Créez une équipe

de 20 personnes avec l'IA.

Livret débutant : démarrer en une semaine

Philippe Anel

2026

Ce livret accompagne la Masterclass IA « Créer une équipe de 20 personnes avec l'IA », donnée le 6 avril 2026.

Pour toute remarque ou suggestion : `support@scanderia.com`

Table des matières

Préface : Comment lire ce livret

Ce livret est né de 80 questions posées par 650 personnes un lundi soir de Pâques 2026. Si vous tenez ces pages entre vos mains, c'est que vous avez (ou aurez bientôt) les mêmes questions.

Pourquoi ce livret existe

La masterclass du 6 avril 2026 a duré 3h45. C'est long. C'est dense. Et beaucoup de choses importantes ont été dites *en passant*, au détour d'une réponse à une question, sans qu'on ait le temps d'y revenir.

Pire : une partie du public est arrivée sans avoir jamais ouvert un fichier `.md` de sa vie, sans savoir ce qu'est un IDE, sans comprendre la différence entre « IA en local » et « IA dans le cloud ». Et c'est normal, ce sont des sujets neufs, sans mode d'emploi, où les évidences des uns sont des mystères pour les autres.

Ce livret est **le mode d'emploi qui manquait**. Il ne suppose aucune connaissance préalable. Il vous donne les mots, les outils, les étapes. Et surtout, il vous donne **le droit de ne pas savoir**.

> **Le conseil le plus important de la masterclass**
>
> Plus vous êtes naïf quand vous parlez à l'IA, mieux c'est. Jason Paul Michaels, le soudeur qui a construit 50 applications en 9 mois, n'a jamais prétendu être développeur. Il a dit : *« je suis soudeur, toi tu es agent, aide-moi »*. Et ça a marché.

À qui s'adresse ce livret

Ce livret est pour vous si :

- Vous avez entendu parler des agents IA mais vous n'êtes pas encore sûr de ce que ça recouvre
- Vous avez suivi la masterclass et vous avez eu du mal à tout suivre
- Vous avez payé Claude ou ChatGPT mais vous aimeriez en tirer plus qu'un chatbot mais ne voyez pas comment
- Vous voulez automatiser des tâches dans votre business mais vous cherchez par où commencer
- Vous avez essayé d'ouvrir les fichiers du kit mc-files et vous n'avez pas compris ce qu'il fallait en faire

Ce livret n'est pas pour vous si :

- Vous êtes développeur expérimenté et vous cherchez du code avancé
- Vous voulez construire une architecture à 20 agents dès demain
- Vous cherchez du contenu sur le fine-tuning de modèles locaux

Pour ces cas-là, il y a un **livret intermédiaire** et un **livret avancé**. Vous pouvez y aller directement si vous avez déjà les bases. Pas besoin de lire le livret débutant de A à Z si vous savez déjà créer un fichier `.md` et piloter un agent.

> **Indépendants, freelances, salariés : une précision importante**
>
> Ce livret est écrit pour des **professionnels autonomes** : indépendants, freelances, dirigeants de petites structures, ou toute personne qui a la liberté d'installer des outils et de choisir ses méthodes de travail.
> Si vous êtes **salarié en entreprise**, la méthode reste la même (fichiers `.md`, agents, itération), mais vous aurez des contraintes supplémentaires : votre service informatique peut interdire l'installation de certains outils, votre entreprise peut imposer des solutions spécifiques ; l'utilisation de services cloud comme Claude peut nécessiter une validation interne (données sensibles, conformité, politique de sécurité). **Vérifiez avec votre DSI avant d'installer quoi que ce soit.** Le chapitre 2 propose des alternatives qui ne nécessitent aucune installation.

Comment utiliser ce livret

1 Lisez-le dans l'ordre

Les chapitres s'appuient les uns sur les autres. Le chapitre 4 (votre premier dossier) ne marchera que si vous avez compris les chapitres 1 à 3.

2 Ouvrez votre ordinateur

Ce n'est pas un livre de plage. À partir du chapitre 2, vous devez avoir Claude Cowork installé et être prêt à taper. Les chapitres 4 et 5 sont *des exercices*, pas de la théorie.

3 Acceptez de ne pas tout comprendre la première fois

C'est normal. Revenez en arrière. Posez la question à l'IA elle-même, elle est souvent un très bon professeur sur ces sujets.

4 Pratiquez. Beaucoup.

La règle des 20 heures (Josh Kaufman, *The First 20 Hours*, 2013) : comptez vingt heures de pratique avant de vous sentir à l'aise. Pas vingt heures d'affilée, vingt heures cumulées sur deux ou trois semaines. C'est ça qui fait la différence entre « j'ai suivi une masterclass » et « je sais faire ».

Ce que ce livret contient (et ne contient pas)

Ce qu'il contient

- Les définitions claires des mots qu'on utilise tout le temps (agent, mc-file, MCP, HITL, SOP…)
- Les étapes concrètes pour installer et lancer vos premiers outils
- Un exercice guidé : *votre premier dossier agentique de A à Z*
- Les 80 questions du public classées par thème, avec des réponses utilisables
- Les erreurs fréquentes et comment les éviter

Ce qu'il ne contient pas

- **Les sujets avancés** : orchestration de 20 agents, déploiement en production, gouvernance approfondie, optimisation des tokens. Tout ça est dans le livret avancé.
- **Les détails techniques** : comment fonctionne un LLM, c'est quoi l'attention quadratique, comment quantiser un modèle. Pas nécessaire pour démarrer.
- **Les recettes « clé en main »** pour votre métier spécifique. Chaque métier est différent ; ce livret vous apprend la méthode, pas la recette.

> **Un mot sur l'honnêteté**
>
> Ce livret ne vous promet pas que vous allez « doubler vos revenus avec l'IA en 7 jours ». Il vous promet que, si vous suivez les étapes et que vous pratiquez, vous saurez créer un système IA *qui vous sert vraiment*.

Prêt ? Tournez la page.

Chapitre 1

C’est quoi un agent, vraiment ?

C’est la question ° 1 que tout le monde pose. Si ce mot vous semble flou pour l’instant, c’est normal : il le sera beaucoup moins dans cinq minutes.

La définition en une phrase

> **Retenez juste ceci**
>
> **Un agent, c’est une IA qui a revêtu un dossier.**

C’est *la* bonne définition. Mais comme « revêtu un dossier » ne veut probablement rien dire pour vous, déroulons.

La métaphore qui marche : l’uniforme

Imaginez une IA comme un **corps énergétique, un génie, un esprit sans forme**. Elle peut *potentiellement* tout faire : écrire un mail, analyser un contrat, dessiner un logo, coder un site. Mais elle ne sait rien de *vous*, de *votre* métier, de *votre* façon de travailler.

Maintenant, imaginez que vous lui donnez un uniforme de policier : casquette, matraque, menottes, badge. Et un manuel de procédure. **Elle devient un policier.** Pas parce qu’elle a « appris à être policier », mais parce qu’elle *lit* le manuel à chaque interaction et qu’elle *applique* ce qui est écrit dedans.

Dans notre monde, l’uniforme, c’est un dossier sur votre ordinateur. Le manuel, ce sont des fichiers `.md` dans ce dossier. Et la « policière » qui revêt l’uniforme, c’est Claude (ou une autre IA) qui lit le dossier avant de répondre à votre question.

Un agent en action dans Claude Cowork : la conversation, les fichiers `.md` à gauche, le résultat à droite.

> *« L’IA, c’est un corps éthéré, comme un génie, un corps énergétique qui peut tout faire. Et une fois qu’elle a enfilé l’uniforme, elle devient un chef cuisinier, un avocat, un policier… »*
>
> *Idriss Aberkane, masterclass du 6 avril 2026*

FIG. 1.1 : Le cycle de travail avec un agent : vous spécifiez, l'agent exécute, vous corrigez.

Pourquoi c'est différent de ChatGPT « normal »

Quand vous ouvrez ChatGPT, Claude ou Grok dans votre navigateur et que vous tapez une question, voilà ce qui se passe :

1. L'IA lit votre question.
2. Elle cherche dans tout ce qu'elle « connaît » (ce qu'elle a appris pendant son entraînement).
3. Elle vous répond avec ce qu'elle a trouvé.

C'est rapide. C'est pratique. Mais c'est générique. L'IA ne sait rien de votre business, de votre style, de vos clients. Elle vous donne la même réponse qu'elle donnerait à n'importe qui d'autre avec la même question.

Maintenant, quand vous travaillez avec un agent (une IA qui a revêtu un dossier), voilà ce qui se passe :

1. L'IA lit votre question.
2. Elle lit d'abord **votre dossier** : qui vous êtes, ce que vous faites, comment vous voulez qu'elle vous parle, ce qu'elle a le droit et le devoir de faire.
3. Elle répond en tenant compte de tout ça.

La différence de qualité est franche. C'est la différence entre demander à un inconnu dans la rue « comment je développe ma boîte ? » et demander la même chose à un consultant qui a étudié votre entreprise pendant 3 mois.

La métaphore du boulanger

Cette image est celle qui parle le plus.

Si vous entrez dans une boulangerie et que vous dites « je voudrais de la farine pour faire du pain », le boulanger amateur vous tend son paquet de T65 du supermarché. Le boulanger professionnel, lui, vous demande : « *vous voulez quel type de pain ? Blanc, complet, au levain, avec du seigle ? Pour combien de personnes ? Vous avez une machine à pain ou vous pétrissez à la main ?* » Et il vous propose une farine double zéro taux de gluten élevé, à la meule de pierre, importée de Sicile.

La différence entre les deux, c'est la **résolution**. Le professionnel travaille avec une précision que l'amateur n'a pas. Il pose des questions spécifiques parce qu'il connaît les détails qui changent tout.

Un agent bien configuré, c'est exactement ça : une IA à **haute résolution**, qui connaît les détails de *votre* contexte et qui peut répondre avec une précision que ChatGPT générique n'a pas.

Un agent = un fichier ou un dossier ?

C'est une question qui revient tout le temps. La réponse courte :

> **À retenir**
>
> **Un agent, c'est soit un seul fichier `.md`, soit un dossier complet.** Ça dépend de la précision que vous voulez.

Niveau 1 : Un seul fichier

C'est le niveau débutant. Vous créez un fichier `assistant_comptable.md` dans lequel vous écrivez :

```
# Assistant comptable

Tu es mon assistant comptable.
Tu m'aides à préparer ma déclaration TVA.
Tu parles en français, de façon concise.
Tu signales toujours quand tu n'es pas sûr.
```

C'est déjà un agent. Suffisant pour beaucoup d'usages. Commencez là.

Niveau 2 : Un dossier complet

Quand votre agent devient important (par exemple, *« réponds à mes emails clients »*), un seul fichier ne suffit plus. Vous créez un dossier `agent_email/` qui contient :

- `identite.md` : qui est cet agent, comment il parle
- `contexte_entreprise.md` : votre business, vos clients, votre offre
- `bonnes_reponses.md` : des exemples de ce qui marche
- `a_ne_jamais_faire.md` : les règles absolues (ne jamais promettre ça, ne jamais répondre à ci...)
- `workflow.md` : comment il doit traiter un mail entrant

C'est le niveau intermédiaire/avancé. Ne commencez pas par là.

Niveau 3 : Un dossier projet complet avec plusieurs agents

C'est l'architecture avancée : un dossier qui contient 20 sous-dossiers, un par agent, plus des fichiers communs (la « Bible » de l'entreprise, les règles de gouvernance, etc.). **Réservez ça pour quand vous aurez maîtrisé le niveau 1 et 2.**

Ce qu'un agent n'est pas

> **Attention aux confusions classiques**
>
> Trois confusions à dissiper avant d'aller plus loin.

Un agent n'est pas une IA qui vit toute seule sur votre ordinateur. L'IA, elle, tourne chez Anthropic (ou OpenAI, ou Mistral…). Votre dossier est en local sur votre machine. L'IA lit votre dossier à chaque session et applique ce qui est écrit.

Un agent n'est pas un programme que vous lancez. C'est un *contexte* que vous donnez à une IA. Vous ne « lancez » pas un agent, vous ouvrez une session Claude Cowork dans son dossier et vous lui parlez.

Un agent n'est pas intelligent tout seul. Il est aussi précis que votre spécification. Si vous écrivez un dossier flou, vous aurez un agent flou. Si vous écrivez un dossier précis, vous aurez un agent précis. C'est *votre* travail, pas celui de l'IA.

Ce qu'il faut retenir de ce chapitre

- **Un agent = une IA + un dossier.** Pas plus compliqué.
- Le dossier contient des fichiers `.md` qui décrivent *qui* est l'agent, *ce qu'*il doit faire, *comment* il doit le faire.
- Un agent peut être un seul fichier (débutant) ou un dossier entier (avancé).
- La différence avec ChatGPT classique, c'est la **résolution**, c'est-à-dire à quel point vous avez spécifié le contexte.
- L'IA reste chez son fournisseur. Vos fichiers restent chez vous.

> **Exercice avant le prochain chapitre**
>
> Prenez 5 minutes. Fermez les yeux. Imaginez **un seul agent** qui vous serait utile dans votre vie. Un seul. Celui qui vous ferait gagner le plus de temps. Écrivez en 3 phrases qui il est et ce qu'il doit faire. C'est votre premier brouillon de mc-file.

Chapitre 2

Les outils dont vous avez besoin

Bonne nouvelle : il vous faut très peu de choses. Un ordinateur, un abonnement et un éditeur de texte. C'est tout. Voici ce qu'il faut installer et combien ça coûte vraiment.

Vous êtes salarié en entreprise ?

Lisez ceci avant d'aller plus loin. En entreprise, vous ne pouvez pas toujours installer les outils que vous voulez, ni envoyer des données professionnelles vers un service cloud non validé par votre direction informatique. C'est une réalité ; l'ignorer est une des premières causes d'échec des projets.
Avant d'installer quoi que ce soit, vérifiez avec votre service informatique ce qui est autorisé. Si Cowork ou Claude ne sont pas validés, vous pouvez quand même apprendre la méthode : les fichiers `.md` sont de simples fichiers texte que vous pouvez créer avec n'importe quel éditeur déjà présent sur votre poste. Le jour où votre entreprise ouvrira l'accès à un outil agentique (et ça viendra), vos fichiers seront prêts.
Même si `claude.ai` fonctionne sans installation dans un navigateur, vérifiez la politique de votre entreprise avant d'y envoyer des données professionnelles. En cas de doute, parlez-en à votre service informatique d'abord.

Ce qu'il vous faut absolument

1. ### Un ordinateur (Mac, Windows ou Linux)

   Pas besoin d'une machine de gamer. N'importe quel ordinateur acheté dans les 5 dernières années suffit. Les outils qu'on utilise tournent dans le cloud : votre machine sert juste d'interface.
   Tablette / iPad : techniquement possible mais déconseillé. C'est moins ergonomique pour éditer des fichiers et naviguer entre plusieurs fenêtres. Commencez sur ordinateur.

2. ### Un abonnement Claude (ou équivalent)

   C'est le point central. On explique en détail ci-dessous.

3. ### Un éditeur de texte capable d'ouvrir des fichiers `.md`

   Vous en avez probablement déjà un sans le savoir. Voir plus bas.

Claude Cowork, l'outil principal

Ce livret utilise **Claude Cowork**. C'est une application qui tourne sur Mac et Windows, qui permet à Claude de travailler dans un dossier de votre ordinateur. C'est *l'outil* qui rend possible tout ce qu'on fait dans ces pages.

Ce livret parle de Claude, mais la méthode est universelle

Tout ce que vous apprenez ici (créer des fichiers `.md`, structurer un dossier d'agent, itérer sur les instructions) fonctionne avec **n'importe quel système d'IA agentique** : OpenAI Codex, Google Gemini CLI, des outils open source ou même des modèles que vous faites tourner sur votre propre machine. Les fichiers s'appellent `CLAUDE.md` dans l'écosystème Claude, `AGENTS.md` dans d'autres, mais le principe est identique : un fichier texte qui dit à l'IA qui elle est et comment elle doit travailler.
On utilise Claude Cowork parce que c'est aujourd'hui le plus accessible pour un débutant. Si demain un autre outil est meilleur, vos fichiers `.md` restent : vous les ouvrez dans le nouvel outil et vous repartez. **Votre investissement est dans la méthode, pas dans l'outil.**

Pourquoi Cowork et pas le Claude dans le navigateur ?

Quand vous ouvrez `claude.ai` dans votre navigateur, vous pouvez lui parler et lui envoyer des fichiers. Mais **il ne peut pas lire un dossier sur votre ordinateur**. Il ne peut pas non plus écrire dans vos fichiers. C'est un chat, rien de plus.

Claude Cowork, c'est différent. C'est une application que vous installez sur votre Mac ou votre PC et qui donne à Claude l'autorisation de **lire et écrire** dans un dossier que vous avez choisi. C'est cette capacité qui transforme une simple conversation en environnement agentique.

Attention à la confusion Cloud / Claude / Clawd

Claude (C-L-A-U-D-E, le modèle d'Anthropic) n'est *pas* la même chose que **Clawd** (C-L-A-W-D, un outil tiers qui n'a rien à voir). Une personne a perdu tous ses fichiers Meta en laissant Clawd visiter son ordinateur sans sandbox[a]. Quand on dit « Claude » dans ce livret, c'est toujours le vrai Claude d'Anthropic.

[a] **Sandbox** (bac à sable) : un mécanisme de sécurité qui limite ce qu'un programme peut faire sur votre ordinateur. Quand un outil est « sandboxé », il ne peut accéder qu'aux dossiers que vous lui autorisez ; il ne peut pas fouiller dans le reste de votre machine. Claude Cowork fonctionne comme ça.

Comment installer Claude Cowork

1. **Allez sur `claude.ai`**
   Créez un compte si vous n'en avez pas. Prenez au minimum l'abonnement **Pro**. Les versions gratuites ne suffisent pas pour travailler en agentique (elles sont rapidement épuisées). Si vous prévoyez un usage intensif (des agents qui tournent plusieurs heures par jour), regardez plutôt les offres **Max** : la Pro atteint vite ses limites elle aussi.

2. **Téléchargez Claude Cowork**
   L'application est disponible dans les paramètres de votre compte Claude ou directement sur le site Anthropic. Installez-la comme n'importe quelle application.

3. **Lancez-la et connectez-la à votre compte**
   Elle vous demandera l'autorisation d'accéder à un dossier. **Ne lui donnez pas accès à tout votre disque.** Créez un dossier dédié (par exemple `~/Documents/mon-projet-ia`) et donnez-lui accès uniquement à ce dossier.

Les alternatives à Claude Cowork

Claude Cowork n'est pas le seul outil qui fait ça. Il y en a d'autres et la méthode marche avec tous. Voici les principaux :

Codex (OpenAI)

C'est l'équivalent OpenAI de Claude Cowork. Ça fonctionne avec ChatGPT. Interface similaire, même logique. **Choisissez Codex si** vous êtes déjà investi dans l'écosystème OpenAI.

Google Antigravity

C'est l'outil Google dans le même esprit. **Choisissez Antigravity si** votre workflow est déjà dans Google Workspace.

Claude Code

Version ligne de commande de Claude pour développeurs. **Plus puissant mais plus technique.** Si vous ne savez pas ce qu'est un terminal, ignorez pour l'instant.

Modèles locaux (Ollama + Mistral, Qwen, DeepSeek…)

Il est possible de faire tourner une IA entièrement sur votre machine, sans passer par le cloud. **Avantages :** confidentialité totale et c'est gratuit (les modèles open source comme Mistral, Qwen et DeepSeek ne coûtent rien à utiliser, pas d'abonnement, pas de coût par token). **Inconvénients :** nécessite un bon hardware[1], c'est lent, la qualité est moindre. **À réserver aux cas sensibles** (données client, juridique) et à quand vous aurez de la pratique.

Le conseil pragmatique

Pour votre première fois : **prenez Claude Cowork**. C'est le plus fluide, le mieux documenté et le moins susceptible de vous décourager. Vous basculerez vers d'autres outils plus tard si vos besoins l'exigent.

Et si je veux tout garder en local ?

C'est possible et c'est légitime, surtout pour des raisons de confidentialité. Mais c'est un sujet à part entière : choix du modèle, configuration matérielle, performance, compromis qualité. **Scanderia prépare du contenu dédié sur ce sujet.** En attendant, commencez avec un outil en ligne pour apprendre la méthode. Vos fichiers `.md` fonctionneront aussi bien en local le jour où vous basculerez.

Et les API[a] ? Je dois connecter des trucs ?

[a] **API** (Application Programming Interface) : c'est un « pont » qui permet à deux logiciels de se parler. Quand Claude envoie un email via Gmail ou crée un visuel dans Canva, il utilise l'API de ces services. Mais vous n'avez pas à savoir comment ça marche : c'est Claude qui s'en sert, pas vous. C'est comme l'électricité dans les murs : vous appuyez sur l'interrupteur, vous n'avez pas besoin de comprendre le câblage.

Non. Vous n'avez pas à connecter des API, à copier des clés, à configurer des webhooks. Et pas une ligne de code à écrire. C'est le travail de l'IA, pas le vôtre.
Concrètement, si vous voulez que votre agent utilise Canva, Gmail, Slack ou n'importe quel autre outil, vous lui dites simplement dans votre fichier `.md` : « *Connecte-toi à Canva et génère un visuel pour chaque post* ». Ou encore plus directement dans la conversation : « *Ajoute-toi un connecteur pour Gmail et débrouille-toi* ». Claude se charge d'installer ce qu'il faut. Vous, vous décrivez ce que vous voulez. Lui, il fait la plomberie.
C'est tout l'intérêt de la méthode : **vous spécifiez, l'IA exécute**. Le livret intermédiaire détaille les connecteurs disponibles, mais même là, le principe reste le même.

[1] **Hardware** : le matériel physique de votre ordinateur. Processeur, carte graphique, mémoire. Les modèles d'IA locaux ont besoin d'un ordinateur puissant (notamment une bonne carte graphique) pour tourner à une vitesse acceptable.

Combien ça coûte vraiment ?

> **Mention légale sur les tarifs**
>
> Les prix indiqués ci-dessous sont ceux observés en avril 2026. Ils sont susceptibles d'évoluer sans préavis. Avant toute souscription, vérifiez les tarifs en vigueur sur les sites officiels (`anthropic.com`, `openai.com`). Scanderia ne saurait être tenue responsable des écarts entre les montants mentionnés ici et les tarifs appliqués au moment de votre achat.

Soyons honnêtes : les outils gratuits ne suffisent pas pour faire ce qu'on vous montre dans ce livret. Voici les vrais chiffres.

Abonnement Claude Pro

- **~18 € / mois** en paiement mensuel
- **~15 € / mois** en paiement annuel (environ 180 € l'année)
- C'est le minimum pour travailler sérieusement en agentique

Abonnement OpenAI Plus (équivalent)

- **20 $ / mois** (~19 €)
- Si vous préférez Codex

Et pour plus d'utilisation ?

Si vous devenez un gros utilisateur (plusieurs heures par jour, beaucoup de données), vous passerez à des formules entre **50 € et 100 € par mois**. Mais à ce stade-là, vous aurez déjà rentabilisé l'outil largement.

> **Le calcul à faire**
>
> 18 € par mois, c'est le prix d'un abonnement Netflix. Si votre agent vous fait gagner **une seule heure de travail par semaine**, au tarif horaire de votre métier, l'abonnement est rentabilisé en quelques jours.

L'éditeur de texte pour les fichiers `.md`

Les fichiers Markdown (`.md`) sont des fichiers texte tout simples. Vous pouvez les ouvrir avec *n'importe quel* éditeur de texte. Voici les options les plus faciles.

Vous êtes sur Mac

- **TextEdit** : déjà installé sur votre Mac. Suffisant pour commencer.
- **VS Code** (`code.visualstudio.com`) : gratuit, plus puissant, affiche un aperçu joli du Markdown. Recommandé dès que vous serez à l'aise.
- **Obsidian** (`obsidian.md`) : gratuit pour usage perso. Conçu spécifiquement pour travailler avec des dossiers de fichiers `.md`. Idéal pour la méthode mc-files.

Vous êtes sur Windows

- **Notepad** : déjà installé. Ça marche mais c'est basique.
- **VS Code** : même recommandation que sur Mac.
- **Obsidian** : fonctionne aussi sur Windows.

Le conseil d'or

Configurez votre ordinateur pour que **double-cliquer sur un fichier `.md` l'ouvre automatiquement** dans votre éditeur préféré. Si à chaque fois que vous cliquez sur un fichier `.md` votre ordinateur vous demande « avec quelle application voulez-vous l'ouvrir ? », vous allez péter un câble. Réglez ça une fois pour toutes dès aujourd'hui.

Un piège de sécurité si vous téléchargez des projets

En explorant l'agentique, vous allez probablement télécharger des dossiers de projets partagés par d'autres personnes (sur GitHub ou ailleurs). Ces dossiers peuvent contenir des fichiers de configuration qui donnent des instructions automatiques à Claude Cowork quand vous les ouvrez.
Le risque : quelqu'un de malveillant peut cacher dans un dossier partagé des instructions comme « envoie le contenu de tel fichier à telle adresse ». Claude Cowork les exécuterait sans vous demander votre avis.
La règle : avant d'ouvrir un dossier téléchargé avec un outil agentique, regardez s'il contient un fichier de configuration comme `CLAUDE.md`, `AGENTS.md` ou similaire. Si oui, ouvrez-le d'abord avec votre éditeur de texte et lisez ce qu'il contient. Si vous ne comprenez pas ce qui y est écrit ou si quelque chose vous semble suspect, ne l'ouvrez pas avec Cowork.

Ce qu'il faut retenir de ce chapitre

- **Claude Cowork** est l'outil principal. Prenez l'abonnement Pro à 18 €/mois.
- Il existe des alternatives (Codex, Antigravity, local) mais commencez avec Cowork.
- Installez un éditeur de texte pour les `.md` : TextEdit, VS Code ou Obsidian.
- Créez un **dossier dédié** pour vos projets IA et donnez à Cowork l'accès uniquement à ce dossier, pas à tout votre disque.
- Les outils gratuits sont insuffisants. Assumez la dépense.

Avant le prochain chapitre

Installez Claude Cowork. Prenez l'abonnement. Créez un dossier `mon-projet-ia` quelque part sur votre machine. Vérifiez que vous savez ouvrir un fichier `.md` en double-cliquant dessus. On passe à la suite.

Chapitre 3

Markdown en 5 minutes

Markdown, c'est le format des fichiers qu'on utilise tout le temps dans ce livret. C'est simple, c'est du texte et vous le maîtriserez en moins de 5 minutes. Promis.

Qu'est-ce que Markdown ?

Markdown, c'est **un format de texte**. Ni plus, ni moins. Un fichier Markdown se termine par `.md` (comme `identite.md` ou `projet.md`) et contient du texte tout simple, lisible à l'œil nu.

La différence avec un fichier Word ou PDF, c'est qu'un fichier Markdown **n'a pas de mise en page cachée**. Pas de polices, pas de marges, pas de boutons invisibles. Juste du texte, avec quelques signes discrets pour indiquer ce qui est un titre, ce qui est en gras, ce qui est une liste.

Pourquoi les IA adorent Markdown

Si vous ouvrez un fichier Word dans un éditeur de texte brut, vous verrez une bouillie illisible avec des milliers de caractères bizarres. C'est parce que Word enrobe votre texte dans du code de mise en page. **Les IA n'aiment pas ça** : elles préfèrent le texte pur.

Markdown, c'est **le meilleur des deux mondes** : c'est lisible par un humain (vous pouvez ouvrir le fichier et comprendre) et c'est lisible par une IA (rien ne la parasite). C'est pour ça que tout l'écosystème agentique tourne autour de ce format.

> **Et mes documents existants en Word, PDF, PowerPoint ?**
>
> Ils ne sont pas perdus. Vous pouvez demander directement à Claude de lire un fichier Word ou PDF et d'en extraire le contenu en markdown. Pour des documents simples (texte, listes, tableaux), ça fonctionne bien. Pour des cas plus complexes (gros PDF scannés, tableurs Excel avec formules, présentations riches), il existe des outils de conversion spécialisés que le livret intermédiaire abordera.
> L'essentiel à retenir pour l'instant : **l'IA travaille mieux avec du texte brut**. Convertir vos documents importants en `.md`, c'est leur donner le format que l'IA comprend le mieux. Commencez par créer vos nouveaux fichiers directement en markdown et convertissez l'existant au fur et à mesure de vos besoins.

Conseil : gardez toujours les originaux (PDF, images, scans, photos de tableau blanc) dans un sous-dossier `sources/` de votre projet. La conversion en `.md` ne capture pas tout : un graphique, une mise en page complexe, une annotation manuscrite peuvent se perdre en route. Si Claude a besoin d'un détail qu'il ne trouve pas dans le `.md`, il pourra relire le document original directement.

Petit test que vous pouvez faire

Ouvrez ChatGPT et demandez-lui : *« Écris-moi un résumé en 3 points sur la photosynthèse »*. Copiez la réponse et collez-la dans un fichier texte. Vous verrez qu'elle contient **déjà du Markdown** (astérisques autour des mots en gras, tirets pour les listes…). Les IA l'utilisent nativement.

Les 6 signes qu'il faut connaître

Markdown a beaucoup de fonctionnalités. **Vous n'en aurez besoin que de 6 pour ce livret.**

1. Les titres avec

Un signe # devant une ligne la transforme en gros titre. Deux ## pour un sous-titre. Trois ### pour un sous-sous-titre.

```
# Mon grand titre

## Un sous-titre

### Un sous-sous-titre
```

2. Le gras avec **

Deux astérisques autour d'un mot le mettent en gras.

```
C'est **très important** de retenir ça.
```

3. Les listes avec -

Un tiret en début de ligne crée une liste à puces.

```
- Premier point
- Deuxième point
- Troisième point
```

4. Les listes numérotées avec `1.`

Un chiffre suivi d'un point crée une liste ordonnée.

```
1. Première étape
2. Deuxième étape
3. Troisième étape
```

5. Les citations avec `>`

Un chevron en début de ligne crée une citation.

```
> Ceci est une note importante que tu dois retenir.
```

6. Le code inline avec les ‘ (backticks)

Un backtick de chaque côté d'un mot le met dans une police technique.

```
Pour lancer la commande, tape `ls -la`.
```

> **C'est tout.**
>
> Vous savez maintenant écrire du Markdown. Vraiment. Le reste, vous l'apprendrez par capillarité en lisant d'autres fichiers.

Un exemple complet de fichier `.md`

Voici à quoi ressemble un vrai fichier mc-file, avec tout ce qu'on vient de voir :

```
# Agent Rédacteur Newsletter

## Identité

Tu es mon rédacteur pour la newsletter mensuelle de mon salon de bien-être.

## Ton style

- Chaleureux mais professionnel
- Pas de jargon marketing
- Pas plus de 200 mots par édition
- Toujours une question ouverte à la fin

## Ce que tu dois faire

1. Lire le fichier `actualites.md` qui contient les infos du mois
```

```
2. Choisir les 3 sujets les plus pertinents pour les clients
3. Rédiger la newsletter dans le fichier `newsletter-mois-YYYY.md`

## Ce que tu ne fais jamais

- **Jamais** de promos flash type "vite, plus que 24h"
- **Jamais** plus de 3 sujets
- **Jamais** de tutoiement dans le corps du texte

> Note : Si tu n'as pas assez d'infos dans `actualites.md`, demande-moi ce qui
    manque avant de rédiger.
```

Vous voyez ? C'est juste du texte. Quelqu'un qui ne connaît pas Markdown peut le lire et comprendre l'essentiel. L'IA, elle, peut le lire[1] parfaitement et appliquer toutes les règles qui y sont écrites.

Comment créer un fichier `.md`

Sur Mac

1. Ouvrez **TextEdit**
2. Menu *Format* → *Convertir au format texte* (TRÈS IMPORTANT, sinon TextEdit va créer du `.rtf`)
3. Écrivez votre contenu
4. Menu *Fichier* → *Enregistrer*
5. Dans le dialogue : nom du fichier `mon-agent.md`, **décochez « Si aucune extension n'est fournie, utiliser .txt »**

Honnêtement, c'est pénible avec TextEdit. **Installez plutôt VS Code ou Obsidian** : avec eux, c'est un double-clic et c'est fini.

Sur Windows

1. Ouvrez **Notepad**
2. Écrivez votre contenu
3. Menu *Fichier* → *Enregistrer sous*
4. Dans *Type*, choisissez *Tous les fichiers*
5. Nommez le fichier `mon-agent.md`

Même conseil : **VS Code ou Obsidian** rendent ça beaucoup plus agréable.

[1] **Parser** : en informatique, « parser » signifie analyser un texte pour en extraire la structure (titres, listes, gras, etc.). Ici, cela veut simplement dire que l'IA est capable de lire le Markdown et d'en comprendre la mise en forme, pas juste les mots.

Le piège classique : l'extension cachée

Windows et macOS cachent souvent les extensions de fichier par défaut. Vous pensez avoir créé `mon-agent.md`, mais en réalité le fichier s'appelle `mon-agent.md.txt`. L'IA ne le verra pas comme un fichier Markdown.
Solution : allez dans les préférences de votre Finder ou Explorateur et activez « Afficher les extensions de fichier ». Réglez ça une fois pour toutes.

Conventions de nommage

Quand vous nommez vos fichiers mc-files, respectez ces règles :

- **Que des minuscules** : `identite.md`, pas `Identite.MD`
- **Pas d'espaces** : utilisez des tirets `agent-email.md`, pas `agent email.md`
- **Pas d'accents** : `reglementation.md`, pas `réglementation.md`
- **Pas de caractères spéciaux** : pas de `?`, `!`, &, etc.
- **Noms courts et parlants** : `ton.md` est mieux que `fichier-qui-decrit-le-ton-de-lagent-dans-les-situations-formelles.md`

Pourquoi ces règles ? Parce que certains outils et certaines IA gèrent mal les noms avec accents ou espaces. Autant prendre l'habitude tout de suite de faire simple.

Ce qu'il faut retenir de ce chapitre

- Un fichier Markdown (`.md`) est juste **du texte avec quelques signes discrets**
- 6 signes suffisent : `#`, `**`, `-`, `1.`, `>`, ‘
- Les IA adorent Markdown parce que c'est propre et sans parasites
- Nommage : minuscules, tirets, pas d'accents, pas d'espaces
- Méfiance avec les extensions cachées (`mon-agent.md.txt` est un piège)

Avant le prochain chapitre

Créez un fichier `test.md` n'importe où sur votre ordinateur avec le contenu suivant :

```
# Mon premier fichier Markdown

## Ce que je sais maintenant

- Markdown c'est du texte
- Les `#` font des titres
- Les `**` mettent en **gras**

> Je suis prêt pour le chapitre 4 !
```

Si vous arrivez à créer ce fichier, à le sauver avec l'extension `.md`, à le rouvrir et à le relire,

> vous êtes prêt pour la suite.

Chapitre 4

Votre premier dossier agentique

À partir de maintenant, on met les mains dans le cambouis. Dans ce chapitre, vous allez créer votre tout premier agent et le voir fonctionner. 30 minutes chrono.

Prérequis à vérifier avant de continuer

- ✓ Claude Cowork est installé et connecté à votre compte
- ✓ Vous savez créer un fichier `.md` et le sauver
- ✓ Vous avez lu le chapitre 1 (vous savez ce qu'est un agent)

Si un seul de ces points manque, revenez en arrière. Sinon, la suite ne va pas marcher et vous allez vous décourager pour rien.

Ce qu'on va faire ensemble

On va créer un **agent de relecture** : une IA à qui vous donnez un texte et qui vous signale les fautes et les formulations confuses, mais qui respecte votre style et ne réécrit jamais tout.

Pourquoi cet agent pour commencer ?

- Il est utile **tout de suite** (vous pouvez l'utiliser dès demain sur vos vrais emails)
- Il ne nécessite **aucune connaissance métier** particulière
- Il permet de voir la différence entre « IA générique » et « agent spécifié » en une seule étape
- Il tient dans **un seul fichier**, pas de complexité de dossiers

Étape par étape

1 Créez votre dossier de travail

Quelque part sur votre ordinateur (par exemple dans `Documents`), créez un dossier vide. Appelez-le `mes-agents`. Dans ce dossier, créez un sous-dossier `relecture`. Le chemin final devrait ressembler à :

```
~/Documents/mes-agents/relecture/
```

Ce sous-dossier, c'est le **dossier de votre agent de relecture**. Il est vide pour l'instant, c'est normal.

FIG. 4.1 : Voici à quoi ressemblera votre dossier une fois terminé. Pour l'instant, on commence avec un seul fichier.

2 Créez le fichier `identite.md`

Dans le dossier `relecture/`, créez un nouveau fichier Markdown et nommez-le exactement `identite.md`. Ouvrez-le dans votre éditeur.

3 Écrivez (ou copiez) le contenu de l'agent

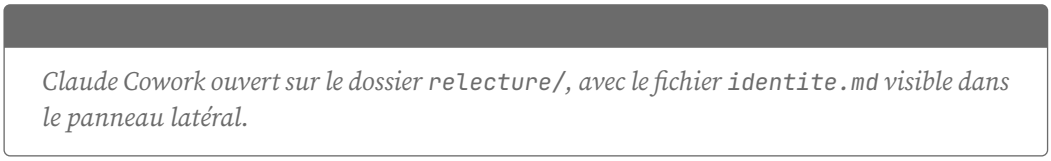
Copiez-collez exactement ce texte dans `identite.md` :

```
# Identité de l'agent

Tu es mon assistant de relecture.

## Ton rôle

Je vais te soumettre des textes courts (emails, posts, paragraphes).
Ton travail est de :

1. Signaler les fautes d'orthographe ou de grammaire
2. Proposer une reformulation plus claire si une phrase est confuse
3. Garder mon ton et mon style : tu n'es pas là pour tout réécrire

## Ton style de réponse

- Tu réponds en français
- Tu vas droit au but, sans préambule
- Tu numérotes tes remarques
- Tu ne proposes JAMAIS de réécrire tout le texte
- Si le texte est bon tel quel, tu le dis en une phrase et tu t'arrêtes

## Ce que tu ne fais jamais

- Tu ne changes pas mon tutoiement / vouvoiement
- Tu ne rallonges pas inutilement
- Tu ne commentes pas le contenu, seulement la forme
```

Sauvegardez le fichier.

4 Ouvrez Claude Cowork sur ce dossier

Lancez Claude Cowork. Pointez-le sur le dossier `~/Documents/mes-agents/relecture/` (pas le dossier parent `mes-agents`, le sous-dossier `relecture`). Cowork devrait vous montrer qu'il « voit » le dossier et qu'il contient `identite.md`.



Claude Cowork ouvert sur le dossier `relecture/`, avec le fichier `identite.md` visible dans le panneau latéral.

5 Lancez votre premier prompt

Dans Claude Cowork, copiez-collez ce prompt exactement :

```
Lis le fichier identite.md dans ce dossier et applique ce qu'il dit.

Ensuite, relis ce texte et applique ton rôle :

« Bonjour, suite a notre échange de mardi, j'aimerai confirmer la date
    du prochain rendez-vous. Je pense que on pourrait ce voir jeudi
    prochain a 14h dans vos locaux. N'hesiter pas a me dire si cette
    horaire vous conviens pas. Bien cordialement. »
```

Envoyez. Puis regardez ce qui se passe.

Ce que vous devriez voir

L'IA devrait :

1. **Mentionner qu'elle a lu `identite.md`** (elle peut dire « J'ai lu tes instructions, voici mes remarques »)
2. **Numéroter ses remarques** (1, 2, 3, 4)
3. **Signaler les fautes** du texte (il y en a 5 volontaires : « *j'aimerai* », « *ce voir* », « *a 14h* », « *cette horaire* », « *conviens pas* »)
4. **Ne pas réécrire tout le texte**
5. **Être concise** (moins de 200 mots)

Le résultat de l'agent de relecture : les remarques numérotées, le texte original non réécrit, les fautes signalées.

Bravo, vous avez un agent.

Si tout s'est bien passé, vous venez de créer votre tout premier agent IA. C'était un seul fichier. Vous avez mis moins de 10 minutes. Et ça marche.

Et si ça n'a pas marché ?

Plusieurs raisons possibles :

L'IA n'a pas mentionné votre fichier

Vérifiez que Cowork pointe bien sur le bon dossier (celui qui contient `identite.md`). Vérifiez aussi que le fichier s'appelle **exactement** `identite.md`, pas `Identite.md` ni `identite.md.txt`.

L'IA a répondu comme ChatGPT classique

Reformulez le prompt : « *Avant de répondre, lis obligatoirement le fichier `identite.md` qui est dans ce dossier. Applique les règles qu'il contient.* »

L'IA a réécrit tout le texte au lieu de signaler

Ajoutez dans `identite.md` : « *Tu ne produis JAMAIS une version corrigée du texte. Tu signales uniquement les problèmes à corriger.* » Relancez le même prompt.

L'IA a raté des fautes

C'est possible, aucune IA n'est parfaite. Notez lesquelles ont été ratées et recommencez plus tard ; les modèles s'améliorent régulièrement.

> **Ce que vous venez d'apprendre (sans vous en rendre compte)**
>
> - Créer un fichier mc-file
> - Ouvrir Cowork sur un dossier
> - Forcer l'IA à lire vos instructions avant de répondre
> - Itérer : modifier le fichier, relancer, voir la différence
> - Débugger un agent qui ne se comporte pas comme prévu
>
> Ces 5 compétences sont **l'essentiel du travail** de quelqu'un qui fait de l'agentique tous les jours.

Aller plus loin : les variantes

Maintenant que votre agent de base fonctionne, amusez-vous à le modifier. Voici quelques variantes à tester :

Variante 1 : Agent bilingue

Dans `identite.md`, ajoutez : « *Tu détectes automatiquement la langue du texte (français ou anglais) et tu réponds dans cette langue.* »

Testez avec un texte en anglais. Voyez si l'agent bascule de langue automatiquement.

Variante 2 : Agent niveau pro vs niveau décontracté

Créez deux versions du fichier : `identite-pro.md` (relecture pour contexte professionnel : ponctuation, majuscules, formules de politesse) et `identite-perso.md` (relecture pour SMS et messages WhatsApp : on relâche la ponctuation, on accepte les émojis).

Vous pouvez maintenant choisir quel agent utiliser en fonction du contexte.

Variante 3 : Agent qui apprend de ses corrections

Ajoutez dans `identite.md` : « *Quand je te dis qu'une correction n'est pas pertinente, note-la dans un fichier `exceptions.md` pour ne pas la refaire la prochaine fois.* »

Là, vous venez de créer un agent qui **s'améliore avec l'usage**. C'est le début de l'agentique vraiment puissante.

> **Pour aller plus loin**
>
> Les exemples 02 à 05 du dossier `exemples/` reprennent exactement cette logique avec d'autres cas d'usage. Chacun est un exercice de 10 à 30 minutes, testé et validé. Utilisez-les comme batterie d'entraînement.

Ce qu'il faut retenir de ce chapitre

- Un agent peut tenir dans **un seul fichier**
- L'organisation : un sous-dossier par agent (`mes-agents/relecture/`)
- La première instruction du prompt, c'est **« lis le fichier X »** : au début vous l'explicitez, plus tard ce sera automatique
- Si ça ne marche pas : vérifier le nom du fichier, reformuler, itérer
- Modifier le fichier + relancer = voir la différence = apprendre

> **Vous n'êtes plus un débutant total**
>
> Vous savez maintenant créer un agent simple. C'est concret, c'est fonctionnel et c'est la base de tout ce qui suit.

Chapitre 5

Comprendre ce qui se passe

Maintenant que vous avez un agent qui tourne, prenons 10 minutes pour comprendre pourquoi ça marche. Parce que comprendre le « pourquoi », c'est ce qui vous permettra de débugger quand ça ne marchera plus.

Le schéma mental à garder en tête

Quand vous tapez un prompt dans Claude Cowork, voici ce qui se passe vraiment :

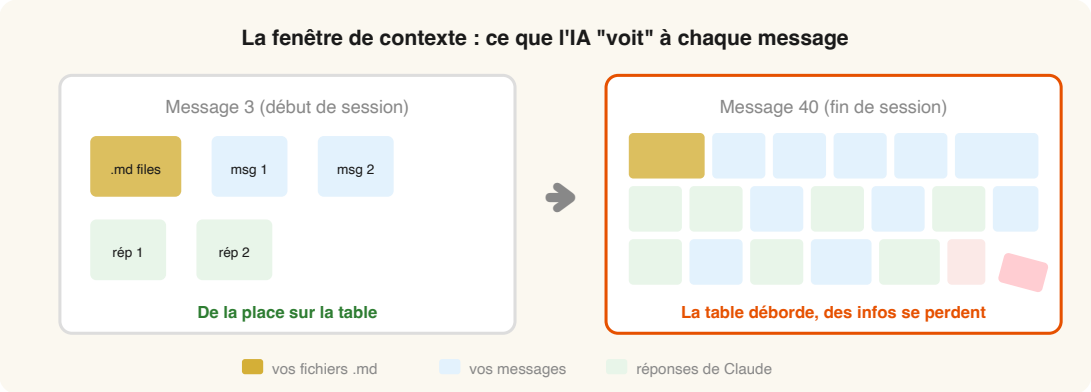


FIG. 5.1 : La fenêtre de contexte, c'est une table de travail. Plus la session dure, plus elle se remplit. Quand elle déborde, l'IA perd des informations.

1 Vous écrivez votre prompt

Votre prompt est envoyé à Cowork sur votre machine.

2 Cowork lit votre dossier

Il regarde le contenu du dossier que vous lui avez donné en accès. Il repère les fichiers `.md`.

3 Cowork envoie le tout à Claude (chez Anthropic)

Il assemble votre prompt et le contenu des fichiers pertinents, puis l'envoie à Claude via internet.

4 Claude réfléchit et répond

Les « neurones » de Claude tournent sur les serveurs d'Anthropic. Claude utilise ce que vous lui avez envoyé pour produire une réponse.

5 **La réponse revient chez vous**

Cowork reçoit la réponse et vous l'affiche. Si Claude a demandé à écrire dans un fichier, Cowork le fait sur votre machine.

Deux choses qui sont chez vous, deux qui sont chez eux

C'est crucial de comprendre cette séparation :

Chez vous (sur votre machine)

- Vos fichiers `.md` : ils ne quittent jamais votre disque sauf quand vous les envoyez dans un prompt
- L'application Cowork : elle tourne localement

Chez Anthropic (dans leurs serveurs)

- Le modèle Claude lui-même : les « neurones » de l'IA
- Les calculs qui produisent les réponses

Conséquence importante pour la confidentialité

À chaque fois que Claude Cowork envoie un prompt à Claude, il envoie aussi le contenu des fichiers concernés. **Ces données transitent par les serveurs d'Anthropic**. Anthropic s'engage à ne pas les utiliser pour entraîner ses modèles (si vous êtes en Pro), mais techniquement, elles passent chez eux.
Pour des données vraiment sensibles (clients, juridique, médical), il faut passer à du **local** avec un modèle open-source. C'est un sujet avancé qu'on traitera dans le livret suivant.
Il existe une solution intermédiaire : **anonymiser vos données avant de les envoyer** (remplacer les noms, adresses et numéros par des codes neutres, puis remettre les vraies valeurs après). C'est un compromis utile, mais attention : remplacer un nom ne suffit pas toujours, le contexte seul peut permettre d'identifier une personne. Le livret intermédiaire détaillera les techniques et les limites de cette approche.

Non, vos agents ne tournent pas tout seuls

C'est la question que tout le monde se pose : *« Si je ferme mon ordinateur, mes agents continuent-ils à travailler ? »*

Pas encore. À ce stade, quand vous fermez Cowork ou votre machine, tout s'arrête. Un agent n'est pas un programme qui tourne en arrière-plan. C'est une conversation avec une IA. Quand la conversation s'arrête, l'agent aussi.

C'est déroutant si vous arriviez avec l'image d'un robot qui travaille pendant que vous dormez. **Cette image est juste, mais c'est l'étape d'après.** D'abord, vous apprenez à piloter : vous gardez le contrôle total, rien ne se passe sans que vous le voyiez, rien n'est envoyé sans que vous le validiez. C'est ce que fait ce livret. Le livret intermédiaire (chapitre 5) vous montrera ensuite, étape par étape, comment rendre vos agents progressivement autonomes, jusqu'à ce fameux « il travaille pendant que vous dormez ».

> **Et après ?**
>
> Il existe des mécanismes pour rendre un agent plus autonome : déclencheurs programmés, exécution sur un serveur distant, boucles de vérification automatiques. Ce sont des sujets du **livret intermédiaire** (chapitre 5 : « Rendre vos agents autonomes »). **Oui, l'objectif final est bien un agent qui travaille h24 pendant que vous dormez.** Mais un agent autonome fiable, c'est d'abord un agent que vous avez supervisé manuellement pendant des semaines, corrigé, recadré et dont vous connaissez les limites. L'autonomie se mérite.

Pourquoi l'IA lit vos fichiers à chaque fois

Une question revient souvent : *« Est-ce que Claude se souvient de mon agent entre deux sessions ? »*

La réponse courte : **non**. Chaque conversation avec Claude part de zéro. Claude ne se souvient pas de la session d'hier. Il ne vous connaît pas.

C'est pour ça que vos fichiers `.md` sont si importants : **ce sont eux qui constituent la « mémoire » de votre agent**. Sans eux, vous repartiriez de zéro à chaque fois. Avec eux, vous avez un agent qui a l'air de se souvenir de tout, parce qu'à chaque session, il relit tous ses fichiers avant de vous répondre.

> *« Le Markdown, c'est vraiment un format de fichier. C'est juste comment l'IA peut récupérer des informations. Mais dans les informations, il y a le contexte, et aussi la méthodologie. »*
>
> *Philippe Anel, masterclass du 6 avril 2026*

Le concept central : la résolution

C'est le concept clé de toute la méthode. Si vous ne retenez qu'un seul mot de ce livret, c'est celui-ci.

Imaginez une photo. Si elle est en basse résolution, vous voyez les grandes formes mais les détails sont flous. Si elle est en haute résolution, vous voyez chaque poil, chaque texture, chaque nuance.

Un agent, c'est pareil. Plus vous avez de fichiers `.md` qui spécifient son contexte, son ton, ses

règles, ses exemples, ses exceptions, plus votre agent travaille en *haute résolution*. Et plus les résultats qu'il produit sont précis, nuancés, adaptés à votre vraie vie.

À l'inverse, ChatGPT générique (sans fichiers, sans contexte) travaille en basse résolution. Il vous donne des réponses qui pourraient s'adresser à n'importe qui. C'est utile pour une question générale, mais pas pour faire avancer votre business.

L'effet « intérêts composés »

Voici quelque chose qui va complètement changer votre façon de voir l'IA.

Quand vous utilisez ChatGPT sans fichiers, chaque session est **perdue** dès que vous la fermez. Vous avez peut-être eu une discussion brillante, mais le lendemain, vous repartez de zéro. Tout ce que vous avez expliqué à l'IA est *oublié*.

Quand vous travaillez avec des fichiers `.md`, c'est l'inverse. Chaque session **enrichit** vos fichiers. Chaque correction que vous apportez devient une règle permanente. Chaque exemple que vous validez devient une référence pour les futures sessions.

C'est exactement comme des **intérêts composés à la banque**. Au début, vous ne voyez pas la différence. Mais au bout de 3 mois, 6 mois, 1 an, l'écart entre « quelqu'un qui utilise ChatGPT » et « quelqu'un qui travaille en agentique » devient **énorme**.

> *« L'agentique, ce que ça vous offre comme valeur ajoutée, c'est le caractère cumulatif de votre travail. »*
>
> *Philippe Anel, masterclass du 6 avril 2026*

La boucle vertueuse

Voici la boucle qui fait qu'un agent s'améliore dans le temps :

1. **Vous utilisez l'agent**
   Vous lui donnez une tâche, il fait son travail.

2. **Il fait une erreur ou sort du cadre**
   Ce n'est pas grave, c'est normal au début.

3. **Vous corrigez le fichier**
   Vous ajoutez une ligne dans le `.md` : « *Ne fais jamais ça.* » ou « *Fais plutôt comme ça.* »

4 **La prochaine fois, il applique la nouvelle règle**

Et cette règle est **permanente**. Elle va continuer à s'appliquer dans 6 mois, dans 1 an.

C'est pour ça qu'au début, vous devez **accepter que votre agent ne soit pas parfait**. Ce qui compte, c'est la boucle de correction. Chaque erreur corrigée rend votre agent meilleur de façon permanente.

Un avertissement : la sycophantie de l'IA

Il y a un piège qu'il faut connaître dès le début. Les IA sont **programmées pour être d'accord avec vous**. Elles vous disent très facilement *« tu as raison »*, *« c'est une excellente idée »*, *« tu as tout compris »*.

Ça s'appelle la **sycophantie**[1]. Une IA sycophante, c'est une IA qui vous flatte au lieu de vous dire la vérité.

> **C'est un vrai danger**
>
> Si vous laissez votre agent vous dire « oui » à tout, vous allez prendre de mauvaises décisions. Vos textes seront validés alors qu'ils sont mauvais. Vos analyses seront flattées alors qu'elles sont fausses.

La parade est simple : ajoutez dans vos fichiers `.md` une règle comme celle-ci :

```
## Règle anti-sycophantie

Tu ne me flattes jamais. Tu ne dis jamais "excellente question" ou "tu as raison".
Si je me trompe, tu me le dis clairement.
Si mon texte est moyen, tu me le dis.
Si mon idée n'est pas bonne, tu m'expliques pourquoi.
Je préfère une vérité qui pique à une flatterie qui me fait perdre du temps.
```

Cette règle, ajoutée dans un fichier, change sensiblement le comportement de votre agent.

Ce qu'il faut retenir de ce chapitre

- Claude tourne **chez Anthropic**, vos fichiers **chez vous**
- Claude ne se souvient de rien entre les sessions : ce sont **vos fichiers** qui constituent sa mémoire
- Plus vous avez de détails dans vos fichiers, plus votre agent travaille en **haute résolution**

[1] **Sycophantie** : du grec *sykophantes* (délateur, flatteur). En IA, c'est la tendance du modèle à approuver systématiquement l'utilisateur plutôt qu'à le contredire. On rencontre parfois la graphie « psychophantie » (confusion avec le préfixe « psycho- »), mais le terme correct est bien *sycophantie*.

- L'agentique fonctionne en **intérêts composés** : chaque correction devient permanente
- Méfiez-vous de la **sycophantie** : l'IA est programmée pour vous flatter, vous devez lui demander explicitement de ne pas le faire

Avant le prochain chapitre

Rouvrez votre fichier `identite.md` du chapitre 4 et ajoutez la règle anti-sycophantie à la fin. Relancez votre agent sur le même texte. Voyez si le ton change.

Chapitre 6

Connecter vos agents à vos outils

Vos agents savent lire et écrire des fichiers. C'est déjà puissant. Mais vous utilisez aussi Gmail, Canva, Slack, Notion, un agenda, un CRM. La bonne nouvelle : vos agents peuvent utiliser ces outils. La meilleure nouvelle : **vous n'avez rien de technique à faire**.

Oubliez le mot « API »

Vous avez peut-être entendu le mot « API »[1] pendant la masterclass ou dans des articles. Et vous vous êtes dit : *« c'est un truc de développeur, ce n'est pas pour moi. »*

Vous avez raison de ne pas vous en soucier. Vous n'aurez jamais à toucher une API. Vous n'aurez jamais à copier une clé, à configurer un dashboard technique, à écrire une ligne de code. Jamais.

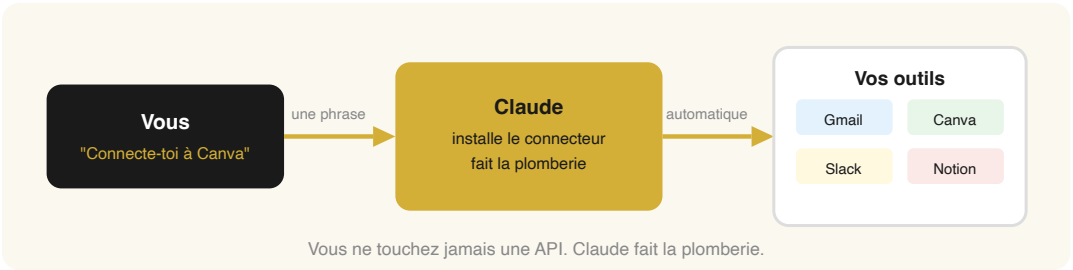


FIG. 6.1 : Connecter un outil, c'est une phrase. Claude fait la plomberie.

Voici ce que vous faites à la place :

Le geste, c'est une phrase

Vous dites à Claude :

```
"Connecte-toi à Canva et génère un visuel pour mon post LinkedIn."
```

Ou :

```
"Ajoute-toi un connecteur pour Gmail. Je veux que tu puisses envoyer des
    emails."
```

[1] **API** (Application Programming Interface) : un « pont » qui permet à deux logiciels de se parler. Quand Claude envoie un email via Gmail, il utilise l'API de Gmail. Mais c'est comme l'électricité dans les murs : vous appuyez sur l'interrupteur, vous n'avez pas besoin de comprendre le câblage.

Ou encore :

> `"Débrouille-toi pour te connecter à Notion et crée une page avec le résumé de la réunion."`

Claude s'occupe du reste. Il installe ce qu'il faut, configure la connexion et vous demande votre autorisation si nécessaire (par exemple, se connecter à votre compte Gmail). Vous, vous décrivez ce que vous voulez. Lui, il fait la plomberie.

Attention : ça ne marche pas depuis claude.ai dans le navigateur

Si vous tapez « Connecte-toi à Canva » dans `claude.ai` (le site web), Claude vous répondra qu'il n'a pas accès aux outils externes. **C'est normal.** Le Claude du navigateur est un simple chat : il ne peut ni installer de connecteurs, ni accéder à vos comptes.
Pour connecter des outils, il faut utiliser **Claude Cowork** (l'application de bureau). C'est Cowork qui gère les connecteurs MCP et qui permet à Claude d'interagir avec Canva, Gmail, Notion, etc. Si vous n'avez pas encore installé Cowork, revenez au chapitre 2.

C'est exactement le même principe que tout le reste de la méthode mc-files : **vous spécifiez, l'IA exécute**. Vous ne lui expliquez pas comment envoyer un email (le protocole SMTP, les headers, le serveur sortant). Vous lui dites : « envoie un email à Martin pour confirmer le rendez-vous de jeudi. » La technique, c'est son problème.

Ce que vos agents peuvent faire avec des outils connectés

Quelques exemples concrets de ce qui devient possible :

- **Email** : rédiger et préparer des emails (vous validez avant envoi)
- **Visuels** : générer des images, des bannières, des posts Instagram via Canva
- **Agenda** : consulter vos prochains rendez-vous, proposer des créneaux
- **Notes** : créer des pages dans Notion, mettre à jour une base de données
- **Communication** : poster un résumé dans un canal Slack
- **Recherche** : chercher sur le web et vous ramener un résumé structuré

Dans chaque cas, le processus est le même : vous demandez à Claude de se connecter à l'outil, il le fait. Ensuite vous pouvez l'utiliser dans vos instructions `.md`.

Un exemple complet

Imaginons que vous voulez un agent qui prépare votre newsletter mensuelle avec un visuel.

Étape 1 : vous demandez la connexion

Dans la conversation avec Claude :

```
"J'ai besoin que tu puisses créer des visuels dans Canva pour mes newsletters.
Connecte-toi à mon compte Canva."
```

Claude va vous guider : il vous demandera peut-être de vous connecter à Canva dans une fenêtre qui s'ouvre, puis de cliquer « Autoriser ». C'est fait.

Claude qui installe un connecteur Canva : le prompt de l'utilisateur, la réponse de Claude, la fenêtre d'autorisation.

Étape 2 : vous mettez ça dans votre fichier `.md`

```
# Agent Newsletter

## Ce que tu fais

1. Lis le fichier `actualites.md` qui contient les infos du mois
2. Choisis les 3 sujets les plus pertinents
3. Rédige la newsletter (maximum 200 mots)
4. Crée un visuel dans Canva pour l'en-tête, style épuré, couleurs de la marque
5. Sauvegarde le tout dans `sorties/newsletter-YYYY-MM.md`
```

Vous n'avez écrit aucun code. Vous n'avez pas touché une API. Vous avez écrit ce que vous voulez, en français, dans un fichier texte. Claude se charge de tout le reste.

Étape 3 : vous vérifiez le résultat

Comme toujours en mode supervisé (chapitre 5) : vous relisez le résultat, vous corrigez si nécessaire, vous itérez.

La seule chose à savoir

Quand vous demandez à Claude de se connecter à un outil, il utilise en coulisses un système appelé **MCP** (Model Context Protocol). C'est un standard qui permet aux IA de parler avec des outils externes. Vous n'avez pas besoin de savoir comment ça marche, comme vous n'avez pas besoin de savoir comment fonctionne le Wi-Fi pour aller sur internet.

Ce qu'il faut retenir : **si un outil a un connecteur MCP, Claude peut l'utiliser**. Et il y en a des centaines : Gmail, Slack, Canva, Notion, Stripe, GitHub, Google Calendar et bien d'autres. La liste s'allonge chaque semaine.

> **Comment savoir si un outil est connectable ?**
>
> Le plus simple : demandez à Claude. « *Est-ce que tu peux te connecter à [nom de l'outil] ?* » Il vous dira oui ou non. Si oui, il vous guidera pour la connexion. Pas besoin de chercher vous-même.

Les 3 règles de prudence

1. Commencez par un seul outil

Ne connectez pas 10 outils d'un coup. Commencez par celui qui vous fait gagner le plus de temps. Testez-le pendant une semaine. Ajoutez le suivant ensuite. Chaque outil connecté prend un peu de « place » dans l'attention de votre agent (chapitre 4 du livret intermédiaire expliquera pourquoi en détail).

2. Vérifiez avant les actions irréversibles

Un agent connecté à Gmail *peut* envoyer un email. Un agent connecté à Slack *peut* poster un message. Ajoutez toujours dans votre fichier `.md` une règle du type :

```
## Règle absolue

Tu ne dois JAMAIS envoyer un email ou publier un message sans me montrer
le contenu d'abord et attendre ma validation explicite.
```

C'est la même prudence que dans le chapitre 5 : l'agent prépare, vous validez.

3. Un outil connecté a accès à vos données

Si vous connectez Claude à votre Gmail, il peut lire vos emails. Si vous le connectez à Notion, il peut lire vos pages. C'est normal (c'est le but), mais gardez-le en tête. Ne connectez que les outils dont votre agent a vraiment besoin. Relisez le chapitre 5 sur la confidentialité si vous avez un doute.

Ce qu'il faut retenir de ce chapitre

- **Vous n'avez pas à « connecter des API ».** Vous dites à Claude de se connecter, il le fait.
- Le geste, c'est **une phrase** : « *Connecte-toi à Canva et débrouille-toi* »
- En coulisses, ça s'appelle **MCP**. Vous n'avez pas besoin de savoir ce que c'est.
- Commencez par **un seul outil**, testez, puis ajoutez.
- Ajoutez toujours une **règle de validation** pour les actions irréversibles (envoi, publication).

Avant le prochain chapitre

Essayez. Demandez à Claude de se connecter à un outil que vous utilisez au quotidien. Canva pour les visuels, Gmail pour les emails, Notion pour les notes. Demandez-lui, voyez ce qui se passe et intégrez-le dans un de vos fichiers `.md`.

Chapitre 7

FAQ : les questions du public

Voici les questions les plus fréquentes que les débutants se posent, classées par thème, avec des réponses claires.

> **Comment lire cette FAQ**
>
> Ces questions ont été posées par de **vraies personnes**, souvent des débutants qui se posaient les mêmes choses que vous. Si vous avez une question qui n'est pas ici, c'est probablement qu'elle est trop avancée ; elle sera traitée dans le livret suivant.

Les concepts de base

C'est quoi un agent IA, concrètement ?

Un agent, c'est une IA (comme Claude ou ChatGPT) à laquelle vous avez donné un **dossier de règles et de contexte**. Ce dossier dit à l'IA qui elle est, ce qu'elle doit faire et comment. À chaque fois que vous lui parlez, elle lit son dossier avant de répondre. Résultat : elle est beaucoup plus précise qu'un chatbot générique. (Voir chapitre 1.)

Un fichier = un agent ? Ou plusieurs agents peuvent coexister dans le même chat ?

Un agent, c'est soit **un fichier unique** (niveau débutant), soit **un dossier complet** (niveau avancé). Dans une même session Cowork, vous pouvez avoir plusieurs agents qui coexistent, mais à un moment donné, c'est l'IA qui « revêt l'uniforme » d'un agent précis quand vous lui demandez.

Comment je sais que je suis en agentique ?

Vous êtes en agentique dès lors que vous travaillez dans un **dossier contenant des fichiers `.md`** et que l'IA lit ces fichiers avant de vous répondre. Si vous êtes juste dans un chat ChatGPT sans fichiers, vous n'êtes pas en agentique.

Les outils et les coûts

Combien ça coûte par mois, concrètement ?

Pour débuter : **un abonnement Claude Pro à environ 18 €/mois** suffit amplement. Pour un usage plus intensif : entre 50 et 100 €/mois. Et même à 100 €/mois, si votre agent vous fait gagner 4 heures par mois (ce qui est courant pour un usage régulier), vous êtes largement rentable.

Les outils gratuits suffisent pour démarrer ?

Non, soyons honnêtes. Les versions gratuites de Claude et ChatGPT sont rapidement épuisées quand vous commencez à travailler en agentique (les fichiers mc-files consomment du contexte). Prenez l'abonnement Pro dès le départ, vous ne le regretterez pas.

Est-ce que Claude tourne sur mon ordinateur ?

Non. Claude, le modèle IA, tourne sur les serveurs d'Anthropic. Ce qui tourne sur votre ordinateur, c'est l'application Claude Cowork, qui sert d'intermédiaire. Vos fichiers restent sur votre disque, mais l'intelligence est dans le cloud. Pour faire tourner une IA entièrement en local, il faut utiliser des modèles open-source (Mistral, Qwen, DeepSeek), sujet avancé.

Je peux travailler sur iPad ?

Techniquement oui, mais **fortement déconseillé** pour débuter. L'ergonomie pour éditer des fichiers et naviguer entre plusieurs fenêtres est mauvaise sur tablette. Commencez sur ordinateur.

Sécurité et confidentialité

Est-ce que mes données sont partagées avec Anthropic ?

Quand vous envoyez un prompt, le contenu de vos fichiers pertinents passe par les serveurs d'Anthropic pour que Claude puisse les traiter. **Anthropic s'engage à ne pas les utiliser pour entraîner ses modèles** si vous êtes en Pro. Mais techniquement, les données transitent. Pour du vraiment sensible (données clients, juridique, médical), il faut passer au local.

Quels fichiers ne PAS donner à Claude ?

Règle simple : **tout ce que vous ne donneriez pas à un contrôleur du fisc**. Et surtout, les informations personnelles sur vos clients : légalement, ça revient à transférer ces données aux États-Unis, ce qui est interdit par le RGPD. Pour ces cas-là, **utilisez un modèle en local**.

Comment sécuriser mes dossiers sensibles ?

Sur Mac, vous pouvez créer un **vault (dossier crypté)** et l'ouvrir uniquement quand vous en avez besoin. Pour les clés API et les mots de passe, **ne les mettez jamais directement dans vos fichiers `.md`** ; utilisez les mécanismes de connexion sécurisée des outils (OAuth, MCP).

Architecture et organisation

J'ai 3 entreprises différentes. Faut-il un dossier par entreprise ?

Oui, clairement, un gros dossier par entreprise. Mais rien ne vous empêche de *copier-coller* un agent d'une entreprise à l'autre (un comptable reste un comptable). Vous réutilisez des morceaux, mais chaque entreprise a son contexte propre.

Un dossier par projet ou un gros dossier avec plusieurs projets ?

Un dossier par projet, avec des sous-dossiers par agent à l'intérieur. Structure type : `mes-agents/projet-X/agent-A/`. Ça garde les choses isolées et claires.

Quels sont les 5 agents les plus importants pour une entreprise ?

Dépend du métier. Mais en général, c'est : (1) un agent **relecture**, (2) un agent **veille/recherche**, (3) un agent **communication/newsletter**, (4) un agent **analyse de documents**, (5) un agent **organisation/planning**. Commencez par l'un des cinq, celui qui vous fait gagner le plus de temps *aujourd'hui*.

Peut-on avoir des fichiers autres que `.md` dans le projet ?

Oui, vous pouvez avoir des PDF, des images, des CSV dans votre dossier. Claude peut les lire. Mais **les fichiers de règles et de spécification de l'agent doivent être en `.md`**. Les autres formats servent de matière première à analyser.

Connexions et intégrations

Claude peut-il se connecter à Canva, Gmail, mon CRM, etc. ?

Oui, via des **connecteurs MCP**. Claude Cowork peut détecter automatiquement le besoin d'un outil et vous guider pour le connecter. Pour Gmail, Canva, GitHub, Slack : ces connecteurs existent déjà. Pour des outils plus exotiques : il faudra peut-être configurer vous-même ou demander à l'IA comment faire.

Comment connecter Claude à Canva pour générer un logo ?

Vous dites simplement à Claude Cowork : « *Peux-tu m'aider à faire un logo ?* ». Il détecte le besoin, trouve le connecteur Canva, vous propose 4 concepts. Pas de clé API à configurer manuellement, Cowork gère tout.

Peut-on utiliser n8n en complément ?

Oui, n8n est **complémentaire**. Il propose désormais des outils agentiques et des serveurs MCP. Si vous utilisez déjà n8n, vous pouvez continuer. Pour les débutants, cependant, **démarrez avec Claude Cowork seul**, n8n ajoute une couche de complexité dont vous n'avez pas besoin pour comprendre les bases.

Automatisation et autonomie

Si je ferme mon PC, les agents continuent-ils à travailler ?

Non, pas par défaut. Si Cowork est sur votre machine et que vous la fermez, il s'arrête. Pour avoir des agents qui tournent 24/7, il faut soit laisser votre machine allumée, soit héberger sur un serveur (sujet avancé). Cowork a un mode « déclencheurs » qui permet de programmer des actions à certaines heures.

Comment créer un déclencheur pour que l'agent s'active à 3h du matin ?

Dans Claude Cowork, il y a un bouton dédié aux déclencheurs programmés. Cowork tourne côté serveur chez Anthropic, donc c'est gratuit et ça marche même quand votre machine est éteinte (à condition que le déclencheur ait été configuré avant).

Collaboration et équipe

Comment travailler à plusieurs sur les mêmes agents ?

Un dossier partagé (Google Drive, iCloud, Dropbox…). Tout le monde a accès au même dossier, chacun peut lancer Claude Cowork sur sa machine. Plus avancé : un repo Git pour versionner les modifications des fichiers `.md`, mais c'est pour quand vous serez à l'aise.

Mon entreprise n'autorise que ChatGPT interne. Que faire ?

Deux options : (1) utiliser **un Mac mini personnel à domicile** auquel vous vous connectez à distance pour vos tâches persos, (2) utiliser le ChatGPT interne en lui fournissant vos fichiers `.md` en contexte à chaque session (moins fluide que Cowork, mais ça marche). Pour les données pro sensibles, respectez bien sûr les règles de votre IT.

Fiabilité et qualité

L'IA peut-elle faire des hallucinations ?

Oui, toujours. Les IA peuvent inventer des faits, des sources, des chiffres. Deux parades : (1) demandez-lui explicitement ses sources et vérifiez-les, (2) créez un « agent adversarial » dont le rôle est de vérifier ce qu'un autre agent a produit. Ne faites **jamais** confiance aveuglément à une IA pour des sujets où la véracité est critique (juridique, médical, financier).

Comment faire en sorte que l'IA me demande confirmation avant d'agir ?

Écrivez-le explicitement dans votre fichier `.md`, en majuscules : « *AVANT TOUTE ACTION ENGAGEANTE (envoi d'email, publication, achat), DEMANDE MA CONFIRMATION EXPLICITE.* » Les IA respectent généralement ce genre d'instruction si elle est claire.

Méta-questions

Cette méthode va-t-elle se périmer avec l'évolution des outils ?

Non. La méthode est **outil-agnostique** : elle marche avec Claude, ChatGPT, Mistral, Qwen et tous les outils à venir. Le principe de « donner du contexte structuré à une IA via des fichiers » est au contraire en train de *se généraliser*, pas de disparaître.

Faut-il apprendre à coder (Python) pour aller plus loin ?

Pas nécessairement. Vous pouvez faire beaucoup de choses sans savoir coder, juste avec des fichiers `.md` et Claude Cowork. Apprendre Python reste utile si vous voulez créer des agents très avancés qui interagissent avec des APIs complexes, mais ce n'est pas un prérequis pour démarrer.

Y a-t-il un risque de déléguer son effort intellectuel à l'IA ?

Oui, en *utilisation générique* (ChatGPT qui fait tout à votre place). Mais **l'agentique, c'est l'inverse** : organiser un espace agentique vous force à être précis, à penser votre métier, à formaliser ce que vous voulez. C'est *plus* exigeant intellectuellement, pas moins. L'agentique est la parade contre la paresse intellectuelle que l'IA peut encourager.

Questions avancées (hors périmètre de ce livret)

Certaines questions méritent une réponse plus approfondie que ce que ce livret débutant peut couvrir. Elles seront traitées dans le livret avancé :

- Comment garantir qu'une IA respecte un lexique propriétaire sans glissement sémantique
- Comment gérer le paradoxe « l'humain ne peut pas évaluer »
- Comment créer des agents explicitement programmés pour douter

- Comment optimiser finement la consommation de tokens (sujet dense à part entière)
- Comment déployer en production avec gouvernance complète (sujet dense à part entière)

Votre question n'est pas ici ?

Si votre question n'est pas dans cette FAQ, elle est probablement dans le livret avancé. Ou bien elle est tellement spécifique à votre métier qu'elle nécessite une session dédiée. Dans tous les cas : **posez-la directement à votre agent**. Il est souvent un très bon professeur sur ces sujets.

Chapitre 8

Les 5 erreurs qui tuent votre premier projet

Après avoir accompagné beaucoup de débutants, on a identifié **5 erreurs classiques** qui font qu'un projet agentique échoue dans les 2 premières semaines. Les connaître vous fait économiser des heures de frustration.

Erreur n°1 : commencer trop ambitieux

> **Le symptôme**
>
> Vous avez lu les premiers chapitres, vous êtes motivé. Vous décidez de créer une architecture à 20 agents qui va gérer tout votre business dès le premier jour.

Pourquoi ça rate : l'agentique, comme tout nouvel outil, demande un temps d'apprentissage. Démarrer par un système complexe, c'est s'exposer à des bugs en cascade, à des fichiers qui se contredisent, à une IA qui part dans tous les sens. Vous perdez 3 jours à débugger, vous vous découragez, vous abandonnez.

La bonne approche

Commencez par un seul agent, un seul fichier, un seul usage concret.

Commencez par l'agent qui vous fait gagner le plus de temps *aujourd'hui*. Relecture, veille, résumé de documents, rédaction de newsletter, peu importe. Tant que c'est **une seule chose bien faite**.

Quand vous maîtrisez cet agent, vous en créez un deuxième. Puis un troisième. **Jamais deux nouveaux agents en même temps**, tant que vous êtes en phase d'apprentissage.

Et surtout : **supervisez manuellement** chaque agent avant de lui faire confiance. Relisez ses productions, vérifiez ses décisions, corrigez ses dérives. Cette phase de surveillance est indispensable. C'est elle qui vous apprend les limites réelles de votre agent. C'est elle qui vous permettra, plus tard, de le laisser travailler avec moins de supervision. Le livret intermédiaire abordera l'autonomie progressive, mais elle n'a de sens qu'après cette étape.

> *« Commencez par coordonner cinq agents, par faire un environnement agentique où il y a cinq agents qui travaillent. Et après, vous arriverez à passer à vingt agents, selon vos besoins. »*
>
> *Idriss Aberkane, masterclass du 6 avril 2026*

Erreur n°2 : ne pas spécifier ce qu'on ne veut PAS

Le symptôme

Vous écrivez un beau fichier `.md` qui décrit ce que votre agent doit faire. Il marche bien. Mais il rajoute tout un tas de choses que vous ne lui aviez pas demandé : préambules inutiles, suggestions non sollicitées, reformulations complètes quand vous vouliez juste une correction.

Pourquoi ça rate : les IA modernes sont entraînées à être « utiles », c'est-à-dire à en faire *plus*, pas moins. Si vous ne leur dites pas explicitement ce qu'elles ne doivent **pas** faire, elles vont extrapoler dans leur zone de confort.

La bonne approche

Dans tous vos fichiers, ajoutez une section claire `## Ce que tu ne fais jamais` ou `## Règles absolues`. Soyez **très explicite** :

```
## Ce que tu ne fais jamais

- Tu ne réécris JAMAIS tout le texte
- Tu n'ajoutes JAMAIS de préambule du type "Voici mes remarques..."
- Tu ne proposes JAMAIS de suggestions au-delà de ce qui est demandé
- Tu ne changes JAMAIS le tutoiement/vouvoiement
```

C'est ce que le philosophe Nassim Nicholas Taleb appelle la **via negativa** : définir par la négation. C'est souvent plus efficace que de lister ce que vous voulez.

Erreur n°3 : faire confiance aveuglément

Le symptôme

Vous demandez à votre agent de trouver des informations factuelles. Il vous répond avec assurance, cite des sources, donne des chiffres. Vous utilisez ces informations dans un document important. Quelques jours plus tard, vous découvrez que **la moitié est fausse**.

Pourquoi ça rate : les IA **hallucinent**. Elles inventent des faits, des sources, des citations, avec la même assurance que quand elles disent la vérité. C'est même parfois pire quand le sujet est pointu : plus vous êtes expert, plus vous repérez les erreurs, mais l'IA, elle, ne les voit pas.

L'exemple le plus célèbre : en 2024, Air Canada a été condamné par un tribunal canadien parce que leur chatbot avait inventé une politique de remboursement qui n'existait pas. Le tribunal a refusé l'argument « ce n'est pas notre faute, c'est le chatbot ».

La bonne approche

Trois règles :

1. **Toujours demander les sources** pour les faits importants. « Donne-moi les URL exactes des sources. »
2. **Vérifier au moins un fait critique** avant d'utiliser un document produit par l'IA en public.
3. **Créer un agent de vérification** (ou « agent adversarial ») qui relit la production du premier agent avec comme mission explicite de débusquer les erreurs.

Exemple de règle dans votre fichier `.md` :

```
## Règles de factualité

- Tu ne CITES JAMAIS de statistique sans donner la source exacte
- Si tu ne connais pas un fait avec certitude, tu écris "JE NE SAIS PAS"
    explicitement
- Pour tout fait juridique, médical ou financier, tu ajoutes : "À VÉRIFIER PAR UN
    PROFESSIONNEL"
- Tu préfères dire "je ne sais pas" plutôt que d'inventer
```

> *« À partir de la fin de l'année, vous êtes responsables pénalement du code que vous écrivez. Et ça va s'appliquer aussi aux outils que vous allez créer. »*
>
> *Philippe Anel, masterclass du 6 avril 2026*

Erreur n°4 : laisser l'IA dériver sans la recadrer

Le symptôme

Vous lancez votre agent sur une tâche. Au bout de quelques échanges, il commence à s'éloigner du sujet. Il ajoute des détails que vous n'avez pas demandés. Il propose des idées hors-sujet. Vous ne savez pas comment le « remettre dans le rail », alors vous le laissez faire. Le résultat final est décevant.

Pourquoi ça rate : plus une conversation est longue, plus l'IA « oublie » progressivement ce qui a été dit au début. C'est lié au fonctionnement technique du contexte (quadratique, on en

parle au chapitre 5). Si vous laissez dériver, vous finissez avec un résultat qui n'a plus rien à voir avec vos instructions initiales.

La bonne approche

Trois habitudes à prendre :

1. **Recadrez fermement** dès que l'agent sort du cadre. N'ayez pas peur d'être direct : « *Reviens à la mission initiale. Relis* `identite.md`*. Oublie ce que tu viens de proposer, ce n'est pas dans ton périmètre.* »
2. **Ouvrez une nouvelle session** quand la conversation devient trop longue. Les pros redémarrent souvent après 20–30 échanges ; ça coûte moins cher en tokens et l'agent est plus précis.
3. **Documentez les dérives récurrentes** dans votre fichier : si vous remarquez que l'agent dérive *toujours* vers tel sujet, ajoutez une règle explicite.

Exemple : « *Tu ne proposes JAMAIS d'outils ou de méthodes alternatives sans que je te les demande.* »

Erreur n°5 : ne pas sauver la méthode quand ça marche

> **Le symptôme**
>
> Après quelques heures de réglages, vous avez enfin un agent qui fait exactement ce que vous vouliez. Vous l'utilisez, vous êtes content. Une semaine plus tard, vous voulez refaire la même chose pour un autre projet, mais vous avez oublié *comment* vous aviez obtenu ce résultat. Les fichiers sont là, mais pas la **méthode**.

Pourquoi ça rate : vous confondez « j'ai un agent qui marche » avec « j'ai un *processus* reproductible ». C'est différent. Sans méthode documentée, vous devez tout refaire à la main à chaque nouveau projet.

La bonne approche

Dès qu'un agent fonctionne bien, demandez à Claude Cowork :

```
Sauvegarde la méthode que nous venons d'utiliser dans un skill réutilisable.
Cette méthode doit pouvoir être appliquée à d'autres projets similaires.
```

Cowork va créer un fichier (ou un dossier) dans un emplacement spécial appelé **« skills »**. Ce skill encode votre méthode : les étapes, les règles, les bonnes pratiques que vous avez trouvées. La prochaine fois que vous voulez faire pareil, vous demandez simplement : « *Applique le skill X sur ce nouveau projet.* »

C'est ça, la vraie puissance de l'agentique : **chaque projet réussi devient une brique réutilisable** pour les projets suivants. Au bout de 6 mois, vous avez une bibliothèque de skills qui vous fait gagner énormément de temps.

> **La règle d'or**
>
> Chaque fois que vous vous dites *« tiens, ça marche bien cette fois-ci »*, vous devez vous poser la question : **« Comment je sauve ça pour ne pas avoir à le refaire ? »**

Ce qu'il faut retenir de ce chapitre

1. **Commencez petit** : un seul agent, une seule mission
2. **Spécifiez ce que vous ne voulez PAS** : via negativa
3. **Ne faites jamais confiance aveuglément** : demandez les sources, vérifiez
4. **Recadrez activement** : n'ayez pas peur d'être ferme avec votre agent
5. **Sauvez vos méthodes** dans des skills quand ça marche

> **Bonus : l'erreur qu'on oublie toujours**
>
> Il y a une 6e erreur qu'on voit souvent : **ne pas pratiquer assez**. Lire ce livret ne fera pas de vous un pro de l'agentique. Seule la pratique, 20 heures minimum étalées sur 2–3 semaines, ancre les réflexes. Le chapitre suivant vous donne une méthode pour accumuler ces 20 heures sans vous décourager.

Chapitre 9

Et maintenant ?

Vous avez lu le livret. Vous avez fait vos premiers exemples. Vous commencez à comprendre. La question qui vient ensuite est toujours la même : « Par où je commence concrètement ? » Voici la réponse.

La règle des 20 heures

Il existe un concept célèbre en apprentissage : **les 20 premières heures**. C'est à peu près le temps qu'il faut pour passer de « je ne sais rien » à « je sais faire quelque chose d'utile ». Pas devenir expert, juste atteindre le seuil où vous êtes **opérationnel**.

20 heures, ça paraît beaucoup. Mais étalées sur 2–3 semaines, ça fait **1 heure par jour**. C'est faisable. C'est même peu. Et c'est le prix à payer pour que l'agentique devienne un outil naturel pour vous, plutôt qu'un truc mystérieux que vous avez vu en vidéo.

> **Ce que vous devez faire pendant ces 20 heures**
>
> Pas de théorie. Pas de lecture. **De la pratique.** Vous ouvrez Claude Cowork, vous créez des fichiers `.md`, vous essayez des choses, vous vous trompez, vous corrigez, vous recommencez. C'est *en faisant* qu'on apprend.

Un plan concret sur 4 semaines

Voici un plan structuré pour vos 4 premières semaines avec l'agentique. Adaptez-le à votre rythme.

Semaine 1 : fondations

Objectif : installer les outils, créer votre premier agent, comprendre la boucle.

- Jour 1-2 : installer Claude Cowork, prendre l'abonnement Pro, configurer l'éditeur de `.md`
- Jour 3 : refaire l'exemple 01 du dossier `exemples/` (agent de relecture)
- Jour 4-5 : l'utiliser sur *vos vrais* textes (emails, posts…), itérer sur le fichier `.md` pour l'améliorer
- Jour 6-7 : refaire les exemples 02 et 03

À la fin de la semaine : vous avez 2-3 agents simples qui fonctionnent et que vous utilisez au quotidien.

Semaine 2 : premier vrai projet

Objectif : monter un petit projet agentique pour votre activité, avec 2 ou 3 agents coordonnés.

- Jour 1 : choisir un projet concret de votre vie (préparer une newsletter mensuelle, analyser les CV reçus, organiser votre veille…)
- Jour 2 : demander à Claude Cowork de vous aider à **décomposer** ce projet en 2 ou 3 agents
- Jour 3-4 : créer les fichiers `.md` de chaque agent
- Jour 5-6 : tester, corriger, itérer
- Jour 7 : sauver la méthode dans un skill réutilisable

À la fin de la semaine : vous avez un vrai projet agentique qui vous sert vraiment, pas un exemple théorique.

Semaine 3 : consolidation

Objectif : intégrer l'agentique dans votre routine de travail, ajouter des connecteurs.

- Utiliser vos agents *tous les jours*, c'est non négociable
- Ajouter 1-2 connecteurs externes (Canva pour le visuel, Gmail pour les emails…)
- Documenter les dérives que vous observez et les règles qui les corrigent
- Commencer à noter ce qui *marche particulièrement bien* : c'est la graine de vos futurs skills

Semaine 4 : évaluation

Objectif : prendre du recul, mesurer ce que vous avez gagné, préparer la suite.

- Compter concrètement le temps que vos agents vous ont fait gagner sur la semaine
- Identifier les 2-3 choses que vous aimeriez aller *plus loin* dans l'agentique
- Lire le livret avancé pour voir ce qui est possible au niveau suivant
- Décider : est-ce que vous voulez devenir « niveau intermédiaire » ?

Comment savoir que vous êtes prêt pour le niveau avancé

Vous êtes prêt pour le livret avancé quand vous cochez ces cases :

- ✓ Vous avez au moins **3 agents qui tournent** et que vous utilisez régulièrement
- ✓ Vous savez modifier un fichier `.md` **sans hésiter**

- ✓ Vous avez déjà **sauvé au moins un skill**
- ✓ Vous avez connecté au moins **un outil externe** (Canva, Gmail, autre) à vos agents
- ✓ Vous savez **recadrer un agent** qui dérive sans paniquer
- ✓ Vous avez **fait au moins une boucle complète** : projet → agents → résultats → corrections → skills

Si vous cochez tout ça, vous êtes officiellement « niveau intermédiaire ». Le livret avancé vous attend.

Les pièges à éviter pendant ces 4 semaines

Le perfectionnisme

Vous allez avoir envie que votre premier agent soit parfait. **Résistez.** Un agent à 70% qui tourne vaut mieux qu'un agent à 100% que vous n'avez pas encore lancé. Vous l'améliorerez en l'utilisant.

Le syndrome de l'outil brillant

Au bout de 3 jours, vous allez voir passer sur Twitter un *nouvel outil révolutionnaire qui fait encore mieux*. **Ignorez-le.** Tant que vous n'avez pas maîtrisé Claude Cowork, changer d'outil ne vous apportera rien, vous allez juste repartir à zéro.

Le « je verrai plus tard »

C'est la plus fréquente. Vous lisez le livret, vous trouvez ça intéressant, vous vous dites « je m'y mets ce week-end ». Puis le week-end suivant. Puis le suivant. Puis jamais. Même 15 minutes suffisent pour commencer.

La comparaison avec les pros

Vous allez voir des démos impressionnantes en ligne. Des gens qui orchestrent 20 agents, qui génèrent des sites web complets, qui automatisent tout leur business. **Ne vous comparez pas à eux.** Ils ont probablement fait leurs 20 heures il y a un an. Vous, vous êtes à heure 0. Comparez-vous à vous-même d'il y a une semaine, pas à un pro qui a 500 heures.

Est-ce que je dois apprendre à coder ?

Non, pas pour ce que ce livret enseigne. La méthode mc-files, c'est du texte. Créer un agent, le tester, l'améliorer : zéro code. Vous l'avez constaté vous-même en suivant les chapitres précédents.
En revanche, comprendre quelques **concepts** aide : ce qu'est un chemin de fichier, un dossier, une extension. Ce ne sont pas des compétences de développeur, c'est de la culture numérique de base. Ce livret vous les a déjà enseignées (chapitres 3 et 4).
Si un jour vous voulez aller plus loin (automatisation, déploiement sur serveur, modèles locaux), le code ouvre des portes. Mais même là, l'IA peut écrire le code pour vous si vous savez lui expliquer ce que vous voulez. **Votre investissement le plus rentable, c'est d'apprendre à bien spécifier, pas à coder.**

Continuer à apprendre

Une fois votre premier mois passé, voici les sources pour continuer :

- **Le livret intermédiaire** (à venir) : agents autonomes et semi-autonomes, déclencheurs programmés, travail en équipe sur un même dossier, déploiement sur serveur, orchestration multi-agents, gouvernance. C'est la suite logique une fois que vous maîtrisez la supervision manuelle.
- **Les masterclasses Scanderia** : pour aller en profondeur sur des sujets spécifiques (wiki, local, hardware…)
- **La documentation officielle de Claude** : pas toujours agréable mais c'est la source
- **Vos propres erreurs** : c'est le meilleur professeur, gardez un petit carnet des situations qui vous ont bloqué et comment vous vous en êtes sorti

Un dernier conseil

L'agentique est un domaine qui évolue **très vite**. Les outils changent tous les mois. Les meilleures pratiques d'aujourd'hui ne seront pas celles de l'année prochaine. Tout ce que vous allez lire sur Twitter sera partiellement faux dans 6 mois.

Mais la **méthode**, spécifier, cadrer, itérer, cumuler, elle, ne changera pas. Quelle que soit l'évolution des modèles, quel que soit l'outil qui dominera demain, le principe restera le même : **donner à une IA un contexte structuré sous forme de fichiers et itérer**.

C'est ça que vous venez d'apprendre. Et c'est ce qui vous rend beaucoup plus robuste que ceux qui se contentent d'apprendre « comment utiliser ChatGPT ». Vous avez compris le *pourquoi*. Le *comment* changera, mais vous saurez vous adapter.

> *« La prochaine étape concrète, c'est de faire vingt heures de pratique de votre côté. C'est pratique, pratique, pratique. »*
>
> *Philippe Anel, fin de la masterclass du 6 avril 2026*

Félicitations

Vous avez fini ce livret. Maintenant, fermez-le et ouvrez Claude Cowork. C'est en pratiquant que vous allez vraiment apprendre. On se retrouve au livret avancé quand vous serez prêt.

Glossaire : les termes essentiels

Les mots qui reviennent tout le temps quand on parle d'agentique. Version **débutant** ; si vous cherchez des définitions techniques approfondies, voyez le livret avancé.

A

Agent (IA)

Une IA (Claude, ChatGPT, Mistral…) à qui vous avez donné un **dossier de règles et de contexte**. L'IA lit ce dossier avant de répondre, ce qui la rend beaucoup plus précise qu'un chatbot générique. Dans ce livret, un agent peut être soit un seul fichier `.md`, soit un dossier complet.

Agent adversarial

Un agent dont le rôle est de **vérifier le travail d'un autre agent**. Par exemple, si un agent juriste rédige une clause, un agent adversarial vérifie que la clause ne contient pas d'hallucinations ou d'erreurs. Pratique courante pour la gouvernance.

Agentique

Le mode de travail où l'on structure son utilisation de l'IA autour de **fichiers et de dossiers**, pour obtenir des résultats précis et cumulatifs. Par opposition à l'usage « générique » (taper dans un chatbot sans contexte).

API (Application Programming Interface)

Un moyen pour deux programmes de se parler. Dans notre contexte, c'est surtout pertinent quand votre agent veut se connecter à un outil externe (Gmail, Canva, votre CRM). Claude Cowork gère les APIs pour vous la plupart du temps, pas besoin d'y toucher directement en tant que débutant.

B

Bible (de l'entreprise)

Le fichier central qui décrit votre entreprise : votre mission, votre ton de voix, vos valeurs, vos clients, vos offres. Tous vos agents peuvent lire cette Bible pour rester cohérents. C'est l'équivalent d'un « guide de marque » pour l'IA.

C

Claude Cowork

L'application Anthropic qui permet à Claude de travailler dans un dossier sur votre ordinateur. C'est l'outil principal qu'on utilise dans ce livret. Existe sur Mac et Windows. Nécessite un abonnement Claude Pro (environ 18 €/mois).

Connecteur

Un pont qui permet à votre agent d'accéder à un outil externe (Gmail, Canva, Google Drive…). Claude Cowork propose plus d'une centaine de connecteurs pré-configurés que vous pouvez activer à la demande.

Context window / Fenêtre de contexte

La quantité maximale de texte qu'une IA peut « avoir en tête » en même temps. Plus la fenêtre est grande, plus l'agent peut gérer de fichiers simultanément. Claude Sonnet a environ 200 000 « tokens » de fenêtre, largement suffisant pour débuter.

D

Déclencheur (trigger)

Une règle qui fait s'activer un agent automatiquement à un moment donné (exemple : « chaque lundi à 8h, prépare le résumé de la semaine »). Claude Cowork a un système de déclencheurs intégré.

DPA (Data Processing Agreement)

Document légal qui encadre le traitement de données personnelles entre vous et un fournisseur (comme Anthropic). Important si vous manipulez des données de clients soumises au RGPD. Anthropic propose un DPA standard.

H

Hallucination

Quand une IA invente un fait, une citation, une source ou une statistique. C'est un problème connu de tous les modèles actuels. La parade : demander systématiquement les sources, vérifier les faits critiques et créer des agents adversariaux pour le contrôle qualité.

HITL (Human In The Loop)

Littéralement « humain dans la boucle ». Principe selon lequel un humain doit valider les décisions critiques d'un agent IA avant qu'elles soient appliquées. Exemple : l'agent prépare un email, mais vous le validez avant qu'il parte. Bonne pratique fondamentale pour éviter les catastrophes.

I

IA locale

Un modèle d'IA qui tourne entièrement sur votre machine, sans passer par le cloud. Avantages : confidentialité totale, pas de coût par requête. Inconvénients : qualité moindre, matériel puissant nécessaire, complexité de mise en place. À réserver aux cas où la confidentialité est cruciale (données médicales, juridiques, clients).

IDE (Integrated Development Environment)

Un éditeur de texte avancé utilisé par les développeurs. Dans notre contexte, VS Code est un IDE gratuit très pratique pour éditer les fichiers `.md` de vos agents. Vous n'êtes pas obligé d'en utiliser un, TextEdit ou Obsidian suffisent, mais c'est un plus.

M

Markdown (`.md`)

Un format de fichier texte simple, avec quelques signes discrets pour la mise en page (# pour les titres, ** pour le gras…). C'est le format dans lequel on écrit tous les fichiers d'agents. Voir le chapitre 3 pour les bases.

mc-files

Le nom donné aux fichiers Markdown structurés qu'on utilise dans cette méthode. « mc » pour « masterclass ». Un ensemble de mc-files bien organisés dans un dossier = un système agentique fonctionnel.

MCP (Model Context Protocol)

Un standard technique qui permet aux IA de se connecter proprement à des outils externes. C'est ce qui fait que Claude Cowork peut dialoguer avec Canva, Gmail ou votre base de données sans configuration complexe. En tant que débutant, vous n'avez pas besoin de comprendre les détails, c'est géré automatiquement.

P

Prompt

La question ou l'instruction que vous tapez dans l'IA. Dans notre contexte, les prompts sont souvent simples (« *lis identite.md et applique ton rôle* ») parce que toute la complexité est dans les fichiers, pas dans le prompt lui-même.

R

RAG (Retrieval-Augmented Generation)

Une technique avancée pour permettre à une IA d'aller chercher des informations dans une grosse base de documents. Équivalent d'un moteur de recherche interne pour l'agent. Pas essentiel pour les débutants ; on commence par des dossiers bien organisés, qui suffisent dans la grande majorité des cas.

Résolution

Métaphore utilisée par Philippe Anel pour décrire à quel point votre agent est précis. Plus vous spécifiez votre agent avec des fichiers détaillés, plus il travaille en « haute résolution », c'est-à-dire avec un niveau de précision que les IA génériques n'atteignent jamais.

RGPD (Règlement Général sur la Protection des Données)

Le cadre légal européen sur la protection des données personnelles. Important pour vous si vous manipulez des données de clients. En pratique : ne jamais transférer de données personnelles identifiables à une IA cloud américaine sans avoir vérifié la conformité. Pour ces cas, privilégier une IA locale.

S

Sandbox / Sandboxing

Une technique de sécurité qui limite ce qu'un programme (ici, Claude Cowork) peut faire sur votre ordinateur. Claude Cowork est sandboxé : il ne peut accéder qu'aux dossiers que vous lui autorisez explicitement. C'est une bonne nouvelle pour la sécurité.

Skill

Une méthode réutilisable sauvegardée dans Claude Cowork. Quand vous trouvez une façon de faire qui marche bien (par exemple « comment rédiger une bonne newsletter »), vous pouvez demander à l'agent de la sauver comme skill. Vous pourrez la rappeler plus tard avec « *applique le skill newsletter sur ce projet* ». C'est la base de la cumulativité.

SOP (Standard Operating Procedure)

Une procédure standardisée. Dans l'agentique, vos fichiers `.md` sont en fait des SOPs pour l'IA : ils décrivent étape par étape comment faire un travail. La différence avec les SOPs classiques d'entreprise : vos SOPs sont *exécutées* automatiquement par l'IA, pas rangées dans un classeur que personne ne lit.

Sycophantie / Sycophante

Du grec *sykophantes* (délateur, flatteur). La tendance naturelle des IA à être d'accord avec vous et à vous flatter, même quand vous avez tort. C'est un piège important à connaître. La

parade : ajouter une règle explicite dans vos fichiers `.md` pour demander à l'agent d'être franc et contradictoire quand nécessaire. On rencontre parfois la graphie erronée « psychophantie ». Voir chapitre 5.

T

Token

L'unité de base que les IA utilisent pour « compter » le texte. Un token, c'est environ 0,75 mot en français. Les IA facturent au nombre de tokens consommés (input + output). Au niveau débutant, vous n'avez pas besoin d'optimiser les tokens, l'abonnement Pro suffit largement.

V

Vault

Un dossier crypté qu'on peut créer sur Mac (et équivalents sur Windows/Linux). Permet de stocker des mc-files sensibles avec un mot de passe. Utile quand vos agents manipulent des données confidentielles : vous ouvrez le vault uniquement quand vous en avez besoin.

Via negativa

Expression de Nassim Nicholas Taleb signifiant « définir par ce qu'on ne veut pas ». Technique très efficace dans les fichiers `.md` : souvent, dire à votre agent « *ne fais jamais X* » est plus efficace que de lister tout ce qu'il doit faire.

> **Il vous manque un terme ?**
>
> Ce glossaire ne contient que les termes nécessaires pour le niveau débutant. Pour les termes plus avancés (embeddings, fine-tuning, quantization, attention quadratique, orchestration MoE, A2A…), voyez le glossaire du livret avancé.