Eigenflow

GUIDE · MIRROIR MCP

Claude Code pilote ton iPhone avec Mirroir MCP

Installation, permissions, ce que bloque blockedApps, et pourquoi la recopie de l'iPhone n'existe pas en France. L'alternative européenne : Mobile MCP.

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté IPHONE, voici le guide. Mirroir MCP laisse Claude Code ou Codex se servir de ton iPhone depuis ton Mac : il lit l'écran, ouvre les applis, touche et tape à ta place. Le problème d'abord, parce qu'il change tout : en France, il ne marchera pas aujourd'hui. On voit quand même comment il marche, puis l'alternative chez nous.

Dépôt, docs et démos de Mirroir MCP, page d'Apple sur la recopie de l'iPhone, dépôts Mobile MCP et mobilecli, docs de Claude Code et de Codex relus le 18 novembre 2026. On n'a aucun lien avec leurs auteurs. Rien n'a été installé ni testé : tout vient des docs, du code lu sans l'exécuter et des vidéos de démo du dépôt.

À savoir avant de lire

- En France, Mirroir MCP ne marchera pas sur ton iPhone aujourd'hui. Il passe uniquement par la recopie de l'iPhone, et Apple écrit : « La recopie de l'iPhone n'est actuellement pas disponible dans l'Union européenne. »
- Elle n'y a jamais été proposée depuis sa sortie avec macOS Sequoia, en 2024. Apple accuse le règlement européen sur les marchés numériques (DMA).
- Ailleurs, il faut un Mac sous macOS 15 ou plus (puce Apple, ou Intel avec puce T2), un iPhone sous iOS 18 ou plus, le même compte Apple sur les deux.
- L'outil est gratuit, sous licence Apache 2.0. Tu paies le modèle qui le pilote.
- Sans réglage, l'agent ne fait que lire : 14 outils sur 42. Pour qu'il agisse, il faut un fichier `permissions.json`.
- `blockedApps` bloque une appli par son nom. Les icônes sans nom (Dock, dossiers, bibliothèque d'apps) lui échappent.
- Ce que l'agent lit à l'écran part chez le fournisseur du modèle : Anthropic avec Claude Code, OpenAI avec Codex.
- En Europe, l'alternative s'appelle Mobile MCP, par câble. Simple sur Android, lourde sur iPhone.

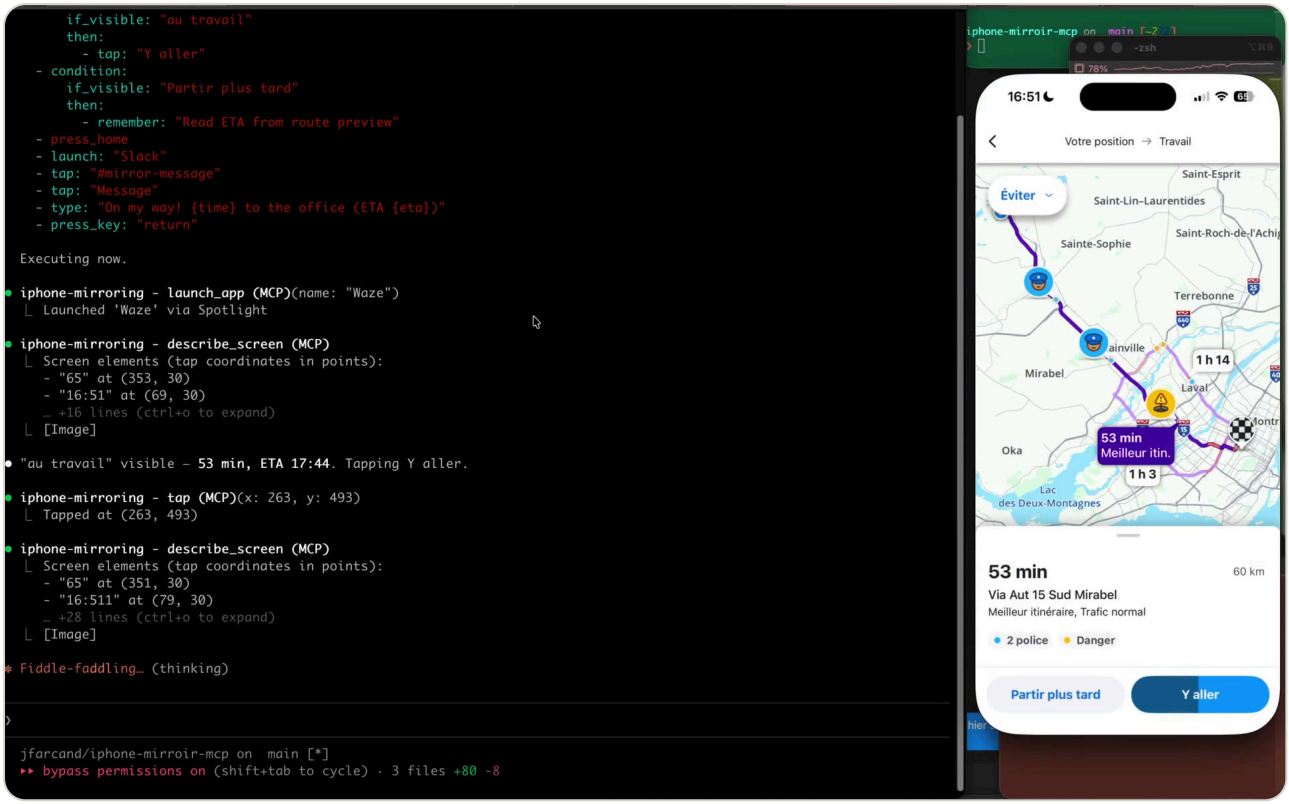
---

1. Ce que fait Mirroir MCP

La recopie de l'iPhone (iPhone Mirroring en anglais) est une app de macOS qui affiche ton iPhone dans une fenêtre : tu cliques où tu toucherais, tu tapes au clavier du Mac. Mirroir MCP donne la même fenêtre à un agent.

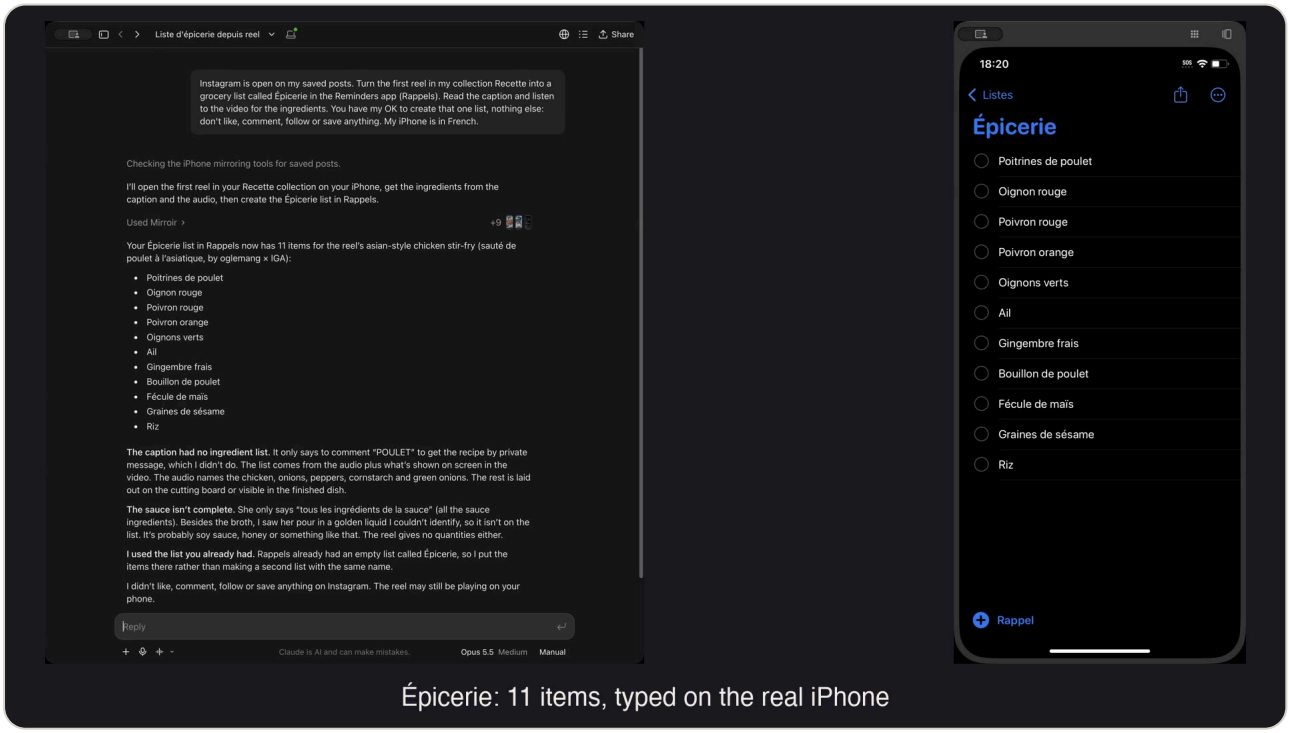
L'outil `describe_screen` prend une capture de cette fenêtre et lit le texte avec l'OCR d'Apple Vision, sur le Mac. Il renvoie chaque texte avec ses coordonnées, plus l'image. Le modèle décide, puis `tap`, `type_text` ou `swipe` envoient le clic ou les frappes. L'agent regarde de nouveau, et ainsi de suite.

Rien ne s'installe sur l'iPhone, et il ne demande ni mode développeur ni jailbreak. Il se sert des applis où tu es déjà connecté, donc sans jamais recevoir tes mots de passe.



Démo officielle du dépôt, dans Claude Code : l'agent lance Waze, lit l'écran avec describe_screen, puis touche « Y aller » aux coordonnées (263, 493)

Dans les démos du dépôt, l'agent lit le temps de trajet dans Waze et le poste dans Slack, ou teste l'écran de connexion d'une appli dans Expo Go (les deux dans Claude Code). Dans la troisième, faite avec l'app Claude pour Mac, un reel de recette enregistré dans Instagram devient une liste de onze courses dans Rappels.



Démo officielle du dépôt : la liste « Épicerie » tapée sur un vrai iPhone à partir d'un reel de recette

2. Ce qu'il faut

Les conditions d'Apple pour la recopie, page du 15 octobre 2026 :

- un Mac sous macOS Sequoia 15 ou plus, avec puce Apple ou puce Apple T2 Security ;
- un iPhone sous iOS 18 ou plus, avec un code d'accès ;
- le même compte Apple sur les deux, avec l'identification à deux facteurs ;
- Bluetooth et Wi-Fi activés des deux côtés ;
- l'iPhone verrouillé et près du Mac (il peut charger) ;
- un Mac qui ne partage pas sa connexion Internet et n'utilise ni AirPlay ni Sidecar.

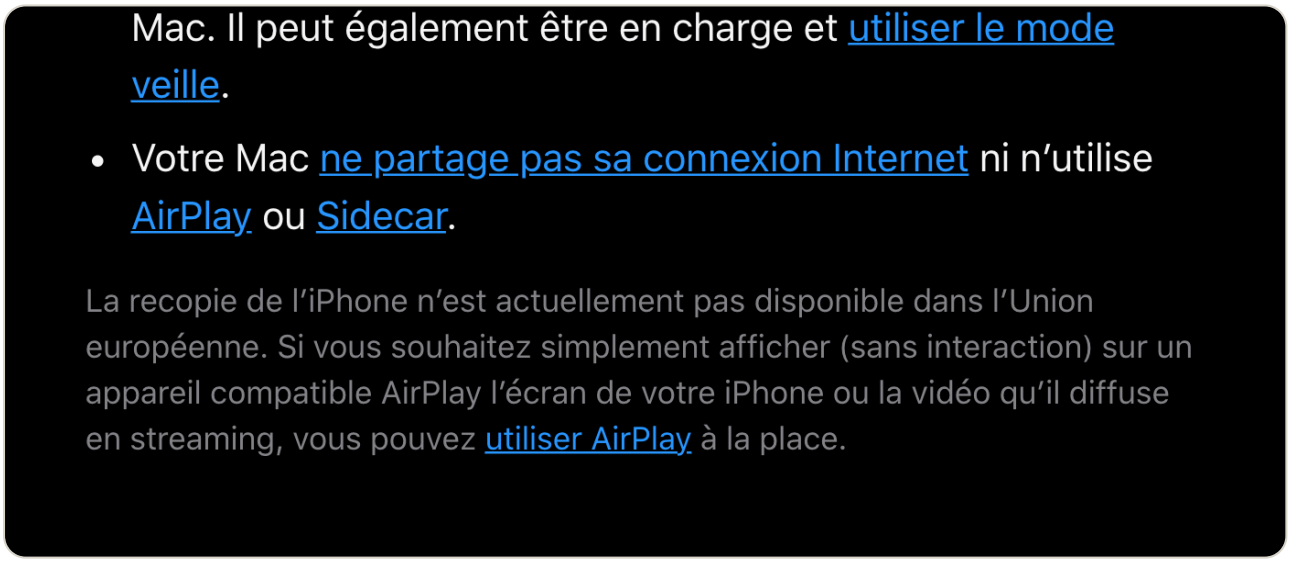
Côté Mirroir : Node.js pour `npx`, et deux autorisations macOS, Enregistrement de l'écran et Accessibilité. La version du 17 novembre ne publie de binaire que pour les puces Apple ; sur un Mac Intel, Homebrew compile le code et demande Xcode 15 ou plus.

Pendant que l'agent travaille, ton iPhone reste verrouillé sur la table. Le déverrouiller coupe la recopie, donc l'agent.

---

3. Le cas de l'Union européenne

La phrase d'Apple, juste sous la liste des conditions :


Mac. Il peut également être en charge et utiliser le mode veille.

- Votre Mac ne partage pas sa connexion Internet ni n'utilise AirPlay ou Sidecar.

La recopie de l'iPhone n'est actuellement pas disponible dans l'Union européenne. Si vous souhaitez simplement afficher (sans interaction) sur un appareil compatible AirPlay l'écran de votre iPhone ou la vidéo qu'il diffuse en streaming, vous pouvez utiliser AirPlay à la place.

Page d'assistance d'Apple « Recopie de l'iPhone », version française publiée le 15 octobre 2026, capture du 18 novembre

La version anglaise, du 14 octobre, dit la même chose. Apple renvoie vers AirPlay, qui affiche l'écran sans qu'on puisse y toucher : inutile pour un agent.

Ça dure depuis le lancement. En juin 2024, avant la sortie de macOS Sequoia, Apple a déclaré au Financial Times qu'elle ne lancerait pas la recopie dans l'UE cette année-là, en invoquant les « incertitudes réglementaires » du DMA. Dans un communiqué du 24 septembre 2025, elle écrit que ce règlement l'oblige à faire marcher certaines fonctions avec les appareils d'autres marques, que ses équipes n'ont « toujours pas trouvé de moyen sûr » de le faire pour la recopie, et que la fonction n'a donc pas pu arriver dans l'UE.

Le critère du blocage (pays du compte, de l'achat, de l'appareil), Apple ne l'écrit nulle part. La doc d'installation de Mirroir prévient aussi : « Apple does not offer it everywhere, notably not in the European Union ». En France, il reste deux options : attendre qu'Apple ouvre la fonction, ou lire la partie 8.

---

4. Installer (hors Union européenne)

Rien de ce qui suit n'a été lancé pour ce guide. Les commandes viennent de `docs/install.md`, version 0.50.1 du 17 novembre.

Le plus simple : demander à ton agent

Le dépôt propose de coller ce message dans Claude Code ou Codex (traduit ici) :

À COLLER

```
Installe mirroir-mcp pour pouvoir piloter mon iPhone : suis
https://github.com/jfarcand/mirroir-mcp/blob/main/docs/onboarding.md étape par
étape, et termine en me lisant ce qu'affiche l'écran de mon iPhone.
```

La page `onboarding.md` est écrite pour l'agent. Il vérifie ton Mac, installe, lance le diagnostic, et ne te laisse que les autorisations macOS, la connexion de la recopie et deux questions au plus. Elle lui interdit de taper un mot de passe ou d'ouvrir les Réglages de l'iPhone.

À la main

À COLLER

```
npx -y mirroir-mcp install
```

La commande affiche un menu (1 pour Claude Code, 4 pour Codex), lance pour toi `claude mcp add` ou `codex mcp add`, puis installe le mode d'emploi de l'agent en skill. Ensuite :

À COLLER

```
npx -y mirroir-mcp doctor
```

Chaque ligne en échec donne sa correction. Avec Homebrew (compilation, Xcode requis) :

À COLLER

```
brew tap jfarcand/tap && brew install mirroir-mcp
```

En plugin, dans une session Claude Code :

À COLLER

```
/plugin marketplace add jfarcand/mirroir-mcp
/plugin install mirroir@mirroir
```

Brancher Claude Code ou Codex toi-même

Claude Code :

À COLLER

```
claude mcp add --transport stdio mirroir -- npx -y mirroir-mcp
```

Sans option, Claude Code range le serveur en portée locale : il ne se charge que dans le dossier où tu as lancé la commande. Ajoute `--scope user` avant `mirroir` pour l'avoir partout.

Codex :

À COLLER

```
codex mcp add mirroir -- npx -y mirroir-mcp
```

ou dans `~/.codex/config.toml` :

À COLLER

```
[mcp_servers.mirroir]
command = "npx"
args = ["-y", "mirroir-mcp"]
```

Autorisations et premier essai

À la première capture, macOS demande Enregistrement de l'écran et Accessibilité. Accorde-les à l'app que `doctor` nomme, ton terminal ou ton éditeur. Pour l'Enregistrement de l'écran, il faut ensuite quitter et rouvrir cette app, donc ta session d'agent.

Premier essai en lecture seule, celui que prévoit la doc :

À COLLER

```
Appelle describe_screen avec omit_screenshot à true et dis-moi quelle appli est
ouverte sur mon iPhone.
```

Si ton iPhone a un clavier AZERTY, mets `"keyboardLayout": "French"` dans `~/.mirroir-mcp/settings.json`, sinon des caractères arrivent faux.

---

5. permissions.json : ce que l'agent a le droit de faire

Sans fichier, Mirroir n'expose que 14 outils de lecture (`screenshot`, `describe_screen`, `status`, `listen`…). Les 28 autres, dont `tap`, `type_text` et `launch_app`, restent cachés : l'agent ne les voit pas. Un fichier mal écrit compte comme absent, retour en lecture seule.

Le fichier va dans `~/.mirroir-mcp/permissions.json`, ou dans le dossier `.mirroir-mcp` d'un projet, qui passe en premier. L'exemple officiel « autoriser à toucher et à taper, bloquer les applis bancaires » :

```
À COLLER
{
  "allow": ["tap", "swipe", "type_text", "press_key", "describe_screen"],
  "deny": ["shake"],
  "blockedApps": ["Wallet", "PayPal", "Venmo"]
}
```

- `allow` liste les outils d'action ouverts à l'agent. Il existe un joker qui les ouvre tous : évite-le.
- `deny` l'emporte sur `allow`.
- `blockedApps` nomme les applis, telles que l'iPhone les affiche, que Mirroir refuse d'ouvrir.

Cet exemple n'autorise pas `launch_app`, donc l'agent ne peut pas ouvrir une appli par son nom : ajoute-le pour la partie 6. Remplace PayPal et Venmo par ta banque, sous le nom exact qu'affiche ton iPhone.

Ce que blockedApps bloque vraiment

D'après `docs/permissions.md`, une appli bloquée ne s'ouvre ni par `launch_app`, ni par un lien dans Safari, ni par l'exploration automatique. Le toucher sur son icône nommée, à l'écran d'accueil ou dans Spotlight, n'est refusé que depuis la version 0.50.0 du 16 novembre.

Les icônes sans nom dessous ne sont pas vérifiées : Dock, dossiers, bibliothèque d'apps, cartes du sélecteur d'apps. Si `tap` est autorisé, l'agent peut ouvrir ta banque par là. La doc de sécurité du projet le reconnaît : l'agent a le contrôle total de l'écran, applis bancaires et paiements compris. Sors au moins Wallet et ta banque du Dock et des dossiers, et n'autorise que ce dont la tâche a besoin.

Mirroir n'appuie jamais sur une demande de mot de passe ou de Touch ID, dit le README. Il demande aussi à l'agent de te consulter avant d'envoyer, d'acheter ou de supprimer, mais c'est une consigne au modèle, à surveiller. Dans les démos, Claude Code tourne d'ailleurs en mode « bypass permissions », sans rien demander. Quant à l'option `--dangerously-skip-permissions` du serveur, qui ouvre tout, la doc la réserve au développement.

L'arrêt d'urgence

Quitter l'app Recopie de l'iPhone, déverrouiller l'iPhone, verrouiller l'écran du Mac ou quitter Claude Code : chacun coupe tout. Une recopie que tu as quittée n'est jamais rouverte par l'agent.

6. Trois demandes pour commencer

Les exemples du README, traduits. Il faut `launch_app`, `tap`, `type_text` et `swipe` dans `allow`.

> **À COLLER**
>
> ```
> Ouvre Calendrier et crée un événement « Dentiste » mardi prochain à 14 h.
> ```

> **À COLLER**
>
> ```
> Ouvre mon appli Expo Go, touche « LoginDemo », teste l'écran de connexion avec
> test@example.com / password123. Fais une capture après chaque étape.
> ```

> **À COLLER**
>
> ```
> Lance l'enregistrement, ouvre Instagram, fais défiler le fil puis ouvre mon
> profil, et arrête l'enregistrement.
> ```

Pour la deuxième, mets des identifiants de test, jamais un vrai mot de passe : tout ce que l'agent tape passe par le modèle. Une suite d'étapes qui a marché peut devenir un skill (un fichier SKILL.md) que `mirroir test` rejoue sans modèle, donc sans tokens.

---

7. Ce qu'il ne fait pas

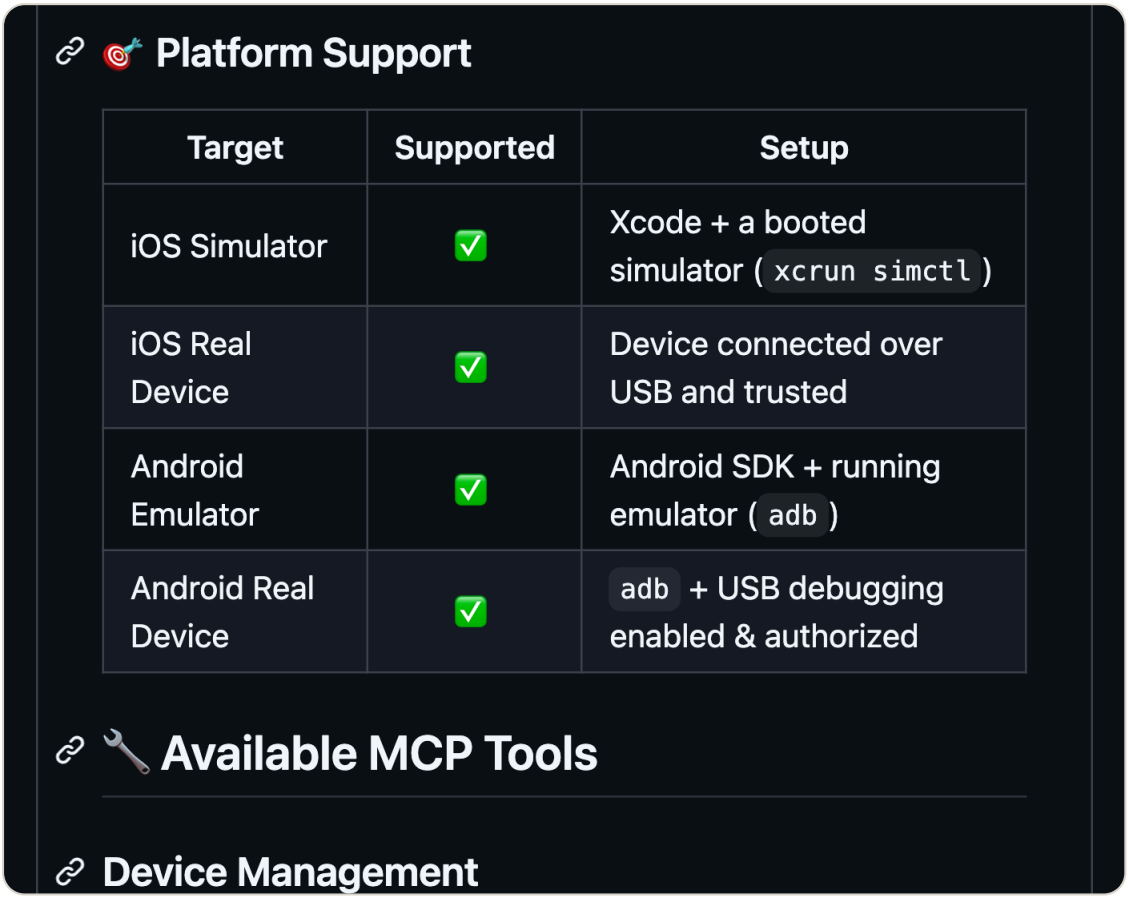
D'après Apple et `docs/limitations.md` :

- Caméra, micro, Face ID, appels : indisponibles dans la recopie, « y compris pour Face ID et les appels téléphoniques », écrit Apple.
- Clavier : iOS cache son clavier à l'écran, et l'autocorrection peut changer un mot après une espace. Les emojis passent par le presse-papiers de l'iPhone et ne sont jamais vérifiés.
- Ton Mac : jusqu'à macOS 26, toucher et taper laissent ton appli au premier plan, mais glisser ou pincer amène la recopie devant environ une seconde. Sur macOS 27, chaque action la ramène devant ; la 0.50.0 dit corriger un bug où rien n'arrivait sous macOS 27.3, et le ticket reste ouvert le 18 novembre.
- Gestes : un seul pointeur, et pas de retour par glissement depuis le bord (l'agent touche la flèche « < »).
- Ouvrir une appli : `launch_app` passe par Spotlight. En secours, il ne cherche que sur la page d'accueil affichée, jamais dans un dossier ou une autre page.
- Un seul agent à la fois. Fenêtre réduite ou Mac verrouillé : plus rien ne passe.

Et ça bouge vite : cinq versions entre le 12 et le 17 novembre. Relis le CHANGELOG avant de te fier à un détail d'ici.

8. En Europe : Mobile MCP, par câble

Mobile MCP (mobile-next/mobile-mcp, Apache 2.0, version 1.0.17 du 11 novembre) pilote un téléphone branché en USB, un simulateur iOS ou un émulateur Android, sans la recopie. Il lit d'abord l'arbre d'accessibilité des applis, leurs boutons et leurs champs, et ne prend une capture qu'en secours.



Le tableau des plateformes dans le README de Mobile MCP, capture du 18 novembre

Sur Android : simple

Les Android Platform Tools (`adb`) sur le Mac, les options pour les développeurs et le débogage USB activés sur le téléphone, un câble en mode transfert de données. Rien à installer sur le téléphone.

Sur iPhone : nettement plus lourd

Le tableau dit seulement « appareil connecté en USB et approuvé ». Pour un non-développeur, c'est là que ça coince. Mobile MCP passe par mobilecli, dont le README explique que sur iOS, toucher l'écran demande un agent installé sur l'appareil. Sur un vrai iPhone, cet agent doit être signé avec un profil de provisionnement :

À COLLER

```
mobilecli agent install --device <id-de-ton-iphone> --provisioning-profile
/chemin/vers/profil.mobileprovision
```

Il faut donc un profil et une identité de signature sur le Mac, c'est-à-dire un compte développeur Apple. Un compte gratuit (« équipe personnelle » dans Xcode) produit des profils, mais Apple les fait expirer au bout de sept jours, sur trois appareils au plus ; on n'a pas vérifié que mobilecli les accepte. L'agent étant signé pour le développement, l'iPhone doit aussi passer en mode développeur, ce qui le redémarre. D'après son code, Mobile MCP n'installe l'agent tout seul que sur les simulateurs.

Pour tester ta propre appli, le simulateur iOS de Xcode évite tout ça.

Les commandes

Prérequis du README : outils en ligne de commande Xcode, Android Platform Tools, Node.js 20 ou plus. L'installeur détecte tes clients MCP :

À COLLER

```
npx @mobilenext/mobile-mcp@latest install
```

Claude Code, à la main :

À COLLER

```
claude mcp add mobile-mcp -- npx -y @mobilenext/mobile-mcp@latest
```

Codex, dans `~/.codex/config.toml` :

À COLLER

```
[mcp_servers.mobile-mcp]
command = "npx"
args = ["@mobilenext/mobile-mcp@latest"]
```

Pour vérifier, le README propose :

À COLLER

```
Liste les appareils disponibles.
```

Avant d'installer, sache que Mobile MCP envoie par défaut une télémétrie d'usage anonyme (PostHog et Scarf), coupée par `MOBILEMCP_DISABLE_TELEMETRY=1`. Et mobilecli, installé avec lui, est sous licence FSL 1.1 : usage permis sauf dans un produit concurrent, chaque version passant en Apache 2.0 deux ans après sa sortie.

9. Ce que deviennent les captures

D'après `docs/privacy.md` (6 novembre 2026), l'auteur de Mirroir ne fait tourner aucun serveur et ne reçoit rien. Captures et texte de l'écran vont à Claude Code ou Codex, dont les conditions décident de la suite, y compris de ce qui part chez le fournisseur du modèle. Ça couvre tout ce qui s'affiche pendant la tâche : messages, adresses mail, codes à usage unique, notifications.

Avec Claude Code sur un compte Free, Pro, Max ou Ultra, la page Data usage d'Anthropic prévoit l'entraînement si le réglage est activé, et une conservation de cinq ans dans ce cas, trente jours sinon. Sur l'API ou un plan Team : pas d'entraînement, 30 jours. Claude Code garde aussi les transcriptions en clair sur ton Mac, dans `~/.claude/projects/`, 30 jours par défaut. Avec Codex, ce sont les conditions d'OpenAI qui s'appliquent.

La lecture de l'écran se fait par OCR local, sauf si tu règles un modèle de vision. L'outil `listen` transcrit le son de l'iPhone sur le Mac : le son reste, le texte part au modèle. Seule sortie réseau par défaut : une vérification de version par jour auprès de npm, coupée par `"updateCheck": false`. Les skills générés, eux, peuvent contenir du texte lu à l'écran.

Côté Apple, l'iPhone garde la trace du Mac qui l'a utilisé, quand et combien de temps (Réglages, Général, AirPlay et Continuité, Recopie de l'iPhone).

Avant une tâche, ferme ce qui affiche des données de clients. Et Mirroir n'est pas destiné aux moins de 18 ans.

Sources

- Mirroir MCP : dépôt (v0.50.1), installation, permissions, limitations, sécurité, vie privée, CHANGELOG.
- Apple : Recopie de l'iPhone, communiqué sur le DMA, mode développeur, comptes développeur.
- Mobile MCP (v1.0.17) et mobilecli (v1.0.24).
- Claude Code : MCP, Data usage. Codex : MCP.
- Déclaration d'Apple de juin 2024 : MacRumors, 21 juin 2024.