

Build Your AI Operating System (Part 2)

Overview

Taught by Vykintas, this session is the main lecture introducing Claude Cowork, Anthropic's desktop application that runs Claude locally on your computer rather than only in the cloud. Vykintas deliberately avoids complex setup and instead focuses on helping students recognize when and why to reach for Cowork over Claude Chat. The lesson is organized around three "superpowers," each with a simple trigger: working with local files and folders, running long autonomous multi-step tasks, and running scheduled automated tasks. Through a series of live demos (file reorganization, batch image editing, audio format conversion, mobile dispatch, a full autonomous business launch package, and a daily email triage task), students learn to identify the trigger for each pillar and get value from Cowork quickly without elaborate configuration. By the end, a student can recognize which tasks belong in Cowork and start using it productively.

Key Lessons

Mindset Shifts

- **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 manually apply what it tells you.
 - **New thinking:** Claude Cowork runs inside your machine. It reads folders, moves and renames files, creates files, installs tools, and edits directly, without you as the go-between.
 - **Why it matters:** It removes the constant friction of manually moving files between your computer and the browser, which is the biggest pain point for anyone who works with local files.
- **Shift: Simplicity over complex setup**
 - **Old thinking:** Getting value from a powerful tool like Cowork requires elaborate setup, many folders, and configuration shown by YouTube experts.
 - **New thinking:** Start using the tool simply, focus on getting value, and only invest in setup later once you see a use case worth it.
 - **Why it matters:** Time spent on complex setup is time not spent getting value, and it risks confusion from too many tools and edge cases.
- **Shift: From one consultant to a consulting firm**
 - **Old thinking:** Claude works linearly, one task at a time, and stops when it needs your input.
 - **New thinking:** Cowork acts as a supervisor that spins up sub-agents to work in parallel on different parts of a complex task, the way a firm assigns specialists.

- **Why it matters:** This is what lets Cowork finish long autonomous workflows that Claude Chat cannot, delivering many outputs from a single delegated prompt.
- **Shift: Agentic AI is powerful, not a panacea**
 - **Old thinking:** Hand the AI everything in one prompt and it will deliver exactly what you want.
 - **New thinking:** For the highest quality, split a long workflow into separate reviewed tasks with feedback; reserve the one-prompt-everything approach for when speed matters more than quality, or build a system of skills and memory for repeated tasks.
 - **Why it matters:** Vyintas is candid that he hyped Cowork during the demo; being realistic about its limits prevents disappointment and points you to the right approach for your situation.

Frameworks & Principles

- **The Three Pillars of Claude Cowork (when to use it)**
 - **What it is:** A simple decision framework, one trigger per superpower, so you always know when to switch from Claude Chat to Cowork.
 - **Components:**
 - Pillar 1, Local (keyword: local files and folders). Trigger: "I have files and folders on my computer I want Claude to work on."
 - Pillar 2, Long-running tasks (keyword: agentic autonomous work). Trigger: "I have a long multi-step workflow for Claude to execute end to end."
 - Pillar 3, Scheduled (keyword: automated, recurring). Trigger: "I want Claude to run this task automatically without me triggering it every time."
 - **How to apply it:** Before starting any task, ask which trigger it matches. If it hits one, use Cowork; otherwise Claude Chat is fine.
- **The Sub-Agent Architecture**
 - **What it is:** The internal structure Cowork uses for complex long-running tasks, a supervisor (manager) Claude that spawns specialized sub-agents.
 - **Components:**
 - Manager: receives the high-level task and coordinates.
 - Sub-agents: specialized instances each handling one part (research, presentation, financial model), working in parallel.
 - Custom sub-agents (advanced): you can create your own, giving them instructions, tools, and connectors, the way you create skills.
 - **How to apply it:** For a complex deliverable, write one clear prompt listing everything you want; Cowork orchestrates the sub-agents. For repeatable complex work, build skills (and optionally custom sub-agents) so it delivers reliably.

- **When to one-shot a big task versus split it**

- **What it is:** A principle for deciding whether to delegate a large multi-deliverable task in one prompt or break it into reviewed stages. The instructor did not give this a name.
- **How to apply it:** One-shot it 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. Otherwise split into separate tasks with review and feedback between each.

Tools

- **Claude Cowork**

- **What it is:** Anthropic's desktop app running the same Claude "brain" as Claude.ai but with a different "body" that lives on your computer.
- **Used for (in this lesson):** All three pillars: local file operations, the autonomous business launch package, and the daily email triage scheduled task.
- **Notes:** Same app as Claude Chat (a separate tab). Main limitation: your computer must be on, awake, and connected to the internet. Shares skills and connectors with Claude Chat. Cowork projects are separate from Claude Chat projects despite the shared name. Accesses both cloud files (via connectors) and local files.

- **Claude.ai / Claude Chat**

- **What it is:** The cloud-based chat interface, the same Claude brain accessed via browser or the Chat tab.
- **Used for (in this lesson):** The comparison baseline. Vyintas notes you will still use Chat for most daily tasks.
- **Notes:** Cannot touch your computer, cannot run long autonomous workflows, cannot run scheduled tasks. Has a limit of roughly 100 images or PDF pages per chat, which the local-PDF demo ran into.

- **Dispatch (Claude mobile feature)**

- **What it is:** A function in the Claude mobile app (side menu) that sends a task from your phone to Cowork running on your desktop.
- **Used for (in this lesson):** Triggering Cowork from a phone to read a folder of PDFs and return a summary of tools and frameworks for a client call.
- **Notes:** Requires the desktop on and not sleeping. Can return text or files. Do not use it while sitting at your computer (it can take over your screen via computer use and demand repeated permissions); add "don't use computer use unless absolutely necessary" to the prompt.

- **NotebookLM**

- **What it is:** Google's AI research and document tool.

- **Used for (in this lesson):** Comparison. Its slides look nicer but each slide is an image and cannot be edited; Cowork-created slides are less polished but fully editable. Also the destination for the converted .opus audio.
- **Claude Code**
 - **What it is:** Anthropic's tool for more complex coding work.
 - **Used for (in this lesson):** Mentioned as the better choice for reliable video editing, which needs code-defined workflows and libraries.

Demo Walkthroughs

- **Demo 1: File reorganization (Pillar 1, Local)**
 - **Goal:** Rename files by a convention and sort them into subfolders, on your computer.
 - **Step 1: Start the task and select the folder**
 - **Where:** Claude Cowork, New Task.
 - **Do:** Paste the prompt (rename files to start with the lesson number plus a keyword for file type, then lesson name and cohort year). When prompted, select the target folder.
 - **Expected result:** Cowork renames the files, creates Chats, Transcripts, and Workbooks subfolders, and moves files into them, visibly live on your desktop.
- **Demo 2: Batch image edit (Pillar 1, Local)**
 - **Goal:** Resize and crop a folder of speaker photos to match one reference image, speaker centered.
 - **Step 1: Start the task with a reference**
 - **Where:** Claude Cowork, New Task; select the photos folder via "work in a project"; set Act without asking; set model Sonnet 4.6 to save credits.
 - **Do:** Paste the prompt naming the size standard (use Vishen's photo) and any special instruction (your photo is zoomed out, crop and center it).
 - **Expected result:** Cowork evaluates the images, installs the tools it needs in a sandbox, runs the resizing code, and returns standardized photos. Results may be imperfect (Vyintas's own photo cropped poorly); iterate with feedback or build a skill for a repeated task.
- **Demo 3: WhatsApp .opus audio conversion (Pillar 1, Local)**
 - **Goal:** Convert an unplayable WhatsApp .opus file to a standard audio format usable in tools like NotebookLM.
 - **Step 1: Start the task on the audio folder**
 - **Where:** Claude Cowork, New Task; select the folder with the .opus files (the demo also showed you can select more than one folder for a task).
 - **Do:** Paste the audio-fix prompt.

- **Expected result:** Cowork converts the file to a standard, playable format that can be uploaded to NotebookLM.
- **Demo 4: Mobile Dispatch, read PDFs while away (Pillar 1)**
 - **Goal:** From your phone, have desktop Cowork read a folder of PDFs and return the key tools and frameworks.
 - **Step 1: Send a Dispatch message**
 - **Where:** Claude mobile app, side menu, Dispatch.
 - **Do:** Send a message naming the folder path and the task (read all PDFs, list the most important tools and frameworks, keep it short).
 - **Expected result:** The task appears in the Dispatch section of desktop Cowork and runs autonomously, returning a response to your phone. May ask for permissions; may use computer use (visually navigating the screen), which is slower; add "don't use computer use unless necessary." Note that this same demo ran into the Claude.ai limit of roughly 100 images or PDF pages per chat.
- **Demo 5: Autonomous business launch package (Pillar 2, Long-running)**
 - **Goal:** Delegate a full launch package in one prompt and let Cowork work autonomously while you do other things.
 - **Step 1: Prepare inputs and trigger the task**
 - **Where:** A project folder with Inputs (brand doc, offer/packages doc, feedback CSV, headshot) and Outputs subfolders; Claude Cowork, New Task on that folder.
 - **Do:** Paste the prompt listing all deliverables: market research report, ideal client profile, product definition, business plan, Excel financial model, HTML landing page, PowerPoint pitch deck. Specify outputs go to the Outputs folder.
 - **Expected result:** Cowork runs roughly half an hour, spinning up sub-agents, and produces a pitch deck, an Excel financial model (conservative, base, optimistic scenarios, P&L, cash flow), a business plan, an HTML landing page (pulling pricing and a photo from the inputs), and more, in the Outputs folder. May ask permission before deleting a temp file even with Act without asking set, an intentional guardrail. Results are a solid first draft needing review.
- **Demo 6: Daily email triage scheduled task (Pillar 3, Scheduled)**
 - **Goal:** A daily automated task that reviews the inbox and returns a structured priority briefing, set inside a project so feedback improves it.
 - **Step 1: Create a project to house the task**
 - **Where:** Claude Cowork, Projects, New Project, Start from scratch.
 - **Do:** Name it Email Triage; leave instructions and files empty; keep the default folder; memory is on by default.
 - **Expected result:** A project with a local folder, ready to hold the task and its memory.

- **Step 2: Create the scheduled task in the project**
- **Where:** Inside the project, plus button next to Scheduled, New Task, Set up manually.
- **Do:** Name it (Daily Email Triage), add a description, paste the instructions (check Gmail for the last 24 hours, produce a briefing with priority tiers: Critical, Legal/Financial, Action Required, Awaiting Reply, Follow-Up, FYI, Skipped). Project folder is preselected. Set Act without asking, model Sonnet 4.6, frequency Manual to start.
- **Expected result:** The task saves and appears under Scheduled in the project.
- **Step 3: Test it**
- **Where:** The task, Run now.
- **Do:** Trigger it manually.
- **Expected result:** Under History the run appears, and Claude returns a structured inbox briefing grouped by priority tier. No need to reconnect Gmail if the connector already exists, since Chat and Cowork share connectors.

Applied Use Cases

- Reorganizing and renaming any messy local folder by a naming convention (the lesson-files demo).
- Standardizing a batch of images to one size, with a skill built if it is a recurring need.
- Converting awkward file formats (the WhatsApp .opus to a NotebookLM-ready audio file).
- Triggering desktop work from your phone while away, to prep for a meeting (the Dispatch PDF read).
- Generating a full first-draft business launch package quickly when speed matters more than polish.
- Creating editable slides, sheets, and documents you keep final control over, as opposed to image-only slides from other tools.
- A daily automated inbox briefing, extendable later to labeling and drafting responses.

Key Takeaways

- Claude Cowork is the same Claude brain as Claude.ai but lives on your computer; Claude.ai lives in the cloud. That single distinction drives when to use each.
- The Three Pillar triggers tell you when to switch to Cowork: local files, a long multi-step workflow, or a recurring automated task. Everything else stays on Claude Chat.
- Cowork can read, move, rename, edit, and create local files, and it installs the tools it needs in a sandbox, which keeps risk low. Folder selection is the main guardrail.
- Cowork uses sub-agents (a consulting firm, not a solo consultant), which is what enables genuinely long autonomous workflows.

- Agentic one-shot delegation is real but not a panacea; split big tasks with feedback for quality, or build a system of skills and memory for repeated tasks. One-shot only when speed beats quality.
- Scheduled tasks work best created inside a project, since project memory lets your feedback improve future runs without editing the instructions by hand.
- Cowork's main limitation is that your computer must be on, awake, and online; Dispatch extends it to your phone but should not be used while you are at the computer.

Tracie Fujikane ·