

Permalink URI:

vscode://joelabenhaim.aicode-ja/chat?sessionId=40bac5e75d374e52bc52ad8fb9d1e345-VS-V%C3%A9rifier-FINALISE

<USER>

[Verify]SPECIF FINALISER LIVESESSION POUR LES SESSIONS PROMPT NORMALES 20260712 204731

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧  scanning recent files

🔧  scanning git: 🗑️ "État Git courant", 🗑️ "Historique récent", 🗑️ "Fichiers depuis la base"

🔧  reading project: 📁 g "Docs programmation ciblées", 📁 g "Docs Prompt ciblées", 📁 g "Docs EasyAgents ciblées", 📁 g "Docs bus frontend", 📄 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

🔧  scanning git: 🗑️ "Statistiques du commit", 🗑️ "Contrôle espaces diff"

🔧  reading project: 📄 e "Ancien canAbortSource", 📄 e "Ancien query windows", 📄 e "Ancien streamKey", 📄 e "Source ID runRef", 📄 e "Résultat start refusé", 📄 e "Erreur body Slimbk", 📄 e "Origine webview exacte", 📄 h "Finalisation registre", 📄 h "Handoff frontend", 📄 h "Activité Stop détaché", 📄 "Preuves chemins critiques"

🔧  scanning git: 🗑️ "Ajouts de any", 🗑️ "Ajouts eslint disable", 🗑️ "Ajouts TODO", ❌ 🗑️ "Imports dynamiques ajout", 🗑️ "Generated du commit", 🗑️ "Résumé changements ciblées"

🔧  reading project: 📄 e "Types any explicites", 📄 e "Types any frontend", 📄 e "Types any bus", 📄 e "Types any shared", 📄 e "Imports dynamiques source", 📄 e "Assertions inconnues", 📄 e "Assertions frontend", 📄 h "Publication outcome stream", 📄 h "Barrière fallback Prompt", 📄 h "Bootstrap query Prompt", 📄 h "Coordinateur panneau", 📄 h "Hydratation preserveLocal", 📄 h "Commande start transaction", 📄 h "Delete claim durable", 📄 "Preuves finalisation et i", 📄 PromptLiveSessionPanelCoordinator.service.ts [L1-L469], 📄 applyHydratedPromptSessionState.util.ts [L1-L220], 📄 Prompt.tsx [L1-L140], 📄 Prompt.tsx [L730-L865], 📄 iaStreamHandlers.ts [L100-L328], 📄 liveSessionStreamHandlers.ts [L100-L259], 📄 SlimbkResult.type.ts, 📄 Slimbk.ts [L1-L220], 📄 slimbkBuildHeader.ts [L1-L180], 📄 stream.endpoint.ts [L1-L240], 📄 aiStopSuffix.util.ts [L1-L260], 📄 h "Réservation staged source", 📄 "Capacité réservation", 📄 NormalPromptLiveSessionSource.service.ts [L180-L305]

🔧  scanning git: 🗑️ "Diff Queue A", 🗑️ "Diff Queue B", 🗑️ "Diff lock store"

🔧  reading project: 📄 r "openCentralPanel"

Verdict global 📄

Joel, cette implémentation est impressionnante : fermer un panneau est devenu une négociation diplomatique entre registres, leases, files d'attente et React — mais la diplomatie semble tenir. 😊

J'ai réinspecté :

- le commit courant 4f184413 ;
- les **189 fichiers modifiés ou créés** depuis la base indiquée ;
- les programming docs liveSession, EasyAgents, isomorphisme, thinking, Queue B et fallback persist ;
- les contrats partagés ;
- les chemins critiques backend, transport, panneau, Activité et frontend ;
- les tests ciblant les principales courses et invariants.

Le dépôt est propre sur la branche courante. Le contrôle Git du diff ne signale aucune erreur d'espace ou de fin de ligne.

1) L'intégralité de la spécification est-elle implémentée ?

Réponse

Oui fonctionnellement, avec une complétude statique estimée à 99 %. 

Je n'ai identifié **aucune fonctionnalité demandée clairement absente**, ni aucun ancien contrat principal encore utilisé :

- aucune occurrence de l'ancien canAbortSource ;
- aucune occurrence de l'ancien queryState.windows ;
- aucune occurrence de l'ancien streamKey ;
- les contrats utilisent bien sourceId ;
- le démarrage frontend retourne bien l'union synchrone accepted ;

- les nouveaux endpoints, coordinateurs, adaptateurs, services et tests existent.
Le cœur de l'identité exacte est effectivement défini dans [LiveSessionRun.type.ts](#) et la clé du run contient bien :
- le locator ;
- le `turnId` logique ;
- le `sourceId` exécutif.

Écart constaté dans la liste exacte des fichiers

La spécification annonçait **160 fichiers**, alors que le commit en touche **189**. Plusieurs fichiers complémentaires ont été nécessaires, notamment pour :

- la gestion des erreurs streaming Slimbk ;
- la propagation des erreurs de Queue A et Queue B ;
- la barrière fallback factorisée dans un harness ;
- le comportement de l'index pending ;
- les fichiers générés issus de la compilation Slimbk.

Exemples complémentaires :

- [SlimbkResult.type.ts](#)
- [Slimbk.ts](#)
- [slimbkBuildHeader.ts](#)
- [useAiStreamQueue.hook.ts](#)
- [usePromptStreamDispatchGateQueue.hook.ts](#)
- [PromptFallbackPersistBarrier.testHarness.tsx](#)

Ces ajouts sont **cohérents avec les comportements exigés** et ne constituent pas des fonctionnalités parasites.

Conclusion de la question 1

La tranche fonctionnelle est intégralement présente ; seule la liste prévisionnelle des fichiers de la spécification n'était pas exhaustive.

2) L'implémentation fait-elle ce qui était prévu ?

Réponse détaillée par fonctionnalité technique

Fonctionnalité demandée	État	Conformité estimée	
Identité exacte <code>locator</code> + <code>turnId</code> + <code>sourceId</code>	✔ Complète	100 %	Contrat strict, clé structurale, propagation ba
<code>streamInstanceId</code> frontend-only par tentative acceptée	✔ Complète	100 %	Généré après le single-flight dans useAiStr
Démarrage synchrone transactionnel	✔ Complet	100 %	Le refus intervient avant toute mutation ; le c
Réservations avant le premier <code>await</code>	✔ Complètes	100 %	Réservation, Activité et son sont alloués sync
Vérification de la webview d'origine exacte	✔ Complète	100 %	Le contexte extension-only transporte <code>HtmlP</code> réservation.
Rejet des anciens messages d'un panneau remplacé	✔ Complet	100 %	Guard webview + locator et tests dédiés à de
Descripteur live strict et détection des doublons	✔ Complet	100 %	Dispatcher brut avec <code>URLSearchParams.g</code>
Frontière React externe pour query invalide	✔ Complète	100 %	Prompt.tsx monte une frontière dédiée ava
Désactivation de l'hydratation classique et de l'auto-run	✔ Complète	100 %	Les queries live valides passent par le bootstr
Buffer raw-live complet, ordonné et non limité	✔ Complet	100 %	Le registre conserve le tableau complet et les
Réservations, époques, retraites et tombstones	✔ Complètes	99 %	Implémentées dans LiveSessionRegistry
Génération destructive et claims opaques	✔ Complètes	100 %	Acquisition, revalidation, éviction par handle
Finalisation post-forwarding en cinq résultats	✔ Complète	100 %	<code>reservationCancelled</code> , <code>handoffReady</code>
Priorité de <code>retiredMissingBaseline</code>	✔ Complète	100 %	Baseline vérifiée avant terminal dans la finali

Fonctionnalité demandée	État	Conformité estimée	
Persistence de chaque <code>meta.error</code> avant <code>append</code>	✓ Complète	100 %	NormalPromptLiveSessionSource.send
Terminal synthétique après <code>throw/EOF</code>	✓ Complet	100 %	Throw produit <code>INTERRUPTED</code> , EOF sans terminaison
Protocole V1 du premier <code>meta.error</code>	✓ Complet	100 %	Header non-200 <code>response: streaming</code> , d
<code>meta.error</code> tardif envoyé dans le <code>body</code>	✓ Complet	100 %	Seul le premier chunk choisit le header ; les s
Détection d'enveloppe Slimbk post-header	✓ Complète	100 %	parseSlimbkStreamBodyError.util.t
FIFO extension unique	✓ Complète	100 %	Chunks ordinaires et terminal V1 sont insérés
Erreurs auxiliaires non court-circuitantes	✓ Complètes	98 %	Activité, son et emit sont essayés séparément
Queue A gardée par tentative exacte	✓ Complète	100 %	Coalescence limitée à une identité <code>sourceId</code>
Queue B gardée par tentative exacte	✓ Complète	100 %	Entrées, suffixes, deferred chunks et continua
Outcome après Queue A, Queue B et <code>commit</code>	✓ Complet	100 %	La finalisation attend les deux queues puis wa
Terminal unique suivi de <code>transportError</code>	✓ Complet	100 %	Aucun terminal synthétique ajouté si termin
Replay <code>replaceTurnAssistants</code>	✓ Complet	100 %	Seuls les assistants du tour exact sont retirés.
Replay <code>continueAssistant</code>	✓ Complet	100 %	Baseline immuable capturée juste avant l'ouv
Retry <code>preserveLocal</code> réellement opaque	✓ Complet	100 %	La branche ne lit ni brouillon, ni attachments
Handoff direct normal Prompt	✓ Complet	100 %	Terminal + outcome completed + stream exac
Panneau coordonné et sérialisation par locator	✓ Complète	100 %	PromptLiveSessionPanelCoordinator
Conservation de l'Activité panneau fermé/source active	✓ Complète	100 %	Projection source/closed conservée, Idle/obse
Idle lorsque le panneau reste ouvert après <code>settlement</code>	✓ Complet	100 %	Transition terminale unique via le runtime Ac
Stop détaché uniquement <code>source/closed</code>	✓ Complet	100 %	Narrowing strict et composant keyé par <code>source</code>
Style exact du bouton Stop	✓ Complet	100 %	<code>SquareStop</code> , <code>small</code> , <code>danger</code> , label <code>Stop</code> , d
Stop normal unique dans le panneau ouvert	✓ Complet	100 %	Aucun Stop de bannière normalPrompt ; le co
Propagation synchrone de <code>stopRequested</code>	✓ Complète	100 %	Runtime Activité mis à jour avant <code>claim.ab</code>
Son exact par run et tentative	✓ Complet	99 %	<code>runRef</code> exact, <code>soundAttemptId</code> , dédup ter
Delete sous claim destructif	✓ Complet	100 %	Stop/settlement avant suppression disque, évic
Retry EasyAgents sous claim	✓ Complet	100 %	Claim maintenu à travers close, éviction, sup
Préservation du comportement EasyAgents	✓ Complète	99 %	<code>sourceId=turnId</code> , observer ownership, de
Composer réarmé pendant replay	✓ Complet	100 %	<code>replayComposerArmed</code> devient vrai après
Send verrouillé pendant replay	✓ Complet	100 %	Surface composer et autorisation Send sont s
Démasquage sans effacement sans <code>fallback ack</code>	✓ Complet	100 %	La finalisation retire le masque sans modifier
Continuation et thinking par <code>streamInstanceId</code>	✓ Complets	99 %	Préparation dédiée, base thinking par tentativ
Programming docs	✓ Complètes	100 %	Les six documents demandés ont été mis à jo
Couverture de tests ciblée	✓ Très	98 %	Tests de races, stale callbacks, terminal+trans

Fonctionnalité demandée

État

Conformité estimée

complète

Conclusion de la question 2

Oui : tous les grands contrats techniques demandés sont présents et les chemins critiques observés suivent l'ordre et les identités exigés.

3) Y a-t-il des bugs ?

Réponse

Je n'ai trouvé aucun bug fonctionnel confirmé dans le code courant. 

Catégorie

Résultat

Bug certain reproductible depuis la lecture du code	0 identifié
Fonction demandée manifestement manquante	0 identifiée
Ancien contrat dangereux toujours actif	0 identifié
Incohérence de typage majeure	0 identifiée
Violation confirmée des identités exactes	0 identifiée
Risque résiduel lié aux courses asynchrones	Élevé par nature, mais fortement testé

Points de vigilance non classés comme bugs

⚠ La spécification sous-estimait les fichiers nécessaires

Le support du header V1 et de l'erreur post-header a nécessité une évolution isolée du framework Slimbk et du générateur client. C'est une **amélioration nécessaire**, mais la liste exacte de fichiers aurait dû les annoncer.

⚠ Queue A et Queue B ont été plus que simplement « annotées »

Leur logique métier principale reste inchangée :

- même ordre FIFO ;
 - même coalescence ;
 - même gating ;
 - même verrouillage ;
 - même replay des éléments différés.
- Cependant, leurs mécanismes internes ont été étendus avec :
- identité de tentative ;
 - générations de reset ;
 - conservation des erreurs ;
 - propagation de l'échec par `whenIdle(...)`.

C'est requis pour satisfaire la finalisation exacte demandée, mais la phrase « aucun changement d'algorithme » de la spécification est trop absolue par rapport à l'implémentation nécessaire.

⚠ Complexité intrinsèque très élevée

Les zones les plus sensibles restent :

1. source terminée pendant une réouverture ;
2. ancien callback après Retry ;
3. terminal consommé suivi d'une erreur transport ;
4. dispose synchronique pendant un open ;
5. Delete pendant réservation ;
6. release de claim pendant cleanup différé ;
7. Queue B verrouillée pendant la finalisation ;
8. source successive réutilisant le même `turnId`.

Les tests couvrent explicitement ces familles de courses, ce qui réduit fortement le risque, sans pouvoir rendre un système distribué entre React, webviews, bus et backend totalement trivial.

Conclusion de la question 3

Aucune correction de bug n'est actuellement justifiée par une preuve statique ; les réserves restantes concernent le risque de concurrence, pas une défaillance démontrée.

4) Quelle est mon opinion sur le code source et la spécification ?

Sur le code source

Note estimée : 94/100. 

Points forts

- séparation nette entre cœur générique et adaptation produit ;
- identités exactes cohérentes de bout en bout ;

- très bon contrôle des callbacks stale ;
- transactions destructives correctement protégées ;
- finalisation centralisée ;
- parsing strict aux frontières ;
- erreurs auxiliaires agrégées sans masquer le chemin principal ;
- commentaires techniques riches et centrés sur les invariants ;
- tests particulièrement nombreux sur les courses difficiles ;
- aucun fallback de compatibilité implicite ajouté ;
- absence des anciennes APIs supprimées.

Le design de `LiveSessionRegistry.service.ts` est robuste : réservations, records, retraits, tombstones, époque de run et génération destructive ont des responsabilités distinctes.

Le frontend est également bien défendu : `useAiStream.hook.ts` n'utilise plus le `turnId` comme identité suffisante et attend correctement les barrières de queue et de commit.

Faiblesses

- la complexité cognitive est très élevée ;
- plusieurs services comportent beaucoup d'états et de transitions ;
- les invariants sont dispersés entre registre, Activité, son, coordinateur, transport et React ;
- le coût de maintenance futur dépend fortement du maintien des programming docs et des tests de courses ;
- certains fichiers restent longs malgré la modularisation.

Cette complexité semble ici **principalement causée par le problème réel**, pas par une surarchitecture gratuite.

Sur la spécification

Note estimée : 92/100. 

La spécification est exceptionnellement précise sur :

- les identités ;
- les ordres de mutation ;
- les barrières ;
- les comportements stale ;
- les chemins destructifs ;
- les erreurs de transport ;
- l'isomorphisme ;
- les tests attendus.

Ses deux faiblesses principales sont :

1. la liste de fichiers n'incluait pas toutes les modifications finalement indispensables ;
2. l'interdiction absolue de changement algorithmique de Queue A/B entre en tension avec les nouvelles exigences de gestion des erreurs et de finalisation.

Conclusion de la question 4

Le code est production-grade et remarquablement défensif, tandis que la spécification est excellente sur les invariants mais légèrement trop rigide et incomplète sur son inventaire technique réel.

5) Résumé du rapport

Dimension vérifiée	Estimation	Verdict	Commentaire
Complétude fonctionnelle	99 %		Aucun besoin utilisateur clairement absent.
Conformité comportementale statique	98 %		Les ordres, identités et transitions critiques correspondent.
Conformité littérale à la liste de fichiers	90 %		189 fichiers réels contre 160 annoncés, principalement des supports nécessaires.
Identités et guards stale	100 %		<code>sourceId</code> , <code>streamInstanceId</code> , <code>webview</code> et générations correctement séparés.
Replay et handoff	99 %		Replay complet, <code>preserveLocal</code> , direct handoff et terminal snapshot présents.
Transport et erreurs	99 %		V1, body error, terminal unique et outcome différé implémentés.
Activité, Stop et son	99 %		Projections exactes, Stop détaché et son par tentative.
Delete et Retry destructifs	99 %		Claims, handles, settlement et éviction durablement ordonnés.

Dimension vérifiée	Estimation	Verdict	Commentaire
Préservation EasyAgents	99 %	✓	Contrat produit historique conservé.
Couverture de tests dans le code	98 %	✓	Très forte couverture des courses et chemins négatifs.
Qualité et maintenabilité	94 %	✓	Architecture propre, mais charge cognitive inévitablement élevée.
Bugs confirmés	0	✓	Aucun défaut démontré par l'analyse statique.
Confiance globale du rapport	93 %	✓	Forte confiance sur les contrats et chemins critiques inspectés.

6) Conclusion en une phrase

La tranche **liveSession** pour Prompt normal est fonctionnellement complète, très fidèle à la spécification et solidement défendue contre les courses critiques, sans bug confirmé nécessitant actuellement une correction.

7) Corrections jugées nécessaires

Aucune correction de code n'est jugée nécessaire sur la base des faits vérifiés. ✓

Le seul ajustement justifiable serait une **correction documentaire de la spécification historique** afin d'y intégrer les fichiers complémentaires Slimbk, Queue A/Queue B et les harnesses de test effectivement nécessaires ; cela ne justifie pas de modifier l'implémentation actuelle.

</ASSISTANT>

SHOW ALLPage 1 / 1