TBSO · OPENSKL · 001

Système d'exploitation d'entreprise agentique

Découvrez notre approche, notre architecture ainsi que les challenges à résoudre impérativement pour concrétiser la construction d'un OS agentique adapté aux besoins métiers concrets de PME et TPE.

ÉRIC LARCHEVÊQUE	TBSO · OPENSKL	MAI 2026	V1.0

§00

RÉSUMÉ

▲ INTRO

Éric Larchevêque est entrepreneur, investisseur et ingénieur, surtout connu comme cofondateur de Ledger et TBSO. Sa perspective sur les systèmes d'entreprise agentiques repose sur 30 ans d'expérience au coeur d'entreprises de technologie.

Votre équipe passe des heures entre Gmail, Slack, Notion, HubSpot et Stripe. Au lieu de prendre des décisions, d'être au service de vos clients ou de développer des produits, elle *navigue entre les interfaces* en transportant du contexte et de l'information entre des outils qui ne communiquent pas entre eux.

Vous savez très bien que vous devez utiliser l'IA. Mais après avoir automatisé quelques workflows, ou utilisé ChatGPT, voire Claude Cowork, vous vous heurtez à un mur. Vos intégrations restent fragiles, vous ne trouvez personne en interne pour les maintenir, et vous n'avez en général pas d'expertise claire sur la suite à donner. Le souci majeur est que les informations qui vont le coeur de votre entreprise sont dispersées sur de nombreux outils différents et il devient vite compliqué de permettre à une IA d'y voir clair.

> Ce document décrit l'**OS agentique d'entreprise** : un changement de paradigme où tout le travail se fait depuis une interface unique : le cockpit, ou control room. Vous avez accès à toutes les informations nécessaires pour évaluer vos priorités et vos tâches. Vous transférez vos objectifs à des agents IA qui vont générer des plans et se connecter directement aux différents outils SaaS.

Plusieurs éléments de cette approche sont déjà proposées par des éditeurs de logiciels (Salesforce, Microsoft, Atos...) qui s'adressent en général plutôt aux grandes enterprises. Mais il n'y a encore rien de concrétisé pour un outil conçu pour des petits et moyennes entreprises (où il n'y a pas de DSI), et il n'y a encore peu de littérature sur les enjeux à résoudre pour faire de cette vision une réalité.

Ce document se concentre sur cinq de ces challenges :

01

Mémoire

Votre entreprise ne dispose pas d'une source unique de vérité. Différents outils détiennent différentes informations. Les agents ne peuvent pas agir correctement sans savoir *quelle source prévaut* pour *quelle information*, et *quand*.

02

Sécurité

Une interface de pilotage unique représente également une surface d'attaque universelle. Un seul e-mail vérolé peut déclencher des actions malveillantes sur tous les systèmes connectés.

03

Latence et UX

Une interface de chat n'est pas adaptée pour gérer une entreprise. Il faut une UX capable d'adresser simultanément la visualisation des KPI, l'affichage de toutes les informations nécessaires ainsi que toute la surveillance et reprise de contrôle potentielle des agents.

04

Supervision humaine

Devoir approuver chaque action d'agent crée une gouvernance ingérable et donc complètement inutile. Une supervision fonctionnelle demande une autonomie minimale et qui apprend de chaque décision humaine.

05

Migration

Réussir à faire la transition est en soi un challenge majeur. Faire fonctionner les outils déjà existants et un nouveau système de manière simultané peut créer ce qu'on appelle une *hallucination institutionnelle*, c'est à dire deux versions concurrentes de la réalité.

Ce document a aussi en lui-même un scope limité. Il ne s'agit pas d'une spécification de produit, mais de notre évaluation interne des enjeux et de ce qui doit être résolu pour que le projet aboutisse.

§01

UNE INTERFACE UNIFIÉE POUR TOUT CONTRÔLER

▲ THESIS

Vers une exécution supervisée

Votre journée ressemble probablement à ceci : un e-mail reçu avec une demande client, une requête CRM pour vérifier son historique, Slack pour demander à votre équipe les tarifs à jour, retour à l'e-mail pour envoyer un devis, puis une tâche créée quelque part pour qu'une personne fasse un suivi. Peut-être la mise à jour d'un Excel ou d'un autre outil, et probablement une note mentale pour la suite, mais qui aura tendance à se perdre.

Aucune de ces étapes n'est difficile en soi. Tout cela prend juste du temps et de l'énergie. Au final, votre travail ne consiste pas à réfléchir, mais à *naviguer*.

Le système d'exploitation agentique remplace cela par une nouvelle interface, le **cockpit**. Imaginez un écran unique où vous voyez vos KPI et vos priorités, exprimez au système ce que vous voulez faire, examinez ce qu'il propose et approuvez son plan. Le CRM, le système de facturation, l'email et l'outil de gestion de projet existent toujours, mais passent en services d'arrière-plan entièrement gérés via les agents IA et le cockpit.

Pour illustrer concrètement ce que cela signifie, considérons un scénario classique où un formulaire de contact est rempli pour demander un devis pour un produit, le tout dans un environnement PME.

MODE 01 · AUJOURD'HUI

1. Clic clic clic

Dans la plupart des petites entreprises aujourd'hui, cette demande arrive dans une boîte de réception ou une file d'attente CRM. Le flux de travail d'un "humain" ressemble alors à ce qui suit :

- Ouvre l'e-mail de soumission du formulaire ou l'enregistrement CRM.
- Vérifie quel produit et quelles options ont été sélectionnés.

- Recherche le prospect dans le CRM (s'il y en a un) pour voir s'il est un client existant.
- Passe à l'email pour demander les informations manquantes ou envoyer une grille tarifaire.
- Envoie un message à l'équipe sur Slack (ou appelle un collègue) pour demander quel type de réduction peut être offert
- Quelqu'un se souvient que ce prospect avait eu une discussion préalable avec un collègue à propos d'une situation spécifique.
- Un nouvel e-mail est rédigé avec une proposition modifiée reflétant ces échanges précédents
- Un champ de feuille de calcul type Excel ou de CRM est mis à jour.
- Une tâche est créée dans un outil de projet pour s'assurer que quelqu'un effectue un suivi, ou une note mentale est prise en espérant un suivi ultérieur.

Comme vous pouvez le constater, tout le boulot consiste à naviguer entre cinq outils et à se souvenir de ce qu'il faut faire ensuite. Le « processus » vit dans la tête de quelqu'un et dans des listes de contrôle éparpillées un peu partout. Le prospect reçoit la meilleure attention possible et la proposition est rédigée avec les informations les plus à jour, mais ce process a pris beaucoup plus de temps et d'énergie que nécessaire.

MODE 02 · AUTOMATISATION DE BASE

2. Automatisation IA intégrée au SaaS

Un consultant ou un projet mené en interne pourrait mettre en place ce type d'intégration :

- La soumission du formulaire déclenche une automatisation qui crée ou met à jour un contact ainsi qu'une affaire de type "Demande de devis" dans le CRM avec les champs de base remplis.
- Le workflow interroge le CRM pour vérifier si le prospect existe déjà et définit un simple indicateur "nouveau vs existant".
- Basé sur des règles codées en dur (type de produit, taille de l'offre, géographie), il sélectionne un modèle de tarification et une fourchette de

remise.

- Un agent IA est appelé avec ce modèle et quelques paramètres pour rédiger un e-mail au prospect, insérant le nom, le produit et le prix dans un schéma classique.
- Le système envoie l'email via Gmail ou l'intégration email du CRM et publie une notification dans un canal Slack dédié aux ventes.
- Une tâche de suivi est créée automatiquement dans le CRM ou l'outil de gestion de projet avec une date d'échéance et un responsable.

On gagne clairement sur le nombre de clics humains, mais l'automatisation est limitée au périmètre du formulaire et d'une poignée de paramètres du CRM ; elle ne peut pas intégrer les informations provenant de conversations sur les différents outils de l'entreprise, ni les promesses spéciales ou les nuances du compte client sans être reconfigurée à chaque changement du processus.

MODE 03 · OS AGENTIQUE

3. OS agentique : ce qui change

Avec une approche basée sur OS agentique, la même soumission de formulaire de contact devient le déclencheur d'une séquence supervisée et orientée par les objectifs plutôt qu'un flux de travail fixe. Voici à quoi cela ressemblerait :

- La soumission du formulaire est ingérée en tant qu'événement dans le cockpit de contrôle. Un agent orchestrateur décide quel sous-agent est le mieux adapté pour le gérer.
- Le sous-agent choisi interroge alors la mémoire de l'entreprise : est-ce qu'il s'agit d'un nouveau client ou d'un client existant, quelle est la valeur actuelle du contrat et quel est l'historique des paiements, quels sont les tickets de support récents, les opportunités ouvertes, ainsi que toute conversation antérieure par email, Slack ou WhatsApp mentionnant ce compte ou ce produit.
- En utilisant ce contexte et les politiques écrites de tarification et de remise de l'entreprise, toutes mises à jour selon les dernières décisions internes, l'agent élabore un plan proposé. Comment classifier l'opportunité, quel prix

> et quelle fourchette de remise correspondent à la fois à la politique et à l'historique, quels risques ou exceptions doivent être signalés, quel responsable interne doit en prendre en charge, et quelles tâches de suivi sont nécessaires.
> - Le cockpit affiche ce plan sous forme de carte structurée proposant une classification, un contexte clé extrait de la mémoire et un message rédigé à l'intention du prospect.
> - Le responsable des ventes peut ajuster le devis ou le message, voir sur quelles décisions antérieures l'agent s'est appuyé, et approuver. L'agent exécute ensuite les étapes autorisées dans les systèmes CRM, email et de gestion des tâches.
> - La mémoire de l'entreprise est automatiquement mise à jour pour refléter les dernières actions et modifications.
>
> Dans cette approche, le traitement d'une demande de devis cesse d'être une chaîne de clics et de décisions ad hoc, ou un flux déterministe principalement rigide, pour devenir une tâche récurrente gouvernée et flexible, prise en charge par le cockpit et supervisée par l'humain.

Pourquoi c’est plus qu’un orchestrateur d’agents

Cinq éléments distinguent cela des orchestrateurs d'agents classiques :

- **Horizontalité** — Les orchestrateurs automatisent un flux de travail ou un domaine. Un OS gère tout de façon horizontale (ventes, support, marketing, stratégie...) sur une même infrastructure.
- **Mémoire** — Les orchestrateurs extraient quelques champs dans un contexte. Un OS maintient une mémoire d'entreprise gouvernée et basée sur des faits et décisions, utilisant des preuves temporelles pour établir une notion de "vérité".
- **Gouvernance** — Dans un orchestrateur, chaque agent a ses propres règles. Dans un OS, une couche centrale de sécurité classe chaque action et applique la supervision humaine de manière uniforme et transverse.
- **Interface** — La plupart des orchestrateurs fonctionnent de manière invisible. L'OS place le cockpit au premier plan, là où se passent les décisions.

- **Adoption** — Un orchestrateur suppose que quelqu'un sait quoi automatiser. L'OS observe l'entreprise, la comprend et suggère par où commencer.

Un orchestrateur d'agents reste un élément constitutif au sein de de l'OS agentique, mais c'est le système d'exploitation qui va faire toute la différence en développant et construisant les différent workflows sans que ses dirigeants ou employés aient besoin de devenir spécialistes en IA.

Pourquoi il doit s'agir d'une salle de contrôle, et non d'un interface de chat

Une entreprise ne peut pas être gérée via une interface type chat. Il existe en effet trois modes de travail nécessitant chacun des UX différentes :

- **Le mode surveillance** — Qu'est-ce qui nécessite une attention particulière ? Quelles sont mes priorités ?
- **Le mode exécution** — Construit le plan et montre le moi, puis met le en oeuvre.
- **Le mode interruption** — Bouton d'arrêt d'urgence, car quelque chose ne va pas.

Une conversation en mode chatbot est bien trop lente pour la surveillance, trop ambiguë pour l'exécution et trop bruyante pour les interruptions. Le cockpit doit ressembler davantage à un terminal Bloomberg qu'à ChatGPT.

Le nouveau rôle du SaaS

Les outils SaaS ne disparaissent pas, mais viennent se cacher en arrière-plan. Les CRM, les systèmes de facturation, les wikis type Notion et les outils de gestion de projets cessent d'être une interface pour devenir une API que le cockpit interroge et met à jour.

§02

ARCHITECTURE DE RÉFÉRENCE

▲ MODEL

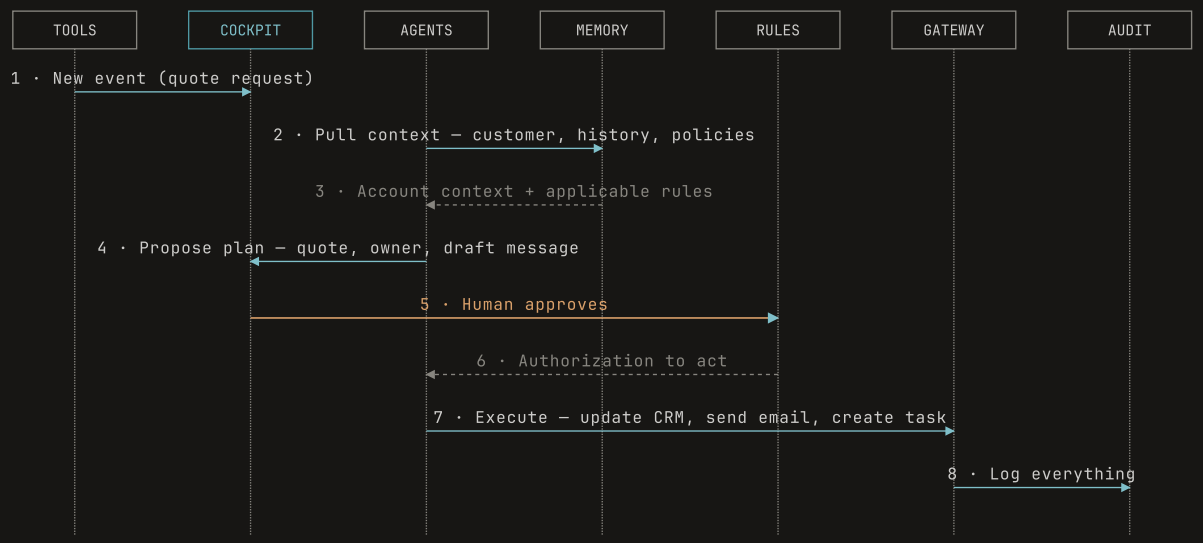
L'OS agentique peut être architecturé comme sept couches conceptuelles qui interagissent.

COUCHE	QUESTION PRINCIPALE	RESPONSABILITÉ PRINCIPALE
Systèmes de référence	Où résident les données opérationnelles aujourd'hui ?	SaaS existants, outils internes, bases de données, documents et plateformes de données.
Passerelle d'intégration	Comment les agents accèdent-ils à ces systèmes de manière sûre et uniforme ?	Serveurs MCP, API, webhooks, connecteurs, adaptateurs, authentification et normalisation.
Noyau de Gouvernance & Sécurité	Qu'est-ce qui est autorisé, par qui, sous quelles conditions ?	Moteur de politique, permissions, classification, jetons de délégation, règles d'escalade, et routage avec intervention humaine.
Mémoire d'entreprise	Que sait l'entreprise et qu'a-t-elle décidé ?	Faits, décisions, rôles, autorité, contexte, validité temporelle, engagements et mémoire institutionnelle.
Observabilité & Audit	Qu'est-ce que le système a réellement fait, et pourquoi ?	Flux d'audit temporel, traces, métriques, explications, approbations, relecture et responsabilité.
Couche d'exécution de l'agent	Qui planifie, raisonne et exécute ?	Planificateurs, orchestrateurs, agents de domaine, agents appelant des outils, décomposition des tâches, exécution et récupération.

COUCHE	QUESTION PRINCIPALE	RESPONSABILITÉ PRINCIPALE
Cockpit	Où les humains voient, décident et interviennent ?	Salle de contrôle pour les plans, approbations, exceptions, surveillance, délégation et supervision opérationnelle.

Voici comment une seule requête se déplace à travers le système :

FIG. 02 · SINGLE-REQUEST · SEQUENCE — RENDERED



SOLID · COMMANDS / WRITES · DOTTED · RESPONSES / RESULTS — FIG. 02 · 10 STEPS

Ce que fait chaque couche

Vos outils existants (Systèmes de référence). Rien ne bouge, rien ne migre. HubSpot reste HubSpot. Gmail reste Gmail. Stripe reste Stripe. Ces systèmes continuent de faire ce qu'ils font, c'est-à-dire stocker des contacts, des e-mails, des factures, des tâches. Ce qui change, c'est que vous cessez de les ouvrir directement pour accomplir votre travail. Ils deviennent des services en arrière-plan que le cockpit lit et écrit.

La couche de connexion (Passerelle d'intégration / Gateway). C'est ce qui permet aux agents d'accéder à vos outils en toute sécurité. C'est un passage

obligatoire garantissant que chaque agent ne touche que ce qu'il est censé toucher, avec des identifiants qui expirent une fois la tâche terminée. Aucun agent ne doit jamais détenir une clé d'accès permanente pour vos outils.

Le moteur de règles (Noyau de Gouvernance et de Sécurité). Avant qu'un agent ne prenne une action, cette couche vérifie si celle-ci est autorisée. Elle classe chaque action proposée, allant de totalement autonome (rédiger un brouillon, rechercher des informations) à nécessitant votre approbation explicite (envoyer un email externe, modifier un prix, déclencher un paiement). Elle apprend également avec le temps, car les actions que vous approuvez régulièrement sans modification nécessiteront progressivement moins votre attention et deviendront entièrement autonomes.

Mémoire d'entreprise. C'est la couche qui distingue l'Agentic OS d'un chatbot ou d'un workflow. Elle cartographie votre entreprise (personnes, projets, produits...) et conserve ce que votre entreprise sait réellement (clients, deals, priorités, décisions, engagements, process...). Contrairement aux approches "naïves" classiques basées sur de la recherche documentaire, la mémoire doit être construite comme une architecture de données gouvernées où chaque élément appartient à un cluster de contexte, possède une source déclarée, un propriétaire et une date. C'est la pièce maîtresse du système.

La piste d'audit (Observabilité & Audit). Chaque action effectuée par le système est enregistrée pour référence future (ce qui a été fait, pourquoi, quels paramètres ont été utilisée, qui l'a approuvé, etc.). C'est ce qui vous permet de demander « que s'est-il passé avec ce truc mardi dernier ? » et d'obtenir une réponse précise. C'est aussi ce qui vous offre une possibilité de retour en arrière en cas de problème, ainsi que la preuve de conformité si quelqu'un en fait la demande (utile pour rejeter la faute sur un agent et organiser son exécution publique en rétribution).

Les agents eux-mêmes (Agent Runtime). C'est ici que le raisonnement et l'exécution ont lieu. Un planificateur détermine ce qui doit être fait, des agents spécialisés gèrent des domaines spécifiques comme les ventes ou le support, et des exécutants légers réalisent l'exécution des diverses étapes individuelles. Ils travaillent en parallèle lorsque c'est possible, consultent la mémoire avant d'agir, et vous sollicitent lorsqu'ils rencontrent une situation nécessitant votre supervision.

Le cockpit. En théorie, le dernier écran que vous utiliserez jamais pour le travail en entreprise. C'est ici que vous voyez ce qui se passe, dites au système ce que vous voulez, examinez ce qu'il propose, et approuvez ce qu'il fait. Les indicateurs clés de performance (KPI), l'état de votre entreprise, les priorités, la liste des éléments nécessitant votre attention, et les plans que les agents sont sur le point d'exécuter (ou sont en train d'exécuter).

§03

CHALLENGE 1 : LA MÉMOIRE D'ENTREPRISE

▲ CHALLENGE-01

CHALLENGE-01

MÉMOIRE D'ENTREPRISE

La mémoire d'entreprise doit être conçue comme un système de **gouvernance des faits**, et non comme un simple index documentaire.

Pourquoi « connecter toutes vos données » ne fonctionne pas

Considérez la question « Qu'est-ce qu'on a promis à ce client la dernière fois ? »

Un système de recherche naïf tentera de synthétiser une réponse à partir des élements suivants :

- Un contrat signé dans Drive.
- Un fil Slack où un ingénieur commercial a promis une fonctionnalité.
- Une note CRM résumant un appel.
- Un fil de discussion par e-mail négociant une remise.
- Une page Notion décrivant le plan standard.

Le vrai problème n'est pas l'hallucination du modèle, c'est que **l'entreprise elle-même a une vérité non résolue**. Il n'existe pas de source unique de vérité car différents champs sont légitimement détenus par différents systèmes et rôles.

Avant que vos agents n'écrivent quelque part, vous devez répondre à quatre questions :

1. Qui est autorisé à affirmer quels faits ?
2. De quel système ou canal ?

3. Pendant combien de temps ces assertions sont-elles considérées comme valides ?
4. Comment les conflits sont-ils détectés, mis en évidence et résolus ?

Les quatre couches de mémoire et leurs limites

Dans un système agentique, il existe quatre couches distinctes de mémoire, chacune avec ses propres limites.

COUCHE MÉMOIRE	OBJECTIF	MODE DE DÉFAILLANCE TYPIQUE
Mémoire de travail	Contexte de tâche actuel et état de raisonnement actif	Troncature de la fenêtre de contexte, entraînant une perte du contexte à court terme pertinent
Mémoire épisodique	Interactions passées, actions et conversations	Résumés qui perdent des détails ou reflètent les biais du modèle
Mémoire sémantique	Connaissances de l'entreprise, faits et informations de référence	Informations obsolètes ou contradictoires sans actualité ni pondération d'autorité
Mémoire procédurale	Savoir-faire opérationnel et flux de travail (« comment les choses sont faites »)	Connaissances restant implicites et non documentées, vivant uniquement dans la tête des personnes

Les agents ne peuvent pas agir en toute sécurité au nom d'une entreprise si ces différentes couches de mémoire ne sont pas explicitement définies et gouvernées de façon transverse.

Mémoire de décision et sourcing d'événements

Les entreprises ne fonctionnent pas grâce aux données, mais **grâce aux décisions**. Cependant, les décisions ne sont en général jamais lisibles facilement pour un algorithme. Elles sont dispersées dans les e-mails, les réunions et les conventions implicites.

Ce qui manque, c'est un graphe contextuel qui capture les faits, les décisions et leurs raisons. Un modèle d'implémentation est le sourcing d'événements : enregistrer les décisions sous forme d'événements en mode ajout séquentiel et reconstruction de l'état actuel à partir de ceux-ci.

Cela a deux implications majeures :

- Les faits ne peuvent pas être écrasés ; ils ne peuvent être que supplantés.
- Chaque réponse fournie par le système peut être retracée à travers une chaîne de preuves.

La constitution de la mémoire

Avant de déployer des agents capables d'écrire partout, une entreprise doit rédiger une **constitution de mémoire**.

Au minimum, il doit définir :

- **Autorité par domaine** — Pour chaque fait important (valeur du contrat, date de renouvellement, statut de paiement, score de santé, propriétaire, etc.), spécification du système et du rôle faisant autorité.
- **Règles temporelles** — Pour chaque champ, spécification des exigences de date de mise à jour et quand les anciennes données doivent être considérées comme obsolètes.
- **Responsabilités d'invalidation** — Qui est responsable de la mise à jour de la mémoire lorsqu'un fait change ?
- **Résolution des conflits** — Comment les conflits sont-ils détectés et qui les résout (qui est le gestionnaire de la mémoire ?).

Ce n'est pas seulement technique. Cela oblige l'entreprise à décider quelle source est fiable pour quel fait. Si cette décision reste implicite, les agents automatiseront la confusion.

Il est possible de pouvoir automatiser une grande partie de cette constitution par la construction d'une hiérarchie des initiateurs de faits, dérivée de l'analyse contextuelle du fonctionnement de l'entreprise elle-même.

§04

CHALLENGE 2 : LE COCKPIT EN TANT QUE SURFACE D'ATTAQUE

▲ CHALLENGE-02

CHALLENGE-02

LE COCKPIT EN TANT QUE SURFACE D'ATTAQUE

Un cockpit agentique unifié est, par conception, une voie unique autorisée vers chaque système critique de l'entreprise. Normalement, un attaquant devait s'introduire pour voler des identifiants, trouver une faille zero-day ou se déplacer latéralement à travers les systèmes. Mais dans un OS agentique, un attaquant a juste besoin de compromettre **l'entrée** que l'agent va lire.

Le problème du corridor de confiance

Lorsque un agent peut lire des entrées qui ont un faible niveau de confiance et déclencher des actions à haut niveau de confiance, alors ses autorisations deviennent une surface d'attaque évidente. Quelques exemples :

- Un email élaboré avec des instructions cachées qu'un agent ingère et utilise pour mettre à jour des prix.
- Une page Notion empoisonnée qui oriente un agent vers la modification d'enregistrements dans le CRM.
- Un outil MCP compromis contenant des méta-instructions malveillantes exécutées au moment de la découverte.

Le cockpit est un **corridor de confiance** entre des surfaces à faible niveau de confiance (emails, documents, tickets, outils externes) et des actions à haut niveau de confiance (paiements, modifications critiques, voire code déployé en production). Si

ce pont n’est pas segmenté et surveillé, le cockpit peut alors devenir la plus grande faiblesse de l'entreprise.

Vecteurs d'attaque

Il existe déjà trois vecteurs actifs exploités dans les écosystèmes MCP réels :

- **Consignes cachées** dans le contenu que l'agent est censé lire (emails, documents, messages).
- **Empoisonnement d'outil** dans les descriptions du serveur MCP lues lors de la découverte des outils, non enregistrées dans les conversations.
- **Mouvement latéral d'agent à agent** via la mémoire partagée et le contexte auquel un agent fait confiance parce qu'un autre agent lui a fait confiance.

En 2026, la majorité des déploiements MCP n'ont pas sécurité en place entre l'agent et les outils, et seulement 8,5 % des serveurs MCP utilisent OAuth pour l'authentification [01] ; 73 % des déploiements d'IA en production sont vulnérables à l'injection de prompt [02]. OpenAI a officiellement qualifié l'injection de prompt de « problème de sécurité de pointe non résolu » [03].

OAuth et le mandataire trompé

La plupart des déploiements MCP actuels utilisent un "jeton" (token). L'utilisateur autorise un agent avec un jeton OAuth à large périmètre ; l'agent le transmet ensuite à chaque serveur. Cela crée un scénario classique du **mandataire trompé** (*confused deputy* en anglais) : l'agent agit avec toute l'autorité de l'utilisateur dans des contextes que l'utilisateur n'a jamais envisagés. [04]

Par exemple, un agent peut avoir besoin d'accéder à Google Workspace pour résumer des e-mails, mais comme il reçoit le jeton OAuth avec toutes les autorisations et le transmet à chaque serveur MCP, un outil compromis peut soudainement lire des documents Drive confidentiels et envoyer des e-mails en votre nom.

Il existe cependant des meilleures pratiques émergentes :

- L'agent est considéré comme un mandataire distinct de l'utilisateur.
- L'agent reçoit un jeton de délégation limité à ses tâches.

- Les jetons sont de courte durée et révocables, et émis par une politique de sécurité ad-hoc.

Le paradoxe de l'universalité et de la compartimentalisation

La proposition de valeur du cockpit est l'universalité : une seule interface pour un accès à tous les systèmes et dans tous les contextes. Mais en même temps, sa sécurité nécessite une forte compartimentalisation, ce qui signifie qu'aucun agent ne peut tout atteindre.

> Cela crée un paradoxe : le même cockpit qui donne accès à tout doit, en pratique, fonctionner comme une somme de compartiments étanches. Chaque connexion doit être associée à une identité bien définie, chaque action doit recevoir une autorisation explicite, et chaque escalade doit emprunter un chemin de transfert formalisé et auditable.

Cela représente un véritable défi. L'architecture doit vivre avec cette contradiction et l'assumer pleinement.

§05

CHALLENGE 3 : LATENCE ET CONCEPTION DE L'UX

▲ CHALLENGE-03

CHALLENGE-03

LATENCE ET CONCEPTION DE L'UX

La latence dans les systèmes agentiques concerne avant tout **la perception du temps** par l'utilisateur et représente un véritable challenge d'exécution.

Trois latences

Il existe trois types de latences perçues :

- **Latence technique** — temps de traitement pour l'inférence LLM, appels d'outils, allers-retours réseau.
- **Latence perceptuelle** — la rapidité ressentie du système, influencée par les progrès visibles et la réactivité.
- **Latence cognitive** — durée pendant laquelle l'utilisateur doit maintenir l'incertitude quant à savoir si le système fera ce qu'il faut ou pas.

Les optimisations techniques (vitesse du modèle, réseaux plus rapides) aident, mais elles ne résolvent pas le problème cognitif. Un indicateur figé pendant une séquence critique semble catastrophique, même si le résultat arrive en trois secondes.

Chat vs CLI : une fausse dichotomie

Les interfaces de chat échouent car le travail en entreprise n'est pas intrinsèquement conversationnel. Dès qu'une instruction devient complexe, avec des paramètres précis et des contraintes de périmètre, elle se prête mal à un échange de questions-réponses.

Les interfaces purement en ligne de commande échouent car l'intention de l'entreprise est ambiguë et riche en contexte. Forcer les utilisateurs à apprendre un CLI (*Command Line Interface*) pour exprimer une intention type "implémenter l'accord avec le client Y" reporte la charge cognitive sur l'utilisateur.

Le modèle correct est **d'abord l'intention, puis la planification** :

1. L'utilisateur exprime son intention en langage naturel.
2. Le système l'interprète, pose des questions de clarification si nécessaire et génère un plan d'exécution structuré avec des paramètres, des contraintes et des points de décision.
3. L'utilisateur examine, modifie et confirme.
4. L'agent exécute ce plan, avec des étapes de contrôle lorsque nécessaire.

Exécution spéculative et parallélisme

L'exécution spéculative est l'une des techniques les plus efficaces pour réduire la latence ressentie dans les systèmes agentiques. Pendant que l'utilisateur examine un plan (souvent 5 à 15 secondes), le système lance déjà, en arrière-plan, les sous-étapes à haute probabilité d'être utiles, dont les résultats seront nécessaires même si l'utilisateur apporte de petits ajustements au plan.

Lorsque l'utilisateur clique sur *Confirmer*, certains résultats sont déjà disponibles. Si le plan est modifié de manière à invalider le travail spéculatif, ce travail est abandonné. Il s'agit d'un emprunt direct à l'exécution spéculative des CPU appliquée aux flux de travail des agents.

De plus, les appels d'outils qui ne dépendent pas des sorties les uns des autres ne doivent jamais être exécutés en série. Les paralléliser peut réduire de moitié la latence perçue sans complexité conceptuelle.

§06

CHALLENGE 4 : BIEN INTÉGRER LES CYCLES DE DÉCISION

▲ CHALLENGE-04

CHALLENGE-04

BIEN INTÉGRER LES CYCLES DE DÉCISION

HITL (*Human-In-The-Loop*) est souvent invoqué comme un garde-fou pour maintenir les agents IA sous un contrôle et une supervision stricts. Mais sans une conception bien étudiée, cette approche devient au mieux un goulot d'étranglement et au pire un enfer à utiliser.

Fatigue de confirmation et illusion de contrôle

La fatigue de devoir tout confirmer est une catastrophe en matière de gouvernance. Voici dans les faits ce qui se passe lorsque les utilisateurs reçoivent trop de demandes de validation :

- Ils approuvent plus vite qu'ils ne lisent.
- Ils considèrent les approbations comme une étape mécanique et se mettent à auto-clicker.
- Le journal d'audit enregistre des approbations qui n'ont jamais été réellement accordées.

Ce résultat est en réalité pire que de donner une autonomie total aux agents. L'entreprise ne gagne ni en efficacité ni en contrôle. Pour y remédier, il existe trois modèles courants de supervision humaine :

- **HITL (Human-In-The-Loop)** — l'agent propose des actions, mais l'exécution est bloquée jusqu'à ce qu'un humain approuve explicitement.

- **HOTL (Human-On-The-Loop)** — l'agent exécute de manière autonome, dans des limites prédéfinies, tandis que les humains supervisent, surveillent et conservent l'autorité de contrôle.
- **HIC (Human-In-Command)** — les humains définissent les objectifs, les contraintes et la gouvernance, tandis que le système fonctionne de manière continue et autonome sauf en cas d'escalade ou d'override nécessaire.

Classification d'action à quatre niveaux

La solution : classer les actions selon leur réversibilité et l'ampleur des dégâts en cas d'erreur :

NIVEAU	CARACTÉRISTIQUES	POSITION D'APPROBATION	EXEMPLES
Niveau 1 Autonome	Réversible, faible portée	Exécuter et enregistrer	Rédiger des e-mails internes, résumer des documents, rechercher des informations
Niveau 2 Notification	Réversible, portée modérée ; systèmes réels impactés	Exécuter, mettre en avant pour révision	Mettre à jour les champs CRM non critiques, créer des tâches, planifier des réunions internes
Niveau 3 Approuver	Partiellement réversible ou sensible	Mettre en attente pour approbation humaine	Envoyer des e-mails externes, modifier les prix sur un compte, créer un nouveau fournisseur
Niveau 4 Gouvernance	Irréversible ou à fort impact	Approbation multipartite	Exécution de paiement, signature de contrats, modifications d'accès, suppression de données

Les actions T3 et T4 doivent rester une petite part du total des actions. Si ce n'est pas le cas, le périmètre de l'agent est trop large et submergera l'utilisateur sous un volume d'approbations vite ingérable.

Approbations en mode check-list

Pour les actions T3 et T4, de simples boutons Approuver/Refuser encouragent la validation automatique. Un meilleur modèle est une liste de contrôle type check-list que l'approbateur doit compléter :

1. Je confirme que l'intention de cette action est celle que j'ai autorisée.
2. Je confirme que les sources de données utilisées sont appropriées.
3. Je confirme que cette action est conforme au périmètre déclaré de l'agent.
4. Je comprends quels systèmes seront affectés et le rayon d'impact.
5. Je sais comment annuler cette action si nécessaire.

Cela prend 30 à 60 secondes, mais cela est approprié pour des actions à forts impacts et engageant la responsabilité claire de l'utilisateur.

HITL comme signal d'apprentissage

Les systèmes HITL échouent quand la supervision humaine devient une corvée répétitive, plutôt qu'un retour d'expérience qui permet au système d'apprendre et de s'améliorer.

Une architecture correcte doit considèrer chaque décision HITL comme un signal d'entraînement :

- Approbations sans modification → candidats à la promotion T1/T2.
- Approbations avec modifications → révèlent des erreurs de calibration dans le raisonnement ou les demandes de l'agent.
- Refus → indiquent des définitions de périmètre trop larges ou mal alignées.
- Actions approuvées qui produisent ensuite de mauvais résultats → indiquent que la politique ou les modèles de risque doivent être révisés.

Cela crée un **cercle vertueux HITL** : chaque revue humaine aide à recalibrer le système, réduisant progressivement le nombre d'actions nécessitant une intervention utilisateur.

§07

CHALLENGE 5 : RÉUSSIR LA MIGRATION

▲ CHALLENGE-05

CHALLENGE-05

RÉUSSIR LA MIGRATION

L'écart de fonctionnement et d'habitude entre un travail centré sur les interfaces des SaaS actuels et un OS d'entreprise agentique avec interface unifiée est majeur et va évidemment créer des défis de migration.

Les données empiriques ne vont pas dans le bon sens

- IDC : 54 % des preuves de concept en IA n'atteignent jamais la production. [05]
- MIT NANDA : échec des pilotes Generative AI autour de 95 %. [06]
- Enquête PwC auprès des CEO : 56 % des CEO ne constatent aucun impact financier des investissements dans l'IA malgré son adoption. [07]
- Gartner : plus de 40 % des projets d'IA agentique devraient être annulés d'ici 2027. [08]

Cependant, les petites entreprises ont un avantage structurel. Le fondateur décide, l'équipe s'aligne en quelques jours et non en plusieurs mois, et la pile d'outils se compose de 5 à 15 produits SaaS, et non de 200.

Ce schéma d'échec n'est pas inévitable, à condition d'adopter une approche progressive et d'éviter un piège bien précis.

Le seul piège à éviter : la double réalité

Dans une transition naïve, à la fois les anciennes interfaces (Notion, Salesforce, Gmail, Stripe) et le cockpit sont actifs et peuvent écrire dans la "vérité opérationnelle". Cela

crée une double réalité où deux systèmes sont tous deux autoritaires en pratique mais aucun n'a formellement raison.

Une prévision basée sur des données partiellement migrées peut être à la fois précise et erronée. Une réponse du support basée sur la mémoire du cockpit peut contredire ce que quelqu'un a configuré manuellement dans l'outil cinq minutes plus tôt.

> Le résultat n'est pas une hallucination au sens classique des LLM. C'est une **hallucination institutionnelle**, plusieurs vérités incohérentes écrites dans les systèmes par plusieurs humains et agents.

La solution est simple en principe : la mémoire de l'entreprise doit rester aussi proche que possible du temps réel, et chaque champ important doit avoir une source d'autorité déclarée. Si quelqu'un met à jour le CRM, le cockpit doit le savoir rapidement ; si Slack, l'email et le CRM sont en désaccord, le système doit savoir quelle source est autorisée à décider.

Le modèle d'adoption : connecter, observer, agir, étendre

La migration vers un OS agentique pour une petite équipe devrait suivre un rythme naturel :

01	**Ingestion (jours 1-2)** Branchez vos outils existants (CRM, e-mail, Slack, documents, facturation...). L'OS ingère vos données et commence à construire et structurer la mémoire de votre entreprise. Vous ne changez encore rien à votre façon de travailler.
02	**Observation (semaines 1-3).** Utilisez le cockpit en mode lecture seule. Posez des questions, mettez en lumière les conflits, voyez ce que le système comprend de votre entreprise. C'est ici que vous découvrez les lacunes, les données obsolètes et les contradictions. Aucun risque d'exécution, et opportunité d'aider le système à améliorer sa compréhension de votre entreprise.

03

Mise en pratique (semaines 3-4).

Demandez à l'OS de sélectionner un domaine qui pose clairement problème : triage des ventes entrantes, suivi client, relance des factures, reporting hebdomadaire. Laissez le cockpit proposer des plans et exécuter de manière autonome des actions à faible risque, tandis que vous approuvez tout ce qui est sensible.

04

Élargissement (mois 2+).

À mesure que la confiance s'installe, élargissez le périmètre. Laissez l'OS ajouter d'autres domaines, laissez le système gérer plus d'actions routinières de manière autonome, et cessez progressivement de réaliser ce travail directement dans vos outils SaaS.

05

Migration

À un moment donné, vous réalisez que vous n'avez pas ouvert votre CRM directement depuis deux semaines. Il fonctionne toujours, il contient toujours des données, mais vous et votre équipe effectuez naturellement le travail via le cockpit, sans plus jamais vous connecter directement à l'outil.

Le côté politique

Dans une grande entreprise, la migration est politiquement sensible. Elle menace les rôles du middle management et les baronnies de l'information. Dans une petite entreprise, les frictions sont plus légères mais toujours réelles :

- La personne qui « possède le fichier Excel » peut se sentir exposée lorsque ce savoir devient une mémoire partagée.
- Les membres de l'équipe à l'aise avec leurs routines actuelles peuvent résister à changer leurs habitudes.
- Si le dirigeant n'adopte pas visiblement d'abord le cockpit, personne d'autre ne le fera.

La solution consiste à montrer l'exemple, en commençant par un domaine qui fait clairement gagner du temps à tout le monde, et à rendre les premières réussites visibles rapidement. Une fois qu'une boucle fonctionne nettement mieux via le cockpit, l'adoption a alors tendance à s'auto-entraîner.

§08

LIMITATIONS ET QUESTIONS OUVERTES

▲ HONEST

L'OS agentique n'est encore qu'un projet et pas vraiment quelque chose que vous pouvez déployer aujourd'hui. Il reste encore un certain nombre de limitations et de questions ouvertes à résoudre.

Qu'en est-il des coûts ?

Pour une entreprise de 10 personnes utilisant un tel OS quotidiennement, on peut s'attendre à des coûts d'inférence LLM de 200 à 2 000 €/mois selon les volumes de données et le choix du modèle. Auquel il faut rajouter le coût de l'OS lui-même et les abonnements SaaS qui, au moins pour le moment, sont encore nécessaires sous le système.

À première vue, cela peut ressembler à une charge opérationnelle supplémentaire, surtout pour de petites entreprises déjà saturées par les coûts logiciels. La valeur économique se trouve pourtant dans la compression des frais administratifs, la réduction des changements de contexte et la possibilité, pour les dirigeants et employés, de se concentrer sur des tâches à plus forte valeur ajoutée : ventes, exécution, relation client, prise de décision et croissance.

Pour beaucoup de propriétaires de petites entreprises, souvent débordés par la complexité opérationnelle et hésitants sur la marche à suivre en matière d'IA, cela peut au contraire constituer un investissement technologique à fort retour sur investissement.

Cela peut-il fonctionner pour tout le monde ?

Ce modèle convient mal aux entreprises dont l'activité est intrinsèquement non procédurale (agences créatives, R&D très amont). Il est au contraire particulièrement efficace là où les workflows répétitifs existent, mais sont encore répartis entre la tête des équipes et une multitude d'outils.

Confidentialité, RGPD et confiance des employés

Un cockpit qui intègre les e-mails, Slack, les documents, les enregistrements CRM, les données de facturation et les événements du calendrier peut rapidement donner l'impression d'une surveillance s'il est déployé sans précaution. Pour les PME européennes, ce n'est pas seulement une préoccupation culturelle, mais aussi juridique : le RGPD impose des obligations claires concernant la limitation des finalités, la minimisation des données, la base légale, la conservation et le droit à l'effacement, dont aucune ne correspond parfaitement à une couche de mémoire agentielle toujours active.

Avant le déploiement, une entreprise a besoin de réponses explicites à un petit ensemble de questions : ce qui est ingéré, ce qui est exclu par défaut (DM privés, courriels personnels, conversations RH), qui peut interroger quoi, combien de temps la mémoire est conservée, comment les demandes de suppression se propagent dans le graphe de mémoire, et si des données quittent l'UE via les fournisseurs de modèles. Les accords de traitement des données avec les fournisseurs de LLM, les chaînes de sous-traitants et les mécanismes de transfert transfrontalier doivent tous être réglés avant que le cockpit n'accède aux données de production.

La fiabilité et la restauration doivent être intégrées dès la conception

L'exécution agentique échouera. Les agents interpréteront mal le contexte, agiront sur des données obsolètes, choisiront le mauvais outil, déclencheront la bonne action au mauvais moment, ou enchaîneront plusieurs petites erreurs à la suite pour déboucher sur une catastrophe.

Un système d'exploitation agentique nécessite donc de la mise en place de possibilité de retour en arrière (rollback) dès le début. Chaque action exécutée doit être enregistrée avec suffisamment de contexte pour être expliquée, annulée ou compensée. Les actions irréversibles (paiements, suppressions, communications externes, modifications de contrats) nécessitent des voies d'escalade explicites, des modes d'exécution "à blanc" et une possibilité de reprise de contrôle manuel.

La responsabilité reste non résolue

La plupart des déploiements agentiques reposent encore sur des contrats de l'ère SaaS conçus pour des logiciels passifs, et non pour une exécution autonome. Si un agent fixe un prix incorrect pour un produit, envoie un message trompeur au client ou effectue une action causant des dommages, la responsabilité incombera souvent à l'entreprise déployant l'agent, même si l'erreur provient du raisonnement de l'agent.

C'est une question majeure ouverte pour un OS agentique. Les entreprises auront besoin d'une responsabilité interne claire pour les résultats des agents, et les fournisseurs auront besoin de contrats reconnaissant les agents comme des acteurs opérationnels, et non simplement comme des fonctionnalités logicielles. Jusqu'à ce que ce modèle juridique mûrisse, les actions à fort impact doivent rester contrôlées, auditables et liées à une approbation humaine explicite.

État du marché

Nous sommes loin d'être les seuls à faire le constat de ce changement de paradigme. La plupart des acteurs majeurs s'orientent déjà dans cette direction, chacun sous un angle différent :

- **Salesforce** promeut « l'Entreprise Agentique » via Agentforce 360, avec Data 360 comme contexte et Slack comme surface de travail agentique collaborative. [09]
- **Microsoft** étend Copilot Studio à l'orchestration multi-agents à travers Microsoft 365, Fabric, SDKs, et les protocoles Agent-à-Agent. [10]
- **ServiceNow** développe des capacités de Control Tower IA pour gouverner et surveiller les équipes d'agents. [11]
- **Palantir** présente depuis longtemps AIP, Foundry et Apollo comme un système d'exploitation pour les opérations pilotées par l'IA. [12]
- **SAP** et **Workday** transforment les systèmes centraux de références en plateformes d'agents et en couches de gouvernance des agents. [13]:
- **UiPath** étend l'automatisation vers un plan de contrôle agentique via Maestro. [14]
- **Atos**, en Europe, présente l'IA agentique comme une infrastructure de production souveraine via les Sovereign Agentic Studios. [15]

Ces initiatives confirment que l'IA agentique devient un modèle opérationnel à part entière. Mais la plupart des approches restent dirigées par des grands groupes, limitées à un écosystème fermés, fortement axées sur le conseil, ou liées à un système SaaS spécifique.

§09

CONCLUSION

▲ END

Pour un dirigeant d'entreprise, la question n'est plus vraiment de savoir si vous devez utiliser l'IA ou non. Vous avez déjà la réponse, et vous savez très bien que oui car vous n'avez pas le choix. La vraie question devient plutôt *ok, mais je fais comment ?*.

Vous avez quatre choix :

01

Outiller une personne à la fois

Vous pouvez donner à chacun des collègues de votre entreprise un accès à une IA généraliste comme ChatGPT, Claude ou Gemini. C'est le plus simple : rédaction d'e-mails, résumé de documents, brainstorming, analyse de rapports, etc. Chacun avance plus rapidement individuellement. Mais après l'effet "waouh" il est difficile d'aller au-delà des cas d'usage basique.

02

Rendre chaque outil plus intelligent

Vos SaaS sont équipés d'agents IA métier que vous pouvez activer. Votre CRM devient plus intelligent, votre service d'assistance client devient plus intelligent, votre outil de comptabilité devient plus intelligent. Mais chaque agent reste enfermé dans son propre silo.

03

Automatiser et connecter les outils existants

Vous pouvez engager une agence pour créer des intégrations et des workflow IA spécifiques. Cela peut optimiser des flux manuels pénibles comme la qualification de prospects, les relances, le suivi des factures, les mises à jour CRM ou les rapports. Mais cela automatise souvent le désordre existant au lieu de le remplacer. Les outils restent fragmentés, la responsabilité demeure floue, et la mémoire de l'entreprise reste dispersée entre les e-mails, les documents, Slack, les feuilles de calcul et les champs SaaS. Le workflow est sympa, mais l'IA ne comprend toujours pas vraiment l'entreprise : elle reste cantonnée au couloir étroit que l'intégration lui offre.

04

Changer la surface de travail elle-même

Vous pouvez évoluer vers un OS agentique : un cockpit unique où l'entreprise est vue, comprise et pilotée.

L'option 4 est, de loin, la trajectoire la plus intéressante et la plus prometteuse. Ce n'est évidemment pas la plus simple, mais renoncer à la tenter reviendrait à passer la

prochaine décennie à continuer à être un auto-clicker humain entre nos différents outils SaaS.

Notes de bas de page

[01]: Selon l'analyse de NimbleBrains du registre complet MCP, publiée en mars 2026 : « L'état de la sécurité MCP en 2026 » https://nimblebrain.ai/mcp/mcp-security/state-of-mcp-security/

[02]: Selon l'analyse des déploiements d'IA en production par Obsidian Security, l'injection de prompt apparaît dans plus de 73 % des déploiements d'IA en entreprise évalués. Obsidian Security, "Attaques par injection de prompt : l'exploitation IA la plus courante en 2025" https://www.obsidiansecurity.com/blog/prompt-injection

[03]: Le CISO d'OpenAI, Dane Stuckey, a déclaré en octobre 2025 lors du lancement du navigateur ChatGPT Atlas : « l'injection de prompt reste une frontière, un problème de sécurité non résolu, et nos adversaires consacreront beaucoup de temps et de ressources à trouver des moyens de faire tomber l'agent ChatGPT face à ces attaques. » Source : Simon Willison, « Dane Stuckey (CISO d'OpenAI) sur les risques d'injection de prompt » https://simonwillison.net/2025/Oct/22/openai-ciso-on-atlas/

[04]: Le problème du mandataire trompé a été initialement décrit par Norm Hardy dans "The Confused Deputy (or why capabilities might have been invented)" Operating Systems Review, ACM, octobre 1988. https://people.cs.vt.edu/~kafura/cs6204/Readings/ConfusedDeputy.pdf ; Le passage de jetons est explicitement interdit par la spécification officielle MCP, qui stipule : « Si le serveur MCP accepte non seulement des jetons avec des audiences incorrectes mais transmet également ces jetons non modifiés aux services en aval, cela peut potentiellement causer le problème du 'mandataire trompé'. » Spécification officielle d'autorisation MCP, juin 2025 : https://modelcontextprotocol.io/specification/2025-06-18/basic/authorization ; Pour une analyse au niveau de l'implémentation de la vulnérabilité en conditions réelles, voir : Obsidian Security, « When MCP Meets OAuth: Common Pitfalls Leading to One-Click Account Takeover », février 2026. https://www.obsidiansecurity.com/blog/when-mcp-meets-oauth-common-pitfalls-leading-to-one-click-account-takeover

[05]: Lenovo CIO Playbook 2026 (4e édition annuelle), une étude commandée par Lenovo et réalisée par IDC, publiée en janvier 2026, portant sur plus de 3 000 décideurs IT et business dans le monde. Il en ressort que seulement 46 % des POC d'IA ont évolué vers un déploiement à l'échelle de la production. Lenovo Newsroom, janvier 2026. https://news.lenovo.com/pressroom/press-releases/research-reveals-ai-is-paying-off-cios-arent-ready/

[06]: Projet MIT NANDA (MIT Media Lab) a publié *The GenAI Divide: State of AI in Business 2025* en juillet 2025 (la couverture presse a commencé en août 2025), rédigé par Aditya Challapally, Chris Pease, Ramesh Raskar, et Pradyumna Chari. Méthodologie : revue systématique de plus de 300 initiatives d'IA publiquement divulguées, entretiens structurés avec des représentants de 52 organisations, et réponses à un sondage de 153 cadres supérieurs lors de quatre grandes conférences industrielles. Résultat précis du rapport : 95 % des solutions d'IA d'entreprise n'ont eu aucun impact mesurable sur le compte de résultat. Seulement 5 % des outils d'IA personnalisés pour l'entreprise ont atteint la production. Rapporté par Fortune, août 2025 : https://fortune.com/2025/08/18/mit-report-95-percent-generative-ai-pilots-at-companies-failing-cfo/

[07]: Enquête mondiale des PDG PwC n°29, « Diriger dans l'incertitude à l'ère de l'IA », publiée le 19 janvier 2026 à Davos. Réalisée du 30 septembre au 10 novembre 2025, auprès de 4 454 PDG dans 95 pays et territoires. Résultat exact : « plus de la moitié (56 %) déclarent n'avoir constaté ni bénéfices en termes de revenus ni en termes de coûts » liés à l'IA ; « seulement un sur huit (12 %) rapporte les deux » impacts positifs. Rapport complet : https://www.pwc.com/gx/en/ceo-survey/2026/pwc-ceo-survey-2026.pdf

[08]: Communiqué de presse Gartner, 25 juin 2025. « Gartner prévoit que plus de 40 % des projets d'IA agentique seront annulés d'ici fin 2027. » Anushree Verma, Analyste Directrice Senior, Gartner. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

[09]: Salesforce a lancé Agentforce 360 fin 2025, positionnant Slack comme l'interface frontale principale de son écosystème d'IA agentique. Agentforce dans Slack fonde les réponses des agents à la fois sur les données conversationnelles de Slack et les données CRM de Salesforce, permettant aux agents d'effectuer des

actions telles que la création et la mise à jour de canaux, listes et canevas directement dans le flux de travail. Sources : Salesforce EU, « Travailler avec des agents IA dans Slack grâce à Agentforce » https://www.salesforce.com/eu/slack/agentforce/

[10]: Microsoft Copilot Studio s'est étendu à l'orchestration multi-agents, avec des fonctionnalités atteignant la disponibilité générale en avril 2026. La plateforme permet aux agents créés dans Copilot Studio, Azure AI Foundry, Microsoft Fabric et le SDK Microsoft 365 Agents de collaborer à travers les flux de travail en utilisant des protocoles ouverts Agent-à-Agent (A2A). Source : Blog Microsoft Copilot, « Quoi de neuf dans Copilot Studio : mises à jour des systèmes multi-agents » https://www.microsoft.com/en-us/microsoft-copilot/blog/copilot-studio/new-and-improved-multi-agent-orchestration-connected-experiences

[11]: ServiceNow a lancé AI Control Tower lors de Knowledge 2025 (mai 2025) en tant que centre de commandement centralisé pour gouverner, gérer, sécuriser et valoriser tout agent, modèle et flux de travail IA. En mai 2026, ServiceNow a étendu AI Control Tower grâce à une intégration plus approfondie avec Microsoft Agent 365. Sources : ServiceNow Newsroom, « ServiceNow launches AI Control Tower at Knowledge 2025 » https://www.servicenow.com/uk/company/media/press-room/ai-control-tower-knowledge-25.html ; ServiceNow Newsroom, « ServiceNow expands AI agent governance through deeper integration with Microsoft » https://newsroom.servicenow.com/press-releases/details/2026/ServiceNow-expands-AI-agent-governance-through-deeper-integration-wi

[12]: Palantir décrit explicitement son architecture combinée AIP + Foundry + Apollo comme un « système d'exploitation d'entreprise ». Foundry sert de plateforme d'opérations de données, AIP comme plateforme d'IA générative, et Apollo comme plateforme de livraison continue. Source : Palantir, « Integrated platforms: AIP, Foundry, and Apollo » https://palantir.com/docs/foundry/architecture-center/platforms/

[13]: SAP a développé Joule pour inclure plus de 40 agents IA et plus de 2 400 compétences intégrées dans S/4HANA, Ariba, SuccessFactors et IBP, et a lancé un AI Agent Hub au sein de SAP LeanIX pour une gouvernance centrale des agents. Workday a annoncé son Agent System of Record en février 2025, offrant une couche de gouvernance unifiée pour gérer à la fois les agents IA de Workday et ceux de tiers. Sources : SAP News Center, « SAP Business AI : Points forts de la version T1

2026 » https://news.sap.com/2026/04/sap-business-ai-release-highlights-q1-2026/ ; Workday Newsroom, « Workday dévoile son nouveau Agent System of Record » https://newsroom.workday.com/2025-02-11-The-Next-Generation-of-Workforce-Management-is-Here-Workday-Unveils-New-Agent-System-of

[14]: UiPath Maestro™ est officiellement décrit comme « une plateforme d'orchestration cloud-native qui unifie l'automatisation, les agents IA et les interactions humaines en processus métier rationalisés et de bout en bout » utilisant BPMN (Business Process Model and Notation) et DMN (Decision Model and Notation) pour la modélisation visuelle des flux de travail et des règles. https://www.uipath.com/platform/agentic-automation/agentic-orchestration

[15]: Le groupe Atos a lancé Sovereign Agentic Studios le 12 mars 2026, en tant que nouveau modèle opérationnel pour aider les organisations à passer des pilotes d'IA agentique à la production sous contrôle souverain complet. Source : Communiqué de presse du groupe Atos, « Atos Group Launches Sovereign Agentic Studios » https://www.atosgroup.com/en/press/atos-group-launches-sovereign-agentic-studios-bring-ai-safely-production-across-organizations