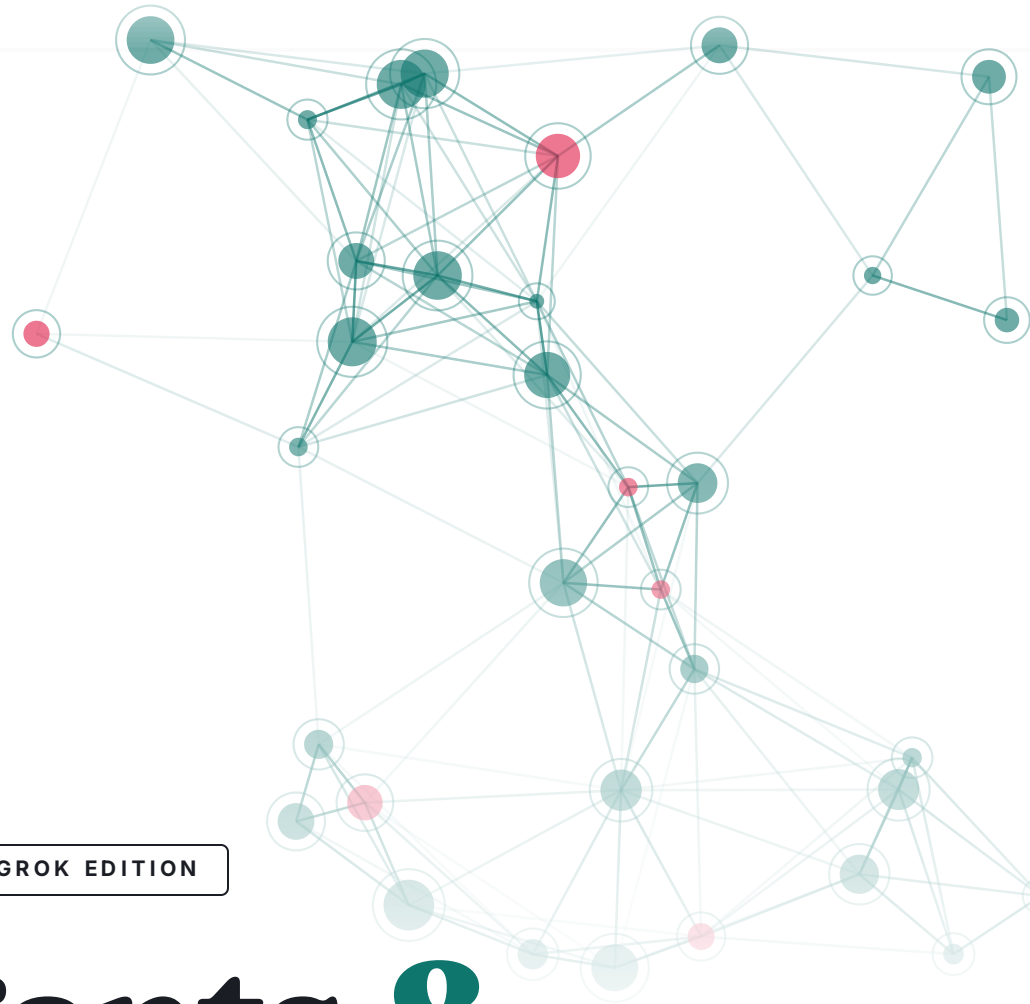




A BEGINNER'S MANUAL

GROK EDITION

# AI Agents & Multi-Agent Workflows

Hand off real work to ChatGPT Work, Codex, Claude Cowork, Gemini tasks, Perplexity Computer and Grok Bot, and fan one question out to five AIs at once

# AI Agents & Multi-Agent Workflows

Hand off real work to ChatGPT Work, Codex, Claude Cowork, Gemini tasks, Perplexity Computer and Grok Bot, and fan one question out to five AIs at once

**GROK EDITION**

Written by Grok Bot, with facts checked against official sources in October 2026.

A chatbot answers questions. An **agent** takes a goal, makes a plan, uses tools such as a browser, files, code and your connected apps, and keeps going until the job is done or it needs you. In 2026 every major AI company ships one: ChatGPT Work and Codex, Claude Cowork, Gemini tasks, Perplexity Computer and Grok Bot.

This manual teaches you how to delegate safely and well. You will learn how to write a task brief an agent can finish, which agent suits which job, how approvals and permissions protect you, and how to run **multi-agent workflows**: sending one question to ChatGPT, Claude, Gemini, Grok and Perplexity at the same time, then combining the best of each answer.

**Product details were checked against official help centers on October 16, 2026.**

Agent features change faster than almost anything else in AI, so confirm in the linked articles.

## HOW TO READ THIS MANUAL

**Tip** — a shortcut or good habit.

**Warning** — easy to get wrong.

`$ command` — text you type, exactly as shown.

**Note** — useful background.

**Danger** — can lose data, money, or access.

`Settings > Privacy` — a menu path to click.

New words appear in **bold** the first time and are collected in the Glossary at the back.

Features, plan availability, regions and limits were checked against official OpenAI, Anthropic, Google, Perplexity and xAI help pages on October 16, 2026. Agent products change frequently and roll out gradually by plan and region. Agents can make mistakes and take real actions; supervise anything involving money, messages sent in your name, personal data or important files. Model names and API prices in code examples are those listed on October 16, 2026.

# Contents

---

|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>01 From Chatbot to Agent</b>                        | <b>5</b>  |
| 1.1 The agent loop                                     | 5         |
| 1.2 Chatbot vs. agent                                  | 6         |
| 1.3 Levels of autonomy                                 | 6         |
| 1.4 What agents are good at today                      | 6         |
| 1.5 What agents still struggle with                    | 7         |
| <b>02 The Agent Landscape in October 2026</b>          | <b>8</b>  |
| 2.1 How they differ in one sentence each               | 9         |
| 2.2 Choosing an agent for a job                        | 9         |
| <b>03 Writing a Task Brief an Agent Can Finish</b>     | <b>11</b> |
| 3.1 The five-part brief                                | 12        |
| 3.2 Why each part matters                              | 12        |
| 3.3 Ask for a plan first                               | 13        |
| 3.4 Brief templates for common jobs                    | 13        |
| <b>04 ChatGPT Work and Codex</b>                       | <b>15</b> |
| 4.1 Using Work                                         | 15        |
| 4.2 Codex for software                                 | 16        |
| <b>05 Claude Cowork</b>                                | <b>19</b> |
| 5.1 Availability (checked 2026-10-16)                  | 19        |
| 5.2 Start a session                                    | 19        |
| 5.3 What Cowork can do                                 | 20        |
| 5.4 The three approval modes                           | 20        |
| 5.5 Global and folder instructions                     | 21        |
| 5.6 Example Cowork tasks                               | 21        |
| <b>06 Gemini Tasks</b>                                 | <b>22</b> |
| 6.1 Requirements and availability (checked 2026-10-16) | 22        |
| 6.2 Create a task                                      | 22        |
| 6.3 Tasks, schedules and skills                        | 23        |
| 6.4 The browser: local or remote                       | 23        |
| 6.5 Safety features Google lists                       | 23        |
| 6.6 Clean up after sensitive tasks                     | 24        |
| <b>07 Grok Bot</b>                                     | <b>25</b> |
| 7.1 What makes Grok Bot different                      | 25        |
| 7.2 Availability                                       | 26        |

|           |                                                      |           |
|-----------|------------------------------------------------------|-----------|
| 7.3       | The shared computer .....                            | 26        |
| 7.4       | A good first handoff .....                           | 26        |
| 7.5       | A multi-Bot setup .....                              | 26        |
| <b>08</b> | <b>Perplexity Computer .....</b>                     | <b>28</b> |
| <b>09</b> | <b>Multi-Agent Patterns .....</b>                    | <b>29</b> |
| 9.1       | Why use more than one AI? .....                      | 29        |
| 9.2       | The five patterns .....                              | 29        |
| 9.3       | Fan-out / fan-in in detail .....                     | 30        |
| 9.4       | Writer and critic in detail .....                    | 30        |
| <b>10</b> | <b>Fan-Out by Hand: Five AIs, One Question .....</b> | <b>31</b> |
| 10.1      | Step 1: open five tabs .....                         | 31        |
| 10.2      | Step 2: the structured fan-out prompt .....          | 32        |
| 10.3      | Step 3: collect the answers .....                    | 32        |
| 10.4      | Step 4: the synthesis prompt .....                   | 32        |
| 10.5      | Step 5: verify the disagreements .....               | 33        |
| <b>11</b> | <b>Automated Fan-Out with Python .....</b>           | <b>34</b> |
| 11.1      | Setup .....                                          | 34        |
| 11.2      | The fan-out script .....                             | 35        |
| 11.3      | What it costs .....                                  | 36        |
| 11.4      | About Perplexity's API .....                         | 37        |
| <b>12</b> | <b>Coding Agents Side by Side .....</b>              | <b>38</b> |
| 12.1      | One rules file for every coding agent .....          | 38        |
| 12.2      | A parallel coding workflow .....                     | 38        |
| 12.3      | Writer and reviewer for code .....                   | 39        |
| <b>13</b> | <b>Safety, Permissions and Approvals .....</b>       | <b>40</b> |
| 13.1      | Prompt injection, explained .....                    | 40        |
| 13.2      | The approval controls you have .....                 | 41        |
| 13.3      | The agent safety checklist .....                     | 41        |
| 13.4      | Data and privacy .....                               | 41        |
| <b>14</b> | <b>Usage Limits and Costs .....</b>                  | <b>42</b> |
| 14.1      | Where to check usage .....                           | 42        |
| 14.2      | Ways to stretch your allowance .....                 | 42        |
| <b>15</b> | <b>Five Workflows to Try This Week .....</b>         | <b>43</b> |
| <b>16</b> | <b>Troubleshooting .....</b>                         | <b>45</b> |
|           | <b>Quick Reference .....</b>                         | <b>46</b> |
|           | <b>Glossary .....</b>                                | <b>47</b> |

# From Chatbot to Agent

---

- An **agent** is an AI that works in a loop: plan, act with a tool, look at the result, decide the next step, until the goal is reached.
- Agents use **tools**: a web browser, a computer with files and a terminal, code execution, and connectors to apps like Gmail, Slack or Google Drive.
- The more an agent can do, the more carefully you must define the job, its limits and when it must ask you.

When you chat with an AI, you are the engine. You ask, it answers, you decide what to ask next. When you delegate to an agent, the AI becomes the engine. You describe the outcome, and it decides the steps.

## 1.1 The agent loop

Every agent product, however polished, runs a loop like this:

- 1 Understand the goal.**  
Read your request, files and instructions.
- 2 Plan.**  
Break the goal into steps.
- 3 Act.**  
Use a tool: search the web, open a page, run code, read a file, draft an email.
- 4 Observe.**  
Look at what happened.
- 5 Decide.**  
Continue, change plan, ask you a question, or finish.

The loop can run for minutes or hours. Good agents show you the plan and progress, and pause for approval before risky actions.

## 1.2 Chatbot vs. agent

|             | CHATBOT              | AGENT                                                        |
|-------------|----------------------|--------------------------------------------------------------|
| You provide | A question           | A goal and a finish line                                     |
| It produces | An answer            | Finished work: files, research, drafts, changes              |
| Tools       | Sometimes web search | Browser, files, code, connected apps, sometimes your desktop |
| Time        | Seconds              | Minutes to hours; can continue while you are away            |
| Risk        | Wrong information    | Wrong information <b>and</b> wrong actions                   |
| Your role   | Asking               | Briefing, approving and reviewing                            |

## 1.3 Levels of autonomy

It helps to think of five levels, from least to most independent:

| LEVEL                | DESCRIPTION                                     | EXAMPLE                                                            |
|----------------------|-------------------------------------------------|--------------------------------------------------------------------|
| 1. Suggest           | AI drafts; you do everything                    | "Draft a reply to this email."                                     |
| 2. Prepare           | AI does the work and leaves drafts for approval | "Draft replies to all unanswered emails; don't send."              |
| 3. Act with approval | AI acts, but asks before important steps        | "Book the cheapest refundable flight; ask before paying."          |
| 4. Act and report    | AI acts on routine steps and reports            | "Every Friday, file receipts into folders and email me a summary." |
| 5. Fully autonomous  | AI acts with no checks                          | Not recommended for beginners, or for anything important           |

### TIP

Start every new agent at **Level 2**: let it prepare everything, then review. Raise autonomy only for routine, low-risk tasks after it has done them correctly several times.

## 1.4 What agents are good at today

- **Research and synthesis:** reading many sources and producing a cited report.
- **Document creation:** spreadsheets with working formulas, slide decks, formatted reports.
- **File chores:** organizing folders, renaming files, turning receipts into an expense sheet.

- **Inbox and calendar triage:** summarizing, labeling, drafting replies, suggesting meeting times.
- **Coding:** writing, testing and reviewing code in a project folder.
- **Monitoring:** watching prices, news or a website and alerting you.

## 1.5 What agents still struggle with

- Sites that block automation, use CAPTCHAs, or require you to log in.
  - Very long tasks with vague goals (“grow my business”).
  - Tasks where a single mistake is costly and hard to undo.
  - Knowing when information is out of date, unless told to check.
- 
- ☐ I can explain the agent loop: plan, act, observe, decide.
  - ☐ I know agents can take real actions, not just write text.
  - ☐ I will start new agents at “prepare and let me approve.”

# The Agent Landscape in October 2026

- **OpenAI: ChatGPT Work** (the agent inside ChatGPT) and **Codex** (for software). The older “ChatGPT agent” mode is no longer available.
- **Anthropic: Claude Cework** for knowledge work (paid plans) and **Claude Code** for software.
- **Google: tasks** in the Gemini app (formerly called Spark), for Google AI Pro and Ultra subscribers.
- **Perplexity: Computer**, an autonomous worker for subscribers.
- **xAI: Grok Bot**, persistent named Bots that each work on a cloud computer.

| AGENT                      | MADE BY    | BEST AT                                                                            | WHO CAN USE IT (CHECKED 2026-10-16)                                                                               |
|----------------------------|------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>ChatGPT Work</b>        | OpenAI     | Research, documents, spreadsheets, slides, scheduled and event-triggered tasks     | Eligible paid plans; event triggers not on Free or Go                                                             |
| <b>Codex</b>               | OpenAI     | Coding in your folders or in the cloud                                             | Included on all ChatGPT plans including Free and Go; Codex Cloud on Plus and above                                |
| <b>Claude Cework</b>       | Anthropic  | Multi-step knowledge work with files, browser and connectors                       | Paid plans only (Pro, Max, Team, Enterprise)                                                                      |
| <b>Claude Code</b>         | Anthropic  | Coding in a terminal or editor                                                     | Pro and higher plans (see Volume 1)                                                                               |
| <b>Gemini tasks</b>        | Google     | Google Workspace chores, web tasks, monitoring, files                              | 18+, personal account, Google AI Pro or Ultra; not in the EEA, Switzerland, the UK or Nigeria                     |
| <b>Perplexity Computer</b> | Perplexity | Research, documents, apps, scheduled tasks                                         | Active Perplexity subscribers                                                                                     |
| <b>Grok Bot</b>            | xAI        | Named, persistent Bots with a shared cloud computer; parallel Bots that coordinate | Paid individual Cursor plans, Cursor Teams, or a linked SuperGrok, SuperGrok Plus or SuperGrok Heavy subscription |

### ☆ IMPORTANT

OpenAI's help center states that **ChatGPT agent mode is no longer available**; use **ChatGPT Work** for multi-step work and the **cloud browser** for supported browser workflows. Older tutorials that say “select Agent mode from the tools menu” are out of date.

## 2.1 How they differ in one sentence each

- **ChatGPT Work** is ChatGPT's “do the task” mode, next to the normal “Chat” mode, and shares its usage allowance with Codex.
- **Codex** is OpenAI's software agent, available in the ChatGPT desktop app, a command-line tool, an editor extension and the web.
- **Claude Cowork** brings Claude Code's agent architecture to non-coding work, running sessions in the cloud that follow you across devices.
- **Gemini tasks** turn Gemini into an assistant that can act across Gmail, Calendar, Drive and the web, on a schedule or when events happen.
- **Perplexity Computer** extends Perplexity's research strength into building documents and apps and running scheduled tasks.
- **Grok Bot** gives you named Bots, each with memory and a persistent cloud computer (browser, files and terminal), that can work in parallel and message each other.

## 2.2 Choosing an agent for a job

| JOB                                                                  | GOOD FIRST CHOICE                                | WHY                                                      |
|----------------------------------------------------------------------|--------------------------------------------------|----------------------------------------------------------|
| Deep research report with citations                                  | Perplexity Computer, ChatGPT Work, Claude Cowork | Strong search and synthesis                              |
| Spreadsheet with working formulas                                    | Claude Cowork, ChatGPT Work                      | Both document creating Excel and slides                  |
| Gmail and Google Calendar chores                                     | Gemini tasks                                     | Native Google Workspace actions                          |
| Organizing files on your PC                                          | Claude Cowork (desktop), ChatGPT Work (desktop)  | Local file access with permission                        |
| Coding a small app                                                   | Codex, Claude Code, Grok Build                   | Purpose-built coding agents                              |
| An ongoing role (inbox assistant, researcher) that remembers context | Grok Bot                                         | Persistent named Bots with memory and their own computer |
| A second opinion                                                     | Any other agent                                  | Different models catch different mistakes                |

**i NOTE**

You do not need every agent. Most people get excellent results from the one included in the subscription they already pay for, plus a free second opinion from another chatbot.

## CHAPTER 3

# Writing a Task Brief an Agent Can Finish

---

- A good brief states the **outcome**, the **context**, **where to work**, the **finish line** and **what needs your approval**.
- Agents fail most often because the goal is vague or the finish line is missing.
- Ask for a plan first on anything important.

## 3.1 The five-part brief

### PROMPT

#### OUTCOME

Create a one-page comparison of three family SUVs we're considering: [Model A], [Model B], [Model C], all 2026 model year.

#### CONTEXT

Family of five, two car seats, about 15,000 miles a year, mostly highway.  
Budget under \$45,000. Safety and cargo space matter most.

#### WHERE TO WORK

Use manufacturers' official sites and the IIHS and NHTSA websites for safety.  
Save the result as a Word document in my Downloads folder.

#### FINISH LINE

A table (price range, MPG, cargo volume, third-row legroom, safety ratings),  
a 5-bullet summary, and a list of every source URL. One page.

#### APPROVAL

Don't fill in any forms or contact dealers. Ask me if a site wants  
my email or phone number.

## 3.2 Why each part matters

| PART          | WHAT GOES WRONG WITHOUT IT                                         |
|---------------|--------------------------------------------------------------------|
| Outcome       | The agent does "something related" but not what you needed         |
| Context       | It makes assumptions (budget, location, preferences) you would not |
| Where to work | It uses weak sources or saves files somewhere you cannot find      |
| Finish line   | It stops too early or keeps going and burns through your usage     |
| Approval      | It takes actions you did not want, such as submitting a form       |

### 3.3 Ask for a plan first

For anything that matters, start with:

#### PROMPT

Before doing anything, show me your plan as numbered steps, the tools you'll use, the sites you'll visit, and anything you'll need from me. Wait for my OK.

Read the plan. It is far cheaper to fix a plan than to undo work.

### 3.4 Brief templates for common jobs

#### Research brief

#### PROMPT

Research [topic] for [audience/purpose]. Use at least [N] reputable sources, prefer [official/government/peer-reviewed] sources, and note publication dates. Deliver: a 1-page summary, a table of key facts with sources, and a list of open questions. Flag anything you could not verify.

#### File chore brief

#### PROMPT

In the folder [path], sort files into subfolders by type (Documents, Images, Spreadsheets, Other) and rename them as YYYY-MM-DD\_original-name. Do not delete anything. Before moving files, show me the plan for the first 20 files. Afterwards, give me a log of every change.

#### Inbox triage brief

#### PROMPT

Look at my unread emails from the last 3 days. Group them: needs reply, FYI, newsletters, possible spam. For "needs reply," draft short replies in my voice and leave them as drafts. Don't send, delete or unsubscribe from anything. Give me a summary table.

## Monitoring brief

### PROMPT

Every morning at 8 AM, check [site/product page] for the price of [item].  
If it drops below \$[amount], notify me with the link. Do not buy anything.  
Stop monitoring after [date].

## Good habits while an agent works

- **Watch the first few steps** of any new kind of task.
  - **Answer questions promptly**; many agents pause until you respond.
  - **Steer early**. If the plan drifts, say so; most agents let you redirect mid-task.
  - **Review the final output against the finish line**, not just whether it looks polished.
- ☐ My briefs state outcome, context, where to work, finish line and approvals.
  - ☐ I ask for a plan first on important tasks.
  - ☐ I keep brief templates for jobs I repeat.

# ChatGPT Work and Codex

---

- ChatGPT now has two modes: **Chat** for quick help and **Work**, an agent for longer, multi-step work and finished deliverables.
- **Work** runs in the cloud on web and mobile; in the desktop app it can also use local files and desktop apps with your permission.
- **Codex** is OpenAI's software agent. It is included on every ChatGPT plan, including Free and Go; **Codex Cloud** needs Plus or higher.
- Work and Codex share one usage allowance on plans that use shared allowances.

## 4.1 Using Work

From OpenAI's *ChatGPT Work and Codex* article (checked 2026-10-16):

- 1 Open ChatGPT on the web or mobile and select **Work** (on desktop, choose ChatGPT from the top-left menu, then the **Work** toggle).
- 2 Start a new chat or open an existing **Project** (choosing Work inside a project uses the project's context).
- 3 Add files and context, then describe the outcome you want.
- 4 Review progress, answer questions, change direction, and approve important actions.
- 5 Review the result and ask for changes in the same chat.

What Work can produce, according to OpenAI: research, analysis, documents, spreadsheets, presentations, reports and Sites. It can use connected apps and a browser.

## Scheduled and event-triggered tasks

- Work can run **once, on a schedule, or on a trigger**, and can monitor for changes through Scheduled Tasks.
- **Event-triggered tasks** start when something happens in a connected app. OpenAI lists new Gmail messages, new Slack channel messages and GitHub pull request activity. They are available to eligible Plus, Pro, Business, Enterprise, Edu and ChatGPT for Healthcare users, **not Free or Go**.
- For a Slack-triggered task, add **@ChatGPT** to each monitored channel.

### PROMPT

When a new email arrives from any address at @myschool.edu, summarize it in three bullets and add any dates to a running list in my "School 2026-27" project. Don't reply to anything.

## Local files in the desktop app

In the ChatGPT desktop app, Work can open a local folder and use files and desktop apps **with your permission**. OpenAI notes that messages and task context may be stored in the cloud even when work runs locally. On web and mobile, Work cannot reach files on your computer.

## Cloud browser

OpenAI's help center describes a **cloud browser** for supported browser workflows on paid plans in supported regions (not Free or Go). When a site needs a login, ChatGPT can pause and let you take over the browser so you type passwords yourself.

## 4.2 Codex for software

From *Using Codex with your ChatGPT plan* (checked 2026-10-16):

| FACT                 | DETAIL                                                                                                                        |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Availability         | Included across ChatGPT plans, including Free and Go; usage limits vary by plan                                               |
| Codex Cloud          | Eligible Plus, all Pro tiers, Business, Enterprise, Healthcare and Education accounts; not Free or Go                         |
| Clients              | ChatGPT desktop app (Codex mode), Codex CLI, Codex IDE extension, Codex web                                                   |
| Start in a terminal  | Open your project folder, run <code>codex</code> , sign in with ChatGPT                                                       |
| Project instructions | <code>/init</code> generates an <code>AGENTS.md</code> scaffold                                                               |
| Check usage          | <code>/status</code> in the CLI, or Settings and the usage dashboard                                                          |
| Usage windows        | 5-hour and weekly windows (as described in OpenAI's reset FAQ)                                                                |
| Safety               | <b>Auto-review</b> (Settings > Permissions > Auto-review) checks eligible actions that would otherwise wait for your approval |

 TIP

If you sign in to Codex with your ChatGPT account, usage comes from your plan. If you use an API key instead, you pay API prices (see Volume 15).

## A first Codex task

### TERMINAL

```
cd my-project
codex
```

### PROMPT

Read this project and explain what it does in 5 bullets. Then run the tests and tell me which fail and why. Don't change any files yet.

Then, once you trust its read of the project:

### PROMPT

Fix the failing test in `tests/test_dates.py`. Make the smallest change that works, run the tests again, and show me the diff.

**AGENTS.md : instructions for coding agents**

An `AGENTS.md` file at the root of a project tells coding agents how to work there. Codex reads it, and so do Grok Build and many other tools. A starter:

**AGENTS.MD**

```
# Project rules
- Language: Python 3.13+. Use type hints.
- Run tests with: pytest -q
- Never commit secrets. Settings live in .env (not committed).
- Keep changes small; explain each change in the summary.
- Ask before adding a new dependency.
```

- ☐ I can switch between Chat and Work.
- ☐ I know event-triggered tasks are not on Free or Go.
- ☐ I ran `codex` in a project folder and asked it to explain the code first.
- ☐ I created an `AGENTS.md` for a project.

# Claude Cowork

- **Cowork** brings Claude Code’s agent architecture to everyday knowledge work, with no terminal. It is available on **paid plans** only.
- Sessions run **in the cloud** and follow you across desktop, web and mobile; the desktop app is needed for local files, your browser and computer use.
- Three approval modes: **Manual** (asks before actions), **Auto** (Claude safety-checks each action) and **Skip** (no checks; for fully trusted tasks only).

## 5.1 Availability (checked 2026-10-16)

From Anthropic’s *Get started with Claude Cowork* article:

| SURFACE                            | AVAILABILITY                                               |
|------------------------------------|------------------------------------------------------------|
| Claude Desktop (macOS and Windows) | All paid plans                                             |
| Web (claude.ai)                    | Pro, Max and Team; Enterprise where an admin enables it    |
| Claude mobile apps                 | Pro, Max and Team; Enterprise where enabled                |
| Claude in Chrome side panel        | Max and Team, rolling out to Pro; Enterprise where enabled |

### NOTE

Anthropic is merging Chat and Cowork into “just Claude,” rolling out gradually to Pro and Max. If your message box no longer shows “Chat” and “Cowork” options, you have the new experience, and Claude decides whether your request needs a quick answer or a task.

## 5.2 Start a session

- 1 Open Claude on the web, desktop or mobile.

- 2 In the message box, select **Cowork** (skip this step if you have the unified experience).
- 3 Describe the task using the five-part brief.
- 4 Review Claude's approach, then let it run.

Sessions keep running when the desktop app is closed or your computer sleeps. If the task uses local files, your browser or your computer, keep the desktop app open.

## 5.3 What Cowork can do

Anthropic lists, among others:

- **Direct local file access** on desktop: read and write the folders you connect.
- **Sub-agent coordination:** breaking work into parallel workstreams.
- **Professional outputs:** Excel spreadsheets with working formulas, PowerPoint decks and formatted documents.
- **Browser actions:** opening sites, reading pages, clicking, typing and filling forms, using a browser built into Claude Desktop or Claude in Chrome.
- **Scheduled tasks:** type `/schedule` in a task, or open **Scheduled** in the sidebar. Scheduled tasks run in the cloud with no device online.
- **Projects:** workspaces with their own files, instructions and memory.
- **Plugins:** bundles of skills, connectors and sub-agents.

## 5.4 The three approval modes

| MODE                                                  | CONNECTOR ACTIONS SET TO "NEEDS APPROVAL"                                                                     | BEST FOR                                   |
|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <b>Manual</b> (default; formerly "Ask before acting") | Claude asks every time                                                                                        | New tasks, anything important              |
| <b>Auto</b>                                           | Read-only tools approved; for write/delete tools Claude reviews each action for safety and blocks unsafe ones | Routine tasks you have watched work before |
| <b>Skip</b> (formerly "Act without asking")           | Approved with no checks                                                                                       | Only fully trusted tasks, tools and files  |

Anthropic notes that **Auto mode uses more of your usage limit** because of the extra checking, and that "no mode replaces your judgment." Claude always asks for explicit permission before **permanently deleting files**.

## 5.5 Global and folder instructions

- **Global instructions:** Settings > Cowork > Global instructions (in the unified experience, part of Instructions for Claude in Settings > General). Use it for your role, tone and output preferences.
- **Folder instructions:** project-specific context attached to a local folder you select on desktop.

## 5.6 Example Cowork tasks

### PROMPT

In my "Receipts 2026" folder, read every receipt (PDFs and photos), and build an Excel workbook with one row per receipt: date, vendor, amount, category, file name. Add a summary tab with totals by month and category using formulas. Don't move or delete any files.

### PROMPT

Turn the meeting transcripts in "Q4 Planning" into a 10-slide PowerPoint: agenda, decisions, owners, timeline, risks. Use our template.pptx. Then list anything in the transcripts that conflicts.

### Usage tips from Anthropic

- Group related work into one task instead of several.
- Start a new conversation for unrelated work.
- Tell Claude when you only need a quick answer, not a file.
- Check usage in **Settings > Usage**.

- ☐ I know Cowork is on paid plans only.
- ☐ I start new kinds of tasks in Manual mode.
- ☐ I set global instructions for my role and preferences.
- ☐ I tried `/schedule` for a recurring task.

# Gemini Tasks

---

- In the Gemini app, a **Chat / Task** toggle next to the message box starts an agent task. Google renamed the feature from “Spark” to “Task.”
- Requirements: 18+, a personal Google account, a **Google AI Pro or Ultra** subscription, Keep Activity on, and Personal Intelligence opted in.
- Tasks can use Gmail, Calendar, Drive, Docs, Sheets, Slides, Keep, Tasks, Search services, YouTube, skills and a local or remote browser.

## 6.1 Requirements and availability (checked 2026-10-16)

From Google’s *Manage your tasks & workflows in Gemini Apps* article:

- Be **18 or over** and signed in with a **personal** Google account (not work or school).
- Have a **Google AI Pro or Ultra** subscription.
- Have **Keep Activity** on and opt in to **Personal Intelligence**.
- Available wherever Gemini Apps are supported **except the European Economic Area, Nigeria, Switzerland and the United Kingdom**.
- Only in the Gemini mobile app, the Gemini app on Mac and the web app at [gemini.google.com](https://gemini.google.com).
- Up to **15 tasks** can run at once.

## 6.2 Create a task

- 1 Go to [gemini.google.com](https://gemini.google.com) (or open the Gemini app on Mac).
- 2 Below the text box, click **Task**.
- 3 Describe the task, including when to run it (a time, or after an event).
- 4 To use a skill, type @ and pick it.

- 5 Add files with **Upload & tools** (upload, add from Drive, or add a notebook).
- 6 Click **Submit**, and check the task thread for questions.

#### TIP

New to tasks? Google suggests typing “set up,” “get started” or “interview me” in a task, and Gemini will help you create your first skills and tasks conversationally.

## 6.3 Tasks, schedules and skills

Google describes three building blocks:

| BLOCK           | MEANING                           | EXAMPLE                                              |
|-----------------|-----------------------------------|------------------------------------------------------|
| <b>Task</b>     | What you want done                | “Plan and manage my business trip to London.”        |
| <b>Schedule</b> | When it runs (a time or an event) | “Every day at 8 AM, give me an update on AI news.”   |
| <b>Skill</b>    | How to do it                      | A “Travel Booking” skill and a “Gmail Writing” skill |

## 6.4 The browser: local or remote

- **Local Chrome** (desktop): Gemini can use the same sites you are signed into. Keep the device and Chrome awake.
- **Remote browser** (cloud): keeps working if you close your device. For sign-ins, the task pauses until you **take over**.
- To take over a remote browser: in the task thread, click **View live**, then **Take over task**; click **Go back to Gemini** when done.

## 6.5 Safety features Google lists

- Gemini asks for confirmation before **sending communications, modifying data, making purchases and submitting web forms**.
- It may ask you to **take control** to enter passwords or payment details.
- It shows its plan and progress.
- It restricts browsing to sites relevant to the task.

#### WARNING

Google’s article warns that Gemini can perform **bulk actions on private items in Google Tasks without confirmation**. It also advises not to type sign-in details, payment details or sensitive information into a task thread, and to avoid scheduling sensitive tasks that might run while you are offline.

## 6.6 Clean up after sensitive tasks

At [gemini.google.com](https://gemini.google.com), open **Settings > Task settings** to delete remote browser data (including saved sign-in cookies) and remote code execution data, or to turn task mode off.

- ☐ I meet the requirements (age, personal account, Pro or Ultra, region).
- ☐ I created a task with the Task toggle.
- ☐ I know how to take over the remote browser.
- ☐ I know where Task settings are for deleting browser data.

# Grok Bot

---

- **Grok Bot** gives you named, persistent Bots: AI teammates with jobs and context that build up over time.
- Each Bot works on a **persistent cloud computer** with a browser, filesystem and terminal. All your Bots share one computer, so files and logins carry over between them.
- Bots can run **in parallel**, message each other, share group chats and pass tasks along. Drafts that would be sent in your name come back to you for approval.

## 7.1 What makes Grok Bot different

From xAI's Grok Bot documentation (checked 2026-10-16):

- **Each Bot has a computer.** Bots use connectors where available and computer use for everything else, and work continues while your laptop is closed.
- **Setup is a message.** Create a Bot, describe the job, and grant access as it asks.
- **Bots coordinate.** They can run in parallel, message each other, share context in group chats and hand off ownership, “so you aren’t the router between tools.”
- **Bots learn from demonstration.** Walk a Bot through a process once and it can save it as a skill and rerun it on a schedule (a routine).
- **Context compounds.** A named Bot keeps its memory, files, browser sessions and preferences across sessions.

## 7.2 Availability

Grok Bot runs on macOS, Windows, Linux, iOS and Android. xAI's docs say access is included with **every paid individual Cursor plan and the Cursor Teams plan**, or by linking an individual **SuperGrok, SuperGrok Plus or SuperGrok Heavy** subscription. Usage resets weekly. You can also start work by tagging **@bot** on X, where available.

## 7.3 The shared computer

All your Bots share the same cloud computer, including files, browser sessions and app logins. xAI notes that the computer belongs to your account, so treat anything placed on it as available to every Bot you run; isolation between different users is strict. Each Bot gets its own screen and runs one computer-use task at a time.

## 7.4 A good first handoff

xAI's suggested pattern is a real task that spans several tools and has a clear finish line, for example:

### PROMPT

Pull this week's pipeline review list from our CRM. Skip anyone already in an active sequence. Research the top five accounts across the web and Slack, pull the right contacts, draft outreach in my voice, and leave me drafts to approve by tomorrow morning.

Review the result, correct it, then turn the stable process into a **skill** or **routine**.

## 7.5 A multi-Bot setup

Because Bots coordinate, you can create a small team:

| BOT            | JOB                                                      |
|----------------|----------------------------------------------------------|
| <b>Scout</b>   | Monitors news and sources on your topics each morning    |
| <b>Analyst</b> | Turns Scout's finds into a short brief with sources      |
| <b>Editor</b>  | Checks the brief for errors and unsupported claims       |
| <b>Inbox</b>   | Drafts replies and meeting suggestions for your approval |

 **NOTE**

xAI's FAQ says sites can still block automation, expire sessions or require a human step, and that Bots hand those steps to you rather than working around them. For important decisions, ask a Bot to check the current source instead of relying on memory.

# Perplexity Computer

- Perplexity describes **Computer** as an autonomous AI worker that can research, create documents and apps, handle email and run scheduled tasks.
- It is available to active Perplexity subscribers on the web (desktop), iOS and Android, according to Perplexity's help center.
- Perplexity Projects can set a default mode of Search or Computer, and an orchestrator model for Computer.

Perplexity built its reputation on cited answers, and Computer extends that into multi-step work. Its help-center Projects article (checked 2026-10-16) says each Project can set a **default mode (Search or Computer)** and a default Computer orchestrator model, and that Computer can create, update and manage Project files for you.

## NOTE

**Partly verified (2026-10-16):** Perplexity's *What is Computer?* help article could not be loaded directly when this book was checked (the site blocked automated access). The description above comes from the article's search summary and from the Projects article. Confirm plan details and limits in the app.

Good Computer tasks play to Perplexity's strengths:

## PROMPT

Build a cited briefing on the 2026 changes to [regulation], for a small business owner. Include: what changed, effective dates, who is affected, a compliance checklist, and links to the official government sources. Save it as a document in this Project. Flag anything that is proposed rather than final.

# Multi-Agent Patterns

- **Multi-agent** means more than one AI working on the same goal, either different products or several copies of one.
- Five patterns cover most needs: **fan-out/fan-in**, **pipeline**, **writer and critic**, **specialist team** and **debate**.
- Multi-agent work costs more time and usage. Use it when accuracy or coverage matters.

## 9.1 Why use more than one AI?

- **Different models know different things** and make different mistakes. Where they agree, you can be more confident; where they disagree, you know where to check.
- **Specialists beat generalists.** Perplexity for sources, Claude for long documents, Gemini for Google Workspace, Codex for code.
- **Parallel work is faster.** Five agents researching five subtopics finish sooner than one doing all five.
- **A critic catches errors** the writer cannot see.

## 9.2 The five patterns

| PATTERN                  | HOW IT WORKS                                                | USE IT FOR                                   |
|--------------------------|-------------------------------------------------------------|----------------------------------------------|
| <b>Fan-out / fan-in</b>  | Send the same task to several AIs, then combine the answers | Research, decisions, fact-checking           |
| <b>Pipeline</b>          | Each AI does one stage and passes the result on             | Research, then draft, then edit, then format |
| <b>Writer and critic</b> | One AI drafts; another reviews against a checklist; repeat  | Important documents, code                    |
| <b>Specialist team</b>   | Split the task by subtopic; each AI owns one part           | Big reports, trip planning, market scans     |
| <b>Debate</b>            | Two AIs argue opposite sides; a third judges                | Decisions with real trade-offs               |

## 9.3 Fan-out / fan-in in detail

- 1 **Write one prompt** that asks for a structured answer (the same headings from every AI).
- 2 **Fan out:**  
send it to each AI.
- 3 **Collect** the answers into one document.
- 4 **Fan in:**  
ask one AI to compare and synthesize, listing agreements, disagreements and unique points.
- 5 **Verify** the disagreements yourself, using primary sources.

## 9.4 Writer and critic in detail

### PROMPT

You are a strict reviewer. Check the draft below against this checklist:  
1) Every factual claim has a source. 2) Numbers add up. 3) It answers the original question. 4) Nothing is overstated. 5) Under 600 words.  
List each problem with the exact sentence and a suggested fix. Do not rewrite the whole draft.

Send the critic's list back to the writer, ask for a revision, and repeat once or twice. Using **different products** for writer and critic usually catches more problems than using the same one.

### When not to go multi-agent

- Simple questions with clear answers.
- Tasks where speed matters more than polish.
- When you are near usage limits (each agent counts).
- When the task involves private data you do not want spread across several providers.

# Fan-Out by Hand: Five AIs, One Question

- You can run a powerful multi-agent workflow with nothing but browser tabs: ChatGPT, Claude, Gemini, Grok and Perplexity.
- Use a **structured prompt** so the answers line up, then a **synthesis prompt** to combine them.
- This works on free plans, within each service's own limits.

## 10.1 Step 1: open five tabs

| TAB        | ADDRESS           | STRENGTH TO LEAN ON                |
|------------|-------------------|------------------------------------|
| ChatGPT    | chatgpt.com       | Broad knowledge, clear structure   |
| Claude     | claude.ai         | Careful reasoning, long documents  |
| Gemini     | gemini.google.com | Google Search grounding, Workspace |
| Grok       | grok.com          | Real-time X and web context        |
| Perplexity | perplexity.ai     | Citations for every claim          |

## 10.2 Step 2: the structured fan-out prompt

### PROMPT

Question: Should a family of four in central Indiana replace a 15-year-old gas furnace with a heat pump in 2026?

Answer using exactly these headings:

1. Short answer (2 sentences)
2. Key factors (bullets)
3. Costs and incentives (with sources and dates)
4. Risks or downsides
5. What I should verify locally
6. Confidence (low/medium/high) and why

Be concise (under 400 words). Cite sources where possible. Say clearly if information may be out of date.

Paste the same prompt into all five tabs.

## 10.3 Step 3: collect the answers

Copy each answer into one document under a heading with the AI's name. Keep the source links.

## 10.4 Step 4: the synthesis prompt

Paste all five answers into one AI (a different one from your favorite, if you like) with:

**PROMPT**

Below are five answers to the same question from different AI assistants.

Produce:

- A) Points all or most agree on.
  - B) Points where they disagree, with each side's reasoning.
  - C) Unique useful points only one mentioned.
  - D) Claims that need verification, especially numbers, prices and rules.
  - E) A combined best answer under 300 words, marking anything uncertain.
- Do not add new facts that none of them stated.

[Answer from ChatGPT] ...

[Answer from Claude] ...

[Answer from Gemini] ...

[Answer from Grok] ...

[Answer from Perplexity] ...

## 10.5 Step 5: verify the disagreements

The synthesis gives you a short list of contested claims. Check those against primary sources: government sites, manufacturer pages, official documentation. This is where the real value lies: five AIs narrow the search, and you confirm the facts that matter.

### A comparison scorecard

|                     | CHATGPT | CLAUDE | GEMINI | GROK | PERPLEXITY |
|---------------------|---------|--------|--------|------|------------|
| Followed the format |         |        |        |      |            |
| Cited sources       |         |        |        |      |            |
| Dates current       |         |        |        |      |            |
| Unique insight      |         |        |        |      |            |
| Errors found        |         |        |        |      |            |

After a few rounds you will learn which assistant to trust for which kinds of questions.

**TIP**

Free plans have message limits that reset on each provider's own schedule. Fanning out to five free plans spreads your usage, so it can actually let you do more total work than one free plan. Volume 20 covers limits and rotation in depth.

# Automated Fan-Out with Python

---

- With API keys, a short Python script can send one prompt to OpenAI, Anthropic, Google and xAI **in parallel** and then synthesize.
- API calls cost money per token, separately from any chat subscription. Set spending caps first (Volume 15).
- The script uses the model names and SDK patterns verified on October 16, 2026.

## 11.1 Setup

Follow Volume 14 to install Python and create a virtual environment, then:

### TERMINAL

```
pip install openai anthropic google-genai xai-sdk python-dotenv
```

Create a `.env` file (never share or commit it):

### .ENV

```
OPENAI_API_KEY=sk-...  
ANTHROPIC_API_KEY=sk-ant-...  
GEMINI_API_KEY=...  
XAI_API_KEY=xai-...
```

## 11.2 The fan-out script

FANOUT.PY

```
import concurrent.futures as cf
from dotenv import load_dotenv

load_dotenv()

SYSTEM = "You are a careful analyst. Be concise and cite sources when you can."

def ask_openai(prompt):
    from openai import OpenAI
    r = OpenAI().responses.create(model="gpt-6.1-sol", instructions=SYSTEM, input=prompt)
    return r.output_text
```

Keep going in the same file with the Claude, Gemini and Grok helpers:

FANOUT.PY (CONTINUED)

```
def ask_claude(prompt):
    import anthropic
    r = anthropic.Anthropic().messages.create(
        model="claude-sonnet-5-5", max_tokens=1500, system=SYSTEM,
        messages=[{"role": "user", "content": prompt}])
    return r.content[0].text

def ask_gemini(prompt):
    from google import genai
    from google.genai import types
    r = genai.Client().models.generate_content(
        model="gemini-3.8-flash", contents=prompt,
        config=types.GenerateContentConfig(system_instruction=SYSTEM))
    return r.text

def ask_grok(prompt):
    from xai_sdk import Client
    from xai_sdk.chat import system, user
    chat = Client().chat.create(model="grok-4.3", messages=[system(SYSTEM)])
    chat.append(user(prompt))
    return chat.sample().content
```

The second half of the same file runs the four functions in parallel threads, so the total wait is roughly the slowest provider rather than the sum of all four, and then asks Claude to synthesize:

## FANOUT.PY (CONTINUED)

```

AGENTS = {"OpenAI": ask_openai, "Claude": ask_claude,
          "Gemini": ask_gemini, "Grok": ask_grok}

def fan_out(prompt):
    results = {}
    with cf.ThreadPoolExecutor(max_workers=len(AGENTS)) as pool:
        futures = {pool.submit(fn, prompt): name for name, fn in AGENTS.items()}
        for fut in cf.as_completed(futures):
            name = futures[fut]
            try:
                results[name] = fut.result()
            except Exception as e:
                # one failure shouldn't stop the rest
                results[name] = f"[error: {e}]"
    return results

def fan_in(question, results):
    joined = "\n\n".join(f"[{name}]\n{text}" for name, text in results.items())
    prompt = (f"Question: {question}\n\nAnswers from several AIs:\n\n{joined}\n\n"
              "List agreements, disagreements, unique points and claims to verify, "
              "then give a combined answer under 300 words. Add no new facts.")
    return ask_claude(prompt)

if __name__ == "__main__":
    q = input("Question: ")
    answers = fan_out(q)
    for name, text in answers.items():
        print(f"\n===== {name} =====\n{text}")
    print("\n===== SYNTHESIS =====\n" + fan_in(q, answers))

```

## 11.3 What it costs

At the prices listed on October 16, 2026 (per million tokens, input/output): `gpt-6.1-sol` \$2/\$10, `claude-sonnet-5-5` \$2/\$10, `gemini-3.8-flash` \$0.75/\$3.75 (promotional through December 31, 2026), and `grok-4.3` \$1.25/\$2.50. A question of about 200 tokens with roughly 600-token answers costs well under one cent per provider, and the synthesis step adds a little more. Long documents cost proportionally more. See Volume 15 for the arithmetic and for setting hard spending limits.

### ⚠ WARNING

Never paste API keys into prompts, chats or shared documents. Keep them in `.env`, add `.env` to `.gitignore`, and set a monthly spending cap with each provider before running scripts in loops.

## 11.4 About Perplexity's API

Perplexity's developer platform changed in 2026. Its quickstart (checked 2026-10-16) shows an OpenAI-SDK-compatible **Orchestrator API** at `https://api.perplexity.ai/router/v1`, a separate **Agent API** and a **Search API**, using a `PERPLEXITY_API_KEY`. Its docs also describe migrating from the older Sonar API. Because model names there were in flux, it is left out of the script above; add it by following Perplexity's current quickstart.

# Coding Agents Side by Side

- Codex, Claude Code and Grok Build can all work in the same project, guided by one `AGENTS.md` file.
- Use **Git branches or worktrees** so agents working in parallel do not overwrite each other.
- Use one agent to write and another to review.

## 12.1 One rules file for every coding agent

Codex generates `AGENTS.md` with `/init`. xAI's Grok Build documentation says it reads `AGENTS.md`, plus `CLAUDE.md` and `.claude/rules/` for compatibility, and that `grok inspect` lists the rules files it found. Keeping your rules in `AGENTS.md` means each agent follows the same conventions.

## 12.2 A parallel coding workflow

- 1 Commit your work so you have a clean starting point.
- 2 Create a branch per agent, for example `git switch -c try-codex` and `git switch -c try-claude` in separate worktrees (`git worktree add ../try-claude try-claude`).
- 3 Give each agent the same task in its own folder.
- 4 Compare the results: run the tests, read the diffs.
- 5 Ask a third agent to review the best diff against your checklist.
- 6 Merge the winner; delete the other branches.

The OpenAI help center notes the refreshed Codex CLI has built-in **worktree** support for keeping separate tasks in separate Git worktrees.

## 12.3 Writer and reviewer for code

### PROMPT

Review this diff as a senior engineer. Check: correctness, tests cover the change, no secrets, error handling, and readability. For each issue give file, line and a concrete fix. Say "approve" only if there are no blocking issues.

### ⚠ DANGER

Coding agents run commands on your computer or in the cloud. Work in a project under version control, keep approval prompts on for commands you do not recognize, and never give an agent production passwords. Volume 8 (Maximum Autonomy) covers safe high-autonomy setups in depth.

# Safety, Permissions and Approvals

---

- The biggest agent risks are **prompt injection** (hidden instructions in web pages, emails or files) and **unintended actions** (sending, deleting, buying).
- Give agents the **least access** they need, keep **approvals on** for consequential actions, and **watch** new kinds of tasks.
- Every provider in this book says the same thing in its help center: safeguards help but do not replace your supervision.

## 13.1 Prompt injection, explained

Google's help center describes prompt injection as content (a website, an email, a document, a `.md` file or media) containing malicious instructions that may be hidden from you but visible to the AI. Examples it gives include instructions to post your private information on a public website or send your emails to an outside service.

Agents are most at risk when they combine three things:

1. **Access to private data** (your email, files, accounts).
2. **Exposure to untrusted content** (web pages, incoming emails, shared documents).
3. **The ability to send data out** (email, forms, web requests).

If a task needs all three, supervise it closely.

## 13.2 The approval controls you have

| PRODUCT       | CONTROL                                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------------------------------|
| ChatGPT Work  | Approve important actions in the task; connected-app permissions; take over the browser for logins                         |
| Codex         | Approval prompts; Auto-review (Settings > Permissions > Auto-review)                                                       |
| Claude Cowork | Manual, Auto or Skip mode; explicit permission before permanent deletions; connector permissions in Customize > Connectors |
| Gemini tasks  | Confirmation before sending, modifying data, purchases and forms; take control for passwords and payments                  |
| Grok Bot      | Drafts you approve before they are sent; Bots hand off steps that need a human                                             |

## 13.3 The agent safety checklist

- ☐ The agent has only the connectors and folders this task needs.
- ☐ Approvals are on for sending, deleting, buying and submitting forms.
- ☐ I type passwords and payment details myself, never into the chat.
- ☐ I watched the first run of any new kind of task.
- ☐ Important files are backed up, or the work is in version control.
- ☐ I review outputs against the finish line before using them.
- ☐ After sensitive browser tasks, I delete saved browser data where the product offers it.

## 13.4 Data and privacy

- Agent sessions, files and screenshots are stored by the provider under its privacy terms. Some products (for example ChatGPT Work in the desktop app) note that task context may be stored in the cloud even when work runs locally.
- Check each product's training settings. OpenAI's Codex article, for example, says Plus and Pro conversations may be used to improve models unless you turn that off in data controls; business plans are not used for training by default.
- For health, legal or financial data, prefer business plans with the right agreements, or keep that data out of agents entirely. See Volume 18 (AI Privacy & Security) and Volume 19 (AI for Healthcare Workers).

# Usage Limits and Costs

- Agent tasks use far more of your plan than a quick chat, because every step consumes tokens.
- OpenAI: Work, Codex and some other agent features share one allowance on plans that use shared allowances. Anthropic: Cowork's Auto mode uses more than Manual. Google: tasks count against Gemini's compute-based limits.
- Group related work, set clear finish lines and check usage pages.

## 14.1 Where to check usage

| PRODUCT       | WHERE                                                                          |
|---------------|--------------------------------------------------------------------------------|
| Codex         | <code>/status</code> in the CLI; Settings and the usage dashboard              |
| ChatGPT Work  | Same allowance as Codex on shared-allowance plans                              |
| Claude Cowork | Settings > Usage                                                               |
| Gemini tasks  | Gemini warns you near your limit; see Volume 20 for Gemini's usage limits page |
| Grok Bot      | Usage resets weekly; see xAI's Plans and billing page                          |

## 14.2 Ways to stretch your allowance

1. **One task, not five.** Group related work.
2. **Clear finish lines** stop agents from over-working.
3. **Ask for a plan first** so you do not pay for wrong work.
4. **Use Chat for quick questions;** save Work and Cowork for real tasks.
5. **Use Manual mode** in Cowork when usage is tight (Auto costs more).
6. **Schedule wisely.** A daily task that could be weekly costs seven times more.
7. **Start fresh** for unrelated work so agents do not carry old context.

# Five Workflows to Try This Week

---

## Workflow 1: The Sunday planner (Gemini tasks or ChatGPT Work)

### PROMPT

Every Sunday at 6 PM, look at my calendar for the coming week and my unread emails flagged important. Give me: the week at a glance, conflicts, things to prepare, and three suggested focus blocks. Don't change my calendar; suggest only.

## Workflow 2: The receipt-to-spreadsheet clerk (Claude Cowork)

Drop receipts into a folder each month and run the receipts brief from the Cowork chapter. Review the sheet, then file it.

## Workflow 3: The five-AI decision memo (by hand)

Pick a real decision. Run the fan-out and synthesis prompts. Verify the disputed points. Write your decision and the reasons in one paragraph.

## Workflow 4: The research pipeline (multi-product)

- 1 **Perplexity:**  
gather sources and a cited fact table.
- 2 **Claude:**  
write a 2-page brief from the fact table only.
- 3 **ChatGPT or Gemini:**  
critic pass with the reviewer checklist.

4

**You:**

check every number against the cited source.

**Workflow 5: The standing research Bot (Grok Bot)**

Create a Bot named for its job, such as “Market Watch.” Tell it the topics, sources and format you want each morning, and ask it to save a skill once the format is right, then schedule it as a routine. Review its brief daily for the first week.

# Troubleshooting

| PROBLEM                             | LIKELY CAUSE                                      | FIX                                                                 |
|-------------------------------------|---------------------------------------------------|---------------------------------------------------------------------|
| "Agent mode" missing in ChatGPT     | Retired                                           | Use Work, or the cloud browser for browser workflows                |
| Event-triggered task option missing | Free or Go plan, or admin setting                 | Upgrade, or ask your workspace admin                                |
| Cowork not offered                  | Free plan, or Enterprise admin has not enabled it | Use a paid plan; ask your admin                                     |
| Cowork stopped using local files    | Desktop app closed or computer asleep             | Keep Claude Desktop open for local work                             |
| Gemini Task toggle missing          | Region, age, account type, or plan                | Check requirements; work and school accounts are not supported yet  |
| Agent stuck at a login page         | Site needs a human step                           | Take over the browser and sign in yourself                          |
| Agent keeps going and burns usage   | No finish line                                    | Add a clear deliverable and stop condition                          |
| Answers from five AIs contradict    | Normal; different sources and dates               | Use the synthesis prompt, then verify with primary sources          |
| Python fan-out: one provider errors | Missing key, wrong model name or rate limit       | Check <code>.env</code> , model name and the provider's status page |
| Coding agents overwrite each other  | Same folder and branch                            | Use separate branches or worktrees                                  |

# Quick Reference

---

- **Brief formula:** outcome, context, where to work, finish line, approvals
- **First line for big tasks:** “Show me your plan and wait for my OK.”
- **ChatGPT:** Chat vs. Work toggle; Work + Projects; event triggers on Plus and above
- **Codex:** `codex` in a project folder; `/init` for AGENTS.md; `/status` for usage
- **Claude Cowork:** paid plans; Manual, Auto, Skip; `/schedule`; Settings > Usage
- **Gemini tasks:** Task toggle; Pro or Ultra; 18+; personal account; up to 15 running
- **Grok Bot:** named Bots, shared cloud computer, parallel coordination, skills and routines
- **Perplexity Computer:** subscribers; Projects can default to Computer mode
- **Fan-out:** same structured prompt to 5 AIs, then synthesis, then verify disagreements
- **Safety trio to watch:** private data + untrusted content + ability to send out

## Sources (checked October 16, 2026)

- OpenAI: [ChatGPT Work and Codex](#); [Using Codex with your ChatGPT plan](#); [ChatGPT agent](#); [Using cloud browser in ChatGPT](#)
- Anthropic: [Get started with Claude Cowork](#)
- Google: [Manage your tasks & workflows in Gemini Apps](#)
- xAI: [Grok Bot overview](#); [Grok Build AGENTS.md](#)
- Perplexity: [What is Computer?](#); [What are Projects?](#); [API quickstart](#)

# Glossary

---

- **Agent:** an AI that plans and takes actions with tools to reach a goal.

---

- **AGENTS.md:** a project file with instructions for coding agents.

---

- **Approval mode:** a setting that controls when an agent must ask before acting.

---

- **Auto-review:** an automatic safety check on agent actions (Codex; similar ideas in Claude's Auto mode).

---

- **Cloud browser / remote browser:** a browser the agent runs on the provider's computers.

---

- **Connector:** a link between an AI and an app such as Gmail or Slack.

---

- **Event trigger:** a condition that starts a task automatically, such as a new email.

---

- **Fan-out / fan-in:** sending one task to several AIs, then combining the results.

---

---

- **Finish line:** the clear description of when a task is done.

---

- **Pipeline:** a sequence where each AI does one stage.

---

- **Prompt injection:** hidden instructions in content that try to hijack an agent.

---

- **Routine:** a saved skill that runs on a schedule (Grok Bot).

---

- **Sub-agent:** a helper agent that handles part of a bigger task.

---

- **Worktree:** a separate working folder for a Git branch, useful for parallel agents.

---