A Methodology for Leveraging AI Across Your Policies and Procedures

Use grounding, verification, and context engineering to bring your policies and procedures alive.

By Earl Cooper (with substantial help from the Claude application), as part of the work to better utilize AI to engage in compliance itself. Found at claudeforcompliance.com.

What grounding, verification, and context engineering are — and why they matter for leveraging AI across your Policies and Procedures

What they are, in summary

1. **Grounding** — the model works only from a verified, verbatim corpus of the regulation, never its memory.
2. **Verification** — every citation is checked against the source by exact match.
3. **Context engineering** — each chat writes its findings as a structured record and hands it off cleanly, so the work is inspectable and many chats across your policy set reconcile into one.

How they let you leverage AI across your P&Ps

Upload each policy and procedure to a chat, all in the same project. Grounding, verification, and context engineering provide the guardrails that enforce the regulatory requirements across the P&Ps in the chats and projects — and further let each chat (a P&P) communicate with the other chats (P&Ps) about its compliance with regulatory requirements.

(This paper is educational and is not legal advice.)

1. Grounding: work performed using actual regulatory text, never regulatory text from memory

What grounding is

Grounding mitigates against hallucinations. A hallucination, in technical terms, is fluent output that is not *grounded* in — not faithful to — a real source, which is a nonstarter for deploying in compliance. Grounding is a method the field has converged on: tie every output to a provided source of truth so answers stay inside the supplied evidence, not the model's memory — the core move of retrieval-augmented generation [1].

In this project, the AI is grounded to a regulatory corpus within the project. That corpus contains the exact regulatory text for mortgage compliance regulations, along with a unique identifier for each piece of regulatory text, which the AI retrieves at the start of its session to ground its work in the same. Verification — step 2 below — is the closing check, where we ask the AI model to recite back the regulatory text it is grounded in, and the citation IDs for any work performed.

Researchers measure grounding as **AIS — *Attributable to Identified Sources** *(Rashkin et al.): every statement about the world must be supported by an independent, provided source [2]. Producing answers* with *citations that can actually be checked is the task the literature calls verifiable generation** [3].

How this is used in AI and policies and procedures

Instead of uploading a PDF of the regulatory text, each regulation is converted into a structured, machine-readable corpus — a table, not prose. Regulatory text is grouped under a **register** (one regulation or source), and each row is a single obligation carrying:

- a **stable ID** (`obligation_id`) — a permanent handle for that one requirement, so a reference points to a specific, retrievable unit of text rather than a vague gesture at "the regulation";
- the **verbatim regulator text** (`paragraph_quote`) — the exact words of the rule, and the *only* thing the model is permitted to cite;
- a **type** that classifies the obligation's force — *do, don't, prohibition, refrain, requirement, notice*;
- **tags and a section label** — subject area and lifecycle markers (e.g., effective date);
- for an update, a **change classification** — `change_action` of *added / changed / removed*, plus what it `supersedes` — so a rule *change* is itself structured data the model lines up old-against-new, not something it has to infer;
- **provenance** — the citing authority, a `source_url`, a `snapshot_id`, and the date it was fetched.

The model is then instructed to work *only* from these rows: cite a rule only by quoting an actual row, verbatim, tagged with its ID.

For an example of this in practice, see the FHA Loss Mitigation update kit.

2. Verification: work output gated against the regulatory text

Grounding and verification are the two ends of a single guarantee. Grounding is the **front-end constraint** — work only from the verified source, and quote it. Verification is the **back-end check** — confirm, after the fact, that the quote actually matches. They act on the same object, the verbatim quote: grounding issues it as the model writes; a

script confirms it once written. A constraint you don't check is only a hope — which is why the method runs both, at opposite ends.

What verification is

The durable result in reliable-AI research is that a *separate* check on a model's output catches errors the generating pass misses — the foundational work trained a distinct verifier, generated candidate answers, kept the one the verifier ranked highest, and accuracy climbed sharply [4]. Drafting, then independently verifying, then revising is the Chain-of-Verification pattern, shown to measurably reduce hallucination across factual tasks [5].

How we do it

We use a string comparison of the quote from the AI chat against the regulatory corpus. Either it passes or it fails.

And when a claim *cannot* be grounded, the model takes the exit — "outside this kit" — rather than fabricate. An explicit abstention path is a known hallucination reducer, and it has a research name: **selective prediction**, or **abstention** — answer only where you are likely correct, decline otherwise [6].

Every change the model proposes is logged with its provision ID and the verbatim text that drove it — so the output is not just a revised document but an audit trail an examiner can follow, consistent with the documentation and traceability the NIST AI Risk Management Framework expects [7].

3. Context engineering: DMs for your policies and procedures

What context engineering is

Context engineering is the discipline of deciding what context a model carries between steps [8]. It helps address stray hallucinations in multi-stage work. The accuracy of multi-stage, multi-chat work benefits significantly when each chat saves its findings (a scratchpad [9]), and separate chats with clean context can access those findings (isolation).

How we use it — a scratchpad in every chat

Use a project folder from Claude/OpenAI/etc. and upload each of your P&Ps to its own chat. This method requires each chat that works on or analyzes the P&P to document its findings, including the regulations that were used in suggesting the change, and the text that was suggested to be changed. Specifically, each P&P under review ends by writing a **structured findings record**: a typed block, not a prose summary, with each AI system or finding tagged by a stable ID, an inherent- and residual-risk rating, and a named

owner for every open question. The chat writes out what it found rather than holding it in context and leaping to a conclusion. (Our "isolate" is one P&P per chat — see §4.)

What it allows

Two things. First, the record is **inspectable** — you see the reasoning, not just the verdict. Second, and more powerfully, it **hands off**: handing the other chats (and P&Ps) in the project *structured knowledge it can query* — not a raw transcript — is what keeps the hand-off from degrading. That hand-off is what turns one-chat discipline into whole-library coordination: because each chat leaves behind a structured record, dozens of isolated chats reconcile into one result — a single regulatory change reviewed across every affected P&P, and a master policy synthesized from what each chat found, without re-reading all of them. Grounding gives the model the right source; context engineering lets many grounded chats add up to one coordinated pass.

4. Build your own, test it out

A Project in Claude/OpenAI/Copilot is a cluster of chats that share a knowledge base. Set up a project, install a skill (this whitepaper includes a skill to update your P&Ps against Fannie Mae's upcoming AI Lender Letter) that enforces the regulatory disciplines, and **upload every policy and procedure as its own chat — one P&P per chat.**

Inside that Project, the three moves run at scale. Every chat **grounds** in the same verified corpus, so each enforces the *same* rule text against its P&P, not its own paraphrase. Every citation is **verified** the same way. And every chat **writes its findings** to a structured record built to hand off — the context-engineering move. Then a synthesis step pulls the findings together: when a regulatory update lands, you drop it into the Project, each chat analyzes whether its policy is affected and what must change, and the results reconcile into one coordinated review instead of forty separate errands.

This is also where the P&Ps can come *alive*. Once they are current, create a second shared Project with the operational people the policies govern — origination, servicing, marketing, vendor management. There, an operator can run an ad-hoc item — a borrower letter, a marketing piece, a vendor communication — through a chat for a fast, grounded first-draft compliance check against the current policy, *before* it goes out. The compliance officer sets the rules the Project enforces and reviews what it surfaces — the gate, never the bottleneck, never bypassed. The P&Ps have changed category: from a binder someone consults after a problem to **part of the methodology by which the firm stays compliant in the first place.** That is the arc the whole method is built toward: **P&Ps as liability → P&Ps as asset → P&Ps as methodology.**

5. The importance of Skills in Project mode

Grounding, verification, and context engineering are *disciplines*. A **Skill** is what makes every chat in the Project apply them identically — the same way in chat #1 and chat #40. Without it, the discipline lives in a prompt you type out or copy and paste.

The nature of skills allows the disciplines to be baked into the project, using the characteristics of skills below:

- **Operating rules — the non-negotiables.** Cite verbatim by ID and never assert a fact the document doesn't support (**grounding**); apply the substring check and the "outside this kit" exit (**verification**); end every run with the exact findings block (**context engineering**). Same rules, every document.
- **`assets/` — the verified rule kit.** The grounded corpus itself (a CSV), shipped inside the Skill so every chat reads the same source.
- **The output contract.** The exact findings block every run must end with, so many chats produce results that merge cleanly — the scratchpad, standardized.

Accordingly, the Project-and-Skill combo enforces grounding, verification, and context engineering, so it isn't left up to copy-and-pasting perfect prompting. The regulations live within the skill, so the skill enforces the regulations across the project, and the skill requires that findings are saved in the correct format — allowing policies and procedures to communicate with each other while mitigating hallucination.

A worked example: Fannie Mae's AI Lender Letter (LL-2026-04)

The method is not theoretical. Fannie Mae's Lender Letter LL-2026-04, published April 8, 2026 and effective August 6, 2026 (120 days later), requires every seller/servicer to govern its use of AI and ML, name an owner, and be ready to show its work [10]. It is a textbook "broad" change: the governing policy is one new document, but AI use shows up in valuation, underwriting, pricing, fraud screening, servicing, document automation, and marketing — so many existing P&Ps need oversight language too, and nobody knows the full scope up front.

Run through the method, that job becomes five coordinated moves inside one Project: **scan every P&P** to inventory where AI lives, **reconcile** that scan against the firm's vendor list and app inventory to catch the AI the documents can't show, **synthesize** the master AI governance policy and risk assessment, **realign** each affected P&P to it, and produce the **change log** an examiner expects. Grounding holds each chat to the verbatim rule; context engineering hands every chat's findings to the synthesis step; verification keeps every citation checkable — and the Skill enforces all three across every chat.

A free, build-it-yourself version of exactly this — the installable Skill, the verified rule kit, and the step-by-step workflow — is available at complianceforclaude.com/fnma-ai-kit. It is the methodology in this paper, made concrete for one regulation.

Closing

The technology question — "can AI help with compliance?" — is the wrong question, because it invites a yes/no answer about a tool. The better question is about *method*: what is the disciplined, defensible way to apply a document-reading machine to a document-heavy obligation?

The answer is the three moves: **ground** the model in a verified corpus, **verify** every citation against the source, and **engineer the context** so each chat shows its work and findings hand off cleanly — then compose them across the whole policy set with Projects and Skills, with the compliance professional as the decision-maker throughout. Do that, and your policies and procedures stop being a cost center you maintain and become an instrument you operate — one that gets more valuable every time the rules change.

And the shape is bigger than this one task. Strip the method to essentials and it is exactly those three moves: **turn the controlling authority into a verified, machine-readable corpus; point AI at it under strict grounding; keep yourself the judge.** That shape fits anywhere a rule meets a document — screening a vendor contract against your own policy, pressure-testing a marketing piece before it ships, triaging a complaint against the regulation it implicates, standing up an exam-readiness file the week a new rule drops. The Fannie letter is just the worked example; the method is the reusable part. If you can name the authority and name the document, you can run this — and the best uses will be the ones you find in your own shop.

References

1. Patrick Lewis et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Advances in Neural Information Processing Systems 33 (NeurIPS 2020), pp. 9459–9474. https://arxiv.org/abs/2005.11401

2. Hannah Rashkin et al. *Measuring Attribution in Natural Language Generation Models*. Computational Linguistics 49:4 (2023), pp. 777–840; arXiv:2112.12870 — defines **AIS (Attributable to Identified Sources)**: model output about the world must be verifiable against an independent, provided source. https://arxiv.org/abs/2112.12870

3. Tianyu Gao et al. *Enabling Large Language Models to Generate Text with Citations*. EMNLP 2023; arXiv:2305.14627 — the **verifiable generation** task and the ALCE

benchmark, scoring answers on citation quality (whether each claim is actually supported by its cited source). https://arxiv.org/abs/2305.14627

4. Karl Cobbe et al. *Training Verifiers to Solve Math Word Problems*. arXiv:2110.14168 (2021) — a separate verifier that judges a model's candidate outputs raises accuracy beyond what the generating pass achieves alone. https://arxiv.org/abs/2110.14168

5. Shehzaad Dhuliawala et al. *Chain-of-Verification Reduces Hallucination in Large Language Models*. Findings of the ACL (2024); arXiv:2309.11495 (2023). https://arxiv.org/abs/2309.11495

6. Amita Kamath, Robin Jia, and Percy Liang. *Selective Question Answering under Domain Shift*. ACL 2020; arXiv:2006.09462 — **selective prediction / abstention**: a model that declines to answer on inputs where it is likely to err maintains accuracy better than one forced to guess. https://arxiv.org/abs/2006.09462

7. National Institute of Standards and Technology. *AI Risk Management Framework (AI RMF 1.0)* and *Generative AI Profile (NIST AI 600-1)*. https://www.nist.gov/itl/ai-risk-management-framework

8. LangChain. *Context Engineering for Agents* (Jul. 2, 2025) — the discipline of choosing what context to carry between agent steps; its four core strategies are **write, select, compress, and isolate** (this method uses *write* — save state outside the context window — and *isolate* — split work across separate entities). The post also surveys the failure modes of overloaded context (poisoning, distraction, confusion, clash), a taxonomy credited to Drew Breunig, *How Contexts Fail, and How to Fix Them* (Jun. 22, 2025). https://www.langchain.com/blog/context-engineering-for-agents · https://www.dbreunig.com/2025/06/22/how-contexts-fail-and-how-to-fix-them.html

9. Maxwell Nye et al. *Show Your Work: Scratchpads for Intermediate Computation with Language Models*. arXiv:2112.00114 (2021) — models perform multi-step tasks markedly better when they write intermediate steps to an explicit scratchpad rather than answering in one pass. https://arxiv.org/abs/2112.00114

10. Fannie Mae. *Lender Letter LL-2026-04: Governance framework on use of artificial intelligence and machine learning* (Apr. 8, 2026; effective Aug. 6, 2026). https://singlefamily.fanniemae.com/news-events/lender-letter-ll-2026-04-governance-framework-use-artificial-intelligence-and-machine-learning

Compliance for Claude builds tools and methods for mortgage compliance professionals using AI. This white paper is educational and does not constitute legal advice. Independent; not affiliated with or endorsed by Fannie Mae, Anthropic, NIST,

or any regulator. External references are provided for authority and context; citation does not imply endorsement by their authors.