

# L22-skill2-obsidian

## Build Your AI Operating System (Part 2)

Instructor: Vykintas

Lesson Link:

Duration: approximately 1h36m

---

### EXECUTIVE SUMMARY

Vykintas introduces **Claude Cowork**, the desktop app that runs the same Claude brain as Claude.ai but lives on your computer instead of the cloud. He deliberately skips complex setup and focuses on one thing: recognizing when and why to reach for Cowork over Claude Chat. The lesson is built on three pillars, each with a simple trigger: local files and folders, long-running autonomous tasks, and scheduled tasks. Live demos carry most of the teaching. It suits anyone who works with local files or wants to automate recurring or multi-step work without building an elaborate system first.

[#claudecowork](#)

[#threepillars](#)

[#agenticai](#)

[#scheduledtasks](#)

[#timestamped](#)

---

### KEY TAKEAWAYS

🧠 | Same Claude brain, different body. Claude.ai lives in the cloud; Cowork lives on your computer. That one distinction drives when to use each.

📁 | The three triggers tell you when to switch to Cowork: local files, a long multi-step workflow, or a recurring automated task. Everything else stays on Chat.

🏢 | Cowork is a consulting firm, not a solo consultant. It spins up sub-agents to work in parallel, which is what enables long autonomous workflows.

⚙️ | "When speed is much more important than quality" is when you one-shot a big task; otherwise split it with feedback, or build a system of skills for repeated work.

🔒 | Folder selection is the main guardrail, tools install in a sandbox, and Cowork still asks before destructive actions like deleting files even when set to act without asking.

📝 | For you: the local-files and scheduled-task pillars map directly onto the Amira and Pepper workflows and your Purium recruiting operations.

---

### MINDSET SHIFTS

These are the conceptual spine of the lesson. Each reframes how to think about working with AI on your computer.

**Shift: From Claude in the cloud to Claude operating your computer**

- Old thinking: Claude is walled off in the browser; you bring files to it and apply what it says by hand.
- New thinking: Cowork runs inside your machine, reading, moving, renaming, creating, and editing files directly.
- Why it matters: It removes the constant friction of shuttling files between your computer and the browser, the biggest pain point for local-file work.

### Shift: Simplicity over complex setup

- Old thinking: Getting value from Cowork requires the elaborate folder-and-config setups YouTube experts demonstrate.
- New thinking: Start simple, get value, and invest in setup only once a use case clearly earns it.
- Why it matters: Time on complex setup is time not getting value, and too many tools and edge cases breed confusion.

### Shift: From one consultant to a consulting firm

- Old thinking: Claude works linearly, one step at a time, and stops for your input.
- New thinking: Cowork acts as a supervisor that spins up specialized sub-agents working in parallel.
- Why it matters: This is what lets it finish long autonomous workflows Claude Chat cannot, producing many deliverables from one prompt.

### Shift: Agentic AI is powerful, not a panacea

- Old thinking: Give the AI everything at once and it delivers exactly what you want.
- New thinking: Split long workflows into reviewed tasks for quality; one-shot only when speed beats quality, or build a system for repeated tasks.
- Why it matters: Vykontas is candid that he hyped Cowork in the demo; knowing the limits points you to the right approach.

#### For you

The "simplicity over complex setup" shift is a useful counterweight to the pull toward elaborate architecture in your ecosystem. Let the default carry you until a workflow repeats and scales, which is also how the Regenerative Build and your skill-first approach already lean.

#mindset

#threepillars

## Pillar 1: Claude works inside your computer (Local)

The first superpower. The keyword is local. Claude.ai cannot touch your computer; it only works with files in the cloud, so you constantly drop files in by hand. Cowork runs directly on your machine: it reads folders, moves and renames files, creates files, installs tools when needed, and edits, all visible live on your desktop [00:10].

#### Vykontas

Claude AI lives in the cloud. Claude Cowork lives in your computer.

The trigger for this pillar: "I have files and folders on my computer I want Claude to work on" [00:18]. When a task hits that, you reach for Cowork.

Three demos make it concrete:

- **File reorganization** [00:12]: a prompt to rename files (lesson number, then a keyword for type, then lesson name and cohort year) and sort them into Chats, Transcripts, and Workbooks folders. Cowork did it perfectly, live on screen.
- **Batch image edit** [00:19]: standardizing speaker photos to one size using Vishen's photo as the reference, speaker centered. Cowork evaluated the images, installed the tools it needed, and ran the resizing. One photo cropped imperfectly, the cue to give feedback or build a skill for a recurring need.
- **WhatsApp .opus conversion** [00:30]: converting an unplayable audio file to a standard format usable in NotebookLM. The demo also showed you can select more than one folder for a task [00:33].

### Security and the folder guardrail

Because Cowork can act on your computer, it insists you select a folder, which scopes what it can touch. Tools it needs install in a sandbox (a virtual environment), not directly on your OS, which keeps risk low. It is not 100 percent safe, but risk is minimized with a trusted service.

### Use case

The file-reorganization demo is the pattern for keeping your own AI Mastery materials, Purium assets, and skill files orderly: a naming convention plus auto-sorting into folders. For anything you repeat, the lesson's own advice applies: turn it into a skill so Cowork does it consistently, which fits how you already work with tracie-skill-creator.

#local

#pillar1

#localfiles

#security

## Mobile Dispatch: trigger Cowork from your phone

Still within the local pillar, Dispatch addresses Cowork's main limitation: it lives on one computer and is not directly on your phone. In the Claude mobile app's side menu, Dispatch sends a task to Cowork running on your desktop [00:42].

The demo: from the phone, point Cowork at a folder of PDFs and ask for the key tools and frameworks, to prep for a client call while away [00:44]. The task appeared in the desktop Dispatch section and ran autonomously. It used computer use (visually navigating the screen), which is slower and demands repeated permissions.

### Do not use Dispatch at your own computer

When Vykintas tried to keep working during the demo, Cowork's computer use took over his screen [00:54]. The lesson: use Dispatch only when away from the machine, and add "don't use computer use unless absolutely necessary" to the prompt so it works in the back end instead.

### Use case

Dispatch fits the times you are away from the M4 Max but want a result waiting: have Cowork pull together context for a Purium prospect call or a coaching session from local files, and return a short brief to your phone.

#dispatch

#pillar1

#mobile

#computeruse

## Pillar 2: Hand it off and walk away (Long-running tasks)

The second superpower. The keyword is long-running or agentic autonomous work. Claude.ai cannot run long workflows autonomously; it stops for your input. Cowork commits to finishing, running agentic execution to completion [00:53].

### Vykintas

See Claude AI as a solo consultant. See Claude Cowork as a consulting firm.

The mechanism is sub-agents. A supervisor Claude spawns specialized sub-agents (research, presentation, financial model) that work in parallel and report back [00:58]. Cowork creates them automatically based on its judgment, and you can also create your own, giving them instructions, tools, and connectors the way you create skills [01:00].

The trigger: "I have a long multi-step workflow for Claude to execute" [01:01].

The demo, triggered at the very start of class and running throughout, was a full business launch package from one prompt [00:02]: market research, ideal client profile, product definition, business plan, an Excel financial model (conservative, base, optimistic, with P&L and cash flow), an HTML landing page, and a PowerPoint pitch deck. It pulled pricing and a photo from an Inputs folder and wrote everything to an Outputs folder, in about half an hour [01:08].

### Editable beats pretty

Cowork's slides are less polished than NotebookLM's, but NotebookLM renders each slide as an image you cannot edit, while Cowork produces a real file you can change element by element [01:05]. For final control over a deliverable, the editable file wins.

### For you

The sub-agent model is exactly why your skill-first approach holds: Cowork's auto-created sub-agents inherit your skills, so the investment in skills pays off without your building agents by hand. Reach for a custom sub-agent only when a skill stops giving you the control you need.

#pillar2

#agenticai

#subagents

#longrunning

## When to one-shot a big task, and when not to

Vykintas is candid that the business-launch demo is not how you would normally work [01:10]. Handing over many deliverables at once skips the input and feedback that produce quality.

### Vykintas

I hyped you up a little bit, but AI still is not ideal.

The honest rule:

- For the best results, split a long workflow into separate tasks: research (review it), financial model (review it), business plan (review it), then the presentation. Feedback between each step.
- One-shot the whole thing only when speed matters more than quality (you accept roughly 90 percent), or when you repeat the task often enough to justify building a supporting system of skills, brain, feedback, and memory.

### Use case

This is the decision rule for your Purium and content builds: a one-off recruiting asset is worth splitting and reviewing stage by stage, but a recurring deliverable (a weekly prospect brief, a repeated onboarding sequence) is worth the system of skills so Cowork can run it reliably end to end.

#agenticai

#quality

#skills

## Pillar 3: Claude on autopilot (Scheduled)

The third superpower. The keyword is scheduled. Claude.ai always needs you to trigger it manually. Cowork can run a task automatically on a schedule [01:17].

The trigger: "I want Claude to do this task automatically without me triggering it every time" [01:18].

The demo was a daily email summary. The better way to build it, which Vyintas demonstrates, is inside a project, because project memory lets your feedback improve future runs without editing the instructions by hand [01:23].

Build sequence, work through it in order:

- ☐ Create a project (Projects, New Project, Start from scratch), named Email Triage; leave instructions and files empty; memory is on by default [01:21].
- ☐ Inside the project, click the plus next to Scheduled, New Task, Set up manually [01:24].
- ☐ Name it (Daily Email Triage), add a description, and paste the instructions (check Gmail for the last 24 hours, return a briefing in priority tiers: Critical, Legal/Financial, Action Required, Awaiting Reply, Follow-Up, FYI, Skipped).
- ☐ The project folder is preselected. Set Act without asking, model Sonnet 4.6, frequency Manual for the first test [01:25].
- ☐ Save, then click Run now to test [01:27].

### Projects on Cowork are not Projects on Claude.ai

Despite the shared name they are separate features with separate lists. They share elements (memory, files, instructions) but are not connected. You can import a Claude.ai project into Cowork (New Project, Import a Project), which recreates it as a local folder [01:22].

### For you

The project-plus-memory pattern is the reusable backbone for any recurring Purium or NLGJ task you want to improve over time: create the project, schedule the task inside it, and your feedback compounds into better runs. The email triage here is the simple version; the advanced one (labeling and drafting replies) is deferred to the implementation lab.

#pillar3 #scheduledtasks #automation #projectmemory

## Limitations worth knowing

- Cowork's main limit: your computer must be on, awake, and connected to the internet [00:36]. It cannot be used easily from your phone or another computer (Dispatch is the partial workaround).
- Claude.ai has a limit of roughly 100 images or PDF pages per chat [00:42], which the local-PDF Dispatch demo ran into; working with the files locally in Cowork sidesteps it.
- It can access cloud files (Dropbox, Google Drive) through connectors, the same as Claude.ai, in addition to local files [00:50].
- For reliable video editing, use Claude Code instead, since that needs code-defined workflows and libraries [01:08].
- Deferred to the implementation lab: the Obsidian co-editing setup and knowledge graph, the advanced email triage (labeling and drafting), and the dashboard-to-sheet demo [00:51].

#limitations #connectors

## Resources mentioned

- The prompt document Vykintas shared at the start (all demo prompts for the session).
- obsidian.md (for the deferred co-editing demo).
- NotebookLM (used as the audio destination and the slide comparison).

---

## TAGS

#aimastery #claudecowork #threepillars #localfiles #agenticai #subagents #scheduledtasks #dispatch  
#automation #projectmemory #security #limitations #mindset #timestamped

---

[Claude Cowork](#) - [Claude.ai](#) - [Claude Code](#) - [Vykintas](#) - [Three Pillars](#) - [Local Files Pillar](#) - [Long Running Tasks](#) - [Scheduled Tasks](#) - [Agentic AI](#) - [Sub-Agents](#) - [Consulting Firm Analogy](#) - [Custom Sub-Agents](#) - [Dispatch](#) - [Computer Use](#) - [Sandbox](#) - [Folder Guardrail](#) - [File Reorganization](#) - [Batch Image Edit](#) - [Opus Audio Conversion](#) - [Business Launch Package](#) - [Excel Financial Model](#) - [HTML Landing Page](#) - [Daily Email Triage](#) - [Project Memory](#) - [Cowork Projects](#) - [Import Project](#) - [NotebookLM](#) - [Editable Slides](#) - [Skills First](#) - [One Shot Versus Split](#) - [100 Image Limit](#) - [Amira](#) - [Pepper](#)