

Gouvernance de l'IA : ce qui peut dérailler entre l'approbation et l'exécution

Par Stephan Pochet

La gouvernance de l'IA exige des contrôles efficaces depuis les informations sources jusqu'à l'approbation et à l'exécution. Cet essai examine la validation humaine, les données, les changements du système, les autorisations et la réponse aux incidents.

Un assistant prépare un paiement à un fournisseur. Une personne vérifie la facture et donne son accord. Avant l'exécution du paiement, les coordonnées bancaires du fournisseur changent.

L'approbation reste-t-elle valable ?

Cette question doit être réglée dès la conception du processus. Si elle surgit pour la première fois au cours d'une enquête, l'organisation tente déjà de reconstituer une décision qu'elle aurait dû maîtriser.

La gouvernance de l'IA prend souvent une forme très concrète dans ce genre de situation. Une politique désigne un responsable. Une évaluation décrit les risques. Pourtant, une modification ordinaire entre la validation et l'exécution peut révéler une faille qu'aucun de ces documents ne résout.

Cinq domaines de travail permettent de réduire ce risque : la validation humaine, la gouvernance des données, le suivi du cycle de vie, les autorisations effectivement appliquées et la surveillance. Ces sujets sont connus. La difficulté consiste à les faire fonctionner ensemble sur une même opération.

Partir de l'action

L'exemple du paiement à un fournisseur est fictif, mais il offre un point de départ utile. Son objectif est identifiable et son résultat compte au-delà de l'application. L'argent parvient au compte prévu, ou il n'y parvient pas. L'organisation doit répondre de ce résultat, même si plusieurs systèmes y ont contribué.

Commencez par délimiter le travail de l'assistant. Préparer une recommandation est une activité. Transmettre des instructions à un service de paiement en est une autre. Une équipe qui regroupe les deux sous le terme d'assistance au paiement peut manquer le moment où l'application obtient le pouvoir d'agir sur la situation d'autrui.

Cette distinction modifie aussi les preuves nécessaires à l'approbation. Un assistant de rédaction utile peut commettre des erreurs qu'une personne corrige avant toute action. Un système autorisé à exécuter doit disposer de contrôles portant sur l'action elle-même. Une

réponse exacte dans un jeu de tests ne prouve pas qu'il respectera les autorisations d'un utilisateur lorsqu'il sera connecté à un compte réel.

Suivez également le travail entre les services. L'équipe applicative peut maintenir l'assistant tandis que la direction financière contrôle les fiches fournisseurs. Une autre équipe peut gérer les identités. Chacune peut accomplir correctement sa mission tout en laissant une faille entre les services. L'examen sert à trouver ces failles tant qu'il est encore possible de les traiter de manière réfléchie.

Un inventaire devient utile lorsqu'il permet de trouver la personne capable de répondre à la prochaine question. S'il ne mentionne qu'un service, cherchez qui peut autoriser une modification ou arrêter une opération. Une personne doit être joignable lorsque le processus rencontre une situation que sa conception n'avait pas prévue.

Délimiter précisément l'approbation

Dans notre exemple, la personne qui valide doit savoir exactement ce qu'elle approuve : le fournisseur, le montant, le compte destinataire et les pièces justifiant le paiement. Le système doit déterminer quelles modifications rendent cet accord caduc.

Faute de quoi, une personne peut approuver une opération et se retrouver tenue pour responsable d'une autre.

Elle doit aussi disposer du temps et des informations nécessaires pour contester la recommandation. Une file de demandes urgentes peut transformer la validation en une succession de clics. Les équipes devraient vérifier, dans des conditions de travail réalistes, si les personnes chargées du contrôle repèrent un montant erroné ou une modification injustifiée du fournisseur.

Testez ensuite un refus. Vérifiez que l'exécution s'arrête réellement, y compris pour les opérations déjà en attente dans un système connecté. Un rejet consigné ne sert guère si le paiement part malgré tout.

Solliciter une personne a un coût

La validation humaine a un coût qu'il faut reconnaître. Chaque demande mobilise de l'attention. Si l'application exige une confirmation pour des étapes courantes et facilement réversibles, les personnes chargées du contrôle risquent d'être moins attentives lorsqu'arrive la demande importante. La conception doit distinguer ces situations selon le risque de l'action.

Examinez ce que la personne peut établir indépendamment. Présenter une explication produite par l'IA à côté d'une recommandation produite par la même IA peut donner une impression de corroboration alors que les deux reposent sur la même source erronée. Donnez accès au dossier d'origine et rendez visibles les divergences non résolues. La personne ne devrait pas découvrir après l'approbation que les pièces étaient incomplètes.

La responsabilité doit aussi avoir une limite pratique. Une personne ne peut pas accepter de manière éclairée toutes les conséquences possibles d'un système qu'elle ne peut pas examiner. Définissez la décision dont elle répond et les contrôles effectués ailleurs. L'approbation devient alors utile à celle qui la donne comme à celui qui examinera le dossier ultérieurement.

En cas d'indisponibilité, une demande importante doit suivre une procédure de remplacement définie. Elle peut rester en attente ou être transmise à une autre personne habilitée. Le choix dépend du processus, mais le silence ne doit pas devenir implicitement un accord. L'organisation doit décider du délai acceptable avant que la pression n'impose une décision informelle.

La source compte autant que la réponse

Un assistant peut trouver les coordonnées d'un fournisseur dans une facture ou les récupérer dans un document partagé. Ces sources ne font pas nécessairement autorité au même titre.

Il faut décider quel dossier fait foi pour le paiement et qui peut le modifier. Cette décision de gouvernance des données a une conséquence opérationnelle immédiate. Si la facture contredit la fiche fournisseur approuvée, le processus doit prévoir comment résoudre cet écart.

Le problème se retrouve dans des applications dont les conséquences paraissent moins importantes. Un assistant interne peut répondre avec assurance à partir d'un manuel remplacé le mois précédent. Supprimer l'ancien fichier du dossier partagé peut laisser une copie dans l'index de recherche. La correction n'est achevée que lorsque l'équipe a vérifié que l'application ne l'utilise plus.

Responsabilité, accès, actualité et procédure de correction sont ici essentiels. Ils déterminent si la personne chargée de la validation peut se fier aux informations reçues.

La correction doit parvenir jusqu'à l'application

La responsabilité des données comprend la capacité à corriger les erreurs. Supposons que la direction financière confirme qu'une fiche fournisseur est fausse. L'équipe doit retrouver l'endroit où l'application l'a obtenue et déterminer si d'autres recommandations en attente reposent sur les mêmes informations. Corriger le dossier de référence marque le début du travail.

Les copies compliquent la réponse. Un cache peut continuer à fournir une ancienne valeur. Un index de recherche peut être actualisé selon un calendrier. Une demande d'approbation en attente peut contenir une recommandation antérieure à la correction. L'équipe doit décider quels éléments invalider et vérifier que la correction les a bien atteints.

L'accès pose un problème comparable. Un salarié peut perdre le droit de consulter un document alors que l'index de l'assistant continue d'en révéler le contenu. Contrôler

l'accès uniquement à l'ouverture de la conversation ne permet pas de savoir si chaque élément récupéré est autorisé. Testez avec des comptes ayant des responsabilités différentes et incluez un changement de fonction.

Gardez une enquête proportionnée. Il n'est pas nécessaire de réunir toutes les informations disponibles sur un utilisateur pour établir quel document a fondé une recommandation. Recueillez ce qui permet de comprendre et de corriger la décision, avec des restrictions adaptées. Une piste d'audit qui diffuse inutilement des informations confidentielles crée un nouveau problème de gouvernance.

L'approbation porte sur un système précis

Le nom d'un modèle ne renseigne l'auditeur que partiellement. Les instructions de l'application, ses sources documentaires, ses outils et ses autorisations déterminent aussi ce qu'elle peut faire.

Supposons qu'un assistant ait été approuvé pour préparer des instructions de paiement. Une version ultérieure ajoute un outil permettant de les transmettre pour exécution. Le modèle peut rester identique, mais les pouvoirs de l'application ont changé. Cette modification doit être portée à la connaissance des personnes chargées de l'évaluation et de l'approbation.

Un inventaire identifie l'application et son responsable. Un registre permet de suivre les versions des modèles. Les dossiers de déploiement relient ces éléments à la configuration réellement en service. Ensemble, ils doivent permettre d'établir ce qui a été approuvé et si le système a respecté ces conditions.

Il existe des limites. Un prestataire peut ne pas fournir assez d'informations sur les versions pour reconstituer tous les aspects d'un résultat antérieur. Consignez cette limite. Ne promettez pas une piste d'audit que le service ne permet pas de produire.

Déterminer quels changements imposent un nouvel examen

Le processus de mise en production doit préciser quand revenir devant les responsables de la gouvernance. Remplacer un modèle de fondation est un cas évident. Élargir la population d'utilisateurs peut compter tout autant. Il en va de même pour l'ajout d'un dépôt documentaire ou l'autorisation de fonctionner sans présence humaine.

Évaluez ces changements selon les actions qu'ils permettent et les personnes concernées. Une version logicielle mineure peut modifier considérablement les pouvoirs du système. À l'inverse, une mise à jour de maintenance peut laisser intacts l'usage approuvé et les limites des contrôles. Traiter tous les changements de la même manière risque de saturer les personnes chargées de l'examen sans renforcer leur attention aux changements importants.

Consignez la décision et son fondement. Une enquête ultérieure doit pouvoir distinguer une modification évaluée et acceptée d'une modification passée inaperçue. Cela aide aussi l'équipe qui reprendra l'application. Elle doit comprendre pourquoi une restriction existe

avant de décider de la supprimer.

Le retour à une version antérieure mérite une répétition. Rétablir un ancien modèle ne reconstitue pas nécessairement l'ancienne application si la base fournisseurs, les autorisations ou le service connecté ont changé. Le plan de reprise doit préciser ce qui peut être restauré sans risque et ce qui exige une procédure manuelle. Un bouton de retour arrière est une promesse à mettre à l'épreuve.

Faire respecter la règle là où elle peut être enfreinte

Si l'assistant ne doit jamais modifier le compte bancaire d'un fournisseur, bloquez cette action dans le système connecté. Une phrase dans ses instructions ne constitue pas un contrôle d'autorisation suffisant.

Les recommandations de l'OWASP sur l'autonomie excessive sont concrètes : limiter les fonctionnalités et les permissions disponibles, puis faire appliquer les autorisations par les systèmes en aval. La validation humaine constitue un contrôle supplémentaire, notamment pour les actions à fort impact. Chaque mesure répond à un besoin distinct. [1]

Pour le paiement, testez les possibilités de contourner le processus prévu. Un autre outil peut-il effectuer la même modification interdite ? Un compte administrateur échappe-t-il à la restriction ? Que se passe-t-il si le service d'approbation est indisponible ?

Les exceptions demandent également de l'attention. Chaque autorisation temporaire doit avoir un responsable qui en consigne le motif. Fixez sa date d'expiration dès son attribution. Sinon, le contrôle suivant risque de révéler qu'une solution d'urgence est devenue le fonctionnement habituel.

Les exceptions révèlent le fonctionnement réel

Imaginez qu'un paiement légitime soit bloqué à plusieurs reprises. La direction financière doit le réaliser, l'équipe applicative subit une pression et un administrateur peut élargir les accès en quelques minutes. Le raccourci technique est simple. La question de gouvernance porte sur ce que deviendra cet accès une fois le problème immédiat résolu.

Le dossier d'exception doit permettre à la personne suivante de comprendre sa portée et la condition qui y mettra fin. Celle qui l'accorde doit également savoir quelles protections restent en place. Si une restriction disparaît entièrement, qualifier le changement de temporaire ne réduit pas l'exposition pendant sa durée.

Examinez aussi les exceptions récurrentes. Des demandes répétées pour le même contournement peuvent indiquer que le processus approuvé ne répond pas à un besoin légitime. Il peut être nécessaire de le repenser et de l'évaluer correctement. Des utilisateurs dépendant de dérogations répétées rendent la politique officielle moins utile pour décrire le travail réel.

Certaines exceptions doivent être refusées. Si l'équipe ne peut pas établir qui recevra un paiement ou faire respecter l'autorisation requise, passer par l'assistant peut être

inapproprié. Une procédure manuelle peut maintenir l'activité pendant la résolution du problème technique. Elle doit, elle aussi, avoir un responsable.

Donner une destination utile à l'alerte

La surveillance doit aider quelqu'un à décider. Une alerte signalant une tentative d'utilisation non autorisée d'un outil doit parvenir à une personne capable d'enquêter et, si nécessaire, de restreindre ou d'arrêter le processus.

Les journaux et les traces peuvent établir quels outils ont été appelés, leurs réponses et la configuration en service. Ils ne révèlent pas automatiquement le raisonnement interne du modèle. Ils peuvent aussi contenir des informations confidentielles : les éléments de preuve doivent donc avoir leurs propres règles d'accès et de conservation.

Notre guide de l'observabilité de l'IA, en anglais constitue un point de départ. Le test opérationnel est simple : provoquer une défaillance maîtrisée, puis suivre sa détection, son escalade, son endiguement et le rétablissement du service. Faites participer la personne qui assure le remplacement en cas d'absence ou intervient en dehors des horaires habituels.

La fonction MANAGE du cadre de gestion des risques de l'IA du NIST comprend la surveillance continue, la réponse et le rétablissement. Ces objectifs doivent se traduire par des responsabilités que les personnes concernées peuvent effectivement exercer. [2]

Déterminer ce qu'un signal permet de prouver

Une mesure de surveillance doit être interprétée. Une hausse des appels d'outils rejetés peut signaler une tentative d'utilisation abusive. Elle peut aussi résulter d'un changement légitime que les règles d'autorisation n'ont pas pris en compte. Le signal justifie une enquête ; il n'en détermine pas l'explication.

Un tableau de bord calme peut tromper pour une autre raison. L'application a peut-être cessé d'envoyer des événements. Vérifiez le dispositif de surveillance lui-même pour que l'absence d'alerte ne soit pas prise pour la preuve d'un bon fonctionnement. Lorsque c'est possible, comparez les traces de l'assistant à celles de l'action dans le système connecté.

La réponse doit prendre en compte les opérations achevées. Arrêter l'assistant empêche certaines activités futures, mais n'annule pas un message externe et ne récupère pas un paiement déjà envoyé. La procédure d'incident doit distinguer l'endiguement de la correction et désigner qui évaluera les conséquences pour les personnes ou les dossiers concernés.

La gouvernance devient alors une responsabilité continue. L'équipe apprend quelque chose sur une défaillance et doit décider ce qui change dans le processus. Il peut s'agir d'une limite d'autorisation, des pièces présentées à la validation ou des conditions imposant un nouvel accord. Clore l'incident sans examiner ces changements laisse la même voie ouverte à une nouvelle défaillance.

Rendre les preuves compréhensibles pour autrui

Un dossier de gouvernance doit être compréhensible par une personne absente au moment de la décision. C'est un test exigeant. Les équipes s'appuient souvent sur un savoir partagé qui n'atteint jamais le dossier : une limite évoquée en réunion, une hypothèse sur les utilisateurs habilités ou la promesse de laisser une fonction désactivée.

Consignez les conditions importantes pour l'approbation. Si l'assistant peut rédiger mais pas exécuter, formulez-le de manière vérifiable par l'équipe de déploiement. Identifiez les preuves de cette limite et ce qui conduirait à la réexaminer. Le dossier doit aider à poser une question précise sur le système en service.

Évitez d'en faire une archive de toute la production du projet. Un dossier volumineux peut masquer l'absence de décision claire. Organisez les preuves autour de l'usage approuvé et rendez leurs limites faciles à trouver. Conservez les résultats des tests avec assez de contexte pour comprendre ce qui a été testé et ce qui était hors périmètre.

Cette discipline aide lors des changements d'équipe. Un nouveau responsable doit pouvoir découvrir pourquoi un contrôle existe sans reconstituer l'historique du projet à partir des messages. Il doit également repérer une hypothèse devenue fausse. La gouvernance se fragilise lorsque la maintenance dépend du souvenir de la personne qui connaissait la réponse l'année précédente.

Suivre une décision jusqu'au bout

Un examen utile peut commencer par une seule opération. Suivez-la depuis la pièce d'origine jusqu'à la recommandation, l'approbation, l'exécution et l'enregistrement final. Modifiez un élément important en cours de route. Rendez un composant indisponible. Demandez à la personne chargée de la validation de refuser.

Cet exercice fait souvent apparaître des questions que des contrôles séparés laissent de côté : qui tranche entre des dossiers contradictoires, quelles modifications exigent un nouvel accord et qui peut intervenir lorsqu'une alerte arrive.

Le paiement est un exemple. La même démarche peut s'appliquer à un assistant qui clôt un dossier, modifie une fiche client ou envoie un message à l'extérieur, avec des contrôles proportionnés aux conséquences.

L'organisation doit pouvoir expliquer comment l'action a eu lieu et pourquoi elle était autorisée. Surtout, les personnes qui font fonctionner le processus doivent pouvoir empêcher une action qu'elles ont décidé d'interdire.

Sources

[1] OWASP : Excessive Agency.

[2] NIST : AI Risk Management Framework Core.

Les scénarios et les méthodes de vérification sont des exemples de mise en œuvre proposés dans cet article.

Questions et réponses

Qu'est-ce qui rend la validation humaine efficace ?

La personne doit pouvoir examiner une action précise, disposer de pièces fiables et de temps, et avoir le pouvoir de la refuser. Les modifications importantes exigent un nouvel examen et le refus doit empêcher l'exécution.

Quelle différence entre inventaire de l'IA et registre de modèles ?

L'inventaire identifie les applications, leurs finalités et leurs responsables. Le registre suit les versions des modèles. Les dossiers de déploiement relient ces éléments à la configuration en service.

Où appliquer les autorisations d'une application d'IA ?

Le système connecté doit faire respecter les autorisations de l'action demandée. Les seules instructions données au modèle ne constituent pas une limite d'autorisation suffisante.

Les journaux expliquent-ils le raisonnement interne du modèle ?

Non. Ils peuvent consigner les actions observables et la configuration, sans révéler automatiquement le raisonnement interne. Les preuves nécessitent aussi des règles d'accès et de conservation.