

How it works: Prompts → Skills + Projects → MCP

Three levels of doing the same job — keeping your policies & procedures compliant when the rules change

A regulation changes; your P&Ps have to change to match. Sometimes that's **one section in one P&P**. Sometimes it's a **firm-wide sweep across dozens of P&Ps** where you don't even know the full scope up front. Same kind of job — three levels of leverage. Pick the level that fits the size and messiness of the change.

Level 1 — Prompting ONE KNOWN CHANGE

Example: a loss-mitigation rule update.

You have a small CSV (or just the text) of the changed rule, and you already know which P&P it touches — your loss-mit P&P. You bring both into a chat and prompt:

"Here's the updated loss-mitigation rule. Here's my loss-mit P&P. Find the sections that reflect the old rule and rewrite them to comply with the new one."

Claude rewrites the affected sections. Done.

When it's enough: one rule, a target you can identify yourself, scope you can eyeball. Fast, zero setup.

Where it runs out: *you* do the targeting and the prompting every time; nothing is reused; if the rule text changes you re-bring it by hand; and it can't answer *"which of my 40 P&Ps does this even touch?"* For that, you need the next level.

Level 2 — Skill + Project FIRM-WIDE, MANY P&PS

Example: the Fannie Mae AI Lender Letter — the hard kind of update.

This one isn't "edit a section." It's a whole workflow:

1. **Scan all the P&Ps** to find where AI/ML touches the work each P&P governs.
2. **Decide where AI is actually in play** — and flag what the documents *don't* confirm (don't assume a "score" or a vendor name means AI).
3. **Assemble** every scan's results.
4. **Chase the missing info** — produce the list of *who to ask and what to ask*.
5. **Draft the master AI governance policy** from everything found.
6. **Re-draft every affected P&P** to align with that master policy.
7. **Handle the vendor-governance and change-log** pieces.

Two ingredients make this possible: the **Skill** and the **Project**.

The Skill = the whole workflow, packaged

The skill carries every staged prompt, the discipline rules (*cite the rule verbatim; never assert AI that isn't evidenced; stay within scope*), and the exact output format. **You don't write prompts** — you just say *"inventory*

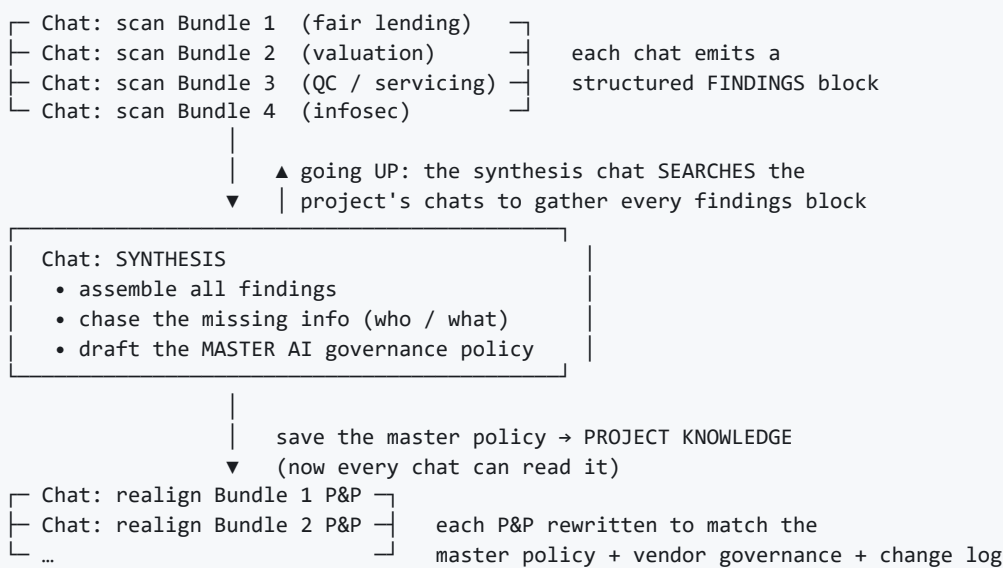
this bundle" or "draft the policy," and the skill runs the right stage the right way, the same way every time. Consistency and rigor come baked in, not improvised per chat.

The Project = shared memory + many chats that pool what they learn

A single chat can't hold a firm-wide job. The Project adds two things:

- **A shared knowledge base.** The regulation CSV lives in the project as a knowledge file, so *every* chat cites the same source — and when the rules change you **swap in a newer CSV to keep it from going stale**, with no reinstalling. (The skill always reads the *fresh*est copy it can find.)
- **Many focused chats that share their findings.** For a job this big you run **one chat per P&P bundle** so each stays focused — then they pool what they found.

Here's how the sharing actually works (this is the part worth getting right):



So it's **fan-out** → **fan-in** → **fan-out**: scan each bundle, pull them all together into one master policy, then push that policy back out so each P&P realigns to it.

The honest mechanics (so you're not surprised): the chats don't hold a live conversation. They share two ways — the synthesis chat **searches the other chats** to pull findings *up*, and the finished policy is **saved to Project knowledge** so the P&P chats can read it on the way back *down*. Cross-chat search can **lag** right after you run the scans, so the reliable habit is to also **paste each chat's FINDINGS block** into the synthesis chat. Same outcome, just don't rely on search alone.

When it's enough: complex, firm-wide changes where you don't know the full scope up front and you need consistency and memory across many documents.

Level 3 — MCP

ALWAYS CURRENT, AUTOMATIC

Level 2 has one soft spot: the regulation CSV is still a **file**. You can swap it to stay current — but *someone has to swap it*. When the rule corpus updates, a new CSV gets sent around and you drop it into the project.

MCP closes that gap. Instead of a file you swap, the regulatory corpus is delivered **into Claude as a live feed** (a connector). The skill/project pulls the rules from that feed, so when the source is updated, **every project is current on the very next run** — no file swap, no reinstall, no "is everyone on the latest version?"

When it's enough: ongoing operation where currency matters and you don't want to babysit updates. **The trade-off:** a connector has to be set up once (usually a quick IT step), where a CSV file does not.

Think of Levels 2 and 3 as two rungs of the *same* "stay current" goal: **swap-the-file** (manual, works anywhere, today) → **live-feed** (automatic, needs a one-time connect).

The ladder, at a glance

	Prompting	Skill + Project	+ MCP
Best for	one known change, one P&P	firm-wide change across many P&Ps	always-current, ongoing operation
You provide	the rule + the P&P, every time	your P&Ps (the skill + CSV do the rest)	your P&Ps
Finding what's affected	you pick the P&P	the scan finds where it touches	same
The workflow	you prompt each step	the skill runs every stage	same
Many-document memory	none	project knowledge + cross-chat findings	same
Keeping rules current	re-bring them each time	swap the CSV knowledge file	delivered live, automatically
Setup	none	install the skill / make a project	+ connect the feed (one-time, IT)

The throughline

It's the same job at every level — **make your P&Ps match the rules**. Reach for a **prompt** when it's a one-off you can target yourself. Reach for the **skill + project** when the change is big and messy and has to stay consistent across many documents. Add **MCP** when you want the rules themselves to stay current without anyone tending the file. You climb the ladder only as far as the job requires.