

Claude Code Agent Teams

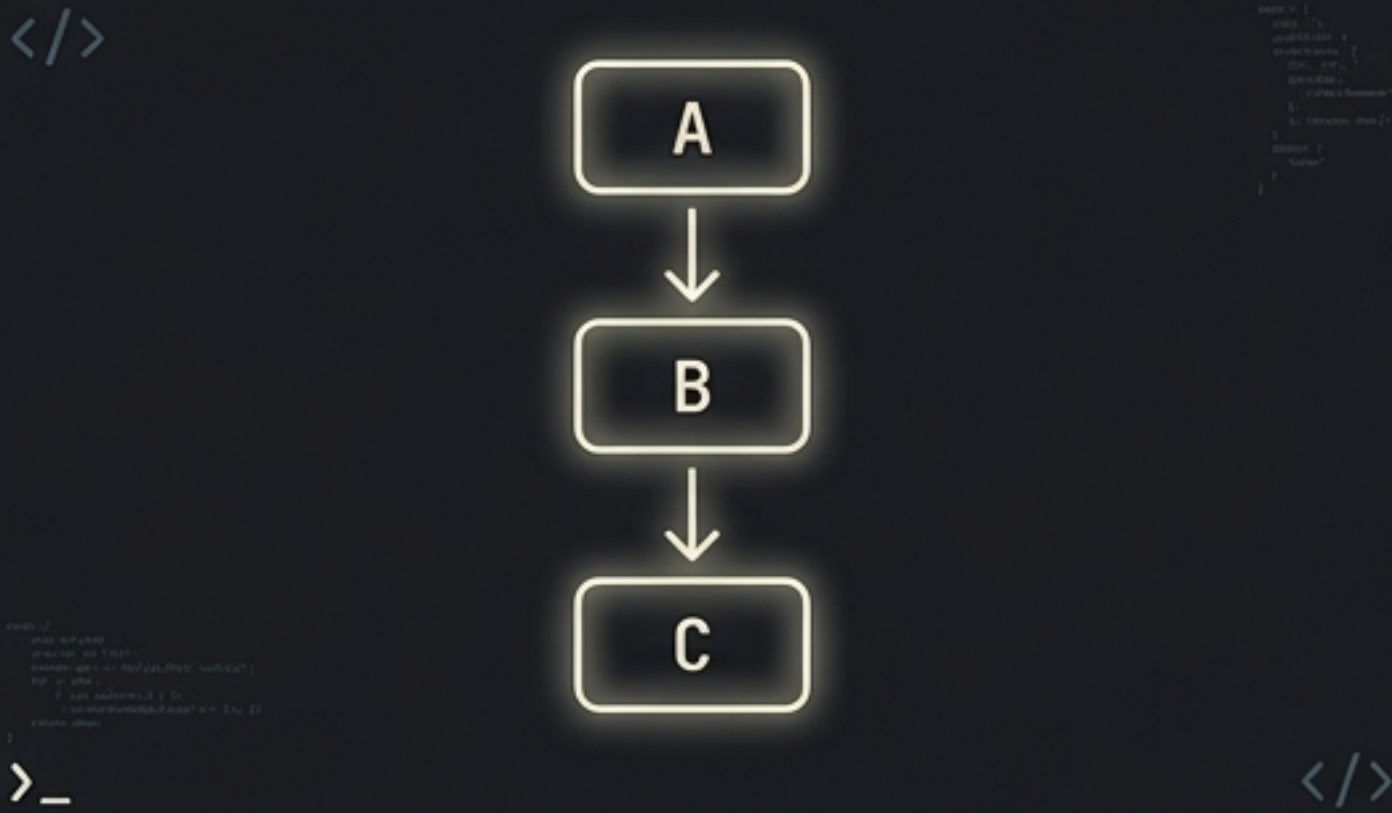
Transformez un Développeur en Équipe Complète

Guide technique de l'orchestration parallèle
et des agents autonomes

```
> $ export CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1
```

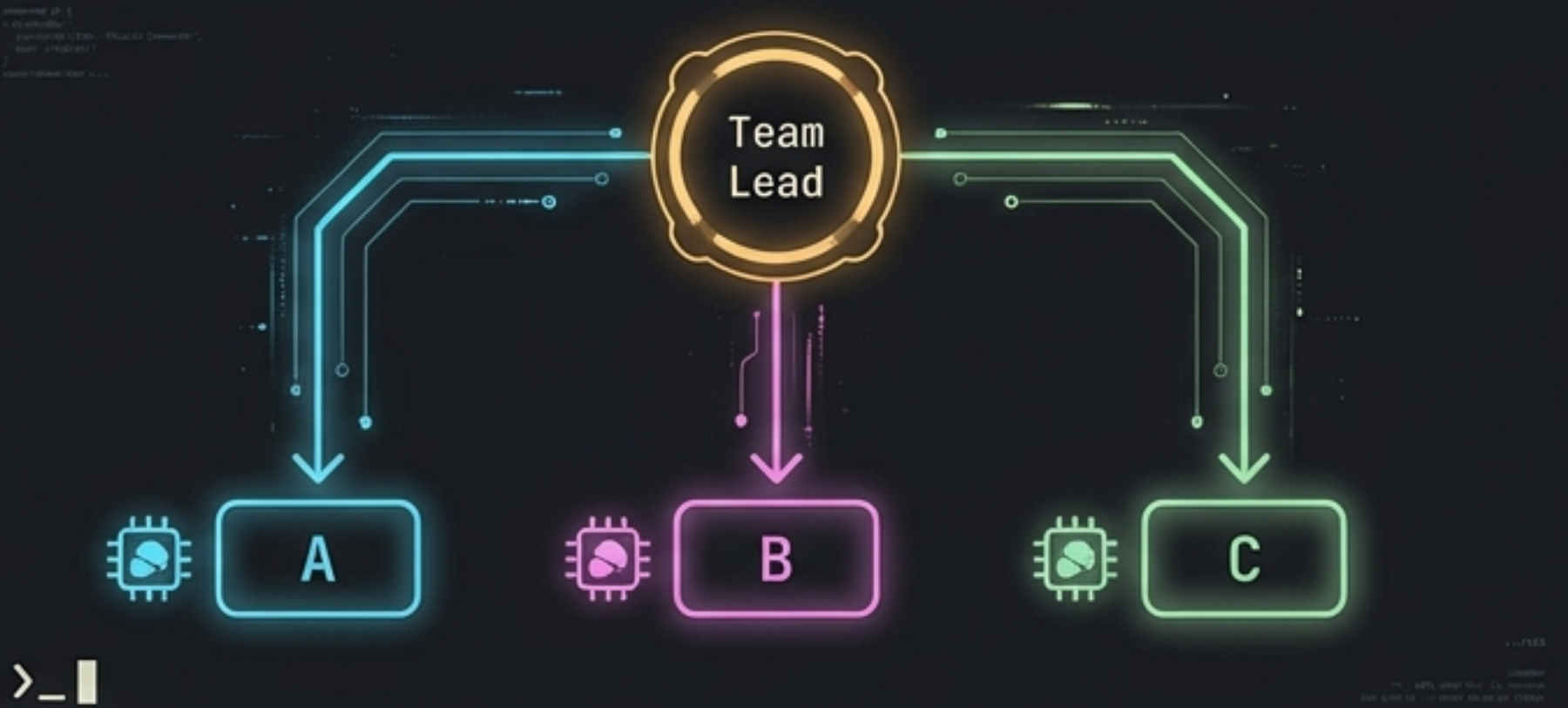

>_ Le Changement de Paradigme : Du Séquentiel à l'Essaim

L'Ancienne Méthode (Séquentiel)



Un agent unique exécute l'étape A, puis B, puis C.

La Nouvelle Méthode (Parallèle / Swarm)



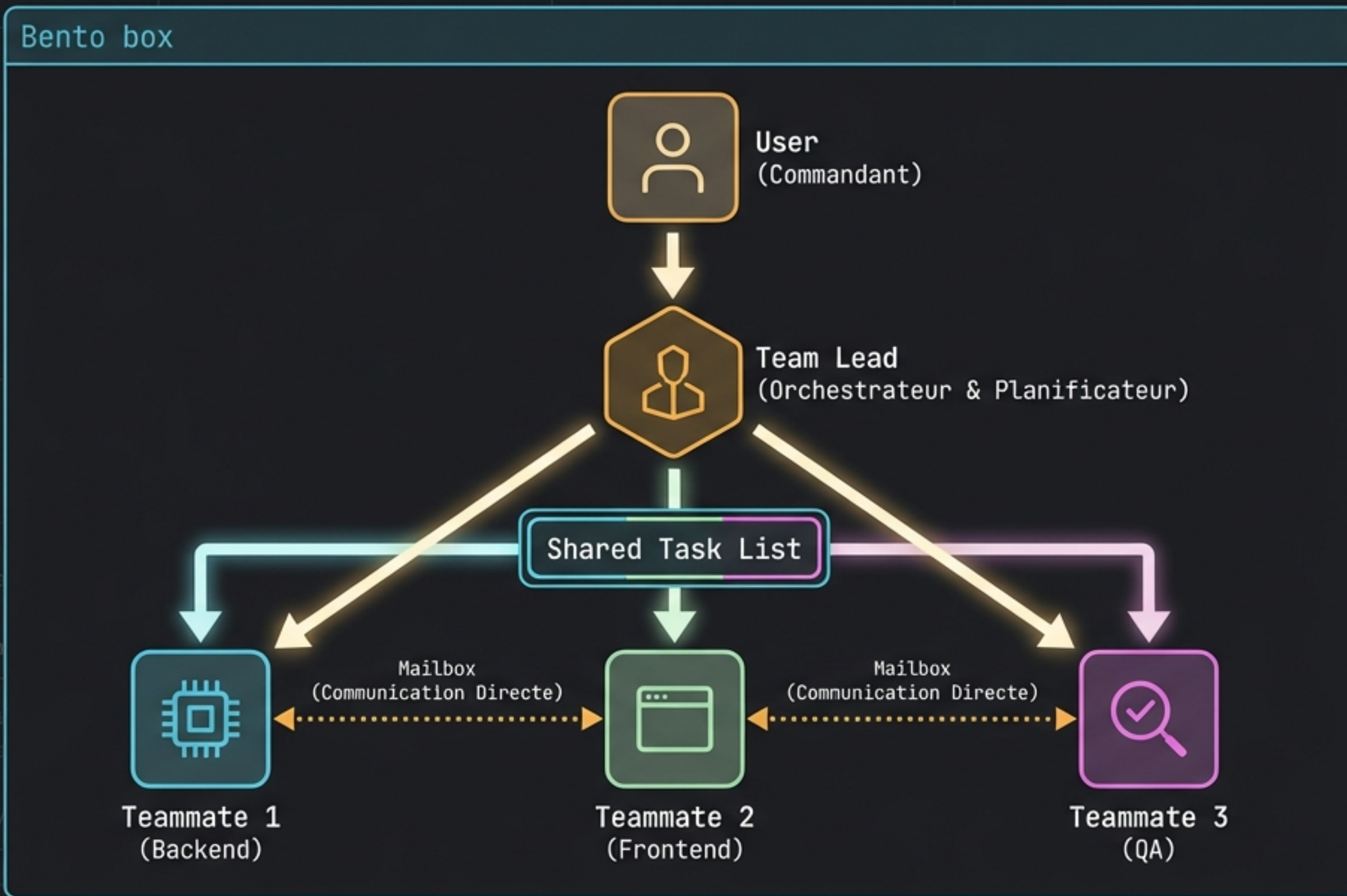
Un 'Team Lead' déploie trois agents spécialisés qui attaquent A, B et C simultanément.

/* C'est incroyable, je pèse mes mots... trois agents qui travaillent en même temps. J'ai l'impression d'être dans un film. — Melvynx */




Rôle : Mission Commander

>_ L'Architecture d'une Équipe



Bento box

- **Team Lead** : Instance principale. Crée le plan et assigne.
- **Teammates** : Instances indépendantes avec leur propre Context Window.
- **Shared Task List** : Liste centralisée accessible à tous.

 Rôle : Mission Commander

>_ L'Interface Visuelle (Tmux & Split Panes)

Pane 1: Team Lead

Team Lead:

> Initialisation de l'équipe.

Team Lead:

> Affectation des tâches : UI à Agent 2,
Widget à Agent 3, Backend à
Agent 4.

Agent 2:

> Reçu. Démarrage de l'interface.

Agent 3:

> En cours. Construction du widget.

Agent 4:

> Connecté à la base de données. En
attente de requêtes.

Pane 2: Admin UI Agent

```
import React from 'react';
import { AdminDashboard } from './components';

const App = () => {
  return (
    <div className='admin-ui'>
      <AdminDashboard />
    </div>
  );
};

export default App;
```

Pane 3: Widget Agent

```
[=====>    ] 65% - Building Assets...

Status: Compiling components. No errors. ✓
```

Pane 4: Backend Agent

```
[16:32:05] INFO: Database connection established.
[16:32:08] LOG: User authentication request
received for 'admin'.
[16:32:10] SUCCESS: User 'admin' logged in
successfully.
[16:32:15] QUERY: SELECT * FROM users WHERE
role='admin';
```

Spécifications Techniques

- **Split Panes** : Chaque agent a son propre terminal.
- **Prérequis** : Tmux (macOS/Linux) ou iTerm2.
- **Fallback** : Mode 'In-process' (Background + Navigation Shift+Up/Down).



Rôle : Mission Commander

>_ Le Cycle de Vie d'une Mission

Bento box



1. Spawn (Déploiement)

Le Lead analyse le prompt et crée l'équipe.



2. Assign (Distribution)

Génération de la Shared Task List. Les agents 'claim' les tâches.



3. Work & Coordinate

Code, tests et messages inter-agents ('Task unblocked').



4. Shutdown

Graceful Shutdown des sessions et nettoyage.



Rôle : Mission Commander

> _ Activation & Configuration

Terminal (Bash)

```
export CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1
```

settings.json

```
{  
  "env": {  
    "CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS": "1"  
  },  
  "teammateMode": "auto" // ou "tmux" / "in-process"  
}
```

Commande de Lancement :

```
> Create a team with 3 teammates to build a refund feature...
```

Note : Fonctionnalité expérimentale, désactivée par défaut.

>_ Mécanismes de Contrôle & Delegate Mode

Bento box



Delegate Mode (Shift+Tab)

Force le Team Lead à la coordination uniquement. Il délègue et ne touche pas au code.



Plan Approval

Exige une validation humaine avant l'exécution critique (ex: modification DB).



Intervention Directe

L'utilisateur peut prendre le contrôle d'un pane spécifique pour corriger manuellement.

> _ Cas d'Usage Stratégiques

Bento box

Recherche & Revue



3 agents analysent une PR simultanément sous des angles différents : Sécurité, Performance, UX.

Features Full-Stack



Parallélisation des couches :
Un agent sur la Base de Données, un sur l'API, un sur le Frontend.

Competing Hypotheses



Scénario Débogage : 5 agents testent 5 théories de bug différentes pour trouver la cause racine plus vite.

>_ Best Practices pour l'Orchestration

Bento box



Context Isolation

Context Isolation

Les teammates ne partagent pas l'historique du Lead. Donnez un contexte précis à chaque spawn.



Atomic Tasks

Atomic Tasks

Découpez le travail. Les instructions vagues entraînent des boucles infinies.



File Conflicts

File Conflicts

Assurez-vous que les agents travaillent sur des fichiers distincts (évitez les race conditions).

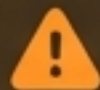


CLAUDE.md

Utilisez ce fichier racine pour les instructions globales d'architecture.

> Limitations & Réalité (Coûts)

⚠ Consommation de Tokens



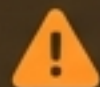
Significativement plus élevée.
Coût proportionnel à la taille de
l'équipe (N agents = N instances).

⚠ Pas de Resume



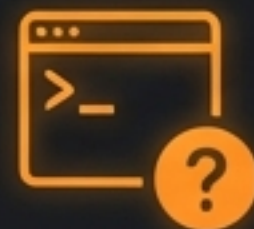
Impossible de restaurer une
session d'équipe avec `/resume`.
En cas de crash, l'état est perdu.

⚠ Single Thread Lead



Le Lead ne peut gérer qu'une
équipe à la fois. Pas de "nested
teams" (équipes imbriquées).

⚠ Orphaned Sessions



Commande de secours si Tmux
ne se ferme pas : `tmux`
`kill-session`.

>_ L'Avenir de la Vitesse de Développement



Nous sommes passés de l'assistance au code à la gestion d'équipe virtuelle.

Bien qu'expérimental, cela permet une exploration parallèle impossible auparavant.

⚠ Start Small : Ne commencez pas par une refonte critique.

Commencez par la Recherche ou l'Audit.

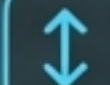
>_ Ready to spawn your first team?

>_ Mémento du Commandant (Cheat Sheet)

Shortcuts 1

Shift+Tab

Toggle Delegate Mode

Shift+Up/Down 

Cycle teammates (In-process)

Ctrl + **T**

Toggle Task List view

Commands 2

> **tmux ls**

List active sessions

> **tmux kill-session**

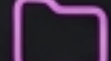
Kill stuck sessions

> **export**

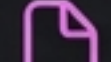
CLAUDE_CODE_EXPERIMENTA
L_AGENT_TEAMS=1

Enable Experimental Teams

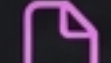
Files 3

 `~/.claude/teams/`

Config location

 `settings.json`

Global config

 `CLAUDE.md`

Project context