

IBM, l'IA et l'autonomie des systèmes

Une modernisation inachevée

Pourquoi l'intelligence artificielle n'a jamais transformé en profondeur les systèmes stratégiques d'IBM — et ce que cela implique pour la modernisation des infrastructures legacy.

ImagineOps

Transformation des systèmes legacy par l'IA

Février 2026

Sommaire

Executive Summary	3
1. IBM et l'IA : une antériorité indiscutable	4
1.1 TD-Gammon (1992)	4
1.2 Deep Blue (1997).....	4
1.3 Autonomic Computing (2001)	4
2. Ce qui n'a pas eu lieu.....	6
3. Le dilemme stratégique.....	7
4. Pendant ce temps : le cloud.....	8
5. Ce que cela signifie aujourd'hui	9
6. Conclusion stratégique.....	10

Executive Summary

IBM n'a pas manqué la révolution de l'intelligence artificielle sur le plan scientifique. Elle l'a abordée très tôt, notamment via ses laboratoires de recherche.

Pourtant, l'IA n'a jamais transformé en profondeur ses systèmes stratégiques — en particulier z/OS et IBM i.

Thèse stratégique : IBM a privilégié un modèle de performance déterministe et de stabilité propriétaire plutôt qu'une autonomisation intelligente de ses systèmes cœur, ce qui a contribué à maintenir un TCO humain élevé et un modèle économique fondé sur la rareté des compétences.

1. IBM et l'IA : une antériorité indiscutable

IBM Research a été un acteur majeur de l'intelligence artificielle dès les années 1980–1990. Trois jalons illustrent cette antériorité — et les choix qui ont suivi.

1.1 TD-Gammon (1992)

Le projet TD-Gammon démontrait qu'un réseau neuronal entraîné par apprentissage temporel-différentiel (temporal difference learning) pouvait atteindre un niveau expert au backgammon sans programmation heuristique explicite.

📄 *Tesauro, G. (1992–1995), IBM Research.*

→ IBM maîtrisait déjà l'apprentissage autonome avant la vague deep learning.

1.2 Deep Blue (1997)

La victoire de Deep Blue contre Garry Kasparov fut un moment médiatique majeur. Mais techniquement, le système reposait sur une recherche d'arbre massive, une évaluation positionnelle déterministe et aucun apprentissage autonome.

📄 *Campbell, Murray et al., IBM Research, 1997.*

Caractéristique	TD-Gammon	Deep Blue
Méthode	Apprentissage par renforcement	Recherche d'arbre + heuristiques
Apprentissage autonome	Oui	Non
Message au marché	L'IA peut apprendre seule	IA = puissance de calcul brute

→ Message implicite transmis au marché : IA = puissance de calcul + performance brute.

1.3 Autonomic Computing (2001)

En 2001, IBM lance l'initiative Autonomic Computing, portée par Paul Horn, Senior VP Research. Le manifeste définit quatre capacités fondamentales pour un système autonome :

Capacité	Description
Self-configuring	Configuration automatique selon les objectifs
Self-healing	Détection et correction autonome des défaillances
Self-optimizing	Amélioration continue des performances
Self-protecting	Défense proactive contre les menaces

📄 *Horn, P. (2001). Manifesto Autonomic Computing, IBM.*

→ *Vision extrêmement avancée pour l'époque — restée largement programmatique.*

2. Ce qui n'a pas eu lieu

Malgré cette vision fondatrice, l'intelligence artificielle n'a jamais pénétré le cœur des systèmes opérationnels d'IBM :

- z/OS n'est pas devenu un système d'exploitation auto-apprenant
- IBM i n'a pas intégré de boucle ML native
- Les systèmes sont restés pilotés par des règles humaines explicites

Certes, des composants d'optimisation existent — WLM (Workload Manager), RMF (Resource Measurement Facility), le self-tuning de DB2. Mais ils reposent sur des politiques déterministes, des heuristiques paramétrées et une optimisation statique. Pas sur un apprentissage adaptatif continu.

Composant IBM	Nature réelle	Ce qui manque
WLM	Règles de priorité paramétrées	Apprentissage contextuel dynamique
RMF	Mesure et reporting	Prédiction et auto-adaptation
DB2 Self-tuning	Heuristiques d'indexation	Optimisation continue par ML

3. Le dilemme stratégique

Un système d'exploitation semi-autonome aurait pu transformer radicalement l'équation économique du mainframe :

Bénéfice potentiel	Impact client	Impact IBM
Réduction des besoins d'experts z/OS	Baisse du TCO humain	Baisse des revenus services
Automatisation de la remédiation	Gain de réactivité	Moins de contrats support
Adaptation dynamique de capacité	Optimisation des coûts	Baisse de la consommation MIPS

Hypothèse stratégique : l'autonomie systémique aurait été économiquement disruptive pour IBM elle-même.

Ce dilemme n'est pas unique à IBM. Il est classique dans l'économie des plateformes : le fournisseur qui réduit la complexité de son système réduit aussi la dépendance de ses clients — et donc ses revenus récurrents. Dans le cas d'IBM, la rareté des compétences mainframe n'est pas un accident : c'est un pilier structurel du modèle économique.

4. Pendant ce temps : le cloud

À partir de 2006, les infrastructures cloud ont progressivement intégré les principes d'autonomie que le mainframe n'a jamais adoptés :

Étape	Innovation	Acteur
2006	Auto-scaling élastique	AWS (EC2)
2010–	ML intégré à l'infrastructure	Google
2014–	Orchestration automatisée des conteneurs	Kubernetes / CNCF
2020–	Observabilité pilotée par IA (AIOps)	Écosystème cloud natif

Les infrastructures modernes sont devenues auto-adaptatives, observables et pilotées par les données. Le mainframe, lui, reste stable, puissant — mais fondamentalement dépendant de l'expertise humaine.

Cloud natif	Mainframe
Auto-adaptatif Observable Piloté par données	Stable Puissant Dépendant de l'humain

5. Ce que cela signifie aujourd'hui

La question stratégique a fondamentalement changé. Le sujet n'est plus de savoir s'il faut quitter le mainframe. La vraie question est désormais :

Peut-on autonomiser intelligemment les systèmes hérités pour réduire le TCO humain sans sacrifier la robustesse ?

L'alliance **Open Source + IA** rend cette transformation possible aujourd'hui, indépendamment des modèles propriétaires historiques. Les avancées en intelligence artificielle générative permettent désormais de comprendre, documenter et transformer du code legacy à une vitesse et un coût sans précédent.

La crise démographique des compétences mainframe accélère cette urgence : l'âge moyen des développeurs COBOL est estimé à **58 ans**, avec un taux de départ en retraite d'environ **10% par an**.

Sources : Micro Focus / Gartner, 2023.

6. Conclusion stratégique

IBM n'a pas manqué l'IA scientifiquement. **Elle a fait un choix stratégique cohérent avec son modèle économique** — un modèle où la complexité système et la rareté des compétences génèrent de la valeur pour le fournisseur.

Mais le marché 2025–2035 est fondamentalement différent :

Facteur de changement	Implication
Pression sur les coûts IT	Les DSI doivent justifier chaque euro de TCO
Rareté des compétences legacy	Le risque RH devient un risque opérationnel
Accélération technologique	L'IA générative transforme l'économie de la migration
Maturité ML opérationnelle	L'automatisation intelligente est désormais viable

**L'autonomie des systèmes legacy
n'est plus une option. C'est une nécessité.**

ImagineOps

Transformation des systèmes legacy par l'IA et l'Open Source
Diagnostic • Transformation forfaitaire • Souveraineté technologique