La gouvernance de l'IA n'est pas une politique. C'est une architecture.

Les frameworks de conformité sont nécessaires. Ils ne sont pas suffisants pour rendre un système régulé structurellement gouvernable.

Twingital Institute / Jérôme Vetillard / Avril 2026

Introduction

Un paradoxe traverse depuis plusieurs années les comités exécutifs du secteur healthcare & life sciences. D'un côté, une grande majorité d'organisations déclarent avoir engagé des démarches structurées de gouvernance de l'IA. De l'autre, une proportion tout aussi significative exprime un sentiment persistant d'insuffisance face aux exigences concrètes de pilotage, de traçabilité, d'audit et de responsabilité associées aux systèmes qu'elles déploient. Des enquêtes récentes de marché et d'écosystème documentent ce décalage de manière récurrente : la DiMe Society a relayé début 2026 qu'une large majorité des dirigeants interrogés considéraient encore l'insuffisance de guidelines comme un obstacle important à l'adoption de l'IA, tandis que le rapport 2026 de Larridin signalait à la fois un faible niveau d'inventaire complet des applications IA, l'absence fréquente d'un cadre de gouvernance opérationnel clair, et un niveau élevé de confiance déclarée dans l'impact de l'IA. Ces enquêtes documentent un malaise structurel.

Ce paradoxe n'est pas purement conjoncturel. Il révèle une confusion rarement formulée avec assez de netteté : la confusion entre politique de gouvernance et architecture de gouvernance.

Ces deux objets ne relèvent pas du même registre. Ils ne correspondent pas à deux degrés de maturité sur un continuum unique. Ce sont deux programmes distincts, qui adressent des problèmes distincts, avec des instruments distincts. La politique de gouvernance organise des responsabilités, des règles, des procédures, des évaluations et des mécanismes d'escalade. Elle encadre le système depuis l'extérieur. L'architecture de gouvernance, elle, définit les propriétés structurelles qui rendent certains comportements traçables, bornés ou inacceptables par construction. Elle conditionne le système depuis l'intérieur.

La thèse de cet article est la suivante : dans les systèmes d'IA opérant en environnement régulé à fort enjeu décisionnel individuel, les frameworks procéduraux de gouvernance sont nécessaires mais structurellement insuffisants. Ils permettent de documenter, d'évaluer, de surveiller et d'organiser la gestion du risque. Ils ne garantissent pas, à eux seuls, que le système déployé soit traçable par construction, borné dans son domaine de validité, ni que la frontière entre automatisation et décision humaine soit maintenue sous pression opérationnelle réelle. Lorsque ces propriétés ne sont pas intégrées à la conception, la gouvernance reste rétrospective, partielle et vulnérable aux contournements que toute organisation sous tension produit mécaniquement.

Cette thèse a un domaine de validité explicite. Elle concerne les systèmes d'IA mobilisés dans des contextes où une décision individuelle peut avoir une portée clinique, réglementaire, économique ou organisationnelle significative : dispositifs médicaux logiciels, aide à la décision diagnostique ou pronostique, prédiction toxicologique réglementaire, triage, allocation de ressources de santé. Elle ne

vaut pas avec la même intensité pour les systèmes à faible enjeu individuel, pour lesquels une gouvernance principalement procédurale peut, dans certains cas, suffire.

I. Clarification terminologique

La première exigence est de stabiliser les termes, puisque le problème commence précisément par leur confusion.

J'appelle gouvernance procédurale l'ensemble des règles, comités, référentiels, évaluations, checklists et processus documentaires appliqués à un système d'IA par des acteurs qui l'observent, l'évaluent et en encadrent l'usage depuis l'extérieur de sa structure. Elle est indispensable. Elle intervient principalement comme dispositif d'encadrement et de contrôle, après que l'architecture du système a été fixée.

J'appelle gouvernance architecturale l'ensemble des propriétés structurelles d'un système qui rendent son comportement traçable, son périmètre de validité explicitable, et ses modes d'action contrôlables au niveau même de son fonctionnement. Elle n'est pas ajoutée au système après conception. Elle est inscrite dans ses patterns de composition, ses flux, ses invariants, ses conditions d'exécution et ses mécanismes d'enregistrement.

Deux distinctions supplémentaires sont nécessaires.

La première oppose auditabilité rapportée et auditabilité native :

- Par auditabilité rapportée, j'entends la capacité à reconstituer après coup une décision à partir d'artefacts produits autour du système : logs applicatifs, métadonnées, outils d'observabilité, approximations locales d'explicabilité.
- Par auditabilité native, j'entends la propriété d'un système dont le fonctionnement ordinaire produit lui-même les éléments nécessaires à la reconstitution fidèle du contexte d'exécution : données engagées, transitions d'état pertinentes, versions des composants, régime décisionnel appliqué.

Il faut ici introduire une nuance que l'on ne trouve pas assez souvent dans la littérature sur ce sujet. L'auditabilité native ne signifie pas explicabilité complète du mécanisme interne d'un modèle statistique. Ce serait une prétention intenable dès que le modèle atteint une complexité non triviale. Elle signifie traçabilité constitutive du contexte, des entrées, des sorties, des transitions et du régime d'exécution de la décision. Elle améliore de façon décisive la gouvernabilité d'un système. Elle ne supprime pas, à elle seule, les difficultés d'intelligibilité des mécanismes inférentiels eux-mêmes. Ce point mérite d'être posé d'emblée, pour ne pas confondre deux problèmes distincts que l'industrie superpose volontiers lorsqu'elle veut esquiver l'un en feignant de résoudre l'autre.

La seconde distinction est celle-ci : un système que l'on peut inspecter n'est pas encore un système qui se rend structurellement inspectable. Les frameworks actuels renforcent surtout les capacités d'inspection. La question posée ici est différente, et plus exigeante : comment concevoir des systèmes dont le fonctionnement normal rend cette inspection fiable, continue et techniquement ancrée ?

II. Pourquoi la gouvernance procédurale domine, et pourquoi cela ne suffit plus

La prévalence actuelle de la gouvernance procédurale n'est ni accidentelle ni absurde. Elle résulte d'une histoire industrielle et réglementaire cohérente.

Les grands cadres de gouvernance de l'IA, qu'il s'agisse du NIST AI Risk Management Framework, de la norme ISO/IEC 42001 ou du corpus d'obligations graduées introduit par le règlement européen sur l'IA, ont été construits selon un principe largement partagé de neutralité technologique. Ce choix est compréhensible du point de vue réglementaire : figer dans un texte juridique ou normatif un unique pattern d'architecture risquerait de verrouiller l'innovation dans un modèle de conception daté. Il est donc logique que ces textes imposent des obligations fonctionnelles, documentaires et organisationnelles sans prescrire un design système particulier. Le NIST AI RMF, par exemple, est explicitement structuré comme un cadre de gestion du risque autour des fonctions Govern, Map, Measure et Manage. L'AI Act, lui, prévoit une application échelonnée selon les catégories d'obligations, avec application générale le 2 août 2026 et calendrier distinct pour certaines catégories, notamment les systèmes à haut risque intégrés à des produits régulés, qui bénéficient d'une transition plus longue.

Ce choix a eu pour effet, dans de nombreuses organisations, de conforter une lecture selon laquelle la gouvernance pouvait être ajoutée après conception, par superposition procédurale et documentaire, sur des systèmes conçus d'abord pour la performance fonctionnelle ou la rapidité de livraison. Puisque le texte ne dit pas comment construire, il devenait tacitement permis de construire d'abord, puis d'ajouter la gouvernance par couches successives. C'est là que se forme le biais structurant.

À ce biais réglementaire s'ajoute un biais culturel. Une grande partie de l'ingénierie logicielle contemporaine a été façonnée par des logiques d'itération rapide et de correction incrémentale. Cette culture est efficace dans des contextes où l'erreur est tolérable, réversible, et dissociée d'un enjeu individuel fort. Elle devient fragile lorsque l'IA intervient dans une chaîne de décision dont les effets doivent pouvoir être justifiés, bornés et audités de manière opposable.

Le paradoxe relevé en introduction prend ici son sens précis. Ces organisations ne manquent pas de politiques. Elles déploient des systèmes qui ne sont pas conçus pour être gouvernés par ces politiques. La gouvernance court après l'architecture, et cette course ne se gagne pas par accumulation de documents.

III. Trois insuffisances structurelles d'une gouvernance purement procédurale

3.1 La traçabilité rétrospective est une reconstruction, pas une preuve

Dans un pipeline ML standard, l'output d'une inférence n'est pas, par défaut, accompagné d'une reconstitution complète et immuable du contexte opératoire qui a rendu cette inférence possible. Ce qui est généralement conservé est une combinaison variable d'inputs, d'outputs, d'horodatages, d'identifiants de version et de logs applicatifs. Ce matériau est utile. Mais il ne garantit pas une reconstitution complète, stable et juridiquement robuste de la décision individuelle.

Lorsque l'on cherche à comprendre après coup une décision particulière, deux familles d'outils apparaissent. La première relève de l'explicabilité post hoc. Des méthodes comme SHAP ou LIME peuvent fournir des indications utiles sur des contributions locales ou sur certaines régularités de comportement du modèle. Mais elles ne sont pas la décision elle-même. Leur intérêt est réel ; leur statut probatoire, lui, doit être manié avec précision. Le problème n'est pas qu'elles seraient inutiles. Le problème est qu'elles sont trop souvent mobilisées comme substitut à une traçabilité que le système n'a pas été conçu pour produire.

La seconde famille relève de l'audit des traces disponibles. Mais un log ne contient jamais autre chose que ce que le système a été conçu pour produire. Dans un environnement fortement régulé, la

distinction entre reconstitution fidèle du contexte d'exécution et approximation interprétative n'est pas une subtilité académique. Elle conditionne la robustesse de l'audit, la qualification d'un incident, et parfois la crédibilité même du dispositif de conformité face à un régulateur qui sait poser les bonnes questions.

3.2 La gouvernance par inventaire reste vulnérable au shadow AI

La gouvernance procédurale repose implicitement sur une hypothèse forte : le périmètre des systèmes à gouverner est raisonnablement connu. Cette hypothèse devient de plus en plus fragile. Les données communiquées par Larridin début 2026 allaient précisément dans ce sens, en signalant qu'une majorité d'organisations interrogées ne disposaient pas d'un inventaire complet de leurs applications IA et que le nombre moyen d'outils déployés était déjà élevé. Ces chiffres ne valent pas démonstration générale à eux seuls, mais ils documentent bien la difficulté pratique du problème.

Il faut ici être précis sur ce que la gouvernance architecturale peut et ne peut pas. Elle peut rendre détectable par construction l'intégration de composants non conformes à l'intérieur du périmètre applicatif institutionnellement intégré. Un composant qui ne respecte pas les invariants d'intégration ne se fond pas silencieusement dans l'architecture centrale ; il génère une anomalie de composition ou demeure hors de la chaîne institutionnelle de décision. Elle ne supprime pas, en revanche, les usages latéraux, opportunistes ou extra-système qui relèvent du shadow AI au sens large. La différence reste néanmoins décisive : elle change profondément le régime de visibilité du problème, et la position de l'organisation vis-à-vis de son risque en devient plus défendable.

3.3 Sous pression opérationnelle, la procédure se dégrade plus vite que la structure

Les politiques de gouvernance sont appliquées dans des organisations réelles, c'est-à-dire sous contrainte de temps, de ressources et de priorités concurrentes. Dans ces conditions, ce qui n'est pas constitutif du fonctionnement métier est plus facilement reporté, dépriorisé ou contourné. La documentation intermédiaire, les revues périodiques de risque, certaines étapes d'alignement transversal subissent une dégradation prévisible lorsque l'organisation passe en mode urgence. Ce n'est pas une question de mauvaise volonté. C'est une question de gravitation organisationnelle.

Une propriété architecturale ne bénéficie pas d'un statut moral supérieur. Elle bénéficie d'un statut technique différent. Si l'inférence suppose la persistance effective des éléments de trace nécessaires à sa reconstitution comme événement institutionnel valide, l'échec de cette persistance n'est plus un simple écart documentaire. Il devient un échec du régime normal d'exécution. Si le passage d'une recommandation à une action automatisée est structurellement contraint, le contournement exige une modification du système, un acte plus coûteux, plus visible, et plus traçable qu'un simple oubli de procédure.

La différence n'est pas absolue. Elle est de coût, de visibilité et de probabilité de contournement. C'est déjà considérable.

IV. Trois propriétés constitutives d'une gouvernance architecturale

Si l'on admet que la gouvernance ne peut pas reposer exclusivement sur la procédure dans les systèmes régulés à fort enjeu, encore faut-il spécifier ce qu'une gouvernance architecturale exige concrètement. Je propose ici trois propriétés minimales, non comme catalogue de bonnes pratiques, mais comme système cohérent dont chaque élément conditionne l'efficacité des deux autres.

Propriété 1 : Traçabilité constitutive.

Chaque inférence ou transition décisionnelle pertinente doit produire, comme condition normale de son exécution, un enregistrement stable des éléments nécessaires à sa reconstitution : données d'entrée pertinentes, version des composants engagés, contexte d'exécution, régime décisionnel appliqué, résultat produit, niveau de confiance et identifiants de corrélation utiles. Le point décisif est le suivant : la trace ne doit pas être pensée comme un sous-produit facultatif du système, mais comme une propriété de son mode de fonctionnement. Un système ne devrait pas pouvoir valider durablement une inférence comme événement institutionnel sans persistance des éléments de trace nécessaires à sa reconstitution.

Propriété 2 : Qualification opérationnelle du domaine de validité.

Une requête ne doit pas être traitée comme si le modèle était universellement valide sur l'ensemble de l'espace d'entrée possible. Le système doit évaluer, au point d'exécution, si l'entrée se situe à l'intérieur du domaine de validité opérationnelle du modèle mobilisé. Cette qualification peut prendre des formes diverses : distance dans l'espace des représentations, estimation de densité, mesure d'incertitude, score de similarité, règles métier. La forme importe moins que la position : la borne de validité doit faire partie du pipeline d'inférence, non être reléguée à une évaluation secondaire ou optionnelle. Un système qui prédit hors de son espace de validité sans le signaler n'est pas un système mal gouverné. C'est un système qui ment sur ses propres certitudes.

Propriété 3 : Séparation structurelle des régimes décisionnels.

Le système doit distinguer explicitement ce qui relève de l'automatisation effective et ce qui relève de la recommandation soumise à validation humaine. Cette distinction ne peut pas dépendre uniquement d'une consigne organisationnelle. Elle doit être encodée dans le design des flux, des interfaces, des permissions, des événements de validation et des mécanismes d'escalade. Transformer une recommandation en action automatisée ne devrait pas être un glissement d'usage ; cela devrait exiger une décision d'architecture, une modification de périmètre et une réévaluation explicite des responsabilités. Les organisations qui délèguent cette frontière à une consigne d'usage découvrent généralement, au premier incident, que la consigne n'a pas résisté à la pression.

V. Auditabilité native et architecture événementielle

La proposition centrale de cet article est l'articulation suivante : certaines architectures rendent l'auditabilité native plus accessible, plus stable et plus économique que d'autres. Parmi elles, l'architecture événementielle associée à des patterns d'event sourcing occupe une place particulière, non parce qu'elle serait le seul chemin possible, mais parce qu'elle est l'un des patterns les plus naturellement compatibles avec cette propriété.

Dans une telle architecture, les changements d'état pertinents sont représentés comme des événements explicites, horodatés, identifiés et persistés dans un journal append-only. L'état courant du système peut être reconstruit comme projection de la séquence des événements jugés pertinents. Ce pattern est généralement présenté pour ses avantages en matière de découplage, de résilience et de scalabilité. Il possède aussi une vertu rarement exploitée jusqu'au bout dans les systèmes d'IA régulés : il rapproche la gouvernance du substrat même de fonctionnement du système. L'événement n'est pas seulement l'unité d'information ; il est l'unité d'audit. Le même mécanisme qui rend les composants coordonnables rend le système gouvernable.

Dans un tel cadre, une inférence pertinente peut être représentée comme un événement ou comme une transition enveloppée dans une séquence d'événements corrélés. Son contexte opératoire, ses dépendances de version, son régime de validation et ses effets en aval deviennent plus faciles à lier dans une chaîne cohérente de reconstitution. Cela ne rend pas transparent le mécanisme interne d'un

modèle profond. La nuance introduite au §I reste entière. Cela rend, en revanche, beaucoup plus robuste la traçabilité de son inscription dans un processus gouverné.

Par ailleurs, dans un système fortement structuré par événements, un composant qui ne respecte pas les schémas attendus, les invariants de validation ou les conditions de persistance nécessaires devient plus difficile à intégrer silencieusement au flux institutionnel. Cela ne supprime ni les usages parallèles ni les composants tiers opaques. Cela renforce la capacité du système central à distinguer ce qui lui appartient de ce qui lui est externe.

VI. Ce que cette approche ne résout pas

Une thèse sérieuse se juge aussi à la clarté de ses limites. En voici quatre, formulées sans atténuation.

La gouvernance architecturale a un coût initial réel.

Concevoir des flux explicites, maintenir des schémas d'événements, persister des transitions pertinentes, qualifier le domaine au point d'exécution, contrôler les transitions entre recommandation et automatisation : tout cela consomme du temps, du stockage, de la discipline d'ingénierie, et parfois de la latence mesurable. Ce coût n'est pas nul. L'argument n'est pas qu'il serait négligeable. Il est qu'il est souvent inférieur, sur le cycle de vie du système, au coût cumulé d'une gouvernance reconstruite après coup sur une architecture qui n'a pas été pensée pour être gouvernable.

Elle ne remplace pas la gouvernance procédurale.

Elle la rend plus crédible et plus efficace. Les obligations de documentation, les rôles, les arbitrages, les responsabilités, les comités, les évaluations de risque, les plans de surveillance post-marché, la gestion du changement : tout cela demeure nécessaire. Une architecture gouvernable n'abolit pas le besoin d'organisation humaine. Elle évite que cette organisation soit condamnée à gouverner un système qui lui échappe techniquement.

Les modèles fondationnels tiers restent partiellement opaques.

Lorsqu'un système intègre un modèle LLM externe via API, la gouvernance architecturale peut encadrer fortement les inputs, les outputs, les contextes d'appel, les usages autorisés et les points de validation humaine. Elle ne rend pas observable l'intégralité du processus interne de génération. L'auditabilité native du système intégrateur n'est pas l'auditabilité native du modèle tiers. Ce point doit être dit sans détour, car l'ambiguïté sur ce sujet est l'une des sources les plus fréquentes de fausse confiance dans les dispositifs de gouvernance actuels.

La frontière décision/recommandation se complexifie dans les systèmes agentiques.

Dans des architectures multi-agents, certaines décisions intermédiaires structurent l'espace des options disponibles pour la décision finale, sans être toujours exposées avec la même lisibilité à l'utilisateur humain. La séparation structurelle des régimes décisionnels reste un objectif valide. Sa mise en œuvre devient plus délicate lorsque la chaîne d'orchestration est elle-même dynamique, distribuée ou adaptative. C'est un problème ouvert dans la littérature, et il serait prématuré de prétendre que la gouvernance architecturale le résout intégralement dans l'état actuel de l'art.

VII. Deux terrains d'implémentation

L'intérêt de la distinction entre gouvernance procédurale et gouvernance architecturale n'est pas purement spéculatif. Il devient particulièrement tangible dans deux types de systèmes développés au sein du Twingital Institute.

Dans le programme Sentinelle IA / PREDICARE, le jumeau numérique du patient est modélisé comme une séquence d'événements physiologiques, thérapeutiques et contextuels. Chaque mesure, chaque prescription, chaque interaction avec le système de santé est représentée comme un événement immuable horodaté, persisté dans un journal d'état. L'état courant du jumeau est calculé comme la projection de cette séquence. Ce choix n'a pas été motivé en premier lieu par un objectif documentaire de conformité, mais par une exigence de modélisation correcte de la dynamique clinique : les événements sont ici plus proches de la réalité clinique que les agrégats figés. Il en résulte un bénéfice de gouvernabilité important : chaque décision prédictive peut être replacée dans une chaîne d'événements reconstituable, non par approximation post hoc, mais parce que la structure du système a été conçue pour cela. Il faut toutefois être explicite sur ce point : un tel choix architectural n'emporte ni présomption automatique de conformité, ni substitution aux exigences du cycle MDR. En matière de dispositifs médicaux, la documentation technique de post-market surveillance relève de l'Annexe III, tandis que l'Annexe XIV encadre l'évaluation clinique et, pour sa partie B, le PMCF. Une architecture plus traçable facilite la gouvernabilité et peut soutenir le dossier réglementaire ; elle ne remplace ni les exigences qualité, ni l'évaluation clinique, ni les obligations documentaires applicables.

Dans le pipeline ToxTwin V1.3, le domaine d'applicabilité est évalué pour chaque molécule soumise au modèle GINEConv OGB, sur la base de sa distance aux voisins les plus proches dans l'espace des descripteurs de représentation moléculaire. Cette évaluation n'est pas un outil de post-traitement optionnel ; elle est un composant du pipeline d'inférence dont le résultat conditionne la présentation de l'output. Ce choix a été motivé par un constat technique précis : un modèle entraîné sur des molécules organiques de bas poids moléculaire produit des scores de confiance peu interprétables pour des structures chimiquement éloignées de son espace d'entraînement. Sans qualification préalable, le système afficherait une confiance élevée sur des cas hors domaine avec la même interface que pour des cas proches du corpus d'apprentissage. C'est une forme de désinformation architecturale. Ce que cette instance illustre est plus modeste, et plus utile, qu'une proclamation générale : la qualification de domaine opérationnelle est intégrable dans un pipeline de prédiction industriel. Elle n'établit pas, à elle seule, la supériorité universelle d'une méthode de définition du domaine d'applicabilité sur toutes les alternatives.

Il faut résister à la tentation de transformer chaque instance cohérente en vérité métaphysique. Ces deux cas ont valeur d'illustration de faisabilité, non de preuve générale.

VIII. Ce que cela implique concrètement

La conséquence la plus importante est un déplacement de séquence.

La plupart des organisations abordent encore la gouvernance de l'IA selon une logique implicite en plusieurs temps : exploration, développement, déploiement, puis encadrement. Cette séquence était peut-être tolérable dans des systèmes décisionnels à faible enjeu individuel. Elle devient fragile lorsque les exigences d'audit, de traçabilité et de maîtrise du régime d'automatisation doivent résister à l'échelle, au temps et à la pression opérationnelle réelle.

L'enjeu n'est donc pas d'ajouter plus de règles ou plus de comités. C'est de faire entrer beaucoup plus tôt, dans les décisions d'architecture, des questions qui sont encore trop souvent traitées comme relevant exclusivement de la conformité : ce qui doit être traçable par construction, ce qui doit être

borné avant exécution, où passe exactement la frontière entre assistance et automatisation, quels composants ont le droit de produire des effets système sans validation humaine tracée. Tant que ces questions arrivent après la stabilisation des patterns d'architecture, la gouvernance reste condamnée à courir derrière des systèmes déjà trop libres pour être réellement gouvernés par les politiques appliquées autour d'eux.

Conclusion

L'industrie n'a pas besoin d'opposer gouvernance procédurale et gouvernance architecturale. Elle a besoin de cesser de croire que la première peut durablement compenser l'absence de la seconde.

Les frameworks de conformité, les référentiels de gestion du risque, les normes de management et les obligations documentaires sont nécessaires. Ils structurent les responsabilités, organisent la surveillance, rendent possible l'audit et constituent un langage commun entre ingénierie, qualité, conformité et direction. Mais ils ne suffisent pas, à eux seuls, à garantir qu'un système d'IA régulé soit effectivement gouvernable au point où il agit, c'est-à-dire au moment de chaque décision individuelle, sous contrainte opérationnelle, dans un périmètre que personne ne contrôle entièrement.

La question centrale n'est donc pas seulement : comment gouverner l'IA que nous déployons ? Elle est plus en amont, et plus exigeante : comment concevons-nous des systèmes dont la gouvernabilité fait partie de l'architecture, et non de la documentation ajoutée autour d'elle ?

C'est moins une question de maturité qu'une question de séquencement. Et en ingénierie comme en médecine, les erreurs de séquencement sont rarement spectaculaires au départ. Elles deviennent surtout coûteuses lorsqu'il est déjà trop tard pour prétendre qu'il ne s'agissait que d'un détail.