I Trained Claude to Run My Admin. Here's the Full Playbook.

A playbook for creating three Claude Cowork scheduled tasks that analyze your meetings, draft your Slack and email replies, and generate structured update proposals for your product docs, all reading from one shared source of truth.

Written by Lance Owide, VP of B2B Product at Commerce

WHAT YOU'RE BUILDING

Three skills. One shared brain.

A set of scheduled tasks in Claude Cowork that run on a recurring cadence and automatically:

- **Analyzes your meetings.** Reads Google Meet transcripts off your calendar, matches each discussion to the right product pack via a project index, generates structured update proposals, and drafts recap emails to attendees.
- **Responds to your Slack.** Pulls DMs and @mentions, drafts replies in your voice, and saves them as Slack drafts for review.
- **Responds to your email.** Drafts replies using your sent-mail voice, flags sensitive topics, and leaves everything as Gmail drafts.
- **Shares a brain.** All three skills read from the same set of product docs, so a decision captured in a morning meeting informs every Slack and email reply for the rest of the day.

You review the drafts in Gmail and Slack, review the update proposals in the Cowork task output, and apply the approved changes to your product docs. Nothing gets sent or written to your docs automatically.

PREREQUISITES

Claude Desktop
Cowork enabled

Google Workspace
Gmail, Calendar & Drive connected

Slack Integration
Connected via Claude

THE THREE SKILLS

Meeting Notes Analyzer

Reads transcripts, generates structured update proposals, drafts recap emails.

Slack Responder

Reads DMs and mentions, drafts replies in your voice, saves as Slack drafts.

Email Responder

Reads inbox, drafts replies in your voice, flags sensitive topics for review.

THE CORE IDEA

The product docs are the shared brain. The skills are the hands.

Most people build AI assistants one task at a time. An email drafter. A meeting summarizer. A Slack bot. Each runs in isolation, does its narrow job, and moves on.

The real leverage shows up when multiple skills read from the **same source of truth**.

What you need in Google Drive

Product packs. One Google Doc per active workstream. Each pack contains the current scope, open questions, decisions, risks, and action items for that project. These are living operational docs, not PRDs.

A project index. One master doc that maps keywords and aliases to product pack Doc IDs. When someone says "PO Automation" or "the purchase order flow thing," the index tells your skills which pack to open.

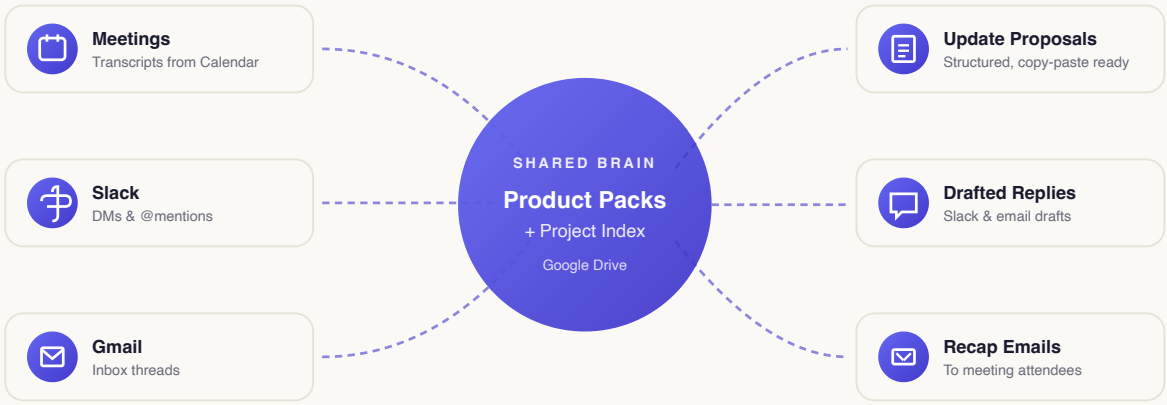
A Meeting Intelligence folder. A folder in Google Drive where the meeting analyzer creates a new dated doc per run. Your archive of structured meeting analysis, update proposals, and decisions over time.

The compounding part: A call happens at 10am. By 10:45, the meeting analyzer has matched the transcript to the right pack, output the new decisions as structured proposals, and drafted the follow-up email. You paste the approved proposals into the pack over coffee. At 11, someone Slacks you about that same decision. The Slack responder reads the updated pack and already knows.

One Project Index. Three skills. All three skill prompts below ask for the SAME `[YOUR INDEX DOC ID]`. Use the same Doc ID in all three places. That is what wires the shared brain together: the meeting analyzer reads the packs and proposes updates; you paste them in; the Slack and email skills read the same packs on every run and draft accurately. Different Doc IDs = three skills that do not know about each other.

THE DATA FLOW

How the three skills work together



Inputs (left) flow through the shared brain (center) and produce outputs (right). The meeting analyzer reads from the packs and generates update proposals for you to review; the Slack and email responders read from the packs to draft replies in context.

SKILL 1 OF 3

Meeting Notes Analyzer

The most involved of the three skills, and the one that pays back the most. Every 30 minutes during business hours, the skill checks your calendar for meetings that ended in the last 4 hours, reads the transcripts, matches them to product packs, generates structured update proposals, and drafts recap emails.

An important note on how updates work. Claude's Google Drive integration can create new Google Docs and read existing ones, but it cannot edit, append to, or comment on existing docs. That's a current limit of the integration, not a Claude limitation. So this skill outputs update proposals in two places: (1) the Cowork task log, structured as copy-paste-ready bullet lists; (2) a new dated "Meeting Intelligence" Google Doc per run containing the full analysis. You review, then paste the approved changes into your product packs manually. It's 2 minutes of clicking per meeting. The skill does everything else.

Before you start: create one Google Doc and one folder

The Project Index. One Google Doc containing three parts: a Configuration section, a Project Mappings table, and a Processed Meetings Log table. Structure it like this:

PART 1: CONFIGURATION

```
INTELLIGENCE_FOLDER_ID: [paste your Meeting Intelligence folder ID here]
LOOKBACK_HOURS: 4
```

PART 2: PROJECT MAPPINGS

Project Name	Aliases	Product Pack Doc ID	Status
PO Automation	purchase order automation, POs, PO flow	1abc...xyz	Active
Cascading Price Lists	CPL, price list fallback	1def...uvw	Active

PART 3: PROCESSED MEETINGS LOG

Meeting Title	Date	Matched Projects	Timestamp
You paste rows here after reviewing each run's output			

The Meeting Intelligence folder. A folder in your Google Drive where the skill will create a new dated Google Doc after each run, containing the full structured analysis (matched projects, update proposals per pack, action items by owner, cross-project impact, risks). Create this folder first, then paste its folder ID into the Configuration section above.

> **The aliases column is doing real work.** When the skill reads a transcript and someone says "the PO flow thing," it needs to know that maps to PO Automation. Put every nickname, acronym, and shorthand you actually use. Build Parts 2 and 3 as real Google Docs *tables* (Insert > Table), not as plain text — the skills look for table rows.

Product Pack template (one per workstream)

Create one Google Doc per active project, placed anywhere you like. The meeting analyzer's Update Proposals reference specific sections by name, so use these section headings verbatim so the proposals slot in cleanly:

```
PRODUCT PACK: [Project Name]
Last updated: [YYYY-MM-DD]

SCOPE
One paragraph: what we are building and for whom.

STATUS
Current state, target launch date, blockers.

OPEN QUESTIONS
- Bullet list of unresolved decisions

DECISIONS
- YYYY-MM-DD: short statement of the decision
- YYYY-MM-DD: another decision

METRICS
- Test set accuracy: X%
```

```
- Pilot volume: Y

RISKS
- Bullet list of known risks

ACTION ITEMS
- Owner to do X by YYYY-MM-DD
- Owner to do Y by YYYY-MM-DD

PILOT CUSTOMER (or KEY STAKEHOLDERS)
Names, companies, contact emails.
```

Keep the pack short — under one page is ideal. The skills read every active pack on every run, so length costs latency and token budget. Each new meeting trims what is no longer relevant rather than appending forever.

How to find a Google Doc ID or Folder ID

Open the doc or folder in Drive. The ID is the long alphanumeric string in the URL between `/d/` (for Docs) or `/folders/` (for folders) and the next `/`. Examples:

```
Doc:
https://docs.google.com/document/d/1AbCdEf_GhIjKlMnOpQrStUvWxYz0123456789AbCdE/edit
Folder: https://drive.google.com/drive/folders/1AbCdEf_GhIjKlMnOpQrStUvWxYz012345
```

Copy that string. Paste it into the highlighted placeholders in the setup prompts below.

STEP 1: THE SETUP PROMPT

Generate your task prompt

Open a new Claude conversation with Google Calendar, Drive, and Gmail connected. Paste this prompt. Customize the **highlighted** sections before running.

```
Create an automated meeting notes analysis skill that runs on a schedule. Here's what I need:

-- Calendar lookup --
Every time this runs, search my Google Calendar for events that ended within the last [LOOKBACK
WINDOW, e.g., 4 hours]. For each event, check if it has attached Google Docs (notes or
transcripts). Skip events with no attachments.

-- Deduplication --
I have a Project Index Google Doc at Doc ID [YOUR INDEX DOC ID]. In that doc is a "Processed
Meetings Log" table with columns: Date, Title, Transcript Doc ID, Matched Project(s),
Intelligence Doc ID. Read that table to identify already-processed meetings. Dedup by the
Transcript Doc ID column (not by title) since titles can vary slightly. At the end of the run,
output the newly-processed meetings as a copy-paste-ready row block so I can manually append them
to the log table.

-- Project matching and memory lookup (REQUIRED) --
The Project Index contains a table mapping project names and aliases to product pack Doc IDs. For
each meeting transcript:
1. Read every active project pack first to load full context. You must actually open each pack
via its Doc ID and read it, not just glance at the title.
2. Scan the transcript against all active projects and tag every project that is meaningfully
discussed (not just mentioned). A project qualifies if a decision was made about it, an action
```

```
item was assigned, scope or timeline was discussed, or dependencies were raised.
3. When generating Update Proposals (next step), compare what was said in the meeting against
what is already in the pack. Only propose NEW information or CHANGES. Do NOT propose adding
anything the pack already documents. This is what makes the shared brain work.

-- Update proposals (IMPORTANT) --
The Google Drive integration cannot edit, append to, or comment on existing Google Docs. Do NOT
attempt to modify product packs directly. Instead, for every matched project, generate a
structured "Update Proposal" block in the task output AND in the new Meeting Intelligence Doc
(see below). Each proposal should specify: (1) the product pack Doc ID and title, (2) the section
of the pack it applies to, (3) the specific text to add/change/remove, (4) the source (meeting +
speaker), and (5) the rationale. Format proposals so I can copy-paste them directly into the
relevant pack section.

-- Meeting Intelligence Doc --
After analyzing each run's meetings, create a new Google Doc in folder ID [YOUR INTELLIGENCE
FOLDER ID] titled "Meeting Intelligence: [YYYY-MM-DD HH:MM] [TIMEZONE]" using the current run
time (e.g., "Meeting Intelligence: 2026-05-31 14:00 PT"). Content: (1) meetings processed, (2)
matched projects per meeting, (3) all update proposals grouped by product pack, (4) action items
grouped by owner with absolute YYYY-MM-DD due dates (interpret any relative dates from the
transcript like "by Monday" as the closest future date), (5) cross-project impact notes, (6) new
risks surfaced, (7) unmatched discussions. Use clear headers so it's scannable.

-- Follow-up emails --
For meetings with 8 or fewer attendees, draft a Gmail recap email to all attendees. Subject:
"Follow up: [Meeting Title]". Write in my voice: warm, direct, action-oriented. Use my name [YOUR
NAME] in the sign-off. Never send. Always create as a draft. Skip the draft for meetings with 9+
attendees.

-- Voice profile --
Before drafting any emails, search my sent Gmail from [DATE RANGE] and study my writing patterns.
Apply that voice to every draft.

-- Safety guardrails --
Never send emails automatically. Never attempt to edit existing Google Docs (the integration does
not support it). If a meeting discusses a project not in my index, include it in the Intelligence
Doc under "Unmatched Discussions" and flag that keywords might need to be added. Flag uncertain
factual claims with [VERIFY] inline next to the claim. Flag drafts that touch sensitive topics
(personnel, compensation, legal, budget commitments to customers) with [PLEASE REVIEW] at the top
of the draft.

-- Schedule --
Run every [FREQUENCY, e.g., 30 minutes] during business hours on weekdays.

-- Output --
Give me the final prompt as plain text that I can paste directly into a Cowork scheduled task. Do
not generate code or an HTML file. Just the prompt.
```

Claude will generate a plain-text task prompt. Save the output. That is your task.

STEP 2: CREATE THE SCHEDULED TASK

From prompt to production

1. Open Claude Desktop and navigate to **Cowork > Scheduled Tasks**.
2. Create a new task. Name it **Meeting Notes Analyzer**.

3. Paste the prompt Claude generated in Step 1 as the task instructions.

4. Set frequency to **every 30 minutes**, restricted to **business hours on weekdays**.

5. Enable the task.

What you'll see after each run

- **Cowork task output.** Meetings processed, matched projects, structured update proposals per pack, action items by owner, and a list of drafts created.
- **A new Meeting Intelligence Google Doc.** Dated and saved in your intelligence folder, containing the full analysis for reference and for pasting into product packs.
- **Gmail drafts.** One per meeting with 8 or fewer attendees, ready for review.

Your 5-minute review workflow. Open the Intelligence Doc, skim the update proposals, copy the ones you agree with into the relevant product pack sections, delete the rest. Add the meeting to the Processed Meetings Log table in your index. Open Gmail drafts, review, send.

SKILL 2 OF 3

Slack Responder

Simpler than the meeting analyzer. Every hour during business hours, the skill searches Slack for DMs and @mentions you haven't responded to, reads the thread context, drafts a reply in your voice, and saves it as a Slack draft in the relevant channel or DM.

```
Create an automated Slack responder skill that runs on a schedule. Here's what I need:

-- Message discovery --
Every time this runs, search Slack for DMs and @mentions directed at me that I haven't responded
to. Focus on messages from the last [LOOKBACK WINDOW, e.g., 24 hours]. Skip messages I've already
replied to, bot notifications, and channels where I'm a passive member.

-- Context loading --
For each message that needs a reply, read the full thread or recent channel history so the draft
has proper context.

-- Memory lookup (REQUIRED before drafting) --
Before drafting any reply, do this in order:
1. Open my Project Index Google Doc at Doc ID [YOUR INDEX DOC ID]. This is the same Index Doc ID
used by the meeting analyzer.
2. Read the Project Mappings table to get the full list of project names, aliases, and product
pack Doc IDs.
3. Scan the message and thread for any project name, alias, or shorthand from the table (case-
insensitive, partial match).
4. For every project mentioned (even in passing), open the matching product pack via its Doc ID
and read it fully.
5. ALSO check the Meeting Intelligence folder ID [YOUR INTELLIGENCE FOLDER ID] for the most
recent Intelligence Doc (sort by Modified date). If its timestamp is more recent than the matched
pack's "Last updated" line, read its Update Proposals for the matched project. Those proposals
reflect decisions the user has not yet pasted into the pack. Prefer them over older pack content
```

```
when they conflict, and append [VERIFY: from unapplied proposal] inline next to any number/date
you draw from there.
6. Use the pack contents (current status, decisions, commitments, action items), augmented by any
fresher proposals from step 5, to make the draft accurate and specific. Reference recent
decisions by name where relevant. Do NOT ask "what is the status?" if the pack or proposals
already answer it.
7. If a project mentioned in the thread is NOT in the index, flag the draft at the top with [NEW
PROJECT: name] so I can add it to the index later.

If no project is mentioned in the thread, draft using thread context and voice profile alone.

-- Voice cloning --
Search my sent Slack messages from [DATE RANGE] to build a voice profile. Study my greetings,
sign-offs, sentence length, and phrases. On Slack I'm more casual than email: shorter sentences,
often no greeting, no sign-off usually, no emojis, no "Dear" or "Best regards".

-- Tone by relationship --
Different audiences get different tones. With [YOUR GROUPS, e.g., direct reports vs. peers vs.
leadership], adapt tone accordingly. Never stiff.

-- Smart filtering --
Only draft replies for messages that need a response. Skip FYI messages, automated notifications,
channel-wide announcements, and messages someone else has answered. For meeting requests, include
my scheduling link: [YOUR SCHEDULING LINK].

-- Safety guardrails --
Never send messages. Always use the Slack draft tool to create drafts. If the Slack draft tool is
not available in this run, output the draft as plain text in the task log with a clear "DRAFT FOR
[channel/DM]:" header so I can paste it manually. Flag uncertain factual claims with [VERIFY]
inline next to the claim. Flag drafts that touch sensitive topics (personnel, compensation,
legal, budget commitments) with [PLEASE REVIEW] at the top of the draft.

-- Schedule --
Run every [FREQUENCY, e.g., hour] during business hours on weekdays.

-- Output --
Give me the final prompt as plain text I can paste into a Cowork scheduled task. Not code, not
HTML. Just the prompt.
```

Create the scheduled task

1. Open Claude Desktop and navigate to **Cowork > Scheduled Tasks**.
2. Create a new task. Name it **Slack Responder**.
3. Paste the prompt Claude generated as the task instructions. Make sure the **[YOUR INDEX DOC ID]** placeholder is replaced with the same Doc ID used by the meeting analyzer.
4. Set frequency to **every hour**, restricted to **business hours on weekdays**.
5. Enable the task.

SKILL 3 OF 3

Scheduled Email Responder

Every hour during business hours, the skill reads your inbox, skips newsletters and automated notifications, drafts replies to the rest in your voice, flags sensitive or complex threads with [PLEASE REVIEW], and marks uncertain factual claims with [VERIFY].

Two hard-won rules. Without explicit instructions, the email skill will quietly do two things wrong: (1) reply only to the sender instead of everyone on the thread, and (2) create drafts that land as standalone messages instead of threading into the original conversation. Both are below in the prompt as Rule 1 (reply-all) and Rule 2 (threading). Don't remove them.

```
Create an automated email drafting skill that runs on a schedule. Here's what I need:

-- Voice cloning from sent emails --
Search my sent emails from [DATE RANGE] and analyze my writing patterns: greetings, sign-offs,
sentence structure, vocabulary, level of formality, and recurring phrases. Use this to build a
voice profile so all drafted replies sound like me.

-- Category-based tone adaptation --
From my sent emails, identify how I write differently to different groups (e.g., [YOUR GROUPS,
e.g., teammates vs. clients vs. leadership vs. external partners]). When drafting a reply, match
the tone to whoever I'm responding to.

-- Memory lookup (REQUIRED before drafting) --
Before drafting any reply, do this in order:
1. Open my Project Index Google Doc at Doc ID [YOUR INDEX DOC ID]. This is the SAME Index Doc ID
used by the meeting analyzer and Slack responder.
2. Read the Project Mappings table to get the full list of project names, aliases, and product
pack Doc IDs.
3. Scan the email subject, body, and full thread for any project name, alias, or shorthand from
the table (case-insensitive, partial match).
4. For every project mentioned, open the matching product pack via its Doc ID and read it fully.
5. ALSO check the Meeting Intelligence folder ID [YOUR INTELLIGENCE FOLDER ID] for the most
recent Intelligence Doc. If its timestamp is more recent than the matched pack's "Last updated"
line, read its Update Proposals for the matched project. Those reflect decisions the user has not
yet pasted into the pack. Prefer them over older pack content when they conflict, and append
[VERIFY: from unapplied proposal] inline next to any number/date you draw from there.
6. Use the pack contents (current status, decisions, commitments, action items), augmented by any
fresher proposals from step 5, to make the draft accurate and specific. Reference recent
decisions by name where relevant. Do NOT ask "what is the status?" if the pack or proposals
already answer it.
7. If a project mentioned in the email is NOT in the index, flag the draft at the top with [NEW
PROJECT: name] so I can add it to the index later.

Updates the meeting analyzer makes to packs (after I paste them in) will be visible here
automatically on the next run. This is what makes the shared brain work.

-- Knowledge extraction from email history --
On top of the project pack lookup above, as you read my sent emails, also remember broader
patterns: how I've handled similar situations, recurring topics that don't have a dedicated pack,
and one-off commitments. Use that context as a secondary source after the pack lookup.

-- Company product knowledge --
I work at [YOUR COMPANY]. When drafting replies to technical questions, reference our public
documentation at [YOUR DOCS URL] and any other publicly available product information. Use web
```

```
search if needed to find current details.

-- Meeting scheduling --
When someone asks to meet or schedule a call, include my scheduling link in the draft: [YOUR
SCHEDULING LINK]. Use this instead of going back and forth on times.

-- Smart filtering --
Only draft replies for emails that actually need a response. Skip newsletters, automated
notifications, marketing emails, requests to move calendar invites, and FYI-only messages.

-- Rule 1: ALWAYS reply-all --
Every draft must include every other participant on the thread, not just the sender. Before
creating the draft, run this recipient extraction in order:
1. Get the most recent message in the thread.
2. Take the From address of that message. This is the first entry of "to".
3. Take every address from the To header of that message. Add them to "to".
4. Take every address from the Cc header of that message. Add them to "cc".
5. Remove my own address ([YOUR EMAIL ADDRESS]) from both "to" and "cc". Remove any alias
variants of my address too.
6. Deduplicate both lists. An address must not appear in both "to" and "cc": if it does, keep it
in "to" and drop it from "cc".
7. If after step 5 the "to" list is empty (the only other participant was me), skip this thread,
do not draft.
The only time "cc" will be empty is if the original message had no Cc header. Reply-only-to-
sender is never acceptable unless the thread literally has one other participant and no Cc field.

-- Rule 2: The draft MUST thread into the original conversation --
Gmail stitches drafts into the right thread by matching the subject line. If the subject is even
slightly off, the draft lands as a standalone message and I have to manually fix it. To force
threading:
1. Take the Subject header of the most recent message in the thread, verbatim.
2. If that subject already starts with "Re:" or "re:" or "RE:" (with or without a space after),
use it exactly as-is. Do NOT add another Re: prefix, do NOT normalize the casing.
3. If it does not start with Re:, prepend exactly "Re: " (capital R, lowercase e, colon, one
space), then append the original subject with no other changes.
4. Do not trim leading or trailing whitespace, do not change punctuation, do not "clean up" the
subject, do not translate it, do not shorten it. Copy it byte-for-byte.
5. Do not include any quoted or forwarded body from the original message in the body field. Only
my new reply text.
6. If the Gmail draft tool supports a replyToMessageId parameter, also pass the message ID of the
most recent message in the thread. This is a belt-and-braces approach: subject-matching is the
primary threading mechanism, replyToMessageId is the backup.

-- Safety guardrails --
Never send anything automatically. Only create drafts using the Gmail draft creation tool. Flag
uncertain factual claims with [VERIFY] inline next to the claim. Flag drafts that touch sensitive
topics (personnel, compensation, legal, budget commitments to customers, hiring/firing, board-
level) with [PLEASE REVIEW] at the top of the draft body.

-- Schedule --
Run every [FREQUENCY, e.g., 1 hour].

-- Output --
Give me the final prompt as plain text that I can copy and paste directly into a Cowork scheduled
task. Do not generate code, an app, or an HTML file. Just the prompt.
```

Create the scheduled task

1. Open Claude Desktop and navigate to **Cowork > Scheduled Tasks**.

2. Create a new task. Name it **Email Responder**.
3. Paste the prompt Claude generated as the task instructions. Make sure **[YOUR INDEX DOC ID]** is the same Doc ID used by the other two skills.
4. Set frequency to **every hour**, restricted to **business hours on weekdays**.
5. Enable the task.

CHOOSING YOUR FREQUENCIES

How often should each skill run?

Skill	Recommended	Best For
Meeting Analyzer	Every 30 min, business hours	Catches meetings within 30 min of ending, before you need the recap
Slack Responder	Every hour, business hours	Frequent enough to feel responsive, infrequent enough to avoid noise
Email Responder	Every hour, business hours	Same logic as Slack. Hourly is the sweet spot.

Start lower, ramp up. For the first week, run each skill every 2 hours so you can review drafts in larger batches and refine the voice. Once you trust the output, tighten the cadence.

WHAT YOU LEARN AFTER A FEW WEEKS

Five lessons from running this setup

- **The hard part isn't the AI. It's the plumbing.** Matching meetings to docs. Keeping voice consistent across skills. Knowing when to draft vs. when to escalate. The prompt is 20% of the work. The other 80% is the rules around the prompt.
- **Scheduled tasks beat chat.** A skill that runs every 30 minutes without you asking is 10x more useful than one you have to summon. The friction of "open Claude, ask, wait" is what kills most assistant use cases.
- **Always draft, never send.** Non-negotiable. Review every output before anything goes out. The skill is your assistant, not your replacement.
- **A project index is the unlock.** One source of truth that tells the skills "this meeting = this doc" is worth more than any prompt tweak.

→ **Start with one skill, then add.** Don't build all three at once. Build the one that saves you the most time. Run it for two weeks. Then add the next one.

QUICK SETUP CHECKLIST

Work through these in order

- [ ] Claude Desktop with Cowork installed
- [ ] Gmail, Google Calendar, Google Drive, and Slack integrations connected
- [ ] Project Index Google Doc created with Configuration, Mappings, and Processed Log sections
- [ ] Meeting Intelligence folder created in Google Drive
- [ ] Folder ID pasted into the Configuration section of the index
- [ ] First product pack Doc created for one active workstream
- [ ] Setup prompt customized (Doc IDs, name, date range, groups, frequency)
- [ ] Meeting Notes Analyzer: task prompt generated and pasted into Cowork
- [ ] Frequency set (every 30 minutes, business hours, weekdays)
- [ ] First run completed, Intelligence Doc reviewed, proposals applied, drafts reviewed
- [ ] Slack Responder: added after meeting analyzer is stable
- [ ] Email Responder: added last, once voice profiles are refined
- [ ] Voice profiles re-generated every 30-60 days to stay current

A NOTE ON PRIVACY

Everything runs on your own accounts

The skills read your emails, meetings, and Slack messages because you asked them to. Drafts stay drafts until you send them. Update proposals stay as structured text in the Cowork output and the Intelligence Doc until you manually paste them into your product packs.

If you're doing this inside a regulated company, check with IT and legal first. Some organizations restrict which AI tools can read corporate email, and some require specific data-handling agreements. Don't assume you can just turn this on without asking.

For everyone else: build one system, review every output, own the process.