Eigenflow

GUIDE · MODS

Les mods de Claude Code contre le compactage

You should know et auto-handoff pas à pas : les allumer, régler le seuil selon ton modèle, et ce que notre test a montré.

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté MODS, voici le guide. Depuis la version 2.1.287, sortie le 1er octobre 2026, Claude Code accepte des mods : des plugins en JavaScript ou TypeScript qui tournent à l'intérieur de Claude Code. Un mod peut dessiner dans l'interface, réagir à chaque appel d'outil, lancer une commande ou appeler un modèle.

On en couvre deux. You should know est fourni par Anthropic, éteint par défaut. auto-handoff est un mod tiers, publié sur GitHub, qui remplace le compactage par une nouvelle session et une fiche de relève. Le 18 novembre 2026, on a relu la documentation de Claude Code, son journal des versions et le code du dépôt (version 0.8.6), puis rejoué sa fiche sur une vraie session à nous, sans installer le mod.

À savoir avant de lire

- Claude Code demande un abonnement payant. En France, Claude Pro coûte 18 € HT par mois (21,60 € TTC), ou 15 € HT par mois payé à l'année. Le forfait gratuit ne l'inclut pas.
- Les mods marchent dans le terminal depuis Claude Code 2.1.287, dans l'onglet Code de l'appli de bureau depuis 2.1.286, et dans l'extension VS Code. Pas dans une session WSL.
- You should know fait relire ta conversation par le même modèle que ta session. Chaque vérification envoie plusieurs milliers de tokens d'instructions : ça puise dans ton forfait.
- auto-handoff est du code tiers qui tourne avec tes droits, sans bac à sable. Lis-le avant de le charger.
- Aucun des deux ne garde tout : le premier signale un oubli, le second remplace le résumé de Claude Code par une fiche.

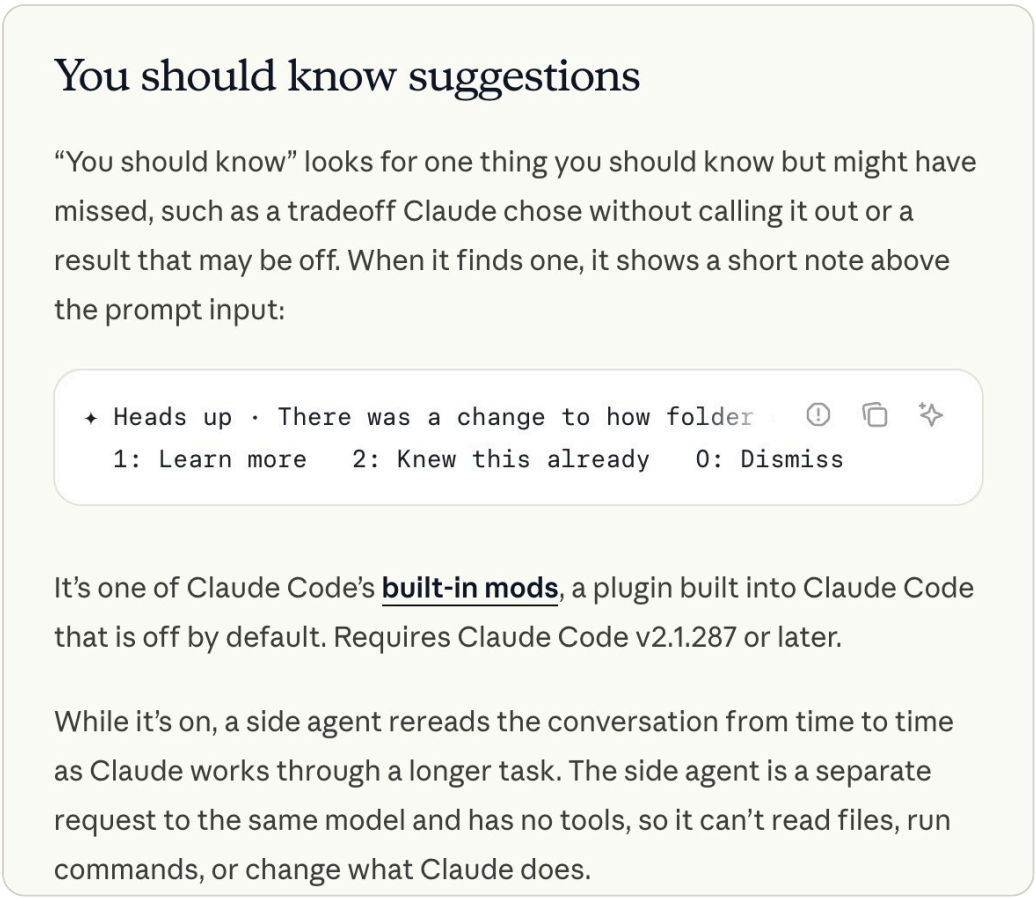
1. Les mods en deux minutes

Un mod réagit à des événements : un appel d'outil, un message envoyé, une partie de l'écran à dessiner. Il peut laisser faire, modifier ou répondre à la place de Claude Code. Certaines fonctions de Claude Code sont déjà des mods, comme `/diff`.

Tape `/plugin` et ouvre l'onglet Installed : les mods d'Anthropic sont rangés sous Built-in, et une ligne grise nomme ceux que tu as ajoutés. Dans `claude -p`, les mods tournent mais ne dessinent rien.

2. You should know, pas à pas

Une fois allumé, un agent annexe relit la conversation de temps en temps pendant une tâche longue. Il cherche une seule chose que tu devrais savoir : un compromis que Claude a fait sans le dire, un résultat qui a l'air faux. S'il en trouve une, une note courte s'affiche au-dessus de la zone de saisie. Cet agent est une requête à part au même modèle, sans outils : il ne lit aucun fichier et ne change pas ce que fait Claude.



La doc de Claude Code : You should know, mod intégré éteint par défaut, version 2.1.287 ou plus (capture du 18 novembre)

L'allumer

Vérifie ta version, il faut 2.1.287 ou plus :

```
À COLLER
claude --version
```

Puis, à l'invite de Claude Code :

```
À COLLER
/plugin enable cc-plugin-you-should-know@builtin
```

Si la réponse dit « is not installed in this project », le mod n'est pas disponible dans cette session. D'après la doc, c'est le cas derrière une passerelle LLM, avec une clé fournie par un apiKeyHelper, avec la variable `DISABLE_TELEMETRY`, et dans la première session après une mise à jour. Il faut une connexion directe, par ton compte claude.ai ou une clé de la Console.

Lire une note

Sous la note, trois choix. Tape le chiffre seul dans la zone de saisie vide.

- 1, Learn more : une explication s'ouvre. Understood la ferme et marque le sujet comme connu, Chat in main session recopie la note dans ta zone de saisie pour en parler à Claude.
- 2, Knew this already : ferme la note et marque le sujet comme connu.
- 0, Dismiss : ferme la note.

Claude ne voit la note que si tu la lui envoies. Les notes suivent ton réglage `language` depuis la version 2.1.290. L'agent ne répète pas les sujets marqués comme connus, même dans tes autres sessions. Pour repartir de zéro, supprime dans `~/.claude/plugins/store/` le fichier dont le nom commence par cc-plugin-you-should-know, quand aucune session ne tourne.

Le couper

```
À COLLER
/plugin disable cc-plugin-you-should-know@builtin
```

Ou dans l'onglet Installed de `/plugin`. Ou encore Turn off suggestions, sur la ligne « Dismissed. » qui suit un Dismiss fait sans ouvrir Learn more (elle reste 20 secondes).

---

3. Pourquoi le compactage fait perdre des détails

Quand la conversation approche de la limite de contexte, Claude Code la remplace par un résumé. D'après sa documentation, le résumé garde tes demandes et les extraits de code importants, mais des consignes détaillées données au début peuvent se perdre.

> When context fills up
>
> Claude Code manages context automatically as you approach the limit. It compacts the conversation into a summary. The summary keeps your requests and key code snippets, but detailed instructions from early in the conversation may be lost. Put persistent rules in CLAUDE.md rather than relying on conversation history.
>
> To control what's preserved during compaction, add a "Compact Instructions" section to CLAUDE.md or run `/compact` with a focus (like `/compact focus on the API changes`).

La doc de Claude Code : ce que garde le compactage et comment le guider (capture du 18 novembre)

Après le résumé, Claude Code relit sur le disque le CLAUDE.md à la racine du projet, la mémoire automatique, le plan, jusqu'à cinq fichiers récemment modifiés et les skills utilisées. Tout le reste passe dans le résumé.

Le moment dépend du modèle. Opus 5.5, Sonnet 5 et suivants, les modèles Fable et Haiku 5.5 ont une fenêtre d'un million de tokens et compactent vers 967 000. Sonnet 4.6 et Opus 4.6 sans contexte étendu compactent à 200 000. Claude Code compacte aussi, sous conditions, une conversation de plus de 200 000 tokens que tu as laissée une heure.

Chez nous, la session principale de Claude Code a été compactée 75 fois depuis le 17 juillet 2026. Le 15 novembre, elle est passée de 894 482 tokens à 58 412.

Les remèdes intégrés, à essayer d'abord

Mets les règles qui doivent tenir dans le CLAUDE.md à la racine du projet : il est relu après chaque compactage. Ajoute-lui une section qui dit quoi garder :

À COLLER

```
# Compact instructions

Garde les décisions prises, les chemins des fichiers modifiés, les commandes de
test et ma dernière demande mot pour mot.
```

Avant une nouvelle tâche longue, compacte toi-même avec une consigne :

À COLLER

```
/compact garde les décisions sur la facturation et la liste des fichiers
modifiés
```

Pour ne résumer qu'une partie : `/rewind`, choisis un message, puis Summarize from here ou Summarize up to here, avec ta consigne sur la ligne « add context (optional) ».

Pour compacter plus tôt (de 100k à 1M, `/autocompact auto` pour revenir) :

À COLLER

```
/autocompact 500k
```

Pour réinjecter un rappel, écrit dans `.claude/rappel.md`, après chaque compactage : un hook SessionStart filtré sur compact, dans le `.claude/settings.json` du projet.

À COLLER

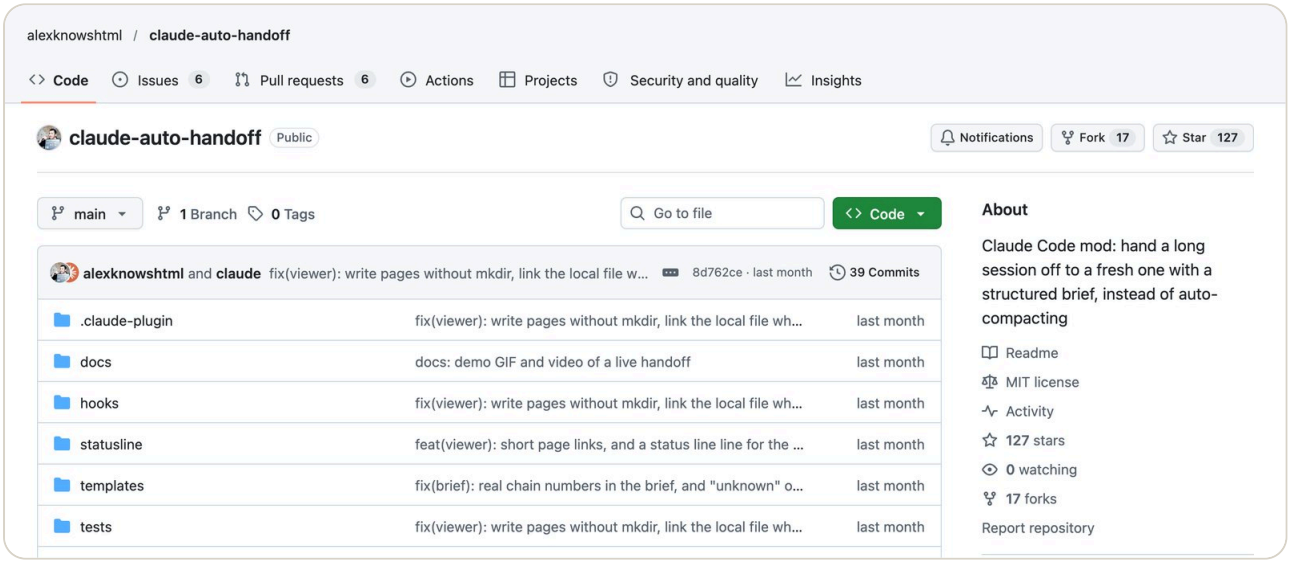
```
{
  "hooks": {
    "SessionStart": [
      { "matcher": "compact", "hooks": [ { "type": "command", "command": "cat
.claude/rappel.md" } ] }
    ]
  }
}
```

Entre deux tâches sans rapport, `/clear` repart à vide et ne coûte rien.

---

4. auto-handoff, pas à pas

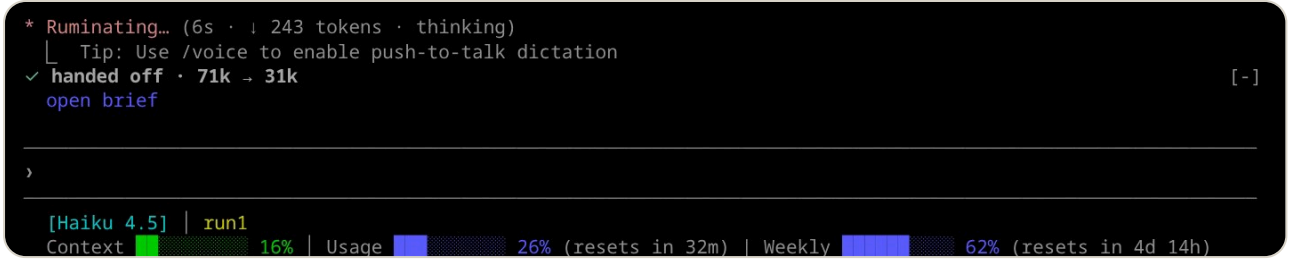
auto-handoff est publié sur GitHub par Alex Hillman, sous licence MIT. Au lieu de résumer sur place, il fait écrire une fiche par Haiku, vide la conversation et relance le travail dans une nouvelle session qui lit cette fiche.



Le dépôt sur GitHub : licence MIT, 127 étoiles et 17 copies au 18 novembre

D’après son README et son code :

- Il mesure le contexte après chaque tour et avant chaque requête. Passé le seuil (160 000 tokens par défaut), il refuse les nouveaux appels d’outils.
- Haiku écrit la fiche à partir des 120 derniers messages, coupés à 2 000 caractères chacun. Les fichiers modifiés et les commits sont relevés par le code.
- Le mod lance `/clear` et envoie à la nouvelle session une ligne qui pointe vers la fiche, rangée dans `~/.claude/state/auto-handoff/`.
- Si le compactage de Claude Code part en premier, le mod l’intercepte et passe le relais à sa place.
- Pas plus de deux relais de suite sans message de ta part.
- Un petit serveur Node affiche chaque fiche en page web, sur ton adresse Tailscale (port 3846) ou sur ta machine seule.



La démo du dépôt : relais à 71k, nouvelle session à 31k, seuil réglé à 80k pour l'essai

L'installer

Il te faut Claude Code 2.1.287 ou plus, git, et Node pour le serveur de pages. Windows marche depuis la version 0.8.5. Récupère le code sans le lancer :

À COLLER

```
git clone https://github.com/alexknowshtml/claude-auto-handoff.git ~/claude-auto-handoff
```

Lis `hooks/register.tsx` et `hooks/brief.ts`. Puis fais lister à Claude Code ce que le mod écoute et ce qu'il demande, sans l'exécuter :

À COLLER

```
claude plugin validate ~/claude-auto-handoff
```

Pour une seule session :

À COLLER

```
claude --plugin-dir ~/claude-auto-handoff
```

Pour toutes tes sessions, dans `~/.claude/settings.json` :

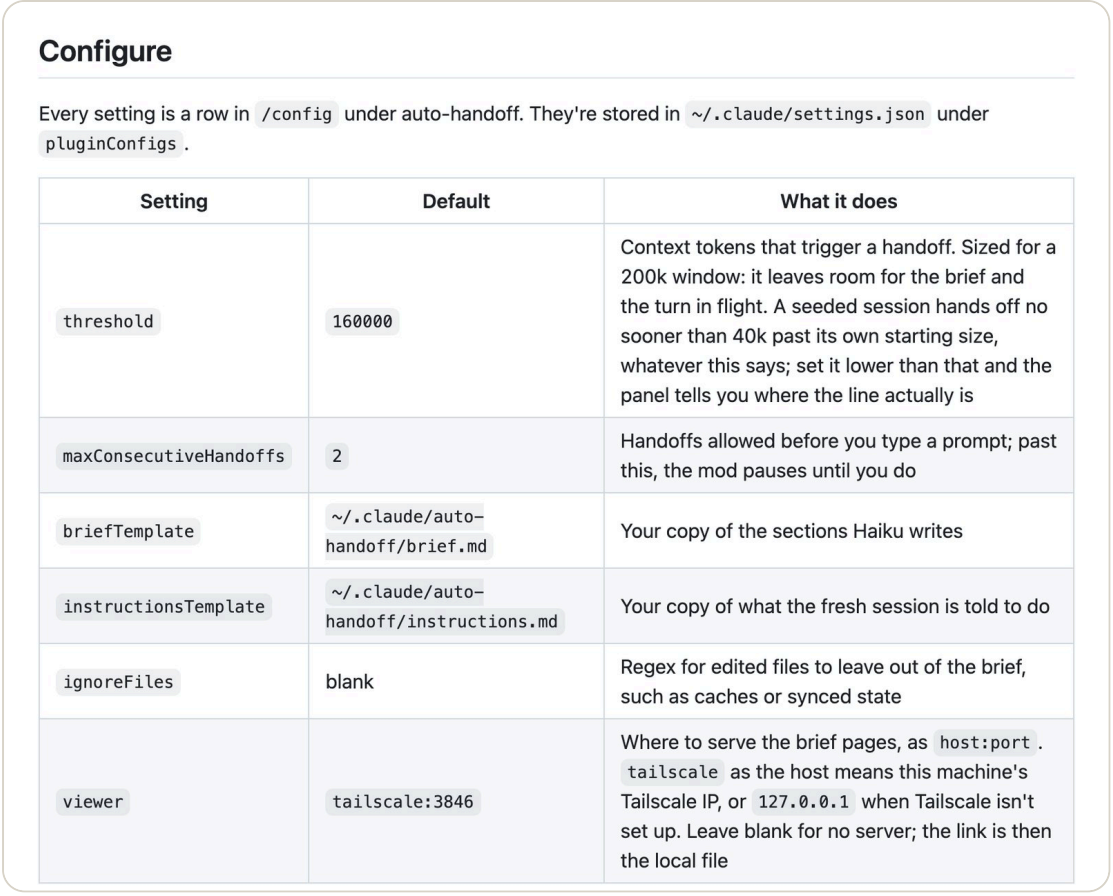
À COLLER

```
{ "env": { "CLAUDE_CODE_PLUGIN_DIRS": "~/claude-auto-handoff" } }
```

Le dépôt n'a pas de fichier de marketplace : pas de `/plugin install` pour l'instant.

Régler le seuil selon ton modèle

Les réglages sont dans `/config`, rubrique auto-handoff. Le seuil par défaut, 160 000 tokens, est prévu pour une fenêtre de 200 000. Sur Opus 5.5, modèle par défaut des forfaits Pro et Max, la fenêtre fait un million : le mod passerait le relais à 160 000, bien avant le compactage vers 967 000.



Configure

Every setting is a row in `/config` under auto-handoff. They're stored in `~/.claude/settings.json` under `pluginConfigs`.

Setting	Default	What it does
`threshold`	`160000`	Context tokens that trigger a handoff. Sized for a 200k window: it leaves room for the brief and the turn in flight. A seeded session hands off no sooner than 40k past its own starting size, whatever this says; set it lower than that and the panel tells you where the line actually is
`maxConsecutiveHandoffs`	`2`	Handoffs allowed before you type a prompt; past this, the mod pauses until you do
`briefTemplate`	`~/.claude/auto-handoff/brief.md`	Your copy of the sections Haiku writes
`instructionsTemplate`	`~/.claude/auto-handoff/instructions.md`	Your copy of what the fresh session is told to do
`ignoreFiles`	blank	Regex for edited files to leave out of the brief, such as caches or synced state
`viewer`	`tailscale:3846`	Where to serve the brief pages, as `host:port`. `tailscale` as the host means this machine's Tailscale IP, or `127.0.0.1` when Tailscale isn't set up. Leave blank for no server; the link is then the local file

Le README : le seuil de 160 000 tokens, prévu pour une fenêtre de 200 000 (capture du 18 novembre)

Garde 160 000 si tu aimes les sessions courtes, moins chères à chaque message. Sinon monte le seuil, à 400 000 par exemple, et regarde dans le journal à quelle taille il passe la main. On n’a testé aucune valeur. Un seuil par modèle est demandé sur le dépôt depuis le 3 novembre (ticket 13).

Changer la fiche

Au premier lancement, le mod copie ses modèles dans `~/.claude/auto-handoff/` : brief.md pour les rubriques de Haiku, instructions.md pour la consigne donnée à la nouvelle session. Modifie ces copies. Chaque titre `##` est une rubrique ; garde « Last Request from the User » et sa ligne Status:. Pour revenir à l’original, supprime ta copie.

Dans notre test, la fiche n’a pas repris les règles données en cours de conversation. La rubrique qu’on ajouterait :

À COLLER

```
## Rules Stated in This Session
Every rule or preference the user stated in the conversation, quoted word for
word.
```

Le couper

Pour une session, serveur compris :

```
À COLLER
env AUTO_HANDOFF_DISABLE=1 claude --plugin-dir ~/claude-auto-handoff
```

Pour de bon, retire la variable de `settings.json`. Sans serveur de pages : vide le réglage viewer dans `/config`, le lien pointe alors vers le fichier.

Lire ses journaux

Tout est noté dans `~/.claude/state/auto-handoff/auto-handoff.log`. Cherche « threshold » (seuil franchi), « tool refused » (appel refusé), « brief written » (fiche écrite) et « haiku brief unusable » (fiche réduite aux faits, quand Haiku a échoué).

---

5. Ce que notre test a montré

On n'a pas installé le mod. Un script Python a refait, d'après `hooks/brief.ts`, la demande du mod à Haiku sur une vraie session à nous, juste avant son compactage automatique du 15 novembre 2026 à 15 h 15. Le modèle de fiche du dépôt est parti à Haiku par `claude -p`, avec le message système exact du mod.

- Haiku a lu les 120 derniers messages, sur 361 depuis le compactage précédent.
- Claude Haiku 5.5 a répondu en 42 secondes : 1 075 mots, 0,0098 dollar au prix de l'API, pris sur l'abonnement.
- Dix rubriques, dont sept points à vérifier avec chacun sa commande, et la dernière demande mot pour mot avec son statut.
- Elle a gardé l'état du moment : deux corrections en attente, quatre brouillons pas encore programmés.
- La liste des commits est restée vide alors que la session en avait fait quatre : `git commit -q` n'affiche pas la ligne que le mod cherche.
- Deux erreurs viennent de notre rejeu : Haiku a pris la date du test pour celle du jour, et cité le dossier temporaire du test.
- Sans le message système du mod, Haiku a refusé. Limite de notre rejeu : le mod l'envoie toujours.

Le résumé écrit par Claude Code au même moment fait 2 826 mots en neuf rubriques, dont tous les messages de l'utilisateur, les erreurs corrigées et huit tâches en attente. Claude Code a gardé en plus les cinq derniers messages mot pour mot. Le compactage n'a pas tout perdu : il a gardé plus que la fiche.

La fiche est 2,6 fois plus courte et dit quoi vérifier. Ses rubriques, tu les choisis. Mais les deux restent des résumés : aucune ne te ramène exactement là où tu étais.

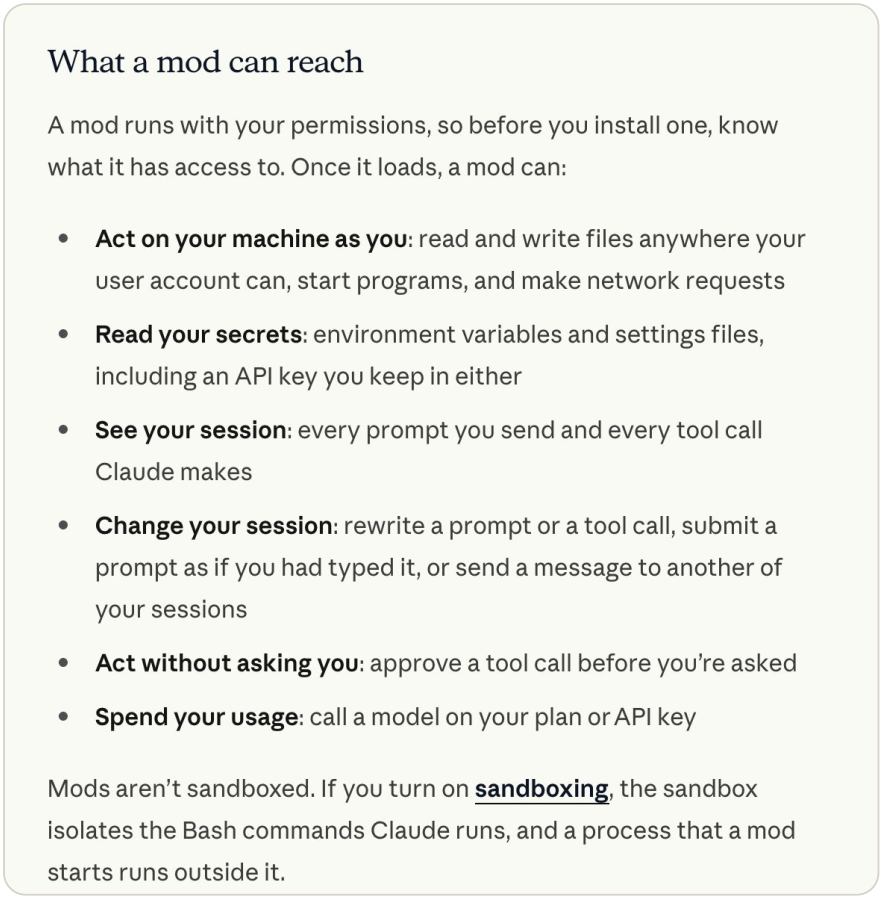
Notre avis : commence par les remèdes intégrés, sans code tiers. auto-handoff vaut l'essai sur une fenêtre de 200 000 tokens, ou si tu veux fixer toi-même ce que garde la fiche. Sur Opus 5.5, règle le seuil avant la première session.

Pas testé : le mod installé (refus d'outils, `/clear`, nouvelle session, pages), un autre seuil, Windows. You should know n'a pas été allumé chez nous.

---

6. Limites et sécurité

- D'après la doc, un mod chargé peut lire et écrire tes fichiers, lancer des programmes, aller sur le réseau, lire tes clés, voir et réécrire chaque message, approuver un appel d'outil à ta place et dépenser ton forfait. Le bac à sable de Claude Code ne l'isole pas.
- La garde intégrée cc-plugin-sec-default ne se charge qu'avec des réglages gérés par une organisation ou une connexion Team ou Enterprise. Sur un Pro ou un Max à ton nom, compte sans elle.



What a mod can reach

A mod runs with your permissions, so before you install one, know what it has access to. Once it loads, a mod can:

- **Act on your machine as you**: read and write files anywhere your user account can, start programs, and make network requests
- **Read your secrets**: environment variables and settings files, including an API key you keep in either
- **See your session**: every prompt you send and every tool call Claude makes
- **Change your session**: rewrite a prompt or a tool call, submit a prompt as if you had typed it, or send a message to another of your sessions
- **Act without asking you**: approve a tool call before you're asked
- **Spend your usage**: call a model on your plan or API key

Mods aren't sandboxed. If you turn on **sandboxing**, the sandbox isolates the Bash commands Claude runs, and a process that a mod starts runs outside it.

La doc de Claude Code : ce qu'un mod peut atteindre une fois chargé (capture du 18 novembre)

- D'après son code, auto-handoff lit la conversation, appelle Haiku, écrit ses fiches, lance `tailscale ip -4` et son serveur Node, tape `/clear` et envoie un message à ta place. On n'y a pas trouvé d'autre requête réseau.
- Les fiches gardent ce que contenait ta session : chemins, adresses de serveurs, identifiants. Elles restent en clair sur ton disque et, avec Tailscale, s'ouvrent depuis tes autres appareils.
- Projet jeune : créé le 3 octobre 2026, dernier commit le 4 octobre, trois contributeurs dont Claude, 6 tickets ouverts, aucune version publiée. Relis les changements avant chaque `git pull`.
- La fiche perd le détail des tâches lancées en arrière-plan (ticket 15), et parfois le rôle donné au premier message (ticket 14).

Sources

- Claude Code : les mods, You should know, le compactage, les seuils, le journal des versions, les prix.
- Le dépôt claude-auto-handoff et son ticket sur le seuil.

Abyane · @abyane_nb · eigenflow.fr

Eigenflow