

Build Your AI Operating System (Part 2)

Overview

Taught by Vykintas, this session introduces Claude Cowork, the desktop application that runs the same Claude model as Claude.ai but operates inside the student's own computer rather than in the cloud. Vykintas states explicitly that the class will not cover complex setup, on the grounds that elaborate configuration consumes the time that should go into getting value. The session is instead organized around three pillars, each a distinct capability with a single plain trigger for when to use it: working with local files and folders, running long autonomous multi step tasks, and running tasks on a schedule. Teaching is carried almost entirely by live demonstrations, six in total, including one that produced an imperfect result and one in which Cowork took control of the instructor's screen. By the end a student can identify which of the three triggers a given task matches and decide whether it belongs in Cowork or Claude Chat.

Key Lessons

Mindset Shifts

- **Shift: From Claude in the cloud to Claude operating your computer**
 - **Old thinking:** Claude works only in the browser. Files on the computer must be carried to it manually, and Claude can only describe what to do rather than do it.
 - **New thinking:** Cowork runs directly inside the machine. It reads folders, moves and renames files, creates new files, installs the tools it needs, and edits in place.
 - **Why it matters:** It removes the manual step of moving files between the computer and the browser, which is the recurring friction for anyone whose working files are stored locally.
- **Shift: Simplicity over complex setup**
 - **Old thinking:** Getting value from a tool like Cowork requires first building an elaborate structure of folders, files, and configuration, as demonstrated by experts online.
 - **New thinking:** Begin using the tool plainly, get value from it, and invest in setup later once a specific use case has shown the investment is worth it.
 - **Why it matters:** Vykintas describes the alternative as a dangerous zone in which the number of tools, features, and exceptions produces confusion, and time goes into building a system rather than getting results from AI.
- **Shift: From one consultant to a consulting firm**
 - **Old thinking:** Claude works linearly on one task at a time and stops when it needs the user's input.

- **New thinking:** Cowork acts as a supervisor that creates sub agents and assigns each of them a separate part of a complex task, working in parallel.
- **Why it matters:** This is the mechanism that allows a long multi deliverable workflow to run to completion in Cowork when the same request would stop partway in Claude Chat.
- **Shift: Agentic AI is powerful but is not a panacea**
 - **Old thinking:** A single prompt containing every deliverable will return exactly what was wanted.
 - **New thinking:** For the highest quality, a long workflow is split into separate tasks with review and feedback between each. The single large prompt is reserved for cases where speed matters more than quality, or where the task repeats often enough to justify building skills and memory underneath it.
 - **Why it matters:** Vyintas interrupts his own demonstration to say he oversold the capability, and states that without either the split approach or a supporting system the output is a first draft requiring review rather than a finished result.

Frameworks & Principles

- **The Three Pillars of Claude Cowork**
 - **What it is:** The organizing structure of the session. Three capabilities of Cowork, each paired with a single trigger sentence so a student can identify from the task itself whether to use Cowork or Claude Chat.
 - **Components:**
 - Pillar 1, Claude works inside your computer. Keyword: local. Trigger: "I have files and folders on my computer that I want Claude to work on."
 - Pillar 2, Hand it off and walk away. Keyword: long running tasks, or agentic autonomous work. Trigger: "I have a long multi step workflow for Claude to execute."
 - Pillar 3, Claude on autopilot. Keyword: scheduled. Trigger: "I want Claude to do this task automatically without me triggering it every time."
 - **How to apply it:** Before starting a task, check it against the three trigger sentences. If it matches one, use Cowork. If it matches none, Claude Chat is the appropriate tool, which Vyintas states will remain the case for the majority of daily tasks.
- **The solo consultant and the consulting firm**
 - **What it is:** The instructor's named analogy for the architectural difference between Claude.ai and Cowork. Claude.ai is a solo consultant. Cowork is a consulting firm.
 - **Components:**
 - Claude.ai: one worker proceeding in sequence, which stops at the point it requires confirmation or further input.

- Cowork: a supervisor that creates sub agents, assigns each a defined piece of the task (for example one for the presentation, one for research), and collects their work.
- Custom sub agents: the user can create their own sub agents rather than relying on the ones Cowork generates, defining instructions, tools, and connectors in the same way skills are created. Vyintas does not demonstrate this, stating he does not want to overcomplicate the session.
- **How to apply it:** For a complex multi deliverable task, provide one prompt describing everything required and allow Cowork to create and coordinate the sub agents itself.
- **When to delegate a large task in one prompt**
 - **What it is:** The instructor's stated rule for choosing between a single large delegation and a split, reviewed sequence. He did not give this a name.
 - **How to apply it:** Use one large prompt in either of two cases. First, when speed matters more than quality and roughly ninety percent of the desired result is acceptable, for example when testing an idea. Second, when the task recurs often enough to justify building a supporting system of skills, brain, feedback, and memory, which Vyintas states will make the same delegation reliable. In all other cases, split the workflow into separate tasks: research first and review it, then the financial model and review it, then the business plan with feedback, and only then the presentation.
- **Folder selection as the security boundary**
 - **What it is:** The principle behind Cowork requiring a folder to be chosen before a task runs. Because Cowork can take the same actions on the machine that the user can, scoping it to a folder limits what it is able to reach.
 - **Components:**
 - Folder scope: the task can act only within the selected folder. More than one folder can be selected for a single task.
 - Sandbox: tools Cowork installs are placed in a virtual environment running on the computer rather than installed into the operating system itself.
 - Retained permission on destructive actions: Cowork requests permission before permanently deleting files even when the task has been set to act without asking.
 - **How to apply it:** Select the narrowest folder that contains the files the task needs. Vyintas states plainly that this does not make the arrangement completely safe, and that the claim is minimized risk when using a trusted service.

Tools

- **Claude Cowork**
 - **What it is:** Anthropic's desktop application, described by the instructor as the same Claude brain as Claude.ai with a different body. It is reached through a separate tab in the same

application as Claude Chat.

- **Used for (in this lesson):** All six demonstrations, covering local file operations, the long running business launch package, and the scheduled email triage task.
- **Notes:** The main limitation is that the computer must be on, logged in, connected to the internet, and not sleeping. Cowork shares skills and connectors with Claude Chat, so a connector established in one is available in the other. Cowork projects are a separate feature from Claude.ai projects despite the shared name. It can reach cloud files through connectors as well as local files.

- **Claude.ai / Claude Chat**

- **What it is:** The cloud based interface, reached through a browser or through the Chat tab of the same application. The instructor uses the two names interchangeably.
- **Used for (in this lesson):** The comparison baseline throughout, and the demonstration of the file upload limit.
- **Notes:** It cannot access the computer, cannot run long workflows autonomously, and cannot run tasks on a schedule. A single chat is limited to roughly 100 images or PDF pages. Vykintas states that for the majority of simple daily tasks students will continue to use Claude.ai.

- **Dispatch**

- **What it is:** A function in the side menu of the Claude mobile application that sends a task to Cowork running on the user's desktop computer.
- **Used for (in this lesson):** Sending a request from a phone for Cowork to read a folder of PDFs on the desktop and report back the tools and frameworks they contained.
- **Notes:** Requires the desktop computer to be on and not sleeping. It can return text or any file Cowork creates. During the demonstration Cowork chose to use computer use, visually navigating and clicking through the screen, which took control of the instructor's machine and required him to message Dispatch to stop the task. He advises adding an instruction not to use computer use unless necessary, and not using Dispatch while at the computer it controls.

- **NotebookLM**

- **What it is:** Google's AI research and document tool.
- **Used for (in this lesson):** Twice. First as the destination the converted audio file needed to be usable in, and as the tool that rejected the original .opus file with an upload error. Second as a comparison for slide generation.
- **Notes:** Its slides are visually better than those Cowork produces, but each slide is generated as an image and cannot be edited directly. Improvement has to be requested through prompting rather than by editing the text.

- **Claude Code**

- **What it is:** Anthropic's tool for more complex coding work. Students had not yet covered it at the time of this session.
- **Used for (in this lesson):** Named only as a recommendation. Asked whether Cowork can batch edit videos, Vyintas answered that it can, but that reliable video editing usually requires workflows defined as code and more complex coding libraries, making Claude Code the better choice.
- **Gmail connector**
 - **What it is:** A connector giving Claude access to an email inbox.
 - **Used for (in this lesson):** The scheduled daily email triage task read the inbox through it.
 - **Notes:** No reconnection was required inside Cowork because the connector already existed on Claude.ai, and the two share connectors.
- **Google Drive integration**
 - **What it is:** An option offered on files Cowork returns in the chat, opening them in Google's applications instead of locally.
 - **Used for (in this lesson):** Opening the generated pitch deck as Google Slides and the business plan as a Google Doc.
- **Obsidian**
 - **What it is:** A local text and note application.
 - **Used for (in this lesson):** Named only as the subject of a deferred demonstration. Vyintas promised to show how to connect Obsidian to Cowork for co editing long text files, and how to establish a knowledge graph, during the implementation lab.
- **A more autonomous competing platform**
 - **What it is:** Vyintas refers to a category of more autonomous agent platforms that had recently generated excitement. The name is not clearly recoverable from the transcript audio, appearing variously as "OpenCloude" and "OpenClaw".
 - **Used for (in this lesson):** Comparison only.
 - **Notes:** He states these carry more significant security issues, and that while Cowork may be less autonomous, its guardrails are a deliberate trade. He also advises that platforms of that kind are the case where a separate computer, separate hard drive, and distinct permissions would be warranted, whereas for Cowork he recommends using the main computer.

Demo Walkthroughs

- **Demo 1: Business launch package (Pillar 2, triggered first and reviewed last)**
 - **Goal:** Delegate a complete multi deliverable launch package in a single prompt and let it run autonomously through the rest of the class.

- **Step 1: Prepare the input and output folders**
- **Where:** A folder on the desktop containing an Inputs subfolder and an Outputs subfolder.
- **Do:** Place the source material in Inputs. In this case a branding document, an offer and packages document, a CSV of student feedback, and a photograph. Leave Outputs empty.
- **Expected result:** Cowork has the reference material it needs and a defined destination.
- **Confidence:** High
- **Step 2: Start the task and select the folder**
- **Where:** Claude Cowork, New Task.
- **Do:** Paste the prompt. As dictated by the instructor:
Create a complete launch package for my AI coaching program for professionals and entrepreneurs. Deliver market research report, ideal client profile, product definition, business plan, Excel financial model, HTML landing page, and PowerPoint pitch deck.
When prompted, select the folder.
- **Expected result:** The task begins and continues running without further input.
- **Troubleshooting:** If a folder is not chosen in advance, Cowork asks for one before starting.
- **Confidence:** High
- **Step 3: Respond to the permission request**
- **Where:** The running task.
- **Do:** Cowork requests permission to permanently delete files in the folder during the task. Approve or decline.
- **Expected result:** The task continues. This request appears even when act without asking has been set, because destructive actions are held back regardless.
- **Confidence:** High
- **Step 4: Review the outputs**
- **Where:** The Outputs folder, and the chat.
- **Do:** Open each deliverable.
- **Expected result:** After roughly half an hour, the folder contains a PowerPoint pitch deck that has pulled the real offers from the input document and the supplied photograph, an Excel financial model with conservative, base, and optimistic scenarios, profit and loss for each, automatically calculated monthly cash flow and an executive summary, a business plan covering products and services, phases, marketing and sales channels, an operations plan and a financial projection summary, and an HTML landing page.
- **Troubleshooting:** The instructor identifies two errors in his own output on review. The market opportunity figure on the pitch deck is inflated, and a satisfaction rating reads ten out of ten because the feedback CSV he supplied contained only the submissions praising him. He states he would correct the rating before use. Results are a first draft requiring review.

- **Confidence:** Medium
- **Demo 2: File reorganization (Pillar 1)**
 - **Goal:** Rename a folder of files to a naming convention and sort them into subfolders.
 - **Step 1: Start the task and select the folder**
 - **Where:** Claude Cowork, New Task.
 - **Do:** Paste the prompt. As dictated by the instructor, it asks Cowork to clean up the files in the folder and change the file names to start from the number of the lesson, then one keyword of what the file is about, whether workbook, transcript, or chat, then the lesson name and the cohort year. Select the target folder when prompted.
 - **Expected result:** Cowork renames each file to the convention, creates Chats, Transcripts, and Workbooks subfolders that did not previously exist, and moves each file into the correct one. The changes are visible live in the folder on the desktop.
 - **Confidence:** High
- **Demo 3: Batch image edit (Pillar 1)**
 - **Goal:** Resize and crop a folder of speaker photographs of differing dimensions so all match one reference image with the subject centered.
 - **Step 1: Configure the task**
 - **Where:** Claude Cowork, New Task. Use the "work in a project" control beneath the prompt field to select the folder. The instructor notes that this control lists computer folders as well as Cowork projects, and that a Cowork project is itself a folder.
 - **Do:** Select the photographs folder. Set permissions to act without asking, since nothing in this task is destructive. Select model Sonnet 4.6 rather than the most powerful model, to conserve credits.
 - **Expected result:** The task is scoped and ready.
 - **Confidence:** High
 - **Step 2: Run the prompt**
 - **Where:** The configured task.
 - **Do:** Paste the prompt. As dictated, it states that the folder contains speaker photos that all need to look the same size, instructs Cowork to use Vishen's photo as the size standard and make all other photos match it exactly, notes that the instructor's own photo is zoomed out and should be cropped correctly, and asks that every speaker be centered.
 - **Expected result:** Cowork evaluates the images, determines which tools it requires, installs them into a sandbox, writes and runs code, and returns the resized photographs.
 - **Troubleshooting:** In the demonstration four of five photographs were correct and the instructor's own was cropped badly. The remedies given were to provide feedback for a one off correction, or to build a skill if the task will recur.

- **Confidence:** Medium

- **Demo 4: Audio format conversion (Pillar 1)**

- **Goal:** Convert a WhatsApp voice file downloaded in .opus format, which would not open on the desktop and which NotebookLM refused to accept, into a standard playable format.
- **Step 1: Select the folder and run the task**
- **Where:** Claude Cowork, New Task.
- **Do:** Paste the audio conversion prompt and select the folder containing the .opus files.
- **Expected result:** Cowork converts the file to a format that plays on the desktop and can be uploaded to NotebookLM.
- **Troubleshooting:** The instructor selected an incorrect folder and observed the interface then displaying one folder "plus one". This revealed that more than one folder can be selected for a single task. Existing selections are removed by unchecking them.
- **Confidence:** High

- **Demo 5: Mobile Dispatch (Pillar 1)**

- **Goal:** From a phone, have desktop Cowork read a folder of PDFs and report back their key content, in preparation for a client call while away from the computer.
- **Step 1: Establish why Cowork is needed**
- **Where:** Claude.ai.
- **Do:** Attempt to upload the folder of PDFs.
- **Expected result:** Claude.ai returns the message that the chat has reached the 100 image limit, including PDF pages, and that a new chat is required. The task cannot be completed there without splitting it across conversations.
- **Confidence:** High
- **Step 2: Send the task from the phone**
- **Where:** The Claude mobile application, side menu, Dispatch.
- **Do:** Send the request. As dictated, it directs Cowork to the Cowork Demos folder on the desktop, then to the folder of text files for the demonstration, states that the sender needs to prepare for a call with a client and does not have access to the files, and asks Cowork to go through all of them and respond listing the most important tools covered and the most important frameworks mentioned, specifying that no long message is needed.
- **Expected result:** The task appears in the Dispatch section of desktop Cowork and runs autonomously. Permissions may be requested on the phone during the run.
- **Troubleshooting:** In the demonstration Cowork chose to use computer use, visually navigating the screen and clicking through folders. This took control of the instructor's machine, prevented him from moving windows, and had to be stopped by messaging Dispatch. He states the fix is to include an instruction not to use computer use unless it is

really necessary, since back end navigation is more efficient, and that Dispatch should not be used while at the computer it controls. The demonstration was not completed.

- **Confidence:** Low

- **Demo 6: Daily email triage as a scheduled task (Pillar 3)**

- **Goal:** A recurring automated task that reviews the inbox and returns a structured priority briefing, built so that feedback improves it over time.
- **Step 1: Create a project to contain the task**
- **Where:** Claude Cowork, Projects, New Project, Start from scratch.
- **Do:** Name the project. Add no instructions and no files. Accept the default project location, which creates a local folder. Memory is on by default.
- **Expected result:** A project exists, backed by a folder on the computer.
- **Troubleshooting:** The instructor begins creating the scheduled task directly, then cancels and takes this route instead. The reason is memory: project memory is a local text file in the project folder, so feedback given after a run is retained and the task improves without the instructions being edited by hand.
- **Confidence:** High
- **Step 2: Create the scheduled task inside the project**
- **Where:** Inside the project, the plus button beside Scheduled, then New Task, then Set up manually. The alternative offered is Create with Claude, which works the same way skill creation does.
- **Do:** Enter a name and a description, and paste the instructions. As dictated, the instructions ask Cowork to check the Gmail inbox for emails received in the last twenty four hours and then produce a structured briefing using the priority tiers critical, legal and financial, action required, awaiting reply, follow up, FYI worth knowing, and skipped.
- **Expected result:** The task is created within the project.
- **Confidence:** High
- **Step 3: Set permissions, model, and frequency**
- **Where:** The task configuration.
- **Do:** The project folder is already selected because the task was created inside the project. Set permissions to act without asking, since a scheduled task that stops to ask defeats its purpose. Select model Sonnet 4.6 to conserve credits. Set frequency to Manual. The available options are manual, hourly, daily, weekdays, and weekly.
- **Expected result:** The task saves and appears as active under Scheduled within the project.
- **Troubleshooting:** The instructor advises always selecting Manual first so the task can be tested, and changing the frequency afterwards.
- **Confidence:** High
- **Step 4: Test the task**

- **Where:** The saved task.
- **Do:** Click Run now.
- **Expected result:** The run appears under History with its trigger time. Claude returns the inbox briefing grouped into the specified priority tiers, including a skipped group listing what it judged irrelevant. The most recent run of any scheduled task also appears under the Scheduled section in the left panel.
- **Troubleshooting:** No reconnection to the inbox is required if the connector already exists on Claude.ai, since connectors are shared.
- **Confidence:** High

Applied Use Cases

- Reorganizing and renaming a folder of files to a naming convention and sorting them into subfolders, as demonstrated with lesson materials.
- Standardizing a batch of images to a single size and framing, with a skill built for the task if it will be repeated.
- Converting a file that arrives in an awkward format into one that other tools accept, as demonstrated with a WhatsApp .opus audio file destined for NotebookLM.
- Preparing for a meeting while away from the computer, by sending a Dispatch request that reads local files and returns a short brief to the phone.
- Producing a full first draft launch package quickly when speed matters more than polish, accepting that it requires review.
- Creating slides, spreadsheets, and documents as editable files when final control over the deliverable is needed, in contrast to tools that return slides as images.
- Working with large data files, spreadsheets, and CSVs, which the instructor identifies as a particular strength of Cowork.
- Recurring tasks run on a cadence, such as a daily inbox briefing or a daily report, set up as scheduled tasks.
- An extended version of inbox handling that labels emails and drafts responses, which the instructor confirms is possible and defers to the implementation lab.

Key Takeaways

- Claude Cowork runs the same Claude model as Claude.ai but lives on the computer rather than in the cloud. That single distinction determines which one a given task belongs in.
- Three triggers decide the question: files and folders on the computer, a long multi step workflow, or a task to be run without being triggered each time. Anything else remains in Claude Chat.
- Cowork can read, move, rename, and edit local files, and can also create new files in any format, including spreadsheets, documents, and presentations.

- Cowork determines what tools a task requires, installs them into a sandbox rather than into the operating system, and runs code without any capability having been configured in advance.
- Folder selection is the primary security boundary, more than one folder can be selected, and Cowork still requests permission before permanently deleting files even when set to act without asking.
- Long autonomous workflows are possible because Cowork acts as a supervisor creating parallel sub agents rather than working in sequence as a single agent.
- Delegating many deliverables in one prompt is a real capability but produces a first draft. Split the workflow with review between stages for quality, or build a supporting system of skills and memory if the task recurs.
- Scheduled tasks are best created inside a project, because project memory retains feedback and improves later runs without the instructions being edited manually.
- The principal limitation is that the computer must be on, logged in, connected, and not sleeping. Dispatch extends reach from a phone but should not be used while at the computer.

Resources & Links

- A document prepared by the instructor and shared with the class at the start of the session, containing every prompt used in the demonstrations. No URL is stated in the transcript.
- obsidian.md, named in connection with the deferred co editing and knowledge graph demonstration.
- NotebookLM, used as the destination for the converted audio file and as the comparison for slide generation. No URL is stated in the transcript.

Tracie Fujikane ·