["buildEasyAgentsBatchLiveSessionQuery"](#),  [frontSessionToPromptMessages.util.ts](#) [L1-L120]
  scanning git:  ["Écarts depuis b42c"](#),  ["Écarts depuis base"](#),   ["Ancêtres commits cités"](#)
  reading project:  r ["handleTypedMessage"](#),  r ["dispatchStreamedCall"](#),  e ["Handlers onStreamed"](#),  e ["Appels handleTypedMessage"](#),  e ["Import handleTypedMessage"](#),  [extTypedMessageBus.broadcastMultiPrompt.test.ts](#),  e ["Mocks contrôleur Prompt"](#),  e ["handleSend mocks Prompt"](#),  e ["handleSubmitEdit mocks"](#),  e ["startLiveObserver mocks"](#),  e ["Starts tests useAiStream"](#),  e ["Assertions stream args"](#),  e ["Docs queryState windows"](#),  [AiQuery.type.ts](#),  e ["Tests fallback ack hook"](#),  [FrontChatSession.type.ts](#),  e ["Mocks module locator Prompt"](#)
  specifying

"SPECIF_FINALISER_LIVESESSION_POUR_LES_SESSIONS_PROMPT_NORMALES_20260712_204731"
 SPECIF_FINALISER_LIVESESSION_POUR_LES_SESSIONS_PROMPT_NORMALES_20260712_204731

Copy

Refine

Code

Verify

Markdown formattedMarkdown source

Spécification — Finaliser `liveSession` pour les sessions Prompt normales

Contexte

La tranche 1 correspond au sous-système fonctionnel introduit au commit `f65adc8e5172cc1e4f9781bf54ffa6dd089d3d33`. La présente réinspection a été refaite depuis zéro sur le dépôt propre courant au commit `c3d66083b46b178e24d4ce497cc3cd5d914a4170` (`c3d66083`). Les commits `b42c734c4178d0725ce98933c085a763288eb697`, `613acafc9ac510e432986f56456d63f185d77347`, `b16c5bdde3b4a27cec2ee3f44c9a471b44465ac6` et `0b9f52cc402ce6d0ae19372fa62a432851c248b0` ne sont pas les ancêtres directs du HEAD courant ; leur base commune pertinente avec le HEAD est `ed3bd94687cc34f6603730689bfc775818881bae`. Le diff entre `f65adc8e5172cc1e4f9781bf54ffa6dd089d3d33` et `ed3bd94687cc34f6603730689bfc775818881bae` est vide. Les écarts entre `b42c734c` et le HEAD concernent l'outillage de spécification ; les écarts supplémentaires depuis `ed3bd946` concernent ce même outillage et une documentation personnelle hors périmètre. Aucun arbre source ciblé par cette tranche n'a changé.

La fondation actuelle fournit déjà :

- un registre process-local de runs exacts ;
- un buffer ordonné complet de `AiStreamedChunk` ;
- des leases de viewer exactes ;
- les barrières `sourceSettled` et `handoffReady` ;
- l'ouverture live-safe sans reset de mémoire Orch ;
- le replay backlog + tail ;
- une protection anti-ABA par époque de locator ;
- le terminal exact et la distinction `TOOL_USE` / vraie fin de tour ;
- les transports `Slimbk` et message bus génériques ;
- un hook frontend générique réutilisant Queue A, Queue B, les locks et les sanitizers Prompt ;
- un adaptateur `EasyAgents` préservant le comportement produit historique.

Le problème utilisateur est qu'un renderer Prompt peut devenir gris, notamment après saturation mémoire Electron. Aujourd'hui, fermer le panneau exécute explicitement `Abort` → `AwaitStop` → `ClearMemory`, détruit la génération et oblige à relancer le dernier message. La tranche finale doit permettre à une génération Prompt normale de continuer dans l'extension, d'être visible dans Activité, d'être rouverte sur sa source exacte et de reconstruire l'historique puis le stream sans perte ni duplication.

La tranche est atomique : capture de source, fermeture, projection Activité, réouverture, replay, Stop, son, finalisation frontend, persistance terminale, nettoyage et protections destructives doivent être livrés ensemble. Aucun état intermédiaire incomplet ne doit être introduit.

Les invariants de conservation restent obligatoires :

- aucun changement d'algorithme dans Queue A, Queue B, `PromptStreamLockStore` ou les sanitizers ; seules leurs entrées, sorties et mutations sont gardées par l'identité de tentative ;
- aucun doublon de timer Activité ni de son lors d'un replay ;
- aucun second bouton Stop dans Prompt ;
- aucun panneau « invisible » : un panneau est `open` ou `closed` ;
- aucun `*.generated.*` modifié manuellement ;
- aucun fichier de documentation personnelle interdit chargé ou modifié ;
- code, tests, commentaires et identifiants ajoutés ou modifiés en anglais ;
- aucun `any`, `eslint-disable`, `TODO`, cast inutile, import dynamique de production ou compatibilité implicite non demandée.

Objectifs décrits par l'utilisateur

Besoin initial exprimé par l'utilisateur

- Livrer toute la tranche finale en une seule implémentation cohérente.
- Permettre à une génération Prompt normale de continuer lorsque son panneau est fermé.
- Conserver l'item dans Activité lorsque le panneau est ouvert, ou fermé avec une source encore active.
- Supprimer l'item d'Activité lorsque le panneau est fermé et la source terminée.
- Conserver l'item Idle si le panneau est ouvert lorsque la source se termine, y compris si le record `liveSession` est retiré faute de baseline ou de vrai terminal.
- Ajouter une troisième ligne Activité avec un bouton Stop uniquement pour `Running/source/closed`.
- Utiliser l'icône Stop existante, `size="small"`, `variant="danger"`, label Stop, avec désactivation immédiate après clic.
- Faire disparaître la ligne Stop détachée dès réouverture du panneau ou fin de la source.
- Réutiliser dans le panneau reconnecté le Stop normal du compositeur.
- Restaurer Prompt normal au plus tôt : après installation de la baseline et armement du replay, autoriser la saisie du prochain brouillon et la préparation des attachments, tout en gardant Send verrouillé.
- Étendre SidebarStore pour représenter des activités de session et non seulement des fenêtres ouvertes.
- Conserver le buffer raw-live complet, sans compression ni limite ajoutée.
- Tester tout code créé, déplacé ou modifié, notamment les courses fermeture/réouverture/Stop/retry transport/sources successives/delete/Retry EasyAgents.
- Mettre à jour les programming docs génériques `liveSession`, EasyAgents, isomorphisme, thinking, Queue B et fallback persist.
- Respecter l'architecture modulaire, le typage strict et toutes les règles de qualité du projet.

Clarifications implicites ajoutées par AICode

- `LiveSessionRunRef` devient `{ locator, turnId, sourceId }`. `turnId` reste la corrélation logique persistante ; `sourceId` identifie une invocation backend précise.
- `streamInstanceId` est frontend-only, change pour chaque stream effectivement accepté et remplace `streamKey`. `replayOpenAttemptId` est également frontend-only, change pour chaque séquence Open initiale ou Retry, et n'est jamais fabriqué pour représenter un stream qui n'a pas démarré.
- Le descripteur live privé comprend exactement `openMode`, `liveSessionTurnId`, `liveSessionViewerLeaseId`, `liveSessionClosePolicy`, `liveSessionSourceId`, `liveSessionAdapter` et `easyAgentsBatchLineNumber`. Toute clé inconnue préfixée `liveSession`, toute présence partielle et tout doublon critique rendent la query invalide.
- Le contrat shared expose une union stricte de descripteurs complets : `easyAgentsBatch + stopOnLastViewerClose + easyAgentsBatchLineNumber + sourceId===turnId`, ou `normalPrompt + continueOnPanelClose` sans numéro de ligne. Extension et frontend utilisent cette même union après détection des doublons.
- Une query live explicite valide conserve toujours priorité. Une query live explicite invalide est rejetée avant le low-level opener et ne se dégrade jamais en ouverture classique. La validation frontend reste une défense en profondeur.

- `Prompt.tsx` est l'unique propriétaire du snapshot brut figé de query. Le dispatcher est pur, utilise `URLSearchParams.getAll(...)`, ne lit jamais la globale router et retourne la classification ainsi qu'un locator strict lorsqu'il est univoque.
- Une frontière React externe traite les queries invalides avant de monter le runtime Prompt. Aucun locator sentinelle ni hook conditionnel n'est autorisé.
- Toute query live explicite, valide ou invalide, désactive l'hydratation classique et tout auto-run. L'injection normalPrompt retire également tous les paramètres `autoRun*`.
- Le contexte d'origine d'un stream est une donnée extension-only non sérialisée : le message bus transmet au handler le `HtmlPage` et la webview exacte ayant émis la demande. Le guard de démarrage compare cette webview à celle actuellement enregistrée pour le locator. Un simple test par locator est insuffisant, car un ancien message en file ne doit jamais démarrer sur un nouveau panneau rouvert avec le même locator.
- `useAiStream.start()` retourne synchroniquement `{ accepted: false } | { accepted: true; runRef; streamInstanceId }`. Aucun buffer, lock, queue, scaffold, placeholder, `uiPending` ou ref de continuation n'est muté sur refus.
- `Send`, `SPECIF`, `Retry`, `Edit`, `Compact context`, `Reply` et `Continue` appliquent tous une préparation pure, appellent `start()`, puis committent exactement un scaffold uniquement après acceptation.
- Les barrières Prompt de clear/focus sont corrélées à `{ turnId, streamInstanceId }`. Si `fallback_persisted` arrive pour cette tentative, le compositeur est effacé comme aujourd'hui. Si la tentative se finalise sans cet ack, le masque transactionnel est retiré mais le texte, les attachments, `nextTurnId`, `pendingEditTurnId` et le mode maximisé restent intacts.
- Les callbacks `onData`, `onComplete`, `onError`, leurs continuations après `await` et le `runMerged` déjà sorti de Queue A sont gardés par `sourceId + streamInstanceId`.
- `completed` ou `transportError` n'est publié, et le verrou d'acceptation n'est libéré, qu'après Queue A idle, Queue B idle et la dernière barrière de commit.
- Si un vrai terminal data a déjà été consommé puis qu'une erreur transport/auxiliaire arrive, le terminal reste unique ; aucun second `meta.error` ni suffixe n'est injecté et l'outcome devient `transportError`.
- Le handoff direct normalPrompt exige terminal data exact, transport normalement complété, Queue A/Queue B drainées et `finalizedStreamInstanceId` correspondant. Il passe directement à normal sans second Open ni seconde hydratation du compositeur.
- Toute réponse Open est gardée par `replayOpenAttemptId`. Une branche `terminal` n'alloue aucun `streamInstanceId`.
- Les retries normalPrompt utilisent une hydratation `preserveLocal` qui ne lit, ne valide, ne normalise ni n'enregistre `promptDraft`, `promptDraftAttachments` ou `meta.nextTurnId`. Les tests utilisent aussi des getters piégés pour prouver l'absence réelle de lecture.
- Le wrapper normalPrompt persiste idempotemment tout `meta.error` reçu depuis Orch avant de l'appendre au registre ou de l'exposer. Cette barrière couvre notamment le chemin auto-accept où Orch peut avoir construit un terminal alors que sa première tentative de patch durable a échoué.
- Tout terminal wrapper synthétique après throw/EOF est créé uniquement sans terminal antérieur, patché durablement avec suffixe canonique, puis appendu. Si la persistance échoue, aucun terminal n'est appendu et le transport échoue.
- Le protocole privé V1 ne concerne que le premier chunk source `meta.error`. Après un header `200_OK`, un `meta.error` reste un body chunk normal.
- Le framework Slimbk actuel sérialise une exception survenue après le header `200_OK` sous forme d'un `SlimbkResult` non-200 dans le body. Le client généré le transmet actuellement au callback `onChunk`. Les handlers extension doivent reconnaître cette enveloppe avant toute logique `AiStreamedChunk`, la transformer en erreur transport et ne jamais la bufferiser, l'émettre ou tenter de la décoder comme protocole source V1.
- Une réservation source revendique le pointeur actif et avance l'époque avant tout `await`. Sa promotion conserve la même époque. Annulation, handoff ou retraite retirent le pointeur, avancent l'époque puis mémorisent l'issue avec l'époque post-retrait.
- Une réservation exacte dupliquée est refusée. Le handler qui n'a pas obtenu la réservation ne peut ni appeler l'endpoint source, ni annuler, promouvoir ou finaliser la réservation du premier handler.
- La finalisation post-forwarding est réservée à `ownership= 'source'` et retourne exactement `reservationCancelled | handoffReady | retiredMissingBaseline | retiredMissingTerminal | stale`.
- Un record sans baseline ou sans vrai terminal est retiré exactement ; aucun record `sourceSettled` non handoffable ne reste bloqué. Si les deux manquent, `retiredMissingBaseline` a priorité.

- Les retraites structurées, utilisées pour fallback/idempotence, restent distinctes des tombstones exactes process-lifetime empêchant la résurrection d'un run key observé.
- Le registre émet après mutation complète deux wake-ups profondément immuables : observer devenu inactif et claim destructif libéré. Les listeners sont isolés et ces événements ne portent aucune décision métier.
- Les suppressions durables et Retry EasyAgents utilisent un claim destructif opaque de locator. Acquisition et libération avancent une génération destructive distincte de l'époque de run. L'éviction publique exige le handle exact et ne libère jamais le claim.
- Un dispose sous claim conserve le mapping détaché et diffère `Abort/Await/ClearMemory`, orphan cleanup, son et cleanup trivial jusqu'au wake-up de release, puis revalidation.
- L'acceptation d'un Stop source met aussi synchroniquement à jour `stopRequested` dans le runtime Activité avant l'abort. Ainsi, si le panneau se ferme juste après le Stop du compositeur, le Stop détaché apparaît déjà désactivé.
- EasyAgents conserve `sourceId=turnId`, son elapsed viewer historique, `last viewer → kill`, son settlement/handoff métier et son UX read-only.

Vue d'ensemble de la méthode choisie

Identité et démarrage transactionnel frontend

1. Chaque commande construit un plan pur sans mutation visible.
2. `useAiStream.start()` acquiert synchroniquement le single-flight ; sur refus il retourne immédiatement sans mutation.
3. Sur acceptation, il génère `streamInstanceId` et, pour une source Prompt, `sourceId`, construit le runRef exact et retourne l'identité.
4. Le caller committe alors le scaffold, les refs de thinking/assistant, `uiPending`, le scroll et les barrières locales.
5. L'effet transport démarre seulement au rendu suivant ; les barrières sont donc armées dans la même pile avant tout callback réseau.
6. Les handlers extension reçoivent un contexte d'origine exact non sérialisé. Ils vérifient que la webview émettrice est encore celle enregistrée pour le locator avant la première mutation.

Cycle backend d'une source Prompt normale

1. Guard exact de panneau par webview d'origine.
2. Réservation source exacte comme première mutation synchrone ; revendication du pointeur actif, bump d'époque et capture de `sourceStartedAtMs`.
3. Allocation synchrone du runtime/timer Activité et de `activityRunId`; la publication est démarrée séparément et reste auxiliaire.
4. Armement du son exact et récupération de `soundAttemptId`.
5. Persistance éventuelle de la préférence SPECIF Code, chargement catalogue, puis capture sans `await` de la baseline de continuation immédiatement avant `askStream(...)`.
6. Priming lazy du `Readable`, promotion réservation → record par `registerRun`, application d'un abort précoce.
7. `continueAssistant` publie immédiatement sa baseline de messages ; `replaceTurnAssistants` attend `fallback_persisted`.
8. Chaque chunk ordinaire est parsé, éventuellement persisté s'il s'agit de `meta.error`, appendu au registre puis exposé.
9. Un throw/EOF sans terminal tente de persister un terminal canonique avant append ; un throw après terminal reste transport.
10. Le service source ne finalise jamais le registre.
11. L'endpoint encode uniquement un premier `meta.error` dans le header V1. Après `200_OK`, les chunks restent dans le body ; une exception post-header devient une enveloppe Slimbk d'erreur dans le body.
12. Le handler extension maintient une FIFO unique pour les chunks ordinaires et le terminal décodé depuis le premier header. Une enveloppe Slimbk d'erreur post-header est interceptée avant la FIFO et classée transport.
13. Les erreurs auxiliaires sont mémorisées dans l'ordre sans empêcher les étapes restantes du chunk ni les chunks suivants.
14. Après retour Slimbk, drainage FIFO, finalisation atomique du registre, arrêt du son exact, puis délégation au coordinateur pour Activité/cleanup.
15. Le frontend publie l'outcome seulement après Queue A, Queue B et commit ; le contrôleur Prompt publie ensuite `finalizedStreamInstanceId` après sa propre finalisation UI.

Replay et handoff

- `replaceTurnAssistants` recharge l'enveloppe persistée récente et retire uniquement les assistants du tour exact.
- `continueAssistant` superpose la baseline de messages immuable sur l'enveloppe persistée la plus récente.
- `not_ready` normalPrompt est retryé avec une seule requête Open en vol et un délai annulable.

- `not_found`, `transportError`, completion sans terminal ou éviction sans terminal conduisent à `liveSessionReplayFailed` pour `normalPrompt`.
- Le handoff `normalPrompt` est direct sur l'état replayé ; `EasyAgents` et les bootstraps déjà terminaux utilisent le snapshot terminal.
- Un retry `normalPrompt` remplace messages et stores d'hydratation mais conserve intégralement le compositeur local, y compris si `Open` répond directement `terminal`.

Registre, retraites, tombstones et claims

Le registre distingue :

- réservation source active ;
- record actif ;
- retraite structurée avec issue + époque post-retrait ;
- tombstone exacte process-lifetime ;
- claim destructif opaque de locator ;
- époque de run et génération destructive indépendantes.

Les `Open` capturent et revalident les deux générations. Un cycle `acquire/release` invalide uniquement l'`Open` en vol. Une nouvelle source rend toute répétition tardive de finalisation ancienne `stale`. L'éviction sous claim retire réservations, records, leases et retraites replayables, conserve les tombstones et ne relâche pas le claim.

Panneaux, Activité, son et Stop

- Le public `openCentralPanel(...)` devient `async`. Il délègue `Prompt` au coordinateur et garde le low-level `opener` privé pour les autres pages.
- Le coordinateur est une classe injectée qui n'importe jamais le low-level `opener`.
- `Open`, `dispose`, `sourceSettled` et `wake-ups` sont sérialisés par locator dans des queues supprimées à `idle`.
- Le low-level `dispose` supprime synchroniquement le panneau du registre avant d'appeler le coordinateur.
- Le coordinateur est l'unique propriétaire de la transition terminale `Activité` d'une source normale.
- `SidebarActivityQueryItem` est une union stricte : `Idle/open` ; `Running/observer/open` ; `Running/source/open|closed`.
- Le runtime `Activité` garde `runRef`, `ownership`, `runId`, `timestamps`, `timer`, `usage` et `stopRequested`; les transitions sont exactes par `runRef + runId`.
- Le `Stop` détaché existe uniquement pour `Running/source/closed`. Le sous-arbre est keyé par `sourceId`.
- Le son utilise `runRef` exact pour la déduplication et `soundAttemptId` pour isoler les tentatives `viewer` successives.

Suppression et Retry destructifs

1. Préchecks UI/Git historiques.
2. Acquisition synchrone du claim destructif juste avant fermeture.
3. Fermeture du panneau ; le `dispose` constate le claim et diffère tout `cleanup` concurrent.
4. Suppression durable : éviction seulement après succès disque.
5. Retry `EasyAgents` : éviction pré-delete historique sous le même claim.
6. Libération en `finally`, bump de génération et `wake-up` du coordinateur.

Liste exacte des fichiers avec détails d'implémentation

Contrats shared

1) `AIcode/src/sharedlib/src/types/liveSessionTypes/LiveSessionReplay.type.ts` — créé

- Définir `ZLiveSessionReplayMode / LiveSessionReplayMode : replaceTurnAssistants | continueAssistant`.
- Documenter la reconstruction frontend de chaque mode.
- 2) `AIcode/src/sharedlib/src/types/liveSessionTypes/LiveSessionRun.type.ts` — modifié
- Ajouter `LiveSessionSourceId` strict trim/non vide et `sourceId` obligatoire au `runRef`.
- Inclure `sourceId` dans la `run` key, sans modifier la locator key.
- Documenter la différence `turnId` logique / `sourceId` exécutionnel.
- 3) `AIcode/src/sharedlib/src/types/liveSessionTypes/LiveSessionPanel.type.ts` — modifié
- Ajouter `LIVE_SESSION_SOURCE_ID_PARAM`, `LIVE_SESSION_ADAPTER_PARAM` et centraliser `EASY_AGENTS_BATCH_LINE_NUMBER_PARAM`.
- Définir `ZLiveSessionAdapter : easyAgentsBatch | normalPrompt`.
- Définir la liste canonique `readonly` des paramètres privés.
- Définir une union stricte `ZLiveSessionPanelDescriptor : EasyAgents exige stopOnLastViewerClose`, ligne positive et `sourceId===turnId`; `normalPrompt` exige `continueOnPanelClose` et interdit la ligne `EasyAgents`.
- Réutiliser les schémas `runRef/sourceId/lease` ; aucun parseur extension/frontend ne redéfinit ces couples.
- 4) `AIcode/src/sharedlib/src/types/liveSessionTypes/LiveSessionOpenResult.type.ts` — modifié
- Ajouter `sourceId` aux branches `live/terminal` et `replayMode` uniquement à `live`.

- Étendre le transport et `superRefine`, y compris les propriétés présentes à `undefined`.
- Conserver les schémas canoniques complets sans rendre `ZChatSessionFile` `deep-strict`.
- Ajouter `ZLiveSessionOpenResultEnvelope` avec `session`: `unknown`, strict sur discriminant et identité, réservé au parse frontend policy-aware.

5) `AICode/src/sharedlib/src/types/liveSessionTypes/__test__/LiveSessionContracts.test.ts` — **modifié**

- Couvrir `run keys`, `sourceId`, `replay modes`, paramètres, liste privée, adaptateurs, couples complets, égalité `EasyAgents` `source/turn`, ligne requise/interdite et propriétés `undefined`.
- Comparer schéma `Open` complet et enveloppe sélective, sans resserrement historique accidentel.

6) `AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts` — **modifié**

- Ajouter fabriques canoniques `SYSTEM_ERROR/INTERRUPTED`.
- Définir encodeur/décodeur `V1` exact : `500_SERVER_ERROR`, code/préfixe `AICODE_LIVE_SESSION_SOURCE_ERROR_V1`, payload strict `{ version: 1, value }`.
- Le décodeur accepte chaîne, `Error` et objet structuré ; il respecte strictement `status` et la présence propre de `errorCode`.
- Extraire une ligne utile de `stack` et reconstruire exactement `AiErrorMetaValue`.

7) `AICode/src/sharedlib/src/types/aiTypes/__test__/aiStopSuffix.util.test.ts` — **modifié**

- Tester réversibilité, statuts/codes, propriété présente à `undefined`, `stacks`, détails multilignes, extras invalides et `chunks` canoniques.

Persistance terminale Orch

8) `AICode/src/extension/src/aiTools/orch/context/memory/core/msg/OrchMessagesMemoryAssistantPatch.ts` — **modifié**

- Ajouter le suffixe canonique idempotent pour `SYSTEM_ERROR` comme pour `INTERRUPTED`.
- Préserver blocs, métadonnées, usages et `sink` ; ne jamais dupliquer un suffixe existant.

9) `AICode/src/extension/src/aiTools/orch/context/memory/core/msg/__test__/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts` — **créé**

- Couvrir `SYSTEM_ERROR`, `INTERRUPTED`, texte absent/présent, suffixe existant, `sink` et hydratation isomorphe.

Contrats Sidebar

10) `AICode/src/sharedlib/src/types/busTypes/ui/SidebarStore.ts` — **modifié**

- Remplacer `SidebarActivityQueryWindowItem` par l'union stricte `SidebarActivityQueryItem`.
- Branches : `Idle/open` ; `Running/observer/open` ; `Running/source/open|closed`.
- Conserver titre, `elapsed`, usage, `updatedAt`, `openedAt` ; inclure `runRef` exact et `stopRequested` seulement pour source.
- Renommer `queryState.windows` en `queryState.items`.

11) `AICode/src/sharedlib/src/types/busTypes/ui/SidebarUiMessages.ts` — **modifié**

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

12) `AICode/src/sharedlib/src/types/busTypes/global/MessageRegistry.ts` — **modifié**

- Ajouter `sourceId` aux streams normal et `liveSession`.
- Ne jamais transporter `streamInstanceId`, `replayOpenAttemptId` ni le contexte `extension-only` de `webview`.

Backend Orch liveSession

13) `AICode/src/extension/src/aiTools/orch/liveSession/liveSession.programming-doc-readme.md` — **modifié**

- Documenter identité, réservations, époques, `replay modes`, protocoles premier-header et erreur post-header-body, barrière de persistance de tout `meta.error`, finalisation post-forwarding, retraites/tombstones, `claims`, `wake-ups`, `panel-origin` exact, `Activité`, son et identités frontend-only.

14) `AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRecord.type.ts` — **modifié**

- Ajouter `baselineOnly`, `replayMode`, `baseline messages`, `timestamp`, `abortRequested`, `ownership source/observer`, réservations, retraites et `tombstones`.
- Définir `LiveSessionPostForwardingFinalizationResult` et le handle opaque `LiveSessionLocatorDestructionClaim`.
- Ajouter génération destructive aux `permits Open`.
- Définir événements immuables `observer-inactif` et `claim-libéré`.
- Définir `snapshots actifs/réservés` sans exposer de `mutable`.

15) `AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRegistry.service.ts` — **modifié**

- Implémenter réservation source exacte, annulation, promotion atomique via `registerRun`, refus de duplication et application d'abort précoce.

- La réservation revendique le pointeur et bump l'époque ; la promotion ne rebump pas.
- Implémenter listings/snapshots, claims exacts d'abort source/viewer et suppression de `canAbortSource`.
- Implémenter claims destructifs, génération destructive, éviction exigeant le handle et blocage reserve/register/Open.
- Finaliser source après forwarding ; retirer tout record sans baseline/terminal, priorité missingBaseline, idempotence dans l'époque puis stale.
- Conserver tombstones après éviction et émettre les deux wake-ups après mutation complète avec listeners isolés.
- 16) `AICode/src/extension/src/aiTools/orch/liveSession/registry/__test__/LiveSessionRegistry.service.test.ts` — **modifié**
- Couvrir réservations, duplicate owner, promotion, abort, baseline modes/messages, finalisation cinq résultats, stale, retraites/tombstones, sources successives, claims, générations, événements, éviction sous handle et waiter réveillé sans terminal.
- 17) `AICode/src/extension/src/aiTools/orch/liveSession/core/LiveSessionService.ts` — **modifié**
- Exposer toutes les nouvelles opérations/snapshots/claims/souscriptions/finalisation.
- Supprimer `canAbortSource` et rendre l'éviction destructive accessible uniquement par claim exact.
- 18) `AICode/src/extension/src/aiTools/orch/liveSession/core/__test__/LiveSessionService.test.ts` — **modifié**
- Couvrir forwarding complet, réservations, sources successives, stale, missing terminal, claims, générations, signaux et tombstones.
- 19) `AICode/src/extension/src/aiTools/orch/liveSession/open/LiveSessionOpen.service.ts` — **modifié**
- Retourner `not_ready` pour réservation exacte et refuser sous claim.
- Revalider époque de run + génération destructive autour de toute lecture.
- Appliquer `replaceTurnAssistants` ou `continueAssistant` sur l'enveloppe récente.
- Contrôler fallback terminal par retraite + époque post-retrait et retourner `sourceId/replayMode`.
- 20) `AICode/src/extension/src/aiTools/orch/liveSession/open/__test__/LiveSessionOpen.service.test.ts` — **modifié**
- Couvrir réservations, baselines, continuation, `sourceId`, retraites, source successive, claim, Open en vol invalidé et record retiré sans terminal.
- 21) `AICode/src/extension/src/aiTools/orch/liveSession/replay/__test__/LiveSessionReplay.service.test.ts` — **modifié**
- Ajouter `sourceId`, isolation des subscriptions, attente de finalisation, éviction sans terminal et éviction publique sous claim.
- 22) `AICode/src/extension/src/aiTools/orch/liveSession/source/LiveSessionSourceLifecycle.service.ts` — **créé**
- Extraire le lifecycle lazy `Readable` OOP générique : priming, first next, callback start sync, consommation, return/destroy, cleanup et agrégation déterministe.
- Permettre au caller d'interdire l'abort thread-level sur échec post-return.
- 23) `AICode/src/extension/src/aiTools/orch/liveSession/source/__test__/LiveSessionSourceLifecycle.service.test.ts` — **créé**
- Couvrir priming, cleanup, settlement unique, erreurs combinées et mode sans abort tiers.
- 24) `AICode/src/extension/src/aiTools/orch/liveSession/source/NormalPromptLiveSessionSource.service.ts` — **créé**
- Mapper `AiQuery` → `replayMode` et `readiness baselineOnly`.
- Capturer la baseline continue sans `await` avant `askStream`, valider l'assistant, primer puis promouvoir la réservation.
- Publier baseline continue immédiatement ou baseline replace sur `fallback_persisted`.
- Avant tout append d'un `meta.error` Orch, appeler idempotement `patchLastAssistantMeta` avec `reason/message/details` et suffixe ; append/yield seulement après succès.
- Cette barrière doit corriger le cas où Orch a émis un terminal auto-accept malgré un premier patch durable échoué.
- Append-before-yield pour tous les chunks ; suivre le vrai terminal.
- Synthétiser `SYSTEM_ERROR/INTERRUPTED` uniquement sans terminal et après persistance exacte ; sinon propager le transport.
- Ne jamais finaliser le registre.
- 25) `AICode/src/extension/src/aiTools/orch/liveSession/source/__test__/NormalPromptLiveSessionSource.service.test.ts` — **créé**
- Couvrir tous les query types, continuation, baseline, ordre, abort, premier error/header, error tardive, incoming `meta.error` déjà persistée ou nécessitant un second patch, échec persistance auto-accept, throw/EOF, throw après terminal et absence de settlement prématuré.

Adaptation EasyAgents

26) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/EasyAgentsBatchLiveSessionAdapter.service.ts` — **modifié**

- `replayMode='replaceTurnAssistants'`, `sourceId=turnId`, `timestamp`.
- Remplacer `canAbortSource` par `claims` exacts.
- Exposer autorité source/réservation, claim destructif et éviction sous `handle`, sans dépendre du coordinateur.

27) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/__test__/EasyAgentsBatchLiveSessionAdapter.service.test.ts` — **modifié**

- Tester identité, `replayMode`, `timestamp`, `claims`, `handle` d'éviction et `wake-up` générique.

28) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunExecution.service.ts` — **modifié**

- Construire tous les `runRefs` avec `sourceId=liveTurnId`, sans changer les transitions Batch.

29) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.live-activity-and-gating.service.test.ts` — **modifié**

- Étendre les assertions d'identité/`replayMode` en préservant `Watchable/Handoff`.

30) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.commit-and-recovery-failures.service.test.ts` — **modifié**

- Ajouter `sourceId` aux assertions `handoff` sans changer l'ordre `commit/cleanup/publication`.

31) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.post-commit-persistence.service.test.ts` — **modifié**

- Ajouter `sourceId` et préserver l'ordre `patch parent` → `publication` → `handoff`.

32) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunChildExecution.service.ts` — **modifié**

- Remplacer le `lifecycle` local par `LiveSessionSourceLifecycleService` injecté ; conserver `parsing/classification/callbacks`.

33) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunChildExecution.service.test.ts` — **modifié**

- Conserver les tests Batch et déplacer les tests `lifecycle` purs.

34) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunManager.service.ts` — **modifié**

- Instancier/injecter le `lifecycle` partagé.
- Migrer `items` et compléter le précheck `Retry` avec l'autorité registre.
- Passer `claims` destructifs et éviction sous `handle` au `Retry`.

35) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunExecution.testHarness.ts` — **modifié**

- Injecter `lifecycle` et `sourceId`.

36) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunLifecycle.service.test.ts` — **modifié**

- Adapter la construction au `lifecycle` partagé.

37) `AIcode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchLastViewerStop.service.test.ts` — **modifié**

- Ajouter `sourceId` et conserver les courses `last-viewer/ancien-nouveau` run.

38) `AIcode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md` — **modifié**

- Documenter `sourceId=turnId`, `lifecycle` partagé, `observer/open`, `elapsed` historique, `last viewer kill`, `claim Retry` et `wake-ups` `observer/claim`.

Endpoints et message bus

39) `AIcode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/open.endpoint.ts` — **modifié**

- Exiger `sourceId` et transporter `sourceId/replayMode`.

40) `AIcode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/open.endpoint.test.ts` — **modifié**

- Couvrir champs, branches et mismatches.

41) `AIcode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/stream.endpoint.ts` — **modifié**

- Exiger `sourceId` et relayer `runRef` complet.

42) `AIcode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/__test__/stream.endpoint.test.ts` — **modifié**

- Couvrir identité, terminal, éviction sans terminal, attente finalisation et erreur `post-header body`.

43) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abort.endpoint.ts` — **modifié**

- Exiger `sourceId` et utiliser le `claim viewer` atomique avec `lease active`.

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

- Couvrir `viewer`, `stale`, `double Stop`, remplacement et `idempotence`.

45) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abortSource.endpoint.ts` — **créé**

- `Stop exact source Prompt` par `locator+turn+source`, réservation ou `record`, `claim` puis `abort` sans `await` intermédiaire.
- Refuser observer, `settled`, `stale`, `double claim` et `absent`.
- Après `claim` accepté, notifier synchroniquement le runtime `Activité sourceStopRequested` avant l'abort ; la publication `Sidebar` reste auxiliaire.

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

- Couvrir réservation, `record`, `stale`, `double clic`, observer, ordre `claim` → `stopRequested` → `abort` et fermeture immédiate du panneau après `Stop`.

47) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/stream.endpoint.ts` — **modifié**

- Ajouter `sourceId`, préserver ordre `SPECIF/catalogue` et déléguer au service `source`.
- Encoder uniquement le premier `meta.error` sous header `V1`.
- Après `200_OK`, envoyer les `meta.error` comme `body chunks` normaux.
- Ne jamais tenter un second header applicatif. Une exception `post-header` est laissée au framework `Slimbk`, dont l'enveloppe `body` sera filtrée par les `handlers extension`.
- Ne pas finaliser le registre.

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

- Tester délégation, premier header `V1`, `meta.error` tardif `body 200`, persistance préalable, absence de synthèse après terminal, absence de finalisation et exception `post-200` sérialisée comme enveloppe `Slimbk body`.

49) `AICode/src/extension/src/messageBus/handlers/ia/liveSessionStreamHandlers.ts` — **modifié**

- Relayer `sourceId` et recevoir le contexte d'origine `extension-only`.
- Refuser une demande dont la `webview` d'origine n'est plus le panneau exact enregistré, y compris si un nouveau panneau du même `locator` existe.
- `Source ownership` : aucun `timer/son` supplémentaire. Observer : ressources uniquement pour ce panneau exact.
- Filtrer toute enveloppe `Slimbk non-200 post-header` avant la `FIFO` et la classer `transport`.
- Utiliser `runRef + soundAttemptId`, allocation `Activité synchrone` et `cleanup exact`.

50) `AICode/src/extension/src/messageBus/handlers/ia/__test__/liveSessionStreamHandlers.test.ts` — **modifié**

- Couvrir `sourceId`, `ownership`, contexte exact, ancien panneau remplacé, observer fermé, publication `Activity` échouée, son `stale`, `body-error post-header`, terminal et éviction.

51) `AICode/src/extension/src/messageBus/handlers/ia/iaStreamHandlers.ts` — **modifié**

- Recevoir `sourceId` et contexte d'origine exact.
- Vérifier `webview + locator` avant réservation ; un ancien message d'un panneau disposé ne doit pas utiliser un panneau rouvert du même `locator`.
- Réserver, allouer `Activité` et son synchroniquement avant tout `await` métier.
- Démarrer publication `Activity` comme tâche auxiliaire sans empêcher l'endpoint.
- Maintenir `FIFO unique` ; `onChunk` filtre d'abord l'enveloppe `Slimbk post-header`, sinon enfile le vrai `chunk`.
- `onError` décode d'abord le protocole premier-header `V1`, sinon mémorise `transport` sans `throw` ; `callback idempotent`.
- Chaque tâche `FIFO` tente `Activity`, son et `emit` séparément, poursuit les `chunks` et agrège les erreurs dans l'ordre.
- Après `call + FIFO` : déterminer `completed/transport`, finaliser seulement si réservation possédée, arrêter le son par `attempt` et appeler `notifyPromptSourceSettled(...)` une seule fois.

52) `AICode/src/extension/src/messageBus/handlers/ia/__test__/iaStreamHandlers.queryDoneSound.test.ts` — **modifié**

- Couvrir origine exacte, panneau remplacé, `sourceId/soundAttemptId`, `Activity` auxiliaire, `FIFO`, premier header `source-error`, `meta.error body`, `body Slimbk error post-header`, terminal puis `transportError`, erreurs auxiliaires non court-circuitantes, `onError idempotent` et conflit de réservation.

53) `AICode/src/extension/src/messageBus/handlers/ia/__test__/iaStreamHandlers.runIdPropagation.test.ts` — **modifié**

- Vérifier ordre réservation → allocation → endpoint/FIFO → finalisation → son → coordinateur, terminal header dans FIFO et transport sans terminal retirant le run.

54) `AICode/src/extension/src/extension/panels/prompt/PromptQueryDoneSound.manager.ts` — **modifié**

- Corréler par runRef exact ; onAiStreamStart retourne soundAttemptId monotone.
- Exiger runRef+attempt pour chunk/stop, ignorer stale, ne jamais créer implicitement un état.
- Dédupliquer terminal par runRef sans réarmer lors d'une reconnexion.
- Conserver l'état tant que source/observer actif ; cleanup uniquement via coordinateur.

55) `AICode/src/extension/src/extension/panels/prompt/__test__/PromptQueryDoneSound.manager.test.ts` — **modifié**

- Couvrir sources successives, attempts du même run, stale, dédup, fermeture et conservation source.

Activité runtime extension

56) `AICode/src/extension/src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts` — **modifié**

- Renommer la classe interne `SidebarQueryActivityRuntimeService`.
- Stocker runRef, ownership, runId, startMs/sourceStartedAtMs, timer, usage et stopRequested par locator.
- Exposer une allocation de début synchrone { runId, publication }.
- Exposer `onSourceStopRequested(runRef)` : mutation interne/store synchrone, broadcast auxiliaire, guard runRef+runId.
- Exposer snapshot readonly pour rehydrate et transition terminale source appelée uniquement par coordinateur.
- Source elapsed depuis sourceStartedAtMs ; observer depuis tentative viewer.

57) `AICode/src/extension/src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.race.test.ts` — **modifié**

- Couvrir sources successives, stale, open/closed, allocation avant broadcast, échec broadcast, stopRequested exact et fermeture après Stop sans réactivation du bouton.

58) `AICode/src/extension/src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.usage.test.ts` — **modifié**

- Couvrir usage, Idle/suppression, missing terminal, publication échouée et arrêt timer exact.

Coordination panneau Prompt

59) `AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelQuery.util.ts` — **créé**

- Parser occurrences brutes, détecter toute query explicite, inconnus `liveSession*`, doublons locator/live et valider avec `ZLiveSessionPanelDescriptor` shared.
- Préserver toute query explicite valide ; rejeter invalide.
- Construire normalPrompt avec lease fraîche, sourceId/adaptateur/policy exacts et suppression complète `autoRun*`.

60) `AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionPanelQuery.util.test.ts` — **créé**

- Couvrir couples shared, égalité source/turn EasyAgents, ligne requise/interdite, doublons, inconnus, ordre, lease, invalid-before-opener et autoRun.

61) `AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelCoordinator.service.ts` — **créé**

- Classe injectée sans import low-level, FIFO par locator.
- `openPromptPanel`, `onPanelDisposed`, `onSourceSettled`, wake-ups observer/claim.
- Préserver explicite valide, rejeter invalide, injecter normalPrompt seulement pour source active/réservée.
- Releases adapter-specific ; normalPrompt libère uniquement lease.
- Unique propriétaire de la transition terminale Activity et des cleanups revalidés.
- Sous claim, conserver mapping détaché et différer tous les cleanups.

62) `AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionPanelCoordinator.service.test.ts` — **créé**

- Couvrir priorités query, races open/close/settle, missing terminal, Idle ouvert, observer wake-up, claim wake-up, réouverture/nouvelle source et cleanup unique.

63) `AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionDetachedStop.service.ts` — **créé**

- Valider synchroniquement projection source/closed exacte, source active/réservée et claim d'abort disponible.
- Partager la transition `onSourceStopRequested` du runtime Activité, puis abort sans await intermédiaire.
- Ne jamais abort une source différente depuis une projection stale.

64) AICode/src/extension/src/extension/panels/prompt/liveSession/__test__/PromptLiveSessionDetachedStop.service.test.ts — créé

- Couvrir accept/refus, reopened, remplacement, double clic, réservation, store stale, claim destructif et ordre stopRequested/abort.

65) AICode/src/extension/src/extension/panels/prompt/PromptThreadCleanup.manager.ts — modifié

- Indexer par locator complet ; séparer register, detach, cleanup historique, cleanup terminal et orphan cleanup.
- Ne jamais abort/clear une source ou un observer actif.
- Conserver mapping détaché pendant les cleanups différés et sous claim ; revalider après chaque await.

66) AICode/src/extension/src/extension/panels/prompt/__test__/PromptThreadCleanup.manager.attachmentsCleanup.test.ts — modifié

- Couvrir scopes, detach, terminal, nouvelle source, observer wake-up, claim wake-up, attachments et idempotence.

67) AICode/src/extension/src/extension/panels/core/openPanel.util.ts — modifié

- Conserver le nom public openCentralPanel(...) mais le rendre async.
- Extraire un low-level opener privé synchrone ; construire le coordinateur module-scoped avec ce port injecté.
- Prompt passe par le coordinateur ; autres pages passent directement par le low-level.
- Exporter notifyPromptSourceSettled(...) pour le handler source.
- Capturer le descripteur initial ; au dispose, supprimer synchroniquement le panneau du registre avant toute délégation.
- Adapter titres/Activity à items/status, injecter subscriptions et runtime, préserver absence de cycle.

68) AICode/src/extension/src/extension/panels/core/__test__/openPanel.util.test.ts — modifié

- Await façade, wiring sans cycle, exact descriptor, reveal, dispose order, sourceSettled, subscriptions, claim et pages non-Prompt.

69) AICode/src/extension/src/messageBus/handlers/ui/uiGenericHandlers.ts — modifié

- Await openCentralPanel ; Prompt est coordonné, autres pages inchangées.

70) AICode/src/extension/src/messageBus/handlers/ui/__test__/uiGenericHandlers.openExternalUrl.test.ts — modifié

- Mocker façade async et préserver URL.

71) AICode/src/extension/src/extension/commands/NewChat.command.ts — modifié

- Await ouverture coordonnée.

72) AICode/src/extension/src/extension/commands/__test__/NewChat.command.test.ts — modifié

- Vérifier await et propagation erreur.

73) AICode/src/extension/src/extension/uriHandlers/registerDeepLinkUriHandler.ts — modifié

- Deep-link chat via façade coordonnée en fire-and-forget avec catch explicite.

74) AICode/src/extension/src/extension/uriHandlers/__test__/registerDeepLinkUriHandler.scope.test.ts — modifié

- Couvrir scopes, façade et guards.

SidebarStore et Activité

75) AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.listeners.manager.ts — modifié

- Transitions atomiques panelOpened, sourceStarted, sourceStopRequested, sourceSettled.
- SourceStarted reçoit run/runtime déjà alloué ; publication séparée.
- Mapping final non-stale : open → Idle, closed → remove ; stale/source différente no-op.

76) AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.mutations.manager.ts — modifié

- windows→items, panelClosed, forceRemove, elapsed et stopRequested exacts.
- Conserver uniquement Running/source fermé ; lors de panelClosed, dériver stopRequested du snapshot registre/runtime exact.

77) AICode/src/extension/src/extension/panels/sidebar/SidebarStore.core.manager.ts — modifié

- Initialiser items et appeler rehydrateQueryActivities.

78) AICode/src/extension/src/extension/panels/sidebar/SidebarStore.rehydrate.manager.ts — modifié

- Fusionner panneaux, observers actifs et sources/réservations ; dédupliquer.
- Interdire observer/closed et Idle/closed.
- Conserver terminal bufferisé non finalisé en Running ; panneau ouvert sans record en Idle.
- Reconstituer title/openedAt par runtime/store/panneau/persistence/default dans cet ordre, sans faire du SidebarStore l'autorité des timers.

- 79) `AICode/src/extension/src/extension/panels/sidebar/__test__/SidebarStore.rehydrate.manager.test.ts` — **modifié**
- Couvrir toutes projections, publication manquante, réservation, observer fermé, terminal pré-forwarding, missing terminal et stopRequested.
- 80) `AICode/src/extension/src/extension/panels/sidebar/__test__/SidebarStore.manager.queryRunningUpsert.test.ts` — **modifié**
- Tester union, transitions, publication auxiliaire, outcomes stricts et sourceStopRequested.
- 81) `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.chatTitles.manager.ts` — **modifié**
- Lire items et préserver fallback title source fermée/running.
- 82) `AICode/src/extension/src/extension/panels/sidebar/SidebarStore.handlers.manager.ts` — **modifié**
- Enregistrer open activity et Stop détaché ; déléguer coordinateur/service.
- 83) `AICode/src/extension/src/extension/panels/sidebar/__test__/SidebarStore.handlers.manager.test.ts` — **créé**
- Vérifier payloads, accepted et délégation.
- 84) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/delete.endpoint.ts` — **modifié**
- Lire items et autorité liveSession.
- Acquérir claim avant fermeture ; fermer/retirer projection sous claim.
- Delete durable sans cleanup concurrent ; éviction seulement après succès avec handle exact.
- Release en finally ; pas d'éviction préventive sur échec disque.
- 85) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/delete.endpoint.liveSession.test.ts` — **créé**
- Couvrir store stale, réservation, claim indisponible, start concurrent refusé, handle exact, succès/échec, tombstones et cleanup différé.
- 86) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/list.endpoint.windowsLock.eperm.test.ts` — **modifié**
- Adapter fixtures items.
- 87) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/__test__/diskAliases.lastWins.endpoints.test.ts` — **modifié**
- Adapter items et préserver delete disque.
- 88) `AICode/src/extension/src/aiTools/orch/context/persistence/__test__/ChatPersistenceWriter.windowsFallback.test.ts` — **modifié**
- Adapter fixture items.
- 89) `AICode/src/extension/src/aiTools/orch/context/memory/core/mem/__test__/OrchMemory.naive_title.user_prompt_specif_multiline.e2e.repro.test.ts` — **modifié**
- Adapter fixture items.
- 90) `AICode/src/frontend/src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx` — **modifié**
- Consommer items/union, ouvrir par nouveau RPC et afficher Stop troisième ligne uniquement source/closed.
- SquareStop, small, danger, label Stop, disableAfterClick, enabled !stopRequested, testIds stables.
- Callbacks via useCallback, objets via useMemo, aucun hook dans map ; sous-composant Stop keyé par sourceId.
- 91) `AICode/src/frontend/src/app/sidebar/sidebar-activity/__test__/SidebarActivityQuery.test.tsx` — **modifié**
- Couvrir projections, open RPC, Stop exact, désactivation, disparition et remount source A → B sans labels fragiles.
- Frontend liveSession et Prompt**
- 92) `AICode/src/frontend/src/app/prompt/liveSession/adapters/NormalPromptLiveSession.adapter.ts` — **créé**
- Binder descripteur normalPrompt shared et politiques not_ready/replayFailed/direct handoff/composer interactif.
- Aucun Stop de bannière ; Stop compositeur unique.
- Initial baseline = hydratation complète ; retries = preserveLocal.
- 93) `AICode/src/frontend/src/app/prompt/liveSession/adapters/__test__/NormalPromptLiveSession.adapter.test.ts` — **créé**
- Couvrir parsing/binding, politiques, surfaces, bannière, handoff et preserveLocal.
- 94) `AICode/src/frontend/src/app/prompt/liveSession/adapters/PromptLiveSessionBootstrap.adapter.ts` — **créé**
- Dispatcher pur recevant snapshot brut ; getAll sur paramètres critiques.
- Importer liste/union shared ; détecter partiels, inconnus, doublons et locator ambigu.
- Retourner classic/valid/invalid, locator strict éventuel et bool explicite ; aucune lecture globale.

- 95) `AIcode/src/frontend/src/app/prompt/liveSession/adapters/__test__/PromptLiveSessionBootstrap.adapter.test.ts` — **créé**
 - Couvrir dispatch, doublons identiques/différents, inconnus, adaptateur, égalité EasyAgents, invalid sans locator et pureté.
- 96) `AIcode/src/frontend/src/app/prompt/liveSession/core/PromptLiveSession.types.ts` — **modifié**
 - Ajouter `liveSessionReplayFailed`, `sourceId` et politiques explicites : composer, Send, historique, cleanup, Stop, handoff, snapshot policy, bannière.
 - Séparer `PromptLiveSessionBannerViewModel` de la surface composer.
 - Représenter `directAfterReplay` vs `terminalSnapshot` et `replayOpenAttemptId` vs `streamInstanceId`.
- 97) `AIcode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.tsx` — **modifié**
 - Consommer le modèle de bannière dédié, compatible viewer EasyAgents et bannière normalPrompt au-dessus du composer.
 - Conserver `status/details/Stop/Retry`, callbacks stables et `testIds`.
- 98) `AIcode/src/frontend/src/app/prompt/liveSession/components/__test__/PromptLiveSessionBanner.test.tsx` — **modifié**
 - Adapter fixtures et ajouter `Retry` normalPrompt sans `Stop`.
- 99) `AIcode/src/frontend/src/app/prompt/liveSession/core/usePromptLiveSession.hook.ts` — **modifié**
 - Maintenir `replayOpenAttemptId` pour bootstrap/retry/timers/unmount.
 - Parser d'abord enveloppe ; full parse pour hydrateFromSnapshot, session opaque pour preserveLocal.
 - Retry `not_ready` normalPrompt ; `not_found/transport/completion` sans terminal/éviction → `replayFailed`.
 - Démarrer observer uniquement après start accepté et conserver `streamInstanceId` exact.
 - Handoff direct sans second Open après terminal data + completed + queues + finalization exacte.
 - Retry terminal sans stream fictif ; `preserveLocal` pour messages/stores seulement.
 - Gérer `transportStopAvailable` et suppression explicite sur terminal/`not_found/reset/Stop`.
- 100) `AIcode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.hydration.test.tsx` — **modifié**
 - Couvrir politiques adaptateur, full vs preserveLocal, draft malformé/getters non lus et stale Open/timer/unmount.
- 101) `AIcode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.handoff.test.tsx` — **modifié**
 - Couvrir EasyAgents terminal, normal direct, Queue B, stale finalization, terminal data + transportError, source error data et absence de second Open.
- 102) `AIcode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.normalPrompt.test.tsx` — **créé**
 - Couvrir `not_ready/not_found`, surface, draft/attachments/nextTurnId, Stop, retry, terminal direct, retry terminal sans stream, enveloppe draft opaque, `replayFailed` abortable et stale attempts.
- 103) `AIcode/src/frontend/src/app/prompt/liveSession/core/__test__/usePromptLiveSession.normalPromptIsomorphism.test.tsx` — **créé**
 - Comparer stream/replay/hydratation pour tous query types, thinking, tools, usage, edits, erreurs, EOF et transport après terminal.
- 104) `AIcode/src/frontend/src/app/prompt/liveSession/adapters/EasyAgentsBatchLiveSession.adapter.ts` — **modifié**
 - Importer constante shared, ajouter `sourceId=turnId/adaptateur`, valider union shared et préserver UX/read-only/elapsed.
- 105) `AIcode/src/frontend/src/app/prompt/liveSession/adapters/__test__/EasyAgentsBatchLiveSession.adapter.test.ts` — **modifié**
 - Adapter query/`sourceId/dispatcher/constante` et policy full hydration.
- 106) `AIcode/src/frontend/src/messageBus/AiStreamStartArgs.type.ts` — **modifié**
 - Définir état interne `runRef+streamInstanceId+outcome`, union de retour start et type `AiStreamFallbackPersistedAck { turnId; streamInstanceId }`.
 - Observer transporte `runRef` complet, lease, `replayMode` et policy Stop.
 - Aucun identifiant frontend-only au backend hors `sourceId` du `runRef`.
- 107) `AIcode/src/frontend/src/messageBus/useAiStream.hook.ts` — **modifié**
 - Single-flight ref synchrone ; IDs générés seulement après acceptation.
 - Retourner union accept/refuse et ne rien muter sur refus.
 - Utiliser `streamInstanceId` comme listener key ; exposer `runRef`, IDs, outcome, terminal data, `transportStopAvailable`, Stop state et ack fallback corrélé à `streamInstanceId`.
 - Garder callbacks/`runMerged` avant/après awaits ; invalider avant reset.
 - Attendre Queue A, Queue B puis commit avant outcome/libération.

- onError sans terminal crée SYSTEM_ERROR ; après terminal conserve terminal unique et outcome transportError.
- Stop source via abortSource, observer via abort lease ; disponibilité consommée avant premier await.
108) AICode/src/frontend/src/messageBus/__test__/useAiStream.hook.liveSessionObserver.test.ts — modifié
- Couvrir IDs, routes, stale callbacks/runMerged, Queue B, outcomes, terminal+error sans doublon, transportStopAvailable et ack corrélié.
109) AICode/src/frontend/src/messageBus/__test__/useAiStream.hook.promptSourceIdentity.test.ts — créé
- Couvrir génération IDs après acceptation, double start même tick, refus sans mutation, observer sourceId fourni, Stop result et callbacks stale.
110) AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.hook.ts — modifié
- Propager start results, runRef/streamInstance/outcome/terminal/transportStopAvailable/fallback ack corrélié.
- StartLiveObserver prépare/committe placeholders seulement après acceptation.
- Lier base thinking à streamInstanceId accepté.
- Exposer finalizedStreamInstanceId, clear transport availability et reset complet.
111) AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunController.liveSessionObserver.test.tsx — modifié
- Couvrir accept/refuse, aucun placeholder sur refus, IDs, base thinking, Stop post-transport et finalization ack.
112) AICode/src/frontend/src/app/prompt/Prompt/usePromptCancelController.hook.ts — modifié
- Séparer isStreaming et transportStopAvailable.
- UI-only cancel seulement si les deux faux ; sinon Stop exact unique.
- Pour replayFailed hors transport, finir le handshake après réponse Stop et conserver disponibilité false.
113) AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptCancelController.stopCore.test.tsx — modifié
- Ajouter cas isStreaming=false + transportStopAvailable=true, double clic et absence de mutation source UI-only.
Continuation et thinking
114) AICode/src/frontend/src/app/prompt/Prompt/preparePromptContinuationStreaming.util.ts — créé
- Préparation pure : assistant non-thinking exact, bases assistant/thinking, UID réutilisé, placeholder thinking unique, séparateur et resets planifiés sans mutation.
115) AICode/src/frontend/src/app/prompt/Prompt/__test__/preparePromptContinuationStreaming.util.test.ts — créé
- Couvrir bases, assistant manquant, séparateur, sources successives, aucune duplication/mutation.
116) AICode/src/frontend/src/app/prompt/Prompt/preparePromptStreamingTurn.util.ts — modifié
- Séparer plan pur et commit ; reset base thinking/séparateur pour replace.
117) AICode/src/frontend/src/app/prompt/Prompt/__test__/preparePromptStreamingTurn.util.test.ts — modifié
- Vérifier plan non committé sur refus et resets replace.
118) AICode/src/frontend/src/app/prompt/Prompt/usePromptThinkingLifecycle.hook.ts — modifié
- Initialiser base thinking par streamInstanceId, append/séparateur unique, reset explicite et cleanup seulement si base vide.
119) AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptThinkingLifecycle.continuation.test.tsx — créé
- Couvrir base, append, tool/text avant thinking, réinsertion, reset et retry même source/nouvelle tentative.
120) AICode/src/frontend/src/app/prompt/Prompt/usePromptToolUse.hook.ts — modifié
- Ne retirer la bulle que si baseThinking vide ; éviter doubles séparateurs.
121) AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.finalization.hook.ts — modifié
- Garder toutes mutations par streamInstanceId, reset bases et publier finalizedStreamInstanceId après finalisation complète.
122) AICode/src/frontend/src/app/prompt/Prompt/usePromptRunCommands.hook.ts — modifié
- Plans purs + start synchrone + commit après acceptation pour Send/SPECIF/Retry/Edit/Compact/Reply/Continue.
- Retourner le résultat strict au caller ; supprimer async artificiels.
- Sur refus, zéro mutation durable et uiPending=false dans la même pile.
123) AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.compactContextNow.test.tsx — modifié
- Adapter acceptation et confirmer replace sans base héritée.
124) AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.retryQuery.attachmentOnly.test.tsx — modifié
- Adapter acceptation et préserver attachment-only.

125) `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.retryQuery.compactContextNow.test.tsx` — **modifié**

- Adapter acceptation et préserver mapping compact.

126) `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.continueResponse.test.tsx` — **créé**

- Vérifier continuation, thinking unique, même turn/nouvelle tentative et refus sans mutation.

Intégration Prompt et documentation

127) `AICode/src/frontend/src/app/prompt/Prompt/Prompt.tsx` — **modifié**

- Séparer frontière bootstrap externe et runtime interne résolu.
- Lire une seule fois le snapshot brut ; aucune décision live via hooks scalaires.
- Invalid sans locator : dialog/close sans contrôleur, hydratation ni autoRun.
- Déduire séparément composer, Send, Stop, historique, cleanup et bannière.
- Autoriser draft/attachments pendant normalPrompt replay selon policy.
- Utiliser transportStopAvailable dans locks/hotkeys/cancel.
- Après Send/Edit accepté, stocker { `turnId`, `streamInstanceId` } dans la barrière de fallback et seulement alors armer submitTxn/focus.
- Effet unique : ack matching → clear historique ; finalization matching sans ack → démasquer/restaurer focus sans effacer texte, attachments, nextTurnId, pendingEditTurnId ou maximized.
- Toute query live explicite désactive hydratation classique et autoRun.

128) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.liveSessionBootstrap.test.tsx` — **modifié**

- Couvrir snapshot brut, doublons, invalid sans runtime, autoRun neutralisé, surface/replay failure et refus start sans barrières.

129) `AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.tsx` — **modifié**

- Supporter composer seul, viewer banner remplaçant composer, ou composer + bannière normalPrompt.
- Bannière normalPrompt au-dessus du textarea, sans second Stop ; VscTextarea reste Stop unique.
- Séparer showComposer, edit, Send lock et banner model.

130) `AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.module.css` — **modifié**

- Ajouter wrapper/espaceur du bandeau normalPrompt, variables VS Code uniquement, aucune CSS inline.

131) `AICode/src/frontend/src/app/prompt/PromptInput/__test__/PromptInput.batchLiveViewerMode.test.tsx` — **modifié**

- Préserver viewer EasyAgents ; ajouter composer normalPrompt + Retry banner sans Stop, attachments et Send lock.

132) `AICode/src/frontend/src/app/easy-agents/easy-agents-batch/core/__test__/EasyAgentsBatch.liveWatch.test.tsx` — **modifié**

- Adapter query source/adaptateur et constante shared.

133) `AICode/src/frontend/src/app/easy-agents/easy-agents-item/core/batch/__test__/EasyAgentBatchSurface.test.tsx` — **modifié**

- Adapter query exacte sans changement UX.

134) `AICode/src/frontend/src/app/prompt/hydration/isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-readme.md` — **modifié**

- Documenter fermeture/réouverture, continuation/thinking, finalization, erreur premier-header, meta.error tardive, enveloppe Slimbk post-header filtrée, persistance de tout meta.error avant append, terminal+transportError unique, direct handoff et preserveLocal opaque.

135) `AICode/src/frontend/src/app/prompt/thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md` — **modifié**

- Documenter base thinking continuation, séparateur unique, guards text/tool et streamInstanceId.

Contrats transverses frontend

136) `AICode/src/frontend/src/app/prompt/Prompt/applyHydratedPromptSessionState.util.ts` — **modifié**

- Union discriminée `hydrateFromSnapshot | preserveLocal`.
- Full : comportement actuel.
- Preserve : aucun setter composer, aucune lecture/validation/normalisation de draft/attachments/nextTurnId ; construire un FrontChatSession assaini uniquement depuis champs consommés.
- Getters piégés sur champs ignorés ne doivent pas être exécutés.
- Retour `void`; doc refs messages toujours enregistrées.

137) `AICode/src/frontend/src/app/prompt/Prompt/__test__/applyHydratedPromptSessionState.util.test.ts` — **modifié**

- Adapter full et couvrir preserveLocal, getters non lus, messages/stores/thread/reset/scroll et retour void.

- 138) `AICode/src/frontend/src/app/prompt/Prompt/usePromptSessionHydration.hook.ts` — **modifié**
- Appeler `hydrateFromSnapshot` ; désactiver pour toute query live explicite.
- 139) `AICode/src/frontend/src/app/prompt/hydration/frontSessionHydration.type.ts` — **modifié**
- Généraliser `allowOrphanToolResults=true` aux baselines live-safe EasyAgents et `normalPrompt`, sans modifier `pairing`.
- 140) `AICode/src/frontend/src/app/prompt/lib/usePromptSessionLocator.hook.ts` — **modifié**
- Ajouter `usePromptRouterQueryStringRawSnapshot()` via `useMemo(..., [])`.
- Conserver parseurs locator existants pour autres consommateurs ; documenter que le snapshot est la voie bootstrap Prompt.
- 141) `AICode/src/frontend/src/app/prompt/lib/__test__/usePromptSessionLocator.hook.test.tsx` — **modifié**
- Ajouter harness prouvant snapshot figé après mutation globale/rerender.
- 142) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.autoRunPrompt.test.tsx` — **modifié**
- Mock snapshot brut classique et union d'acceptation ; préserver `autoRun` classique.
- 143) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.noThumbTurnIdDrift.test.tsx` — **modifié**
- Mock snapshot ; acceptation synchrone Edit ; préserver `turnId` thumbnails jusqu'à `ack`.
- 144) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.restoreAttachments.test.tsx` — **modifié**
- Mock snapshot et préserver restauration attachments.
- 145) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.ensureNextTurnId.stability.test.tsx` — **modifié**
- Mock snapshot/acceptation et préserver `nextTurnId`.
- 146) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearAttachments.test.tsx` — **modifié**
- Vérifier `clear` seulement sur `ack` exact et conservation attachments/`nextTurnId`/`pending` edit lors d'une finalization sans `ack`.
- 147) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearInput.test.tsx` — **modifié**
- Vérifier masque armé après acceptation, `clear` sur `ack` exact, démasquage sans `clear` sur finalization sans `ack` et focus restauré.
- 148) `AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.promptDraft.hydration.test.tsx` — **modifié**
- Mock snapshot/acceptation et préserver `debounce/hydratation`.
- 149) `AICode/src/frontend/src/app/prompt/Prompt/__test__/usePromptRunCommands.startAcceptance.test.tsx` — **créé**
- Harness commun sept commandes ; refus sans mutation, acceptation avec un seul scaffold et résultat synchrone exact.
- 150) `AICode/src/frontend/src/messageBus/prompt-stream-dispatch-lock.programming-doc-readme.md` — **modifié**
- Documenter `streamInstance` guards, Queue A+B idle, `commit` final, `reset` stale interdit et `terminal+transportError` sans doublon.
- 151) `AICode/src/extension/src/aiTools/orch/ask/stream/core/fallback-persist-and-title.programming-doc-readme.md` — **modifié**
- Documenter barrières armées après acceptation.
- Ajouter le cas `accepted` puis finalization sans fallback `ack` : démasquer sans effacer le brouillon/attachments/`nextTurnId`/edit state.
- Atomicité destructive Retry EasyAgents**
- 152) `AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunRetry.service.ts` — **modifié**
- Après guards non destructifs, acquérir `claim` exact juste avant fermeture.
- Tenir le `claim` pendant fermeture, éviction pré-delete, UI `remove` et `delete`.
- Passer `handle` exact à éviction ; `release` en `finally` sur toute issue.
- Préserver toutes règles Git/reconstruction/viewer.
- 153) `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunRetry.testHarness.ts` — **modifié**
- Ajouter mocks stricts `claim/release/autorité/evict(handle)`, sans casts inutiles.
- 154) `AICode/src/extension/src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunRetry.guards.service.test.ts` — **modifié**
- Couvrir `claim` indisponible, ordre, `handle` exact, `release` succès/erreurs, source concurrente refusée et `cleanup` différé.

Barrières de transport et origine exacte révélées par la réinspection

155) AICode/src/sharedlib/src/slimbk/framework/parseSlimbkStreamBodyError.util.ts — créé

- Détecter avant toute logique métier une enveloppe SlimbkResult non-200 apparue dans le body après un header 200.
- Valider status connu, errorMessage string et errorCode optionnel ; toute forme avec un champ top-level status incohérent devient une erreur de protocole, jamais un AiStreamedChunk.
- Ne jamais interpréter cette voie comme protocole source V1, même si texte/code coïncident.

156) AICode/src/sharedlib/src/slimbk/framework/__test__/parseSlimbkStreamBodyError.util.test.ts — créé

- Couvrir enveloppes valides, 200 ignoré, statuts/codes malformés, objets/chunks ordinaires et collision volontaire avec le préfixe V1.

157) AICode/src/extension/src/messageBus/core/extTypedMessageBus.util.ts — modifié

- Définir un StreamHandlerOriginContext extension-only contenant htmlPage et la webview exacte.
- Étendre onStreamed, dispatchStreamedCall et getStreamHandler pour le transmettre comme troisième argument sans toucher aux contrats sérialisés.

158) AICode/src/extension/src/messageBus/core/extMessageBus.util.ts — modifié

- Construire le contexte d'origine dans initializeViewWithMessageBus et le transmettre au chemin stream.
- Exiger ce contexte pour un stream webview ; les appels RPC non-stream restent inchangés.

159) AICode/src/extension/src/messageBus/core/__test__/extMessageBus.streamOriginContext.test.ts — créé

- Prouver que le handler reçoit la webview émettrice exacte, que deux panneaux Prompt ont des contextes distincts et qu'aucune donnée n'est ajoutée au payload sérialisé.

160) AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.module.css — modifié

- Mettre à jour le commentaire de maintenance : le bandeau n'est plus uniquement read-only ni forcément rendu à la place du compositeur.
- Conserver les variables VS Code et le style existant, sauf ajustement strictement nécessaire au modèle réutilisable.

Récapitulatif exact des chemins de fichiers

Créés

- 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

Modifiés

- AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionRun.type.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.hydration.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.hook.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`

Déplacés

Aucun.

Supprimés

Aucun.

Facteur de risque de régression

1. **Risque calculé : 92 %.** Criticité 5, couplage 5, blast radius 5, incertitude 4, faible confiance 3. $Base = 10 \times 5 + 3 \times 5 + 2 \times 5 + 2 \times 4 + 3 \times 3 = 92$; $risque = round(92 \times 5/5) = 92 \%$.
2. **Zones complexes :** start transactionnel, exactitude webview d'origine, sources successives, réservations/époques, Queue A/B, callbacks stale, terminal puis transportError, enveloppe Slimbk post-header, protocole V1 premier-header, persistance de tout meta.error avant append, finalisation post-forwarding, retraites/tombstones, claims destructifs, wake-ups, continuation/thinking, parse query brut, frontière React, preserveLocal sans lecture de draft, transition Activité unique, stopRequested partagé, son par attempt, delete/Retry sous claim et démasquage du compositeur sans fallback ack.
3. **Justification :** la tranche modifie le chemin normal de toutes les requêtes Prompt et plusieurs coordinations asynchrones. Les corrections ajoutées par cette réinspection ferment quatre courses particulièrement dangereuses : un ancien message de stream ne peut plus démarrer sur un nouveau panneau du même locator ; une exception post-header Slimbk ne peut plus être prise pour un `AiStreamedChunk` ; un `meta.error` ne devient handoffable qu'après persistance exacte, y compris le cas auto-accept ; et une tentative acceptée terminée avant `fallback_persisted` ne laisse plus le compositeur masqué indéfiniment. Les identités exactes, finalisations, claims, wake-ups et tests d'isomorphisme restent indispensables pour éviter perte, duplication, résurrection ou cleanup concurrent.

Résumé

- Étendre `liveSession` aux sources Prompt normales sans tuer la source à la fermeture.
- Ajouter `sourceId` backend exact, `streamInstanceId` frontend et `replayOpenAttemptId` Open-only.
- Rendre start synchrone/transactionnel et toutes les commandes sans mutation sur refus.
- Vérifier la webview d'origine exacte, pas seulement le locator, avant toute réservation.
- Réserver avant await, bufferiser avant forwarding, finaliser seulement après FIFO extension et Queue A/B frontend.
- Persister tout meta.error avant append ; encoder seulement le premier meta.error dans le header V1.
- Filtrer les enveloppes Slimbk d'erreur post-header avant qu'elles ne contaminent la pipeline de chunks.
- Conserver un terminal data unique lorsqu'un transportError suit.
- Finaliser en handoff uniquement avec baseline + terminal durable ; retirer sinon.
- Séparer retraites, tombstones, époque de run, claim destructif et génération destructive.
- Sérialiser Activité/cleanup par locator ; Idle si panneau ouvert, suppression si fermé.
- Propager stopRequested à Activité pour que le Stop détaché reste désactivé après fermeture.
- Reconnecter Prompt avec composer interactif, Send verrouillé, Stop unique et Retry banner sans Stop.
- Handoff direct sans second Open ; retries preserveLocal sans lire les champs draft.
- Démasquer sans effacer le compositeur si une tentative acceptée se finalise sans fallback ack.
- Préserver EasyAgents, timers, son, queues, locks, sanitizers et fichiers générés.

Nom du commit

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

`SPECIF_FINALISER_LIVESESSION_POUR_LES_SESSIONS_PROMPT_NORMALES_20260712_204731`

Copy

Refine

Code

Verify

Niveau

Modification appliquée à la spécification

Justification issue de la réinspection



Majeur

Ajout d'un **contexte d'origine extension-only** aux appels streamés, transportant la

Un ancien message resté en file après le dispose pouvait autrement

Niveau	Modification appliquée à la spécification	Justification issue de la réinspection
	webview émettrice exacte jusqu'aux handlers. Le démarrage d'une source ou d'un observer vérifie désormais l'identité du panneau, et non seulement son locator.	démarrer sur un nouveau panneau rouvert avec le même locator.
■ Majeur	Ajout d'une détection partagée des enveloppes d'erreur Slimbk émises dans le body après un header 200_OK . Les handlers les classent comme erreurs transport avant toute logique <code>AiStreamedChunk</code> .	Le client généré transmet actuellement ces enveloppes à <code>onChunk</code> ; sans cette garde, elles pouvaient être émises au frontend comme de faux chunks ou masquer un scénario terminal suivi d'une erreur transport.
■ Majeur	Renforcement du service source <code>normalPrompt</code> : tout <code>meta.error</code> provenant d'Orch doit être persisté idempotemment avant <code>append/yield</code> , pas seulement les terminaux synthétisés par le wrapper.	Cela ferme notamment le chemin auto-accept où Orch peut produire un terminal alors que sa première tentative de persistance a échoué, ce qui aurait rompu l'isomorphisme après hydratation.
■ Majeur	Corrélation de la barrière <code>fallback_persisted</code> à <code>{ turnId, streamInstanceId }</code> et ajout du traitement finalisation sans ack : démasquer le compositeur sans effacer texte, attachments, <code>nextTurnId</code> , état Edit ou maximisation.	Une erreur pré-baseline pouvait laisser <code>submitTxnPending</code> armé indéfiniment et rendre le compositeur inutilisable malgré la fin de la tentative.
■ Mineur	Centralisation d'une union shared stricte de descripteurs de panneau <code>liveSession</code> , incluant les couples adaptateur/policy, la ligne <code>EasyAgents</code> requise ou interdite et <code>sourceId</code> <code>=== turnId</code> pour <code>EasyAgents</code> .	La seule centralisation des noms de paramètres ne garantissait pas que l'extension et le frontend appliqueraient exactement les mêmes règles métier.
■ Mineur	Ajout d'une transition Activité exacte <code>sourceStopRequested</code> , partagée par le Stop du compositeur et le Stop détaché.	Si le panneau se fermait immédiatement après un Stop normal, l'item détaché pouvait brièvement réafficher un bouton actif faute de propagation du claim d'abort vers Activité.
■ Mineur	Canonisation de l'ouverture : <code>openCentralPanel(...)</code> reste le nom public mais devient async, le low-level <code>opener</code> devient privé et le coordinateur reçoit ce port par injection.	Cette précision élimine l'ambiguïté d'implémentation tout en évitant le cycle ESM coordinateur ↔ opener.
■ Mineur	Ajout des tests et fichiers transverses associés, plus correction du commentaire CSS du bandeau <code>liveSession</code> désormais réutilisable avec ou sans compositeur.	La liste des fichiers devait couvrir toutes les nouvelles barrières et ne laisser aucun commentaire de maintenance devenu faux.

</ASSISTANT>