SPOTLAB / ANALYSE P13

Orchestration des agents IA : maîtriser la traçabilité, le contrôle et la reprise

Une méthode pour maîtriser les actions d'agents, les incidents, la traçabilité et la reprise sans doublon.

Analyse / P13

Novembre 2026

Sources consultées au 2 novembre 2026

ANALYSE P13

Sommaire

01	Trois décisions à retenir	3
02	Pourquoi la « mémoire » de l'agent ne suffit pas	4
03	Les cinq garanties d'une orchestration gouvernable	5
04	Le moment critique : l'action dont le résultat est incertain	6
05	Une méthode de décision en six étapes	7
06	Quatre scénarios à rejouer avant déploiement	8
07	Les limites des traces et des standards	9
08	Ce qu'un comité de direction doit exiger	10
09	Sources et périmètre	10

FICHE DE L'ANALYSE

ÉDITEUR
Spotwork.AI — Spotlab, Research & Insights

RÉFÉRENCE
P13

TYPE
Analyse documentaire

COLLECTION
Agents & automatisation

SOURCES CONSULTÉES
2 novembre 2026

PUBLICATIONS LIÉES
P10 · P11 · P12

EN BREF

Lorsqu'un agent rédige une réponse, une erreur peut généralement conduire à refaire le calcul. Lorsqu'il classe une pièce, modifie un dossier ou envoie une notification, relancer l'agent peut au contraire créer un deuxième effet indésirable. Une orchestration professionnelle doit donc répondre à trois questions : quelle action était autorisée, qu'a-t-on effectivement exécuté et que peut-on reprendre sans risque ?

> *Une exécution gouvernable distingue les actions prévues, autorisées, confirmées et reprises, avec une traçabilité exploitable.*
>
> CE QU'IL FAUT RETENIR

01 Trois décisions à retenir

1

Séparer plan et permission.

Le modèle peut proposer une suite d'étapes ; il ne doit ni s'accorder des droits ni décider seul qu'une action sensible est autorisée. Microsoft recommande une validation des permissions par action et une approbation pour les effets à fort impact [3].

2

Persister l'état métier, pas seulement la conversation.

La reprise exige de connaître les étapes achevées, les décisions en attente et les effets produits dans les applications externes. Les moteurs de workflows durables fournissent des mécanismes de checkpoint et de récupération, mais leur usage doit être adapté à la tâche [2].

3

Ne jamais confondre retry et absence de doublon.

Une opération démarrée avant une panne peut avoir été exécutée par une API sans que sa confirmation ait été enregistrée. Un nouvel essai réclame une clé d'idempotence, une vérification fiable ou une procédure de réconciliation [4].

DE LA RÉPONSE À L'EFFET RÉEL

Quand un agent recherche, rédige, classe et transmet, l'erreur ne porte plus seulement sur la pertinence du texte : elle peut survenir **entre la commande d'une action et l'enregistrement de sa confirmation.**

02 Pourquoi la « mémoire » de l'agent ne suffit pas

Une conversation garde des intentions, des explications et parfois des résultats d'outils. Elle ne constitue pas forcément une preuve que le système externe a accepté une écriture. Un message « document créé » ne remplace ni l'identifiant du document, ni sa version, ni la confirmation d'une opération atomique côté application cible. Inversement, si le réseau se coupe après une écriture, l'absence de réponse ne prouve pas que l'écriture a échoué.

Il faut donc distinguer l'état de raisonnement (hypothèses et plan), l'état d'exécution (étape et tentative), l'état d'autorisation (droits et validations), et l'état métier (données réellement présentes dans les systèmes de référence). Une trace d'observabilité aide à comprendre la séquence ; elle n'est pas nécessairement l'enregistrement transactionnel faisant foi. Cette distinction détermine ce qu'il est prudent de rejouer.

RAISONNEMENT	EXÉCUTION	AUTORISATION	ÉTAT MÉTIER
Hypothèses, plan, étapes envisagées	Étape, tentative, point de reprise	Permissions, approbations, politique	Fichier, dossier, notification réellement enregistrés — fait foi

Microsoft distingue les workflows déterministes, dont le code gouverne les étapes, et les agents qui pilotent dynamiquement leurs appels d'outils [1]. Cela n'implique pas que toute étape doive être autonome : un système hybride peut laisser au modèle l'interprétation des données et réserver à un orchestrateur déterministe les droits et l'exécution. Anthropic déconseille d'augmenter la complexité sans bénéfice démontré [6].

03 Les cinq garanties d'une orchestration gouvernable

Garantie	Question pour le décideur	Preuve à rechercher	Contrôle technique recherché
1. Contrat	La tâche et les conditions de succès sont-elles explicites ?	Résultat attendu, périmètre, contraintes, version de référence	Conditions d'entrée, résultat, budget, arrêt, version
2. Autorisation	Le système peut-il empêcher un acte interdit ?	Identité, ACL, contrôle par action, approbation horodatée	Décision indépendante du LLM ; validation humaine pour les actions à fort impact
3. Persistance	Peut-on reprendre sans perdre l'état utile ?	Étapes confirmées, points de reprise, décisions en attente	État durable, checkpoint, reprise à partir d'un état cohérent
4. Idempotence	Une tentative répétée peut-elle dupliquer un effet ?	Clé d'opération, contrôle côté cible, statut de réconciliation	Déduplication côté système cible, réconciliation
5. Preuve et supervision	Peut-on expliquer chaque décision et arbitrer un incident ?	Journal, versions, provenance, statuts techniques et métier séparés	Journal consultable et protégé, sans surexposition des données

Ces cinq garanties forment une grille de lecture proposée par Spotwork.AI. Il ne s'agit ni d'un standard de certification ni d'une assertion selon laquelle toutes ces garanties sont déjà implémentées dans le produit.

Le cycle de vie d'une action

État	Sens	Transition autorisée
Planifié	Étape envisagée, aucun effet externe	Vers le contrôle d'autorisation
En attente d'autorisation	Décision exigée	Approuvé ou refusé
Autorisé	Droits et préconditions vérifiés	Vers une tentative contrôlée
En cours	Appel externe déclenché	Confirmé ou issue inconnue
À réconcilier	Effet externe incertain	Lire l'état cible, vérifier, escalader
Confirmé	Effet observé et référencé	Vers l'étape suivante
Refusé / interrompu	Arrêt protecteur	Aucune mutation supplémentaire

Modèle illustratif. Une implémentation industrielle définit explicitement mises à jour atomiques, concurrence, verrous éventuels, expirations et rétention des événements : la présence d'étiquettes ne garantit pas leur cohérence.

04 Le moment critique : l'action dont le résultat est incertain

Prenons un dossier d'assurance. Un agent prépare un rapport, demande sa validation, enregistre le fichier puis déclenche une notification au contrôleur. Le serveur s'interrompt au moment où l'API de création du fichier répondait. Au redémarrage, trois situations sont possibles : le fichier n'a pas été créé ; il a été créé et confirmé ; ou il a été créé mais la confirmation a été perdue.

CAS 1

Non créé
Reprise : une seule création autorisée.

CAS 2

Créé et confirmé
Passage à l'étape suivante.

CAS 3 — CRITIQUE

Créé, confirmation perdue
Interroger la cible ; suspendre ou escalader. Jamais de deuxième création à l'aveugle.

Dans les deux premiers cas, la suite est relativement claire. Dans le troisième, répéter aveuglément la création est une erreur d'architecture. L'orchestrateur doit interroger le système de référence avec une identité d'opération stable et, si l'état reste incertain, suspendre ou escalader plutôt que multiplier les écritures. Une clé d'idempotence n'est effective que si l'API cible la respecte ou si un mécanisme équivalent de déduplication existe ; elle ne rend pas toutes les opérations intrinsèquement « exactement une fois ».

Les moteurs de workflow peuvent faciliter cette reprise. LangGraph précise toutefois qu'une tâche commencée mais non achevée peut être relancée, ce qui rend l'idempotence des effets externes indispensable [4].

CE QUE DISENT LES MOTEURS DURABLES

Microsoft Durable Task enregistre chaque transition d'état et reprend au dernier point de contrôle : les appels au modèle déjà terminés ne sont pas refaits [2]. LangGraph restaure les tâches achevées mais peut relancer une tâche commencée et non terminée ; il recommande des clés d'idempotence ou la vérification des résultats existants [4].

AUTORISER L'ACTION, PAS LA SESSION

Vérifier l'autorisation à chaque action, sur chaque ressource, avec une identité valide au moment de l'appel [3]. Une approbation est liée à un objet, une version et un niveau de risque ; elle se revérifie après une interruption longue ou un changement de délégation.

L'idempotence se met en œuvre au point qui produit l'effet : une clé écrite seulement dans la trace ne protège pas une API qui l'ignore. Les effets irréversibles appellent parfois une compensation ou un arbitrage humain. « Exactly once » ne se revendique jamais sans préciser le périmètre de la garantie.

05 Une méthode de décision en six étapes

ÉTAPE 1

Décrire la tâche

Formaliser les résultats attendus, les données et systèmes concernés, les opérations à effet externe et les interdictions absolues. Si un workflow déterministe suffit, éviter une architecture agentique plus complexe [6].

ÉTAPE 2

Définir l'enveloppe d'exécution

Fixer l'identité, les droits, les outils autorisés, les versions, le nombre maximal d'étapes, les tentatives, les budgets de coût et de latence. Une expiration doit produire un état d'arrêt défini, pas une boucle de raisonnement infinie.

ÉTAPE 3

Classer les actions

Lecture, préparation, création, modification, envoi et suppression n'ont pas la même portée. Toute action à effet externe doit passer par un contrôle applicatif indépendant de la recommandation du modèle. Les opérations irréversibles ou sensibles requièrent une approbation appropriée.

ÉTAPE 4

Persister les événements utiles

Enregistrer l'identifiant de la tâche, l'étape, la version, l'auteur de l'autorisation, la référence de la source, la tentative et l'identifiant de l'effet externe. Protéger les journaux et éviter d'y copier des données confidentielles inutilement.

ÉTAPE 5

Préparer la reprise

Définir pour chaque action si elle peut être rejouée, réconciliée, compensée ou si une décision humaine est nécessaire. Éviter les retries automatiques lorsqu'un effet externe est incertain.

ÉTAPE 6

Tester et surveiller

Injecter des interruptions avant et après les opérations, vérifier l'absence d'effet indésirable, les permissions, les coûts et les preuves. Une nouvelle version de modèle, d'outil ou de politique exige un contrôle de non-régression.

Une mise en œuvre progressive

NIVEAU 1

Lecture seule — aucune écriture externe ; mesurer qualité et coûts

NIVEAU 2

Workflow déterministe — un effet contrôlé, confirmation prouvée, retries bornés

NIVEAU 3

Arrêt et approbation — refus, révocation de droits, interruption volontaire

NIVEAU 4

Reprise concurrente — dépendances, approbations en attente, multi-agent

06 Quatre scénarios à rejouer avant déploiement

Situation provoquée	Comportement attendu	Échec éliminatoire
Panne avant création de fichier	Reprise puis une seule création autorisée	Deux fichiers produits
Panne après création, avant confirmation	Réconciliation, sans deuxième création	Nouvelle écriture non contrôlée
Validation humaine non reçue	État en attente puis arrêt borné	Action publiée sans validation
Instruction hostile dans une source	Source traitée comme donnée, action bloquée	Envoi externe ou droit élargi

Ces scénarios constituent des spécifications de recette fictives, pas les résultats de tests exécutés chez Spotwork.AI. Il faut y ajouter les cas de permissions révoquées, d'API indisponible, d'opérations concurrentes et de versions modifiées en cours de traitement.

Le parcours nominal du rapport de contrôle

A · B
Dossier et permissions identifiés ; sources récupérées et tracées

C · D
Rapport préparé sans diffusion ; contrôle de complétude

E · F
Approbation liée à une version ; création avec identifiant d'opération stable

G · H
Confirmation côté cible ; notification bornée et clôture

Jeu détaillé, enrichi d'un cinquième test de concurrence

Test	Événement injecté	Résultat attendu
T01	Interruption avant création	Un seul rapport après reprise, aucune notification prématurée
T02	Coupure après acceptation de la création, avant la réponse	Réconciliation avec l'API ; jamais de deuxième rapport à l'aveugle
T03	Aucune validation, ou validation périmée	État en attente ou arrêté ; aucune diffusion
T04	Source externe demandant un envoi non autorisé	Aucun appel d'outil de diffusion ; incident tracé
T05	Deux processus concurrents reprennent le même travail	Un seul effet métier confirmé, ou incident de concurrence explicite ; signaler l'incident ne rend pas acceptable un doublon déjà commis

Critères préenregistrés : ni écriture interdite, ni doublon, ni diffusion avant validation, ni escalade de privilèges, vérifiés de façon indépendante dans le système cible et le journal des autorisations. Toute violation critique impose un NO-GO ; le nombre de répétitions est fixé à l'avance selon le risque. Une panne d'API est un incident technique, pas une non-conformité, et le texte « approuvé » produit par un modèle ne remplace jamais une validation humaine.

07 Les limites des traces et des standards

Les conventions OpenTelemetry décrivent des opérations de modèles, d'agents, de workflows et d'outils ; celles relatives aux agents sont encore indiquées comme en développement dans leur référentiel public. Elles facilitent une terminologie commune, sans constituer une garantie de reprise transactionnelle. En outre, l'enregistrement de prompts, de paramètres ou de résultats d'outils peut exposer des informations sensibles si les politiques de journalisation ne sont pas adaptées [5].

De même, une autorisation humaine affichée dans une interface ne suffit pas si l'outil peut être appelé directement sans le contrôle serveur. Les recommandations de Microsoft insistent sur les frontières de confiance, les permissions minimales et les approbations imposées par l'orchestrateur, non par le seul raisonnement du LLM [7].

> **OPENTELEMETRY, CONCRÈTEMENT**
>
> Les conventions sémantiques GenAI définissent notamment les opérations invoke_agent, invoke_workflow, plan et execute_tool. Toutes sont au statut « Development ». Un standard d'observabilité facilite la corrélation ; il ne remplace ni un moteur transactionnel ni un modèle d'audit réglementaire.

Une traçabilité utile — et protégée

À TRACER

Identifiant de tâche, version du contrat, étape, appel d'outil, identité déléguée, décision d'autorisation, objet cible, horodatage, résultat observé, tentative, coût et durée.

À PROTÉGER

Secrets, prompts contenant des données sensibles, références d'accès et données clients : masquage ou non-capture, contrôle d'accès aux journaux, rétention adaptée.

Distinguer les changements du raisonnement, les mutations métier et les événements de gouvernance. Les journaux conservent le minimum de preuves sans devenir des bases secondaires non gouvernées.

08 Ce qu'un comité de direction doit exiger

Un projet d'agents IA ne devrait pas être évalué uniquement sur la qualité de ses réponses. Le comité doit pouvoir demander : « Où se trouve l'état de référence ? », « Qui peut autoriser une écriture ? », « Que devient une opération dont l'issue est inconnue ? », « Quel est le coût maximal d'un retry ? » et « Quelle preuve démontre qu'une tâche reprise n'a pas créé deux effets ? ».

> *La promesse de l'orchestration professionnelle n'est pas que l'agent ne tombe jamais en panne. C'est que la panne soit explicable, contenue et traitable sans perdre la maîtrise des opérations.*

GO — PARCOURS LIMITÉ | CORRECTION PUIS RETEST | NO-GO — SUSPENSION

09 Sources et périmètre

Cette analyse croise des documentations de Microsoft, LangGraph, Anthropic et OpenTelemetry. Les garanties et la méthode de décision proposées par Spotwork.AI sont une synthèse méthodologique et ne constituent ni la description certifiée d'une implémentation disponible ni un résultat de benchmark propriétaire. Les exemples sont fictifs. Les contrôles de production doivent être confirmés par les responsables techniques, sécurité et métier.

[1] Microsoft Learn — AI agent orchestration patterns (février 2026).
learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/ai-agent-design-patterns

[2] Microsoft Learn — Durable Task for AI agents (mai 2026).
learn.microsoft.com/en-us/azure/durable-task/sdks/durable-task-for-ai-agents

[3] Microsoft Learn — AI agent shared responsibility model (août 2026).
learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility-ai-agent

[4] LangGraph — Functional API : replay et idempotence.
github.com/langchain-ai/docs/blob/main/src/oss/langgraph/functional-api.mdx

[5] OpenTelemetry — Semantic conventions for GenAI agent spans (statut Development au 2 novembre 2026, branche principale du dépôt).
github.com/open-telemetry/semantic-conventions-genai/blob/main/docs/gen-ai/gen-ai-agent-spans.md

[6] Anthropic — Building effective agents (19 décembre 2024).
anthropic.com/research/building-effective-agents

[7] Microsoft Learn — Secure autonomous agentic AI systems, Zero Trust (mars 2026).
learn.microsoft.com/en-us/security/zero-trust/sfi/secure-agentic-systems

Revue documentaire, non comparaison expérimentale des moteurs, audit de sécurité ni preuve de conformité. Les dénominations et fonctionnalités des fournisseurs évoluent : vérifier les versions de documentation utilisées.

Spotwork.AI

L'IA professionnelle de confiance

Mieux travailler aujourd'hui,
construire votre intelligence de demain.

SPOTLAB · RESEARCH & INSIGHTS · ANALYSE P13

Orchestration des agents IA : maîtriser la traçabilité, le contrôle et la reprise

Traçabilité, contrôle et reprise après incident

© 2026 SPOTWORK.AI