PORTFOLIO · AI WORKFLOW DESIGN · OWN SYSTEMS IN DAILY USE · 2026

Where AI Belongs, and Where a Rule Will Do

My own inbox, cleaned in one morning. AI read the mail and proposed; I made every routing decision; seventeen plain Gmail filters now do the work every day with no AI in the loop. The same split, judgment to AI and repetition to rules, runs through four other systems I use (page 2).

107 → 17
FILTERS, OLD SET TO NEW

84 → 7
LABELS, TO SEVEN WORKING LABELS PLUS ONE SUB-LABEL

16,517
THREADS IN THE INBOX BEFORE, 4,543 UNREAD

399 → ~60
INBOX ARRIVALS A MONTH, SIMULATED ON LAST MONTH'S MAIL

BEFORE AND AFTER

	BEFORE	AFTER
Inbox	16,517 threads, 4,543 unread; most new mail went straight in.	The last 30 days plus anything starred. Everything older archived in one action, still searchable. Starred mail sorts first.
Labels	84, about half left behind by mail apps no longer in use; 21 empty.	Seven numbered labels (1 to 7, people first, tools last) and one sub-label for property. 21 empty labels deleted; 57 old ones nested under two archive parents, nothing lost.
Filters	107. Of these, 97 archived mail; none exempted login codes or security alerts.	17. Every rule that moves mail out of the inbox exempts codes, logins and security alerts, so those always reach it.
Who gets through	No list. I kept missing important mail among the rest.	A VIP list built from 180 days of mail I sent: starred, marked important, never spam.
Bills with due dates	Bill reminders arrived through payment apps, unlabelled.	Kept in the inbox under the property sub-label; 446 past threads labelled to match.

THE METHOD, FOUR STEPS

1 · AI AUDITS
Read-only
Headers of 30 days of incoming mail (638 messages, 156 senders), 180 days of sent mail for the VIP list, and all 107 old filters parsed rule by rule.

→

2 · I DECIDE
A triage form
Every sender line pre-filled with a suggestion; I edited only where I disagreed. Then four policy questions: old filters, old labels, login codes, how to match property mail.

→

3 · PLAIN RULES RUN
One file, 17 filters
Imported by hand into Gmail's own settings. From then on the filters do the sorting; no AI reads my mail day to day.

→

4 · CHECK
Before and after import
Routing simulated on last month's mail before import; after a week (due from 15 Oct), a search for important mail that skipped the inbox.

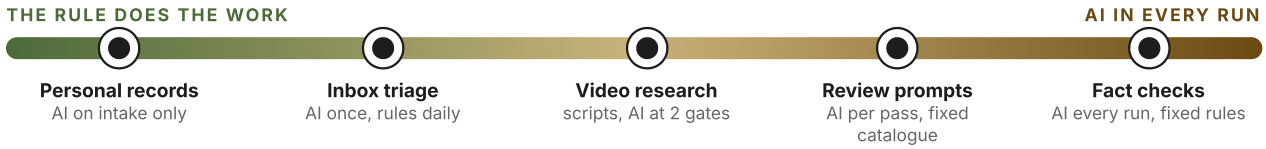
> *The audit changed the plan. Filters that remove junk do not stop important mail being missed, because important mail often comes from new senders. So the rules first decide who gets through, from the people I actually write to, and then move the rest out.*

Role: method, audit design, triage form, filter design and runsheet, with an AI assistant doing the reading and drafting; every routing decision was mine. Personal account: no sender, contact or address appears on this card.

THE DECISION GRID · FIVE SYSTEMS

Each system placed by what its steps actually needed

One question asked of every step. Is the input unstructured, with a choice to make? AI. Is it a call only I can make? Me, at a gate. Does it repeat on a known pattern? A rule, a filter or a script, which costs nothing to run and behaves the same every time.



SYSTEM	AI DOES	I DECIDE	RULE OR SCRIPT DOES	SIZE
Personal records Tax, insurance, investment, property and invoice records, set up so any year can be picked up cold. Made of: playbooks in markdown, CSV registries, naming rules. No code.	On the investment ledger: reads a new statement, compares it with the current file, reports anomalies, then stops.	Says "update" before anything is written; settles open questions.	Naming conventions; one fact in one place; written procedures for each yearly or new-document task; completeness checked by counting the IDs inside receipts against the registry.	7 areas 312 files 20 playbooks and ledgers 19 CSV files
Inbox triage Decides what reaches my inbox and files the rest (page 1). Made of: a one-time audit, a triage form, one filter file.	Once: audits headers and sent mail, proposes labels, a VIP list and a call per sender; simulates the routing.	Every sender call, the label set, the VIP list, four policy questions.	17 Gmail filters, every day, with no AI in the loop.	638 messages audited 17 filters 446 threads backfilled
Video research Turns a topic or a link into research material: video, transcript, description documents, a frame every 3 seconds. Made of: Python and shell scripts, a playbook.	Reads the seed video's transcript to name the real topic; proposes angles; shortlists picks, with reasons on file.	Chooses the angles and approves the picks. The angle choice is a hard gate in code.	Search, download, captions or local transcription, frame slicing with duplicate frames dropped, the index.	4 Python scripts, 1,049 lines 3 collections, 5 runs 38 videos, 38 transcripts 6,113 frames
Review prompts Checks finished work before it ships; keeps my AI working practices in one place. Made of: one triage prompt, two markdown libraries. No code.	Runs one quality pass at a time on the document and reports; never rewrites the work.	Answers the intake questions; picks which passes run, and when.	A 14-pass catalogue in a fixed order. Every library entry must reduce to a rule, a checklist or a template, or it is cut.	9 intake questions 14 passes 39 files 53 practice entries
Fact checks Verifies viral forwards, screenshots and infographics; returns a report, a social graphic and a page of sources. Made of: a playbook with two tracks, two report templates, one graphic script.	Parallel research agents find the primary data or document, compute the real figure and draft a verdict per claim.	Brings the input; decides whether the result is shared.	A fixed five-step verdict scale; every number traces to a URL in the same report; quotes must appear word for word in the source; falsification check mandatory; the graphic is drawn by script, not by AI.	6 phases 5 verdicts 2 runs so far

WHAT THE GRID SHOWS

AI sits where the input is unstructured and a choice is needed: which senders matter, what a video is really about, whether a claim holds.

Anything that repeats on a known pattern became a filter, a script, a registry or a checklist.

Each of my decisions is a gate. Nothing is written to a ledger, downloaded or imported until I say so.

Thin spot, stated: fact checks have two runs so far. The design is tested; the volume is not.

WHAT WENT WRONG, AND THE FIX

Every failure became a rule

Nothing here worked first time. What matters is where each fix landed: in a written rule, a gate or a script, so the same failure cannot come back through a forgetful session.

WHERE	WHAT HAPPENED	THE FIX, NOW STANDING
Inbox access	The Gmail connection could read but not write. Label changes were refused, because the connection had been authorised read-only by default. Filters could not be written through it at all.	Reconnected with label and modify permissions ticked. Filters moved to a file instead: export the old set, rewrite it, import the new one.
Inbox import	Gmail's filter import adds to the existing filters; it does not replace them. The 17 new rules would have run beside the 107 old ones.	Delete all old filters first, then import. Senders still worth routing were carried into the new rules, not lost.
Inbox rule length	Long sender lists pushed single rules toward Gmail's practical length limit.	Two rules split into four; every rule kept under about 800 characters (longest now 777). Gmail does not document the limit, so medium confidence on where it sits.
Inbox scale	Archiving 16,000-plus threads one call at a time through the connector would have meant thousands of calls.	Gmail's own search and select-all did it in one action. The runsheet carries the exact search to paste.
Inbox a silent rule	One old filter had been quietly archiving one person's mail.	Dropped in the rewrite. Found only because all 107 old rules were read before any were replaced.
Video research a skipped decision	A test run on a lighter model chose the topic angles itself instead of asking me.	The angle checkpoint became a hard gate in code: the shortlist script refuses to run without a file quoting my choice. Lesson logged: put must-stops in code, not in the prompt.
Video research environment	The AI's shell cut commands off at about three minutes, and the video site refused captions to its address.	Local transcription as the fallback, in 10-minute chunks; full downloads hand off to my own machine by default.
Personal records misnamed files	Past property-tax receipts had been downloaded under the wrong file names.	Completeness is checked by reading the ID inside each PDF and counting against the registry. File names are not trusted.
Fact checks a missing source	For one run, the original article could not be retrieved.	Failed retrievals are logged with a retry route in a re-check queue; the report claims only what was found.

FALSIFICATION CHECK

The thesis: rules wherever the pattern is known, AI only where judgment is needed. It fails if a rule files an important mail out of sight, or if a rule-run step meets a case it needed judgment for. Checked so far: the routing, simulated on 638 messages before import (senders only, not subjects; medium-high confidence). Not yet checked: the one-week review of mail that skipped the inbox, due from 15 October 2026.

Role: design of all five systems and of the rules, gates and checks in them, built and run with an AI assistant. Counts only throughout: no names of people, contacts or clients, no addresses, no figures from the records themselves.