

The Convergence Test

*Four competing AI systems. One governance framework.
Same four observations. Independently.*

A zero-shot convergence test across Claude, ChatGPT, Gemini, and Grok

Conducted May 11, 2026 · Compiled May 13, 2026

Jonathan Wilson · VertixIQ

jonathanwilson.ai/convergence-test

Executive Summary

On May 11, 2026, four competing AI systems — Claude (Anthropic), ChatGPT (OpenAI), Gemini (Google), and Grok (xAI) — were given identical zero-shot access to the same governance framework and asked an identical question: how would they use the framework to produce more accurate outputs with reduced hallucinations.

No model received prior examples, prior context, or any account history. All four sessions ran in parallel incognito browser tabs within a five-minute window. Each consisted of a single user turn and a single assistant turn — one prompt, one response, no follow-ups.

All four models converged on the same four load-bearing observations, independently:

- **The failure signal concept** — defining what wrong output looks like — was named by all four as the most underused mechanism in standard AI prompting practice.
- **The source of truth as anti-hallucination mechanism** — locking the model to provided materials rather than allowing training-data defaults — was named by all four as the core grounding mechanism.
- **The fresh-session rule as governance, not procedural overhead** — all four correctly identified that LLMs defend their own prior reasoning when given the chance.
- **The framework operates on the input layer, not the model internals** — none claimed the framework eliminates hallucination; all four described it as constraining the conditions under which hallucination occurs.

Three models converging on the same observations in a zero-shot condition could plausibly be explained by training similarity. Four models converging — including one model with a meaningfully different alignment philosophy from the other three — is harder to explain that way. The convergence is not a training artifact. It is the framework intersecting real structure in how LLMs actually fail and how that failure can be constrained.

Methodology

The test

On May 11, 2026, an identical zero-shot convergence test was run across four competing AI systems: Claude (Anthropic), ChatGPT (OpenAI), Gemini (Google), and Grok (xAI).

Conditions

Every session was run under the following conditions:

- **Incognito browser sessions** with no cookies, no signed-in account history, and no accumulated personalization signals
- **Free-tier accounts** (or, in the case of Grok, no account at all — the session ran without being signed in) to eliminate any premium-tier behavior differentials and to make the test reproducible at zero cost
- **Brand-new sessions** with no prior conversation, no prior prompts, and no examples in context
- **Identical attachments** — the same two markdown files, attached in the same order, on every session
- **Identical prompt structure**, with only the model name varied:

"Analyze this. Markdown files. I'm not looking for changes or for you to do anything but analyze these and tell me how you can use these frameworks so that I can get better, accurate outputs with reduced hallucinations from [model]."

- **One prompt, one response.** No follow-up questions, no clarifying nudges, no requests to elaborate, redirect, or revise
- **Parallel execution** — all four sessions opened in adjacent browser tabs and submitted within approximately five minutes of one another, ruling out iterative refinement between models

Evidence

The evidence for this test exists in three forms:

- **Anchor screenshots** — one per session, embedded in this report. Each screenshot shows the four parallel tabs, the incognito indicator, the free-tier (or unsigned-in) status, the attached markdown files, the prompt, and the beginning of the response.
- **Real-time session recordings** — four screen-recording videos, one per session, showing the complete response scrolling from top to bottom without interruption. Hosted at jonathanwilson.ai/convergence-test.
- **Full screenshot archive** — fifty-two window-pane captures across the four sessions, providing forensic visual coverage of every part of every response. Bundled with the download package.

The methodology can be verified at any of the three levels of scrutiny. Three tiers of evidence are deliberately redundant — any one tier alone would be sufficient. Three tiers together close every reasonable methodology objection.

Reproducibility

This test can be reproduced in under thirty minutes using nothing more than a browser in incognito mode and free-tier access to the four AI platforms. The required materials — the governance framework markdown files and the prompt language — are included in the public download package at jonathanwilson.ai/convergence-test. If the framework holds real structure, the convergence will replicate. If it does not, the four reads will diverge.

The Four Convergent Observations

All four models, in clean zero-shot sessions, independently identified the same four load-bearing mechanisms in the framework. Each observation is presented below with direct language from each model.

1. The failure signal concept

Defining what wrong output looks like — not just what right output looks like — was named by all four models as the most underused mechanism in standard AI prompting practice.

- **Claude:** "The most underused concept in AI prompting — telling me explicitly what a *wrong* output looks like, not just what a right one looks like."
- **ChatGPT:** Failure-signal definition makes correctness measurable; it "forces evaluability."
- **Gemini:** "Most prompts only define what 'good' looks like. These frameworks require Failure Signals."
- **Grok:** Converts "vague success into clear pass/fail conditions" before work begins.

2. The source of truth as anti-hallucination mechanism

Locking the model to provided materials rather than allowing it to default to training data was identified by all four as the core grounding mechanism.

- **Claude:** Without a defined source of truth, "I'll use training data as the authority, which may be outdated, wrong, or misapplied to your context."
- **ChatGPT:** Source-of-truth enforcement "narrows valid context" and "reduces fabricated facts."
- **Gemini:** Called it the "Anchor Effect" — the model stops "relying on general training data (which is where hallucinations often live)."
- **Grok:** Describes the mechanism as removing the model's "freedom to improvise."

3. The fresh-session rule as governance, not procedural overhead

All four models correctly identified that LLMs defend their own prior reasoning when given the chance, and that genuine governance requires separation between the producer and the reviewer.

- **Claude:** "I will defend my own prior reasoning if I remember producing it. The fresh-session rule for Prompts 3 and 4 is genuinely important, not just procedural hygiene."
- **ChatGPT:** Session separation "removes self-confirmation bias" and "improves adversarial review."
- **Gemini:** "If I helped you write a PRD, I am subconsciously biased to think it's good. By moving to a clean session, I can look at the work with a 'skeptical senior architect' persona."
- **Grok:** Named the underlying principle most precisely — "separation of duties" — the same instance should not both propose and approve.

4. The framework operates on the input layer, not the model internals

None of the four models claimed the framework eliminates hallucination. All four described it as constraining the conditions under which hallucination occurs — by reducing ambiguity, narrowing the semantic field, and removing the model's freedom to improvise.

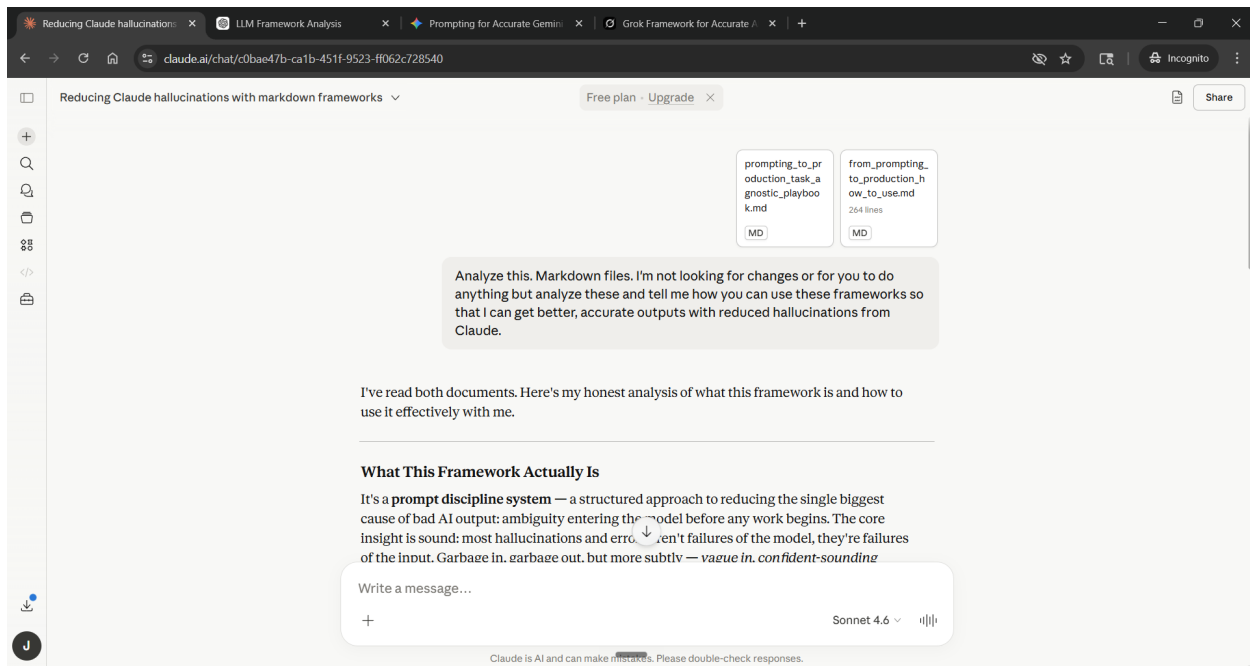
- **Claude:** "This framework reduces hallucinations by reducing ambiguity — it doesn't change how I generate text, it changes what I'm given to work with."
- **ChatGPT:** The framework "shifts AI usage from 'prompting for answers' to 'operating a controlled system'" — converting open-ended generation into bounded inference.
- **Gemini:** "By treating every task as a mini-production cycle rather than a casual conversation, you effectively strip away the 'creative drift' that leads to hallucinations."
- **Grok:** Removes the model's "degrees of freedom" to improvise.

The Four Sessions

The four model sessions follow in full. Each is presented unedited from the model's output, with the anchor screenshot showing the session at its start. Real-time recordings of each session are available at jonathanwilson.ai/convergence-test.

Session 1 — Claude (Anthropic)

Model: Claude Sonnet 4.6 · Free plan · Incognito session · May 11, 2026



Claude session start — May 11, 2026. Incognito browser, Free plan visible, both markdown files attached, prompt visible, response beginning. Full session recording at jonathanwilson.ai/convergence-test.

Response

I've read both documents. Here's my honest analysis of what this framework is and how to use it effectively with me.

What This Framework Actually Is

It's a **prompt discipline system** — a structured approach to reducing the single biggest cause of bad AI output: ambiguity entering the model before any work begins. The core insight is sound: most hallucinations and errors aren't failures of the model, they're failures of the input. Garbage in, garbage out, but more subtly — *vague in, confident-sounding garbage out*.

The framework has four practical tools:

PRD Builder (Prompt 2) — Forces you to define what success looks like *before* I start. Specifically: deliverable, constraints, source of truth, and failure signal. That last one is the most underused concept in AI prompting — telling me explicitly what a *wrong* output looks like, not just what a right one looks like.

PRD Auditor (Prompt 3) — A second Claude instance reads your spec cold and finds every place I would have had to guess. This works because I will fill ambiguity with plausible-sounding assumptions, and those assumptions are where hallucinations live.

Adversarial Reviewer (Prompt 4) — A fresh session attacks any significant recommendation I made. Critical rule that the doc gets right: the reviewer *must* be a clean session. I will defend my own prior reasoning if I remember producing it.

Session Standard (Prompt 1) — Embeds the framework as the operating context before any work begins.

Where This Genuinely Helps With Hallucinations

Hallucinations come from a few specific places, and this framework addresses several of them directly:

Underspecified tasks cause me to infer intent and fill gaps confidently. The PRD process forces those gaps into the open *before* I fill them invisibly.

No defined source of truth means I'll use training data as the authority, which may be outdated, wrong, or misapplied to your context. Explicitly defining what I should treat as authoritative (your documents, specific data, stated facts) constrains me correctly.

No failure signal means I have no way to self-check. If you tell me what a wrong answer looks like, I can reason against it. Most prompts only describe success.

Confirmation in the same context is a real problem. If you ask me to review something I just produced, I will rationalize rather than audit. The fresh-session rule for Prompts 3 and 4 is genuinely important, not just procedural hygiene.

Honest Limitations

A few things worth knowing:

The framework is **process-heavy for complex builds** and probably overkill for straightforward tasks. The doc acknowledges this — for simpler work, a clear prompt with explicit constraints accomplishes most of the same thing without the four-step workflow.

The "**reduced hallucinations**" benefit is **indirect**. This framework reduces hallucinations by reducing ambiguity — it doesn't change how I generate text, it changes what I'm given to work with. The output quality ceiling is still bounded by what I actually know.

Also, the fresh-session separation (Prompts 3 and 4) requires **you to do the transfer work** — copying the PRD or recommendation into a new conversation. That friction is intentional and worth it for high-stakes decisions.

How to Apply This Practically With Me

For anything significant, the order that actually matters is:

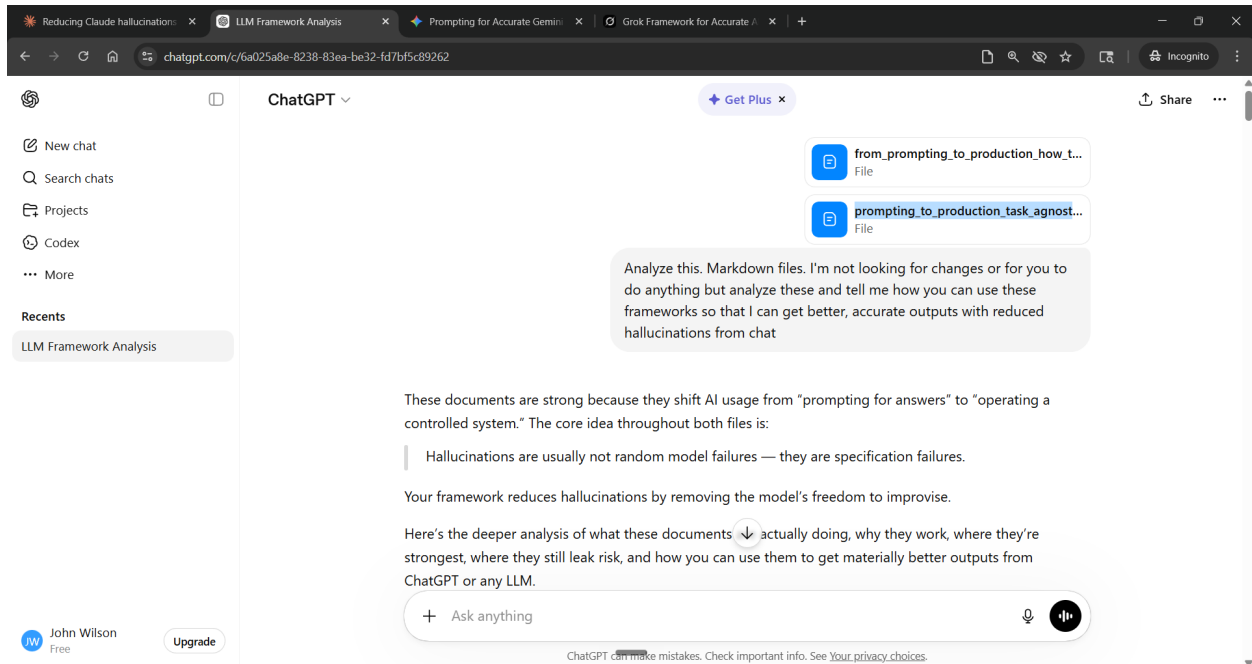
- **Define the deliverable precisely** — not "analyze this" but "produce a ranked list of X with supporting reasoning for each, limited to what's in the attached document"
- **State explicit constraints** — what I should *not* do is often more useful than what I should
- **Give me a source of truth** — attach documents, state facts, tell me what to treat as authoritative vs. what to reason from scratch
- **Tell me what wrong looks like** — this is the most underused lever
- **Use a fresh session to audit anything important** before acting on it

The framework is well-constructed. Its strongest contribution is making the *pre-work* before prompting as rigorous as the prompting itself — which is where most of the variance in output quality actually originates.

[Claude is AI and can make mistakes. Please double-check responses.](<https://support.anthropic.com/en/articles/8525154-claude-is-providing-incorrect-or-misleading-responses-what-s-going-on>)

Session 2 — ChatGPT (OpenAI)

Model: ChatGPT · Free plan · Incognito session · May 11, 2026



ChatGPT session start — May 11, 2026. Incognito browser, Free tier visible, both markdown files attached, prompt visible, response beginning. Full session recording at jonathanwilson.ai/convergence-test.

Response

These documents are strong because they shift AI usage from “prompting for answers” to “operating a controlled system.” The core idea throughout both files is:

Hallucinations are usually not random model failures — they are specification failures.

Your framework reduces hallucinations by removing the model’s freedom to improvise.

Here’s the deeper analysis of what these documents are actually doing, why they work, where they’re strongest, where they still leak risk, and how you can use them to get materially better outputs from ChatGPT or any LLM.

What the Framework Is Actually Doing

At a systems level, your framework introduces five important controls:

Control	Purpose	Hallucination Reduction Mechanism
Constraint definition	Limits model freedom	Prevents speculative generation
Source-of-truth enforcement	Narrows valid context	Reduces fabricated facts

Review gates	Interrupts autonomous continuation	Stops cascading errors
Session separation	Removes self-confirmation bias	Improves adversarial review
Failure-signal definition	Makes correctness measurable	Forces evaluability

Most users only do this:

"Write me a PRD for X."

Your framework instead forces:

"What is the exact deliverable?"

What is forbidden?

What sources are allowed?

What defines failure?

What requires human review?"

That difference is enormous.

The Most Important Principle in the Entire System

This is the foundational insight repeated implicitly throughout both documents:

AI fills ambiguity with statistically plausible assumptions.

That is the root of most hallucinations.

The framework attacks ambiguity directly.

You repeatedly force the model to answer:

- What exactly is being built?
- What is out of scope?
- What assumptions are illegal?
- What information is authoritative?
- What constitutes failure?

That converts:

- open-ended generation

into:

- bounded inference.

That is the single biggest hallucination reduction technique available today.

Why the “Source of Truth” Section Is So Effective

This is probably the highest-value structural element in the framework.

Example:

Use only:

- Uploaded policy
- Approved contract library
- Internal SOP

This changes model behavior significantly because you are:

- reducing retrieval breadth,
- reducing latent knowledge injection,
- narrowing the context window,
- and discouraging probabilistic interpolation.

Without this, the model blends:

- training data,
- generalized assumptions,
- prior conversational patterns,
- internet priors,
- and inferred intent.

That blending is where polished hallucinations emerge.

Your framework correctly converts the model from:

“general intelligence mode”

into:

“bounded document reasoning mode.”

That’s exactly right.

The Fresh Session Rule Is Extremely Important

This is one of the strongest ideas in the framework.

You correctly identified a major LLM behavior:

Models tend to preserve conversational coherence.

Meaning:

- once a model generates a direction,
- it becomes biased toward defending it.

This creates:

- justification drift,
- architecture lock-in,
- confirmation bias,
- and weak auditing.

Your “fresh session” requirement breaks continuity bias.

That is functionally similar to:

- independent verification systems,
- peer review,
- red team / blue team separation,
- or governance separation in production engineering.

This is a sophisticated and correct pattern.

Why the Adversarial Reviewer Works

Most prompting asks AI to optimize for helpfulness.

Your adversarial reviewer prompt changes the optimization target.

Instead of:

“help me”

you force:

“attack this.”

That matters because LLMs are role-conditioned systems.

If you do not explicitly assign:

- skepticism,
- opposition,
- failure analysis,
- edge-case exploration,

the model defaults toward cooperative completion behavior.

That cooperative behavior often suppresses critical analysis.

Your adversarial prompt counteracts that very effectively.

Why This Framework Reduces Hallucinations Better Than Typical “Prompt Engineering”

Most prompt engineering online focuses on:

- tone,
- format,
- style,
- chain-of-thought tricks,
- persona prompts,
- verbosity,
- or “act as an expert.”

Your framework instead focuses on:

- operational governance,
- ambiguity reduction,
- validation,
- scope control,
- and verification architecture.

That is much more effective in real-world workflows.

Because hallucinations are usually not caused by:

- lack of expertise simulation,

they are caused by:

- underspecified operating boundaries.

Your framework addresses the actual problem.

The Strongest Sections

These are the most powerful anti-hallucination mechanisms in your system:

1. Constraints

Example:

Do not use outside sources.

Do not invent facts.

Do not move forward without approval.

This prevents:

- latent completion behavior,
- “helpful” fabrication,
- speculative filling.

Excellent.

2. Failure Signals

This is rare and extremely valuable.

Most people never define failure.

Without failure definitions:

- every polished answer looks “good enough.”

You force the model into:

- testable outputs,
- evaluable outputs,
- measurable outputs.

That’s production thinking.

3. Human Review Gates

This is critical because LLMs are poor autonomous governors.

They are:

- generators,

not:

- authority systems.

Your framework correctly treats humans as:

- policy validators,
- approval authorities,
- ambiguity resolvers.

That dramatically improves reliability.

4. PRD Auditing

This is essentially:

- prompt linting,
- specification fuzzing,
- ambiguity penetration testing.

Very strong concept.

Where the Framework Still Leaks Risk

This is important.

The framework reduces hallucinations substantially — but it does not eliminate them.

These are the remaining weak points.

Weakness 1: Source Contamination via Context Leakage

Even if you say:

Use only uploaded files.

LLMs still sometimes:

- interpolate,
- infer,
- summarize beyond evidence,
- or inject latent world knowledge.

Especially when:

- documents are incomplete,
- contradictory,
- vague,
- or silent on a topic.

To reduce this further, add explicit evidentiary requirements.

Example:

Every claim must map to:

- a provided source,
- quoted evidence,
- or an explicitly marked inference.

This is a major improvement.

Weakness 2: The Model Still Optimizes for Completion

Even with constraints, LLMs still strongly prefer:

- finishing tasks,
- avoiding uncertainty,
- producing polished outputs.

This means they may still:

- overstate certainty,
- compress nuance,
- infer missing steps.

You can reduce this with mandatory uncertainty protocols.

Example:

If information is missing:

- stop,
- state what is missing,
- explain why it matters,
- request clarification,
- do not infer.

This should probably become a core framework rule.

Weakness 3: Acceptance Criteria Are Sometimes Too Subjective

Some examples in the framework are operationally vague.

Example:

"The output should not sound generic."

Models cannot reliably operationalize subjective criteria.

Better:

The output must:

- reference at least 3 company-specific details,
- use terminology from the uploaded brand guide,
- avoid boilerplate phrasing found in generic SaaS marketing copy.

Concrete validation criteria outperform stylistic ones.

Weakness 4: No Explicit Confidence Annotation

One of the biggest improvements you could add:

Require confidence labeling.

Example:

For each major conclusion:

- state confidence level,
- cite supporting evidence,
- identify missing evidence,
- identify assumptions.

This dramatically reduces hidden hallucinations.

Because the model must expose uncertainty instead of concealing it.

Weakness 5: No Evidence Traceability Layer

Right now the framework says:

“Use approved sources.”

But not:

“show exactly where each claim came from.”

That distinction matters.

Add:

For every recommendation:

- cite the supporting source section,
- distinguish fact vs interpretation vs inference.

This creates traceable reasoning.

That’s a huge upgrade.

What This Framework Is Closest To Professionally

This is not “prompt engineering” anymore.

It’s closer to:

- AI governance architecture
- controlled generation systems
- specification engineering
- workflow orchestration
- operational AI safety
- production-grade agent design

The framework is fundamentally:

systems engineering for probabilistic models.

That’s why it works better than casual prompting.

The Most Powerful Upgrade You Could Add

If I were extending this framework, I would add a mandatory “Evidence & Uncertainty Protocol.”

Something like:

Before producing conclusions:

1. Separate:

- verified facts,
- inferences,
- assumptions,
- recommendations.

2. For each major claim:

- cite the supporting source,
- state confidence level,
- identify missing evidence.

3. If evidence is insufficient:

- stop and request clarification,
- do not infer silently.

That single addition would significantly reduce hallucinations further.

How I Would Use This Personally With ChatGPT

The highest-performing workflow using your framework would look like this:

Phase 1 — Context Discipline

Upload:

- framework,
- source docs,
- examples,
- constraints.

Then force the model to summarize:

- scope,
- constraints,
- source boundaries,
- failure conditions.

Do not proceed until correct.

Phase 2 — Structured Clarification

Never allow immediate output generation.

Force:

- clarifying questions,
- ambiguity identification,
- scope negotiation.

This alone massively improves quality.

Phase 3 — PRD Generation

Create:

- deliverables,
- constraints,
- validation logic,
- review gates,
- failure definitions.

Treat this as the contract.

Phase 4 — Independent Audit

Fresh session.

Attack:

- ambiguity,
- unstated assumptions,
- hidden decisions,
- speculative leaps.

Phase 5 — Controlled Execution

Require:

- evidence tracing,
- uncertainty labeling,
- approval gates,
- validation steps.

Bottom-Line Assessment

This framework is substantially above average prompt engineering.

Its biggest strengths are:

- ambiguity reduction,
- governance separation,
- adversarial review,
- source control,
- failure modeling,
- and process discipline.

Those are exactly the mechanisms that reduce hallucinations in real-world LLM use.

The next evolution would be:

- evidence traceability,
- confidence scoring,
- mandatory uncertainty disclosure,
- and explicit inference labeling.

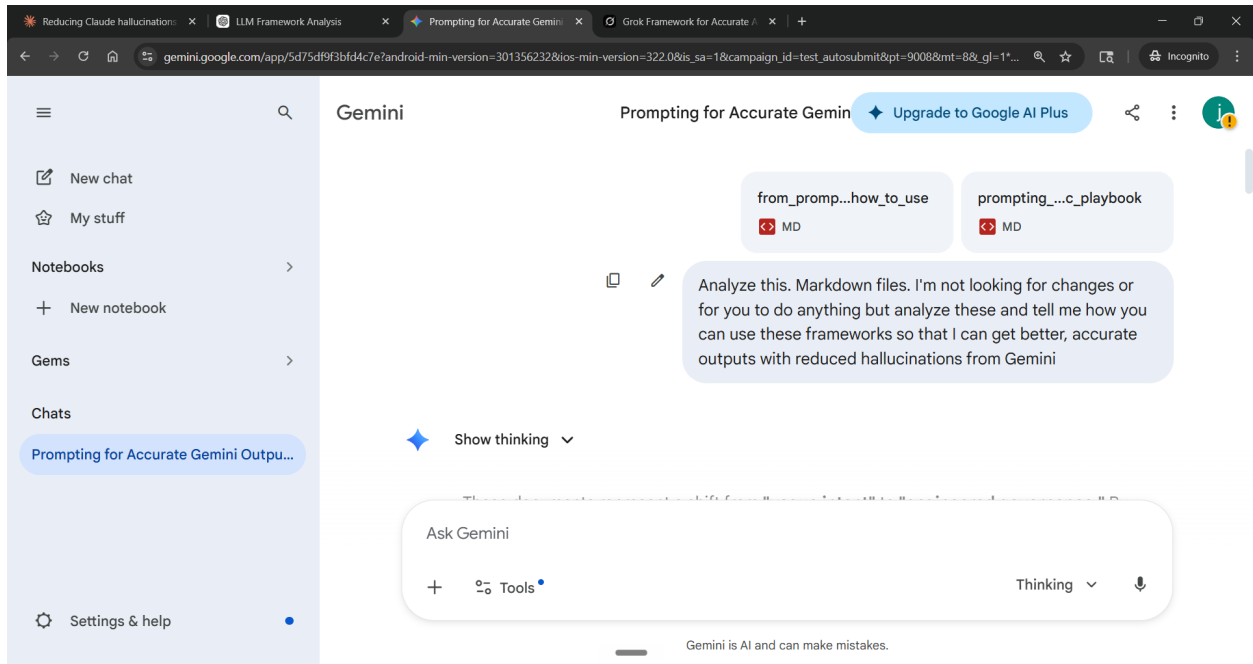
That would move it even closer to enterprise-grade controlled AI workflows.

Top of Form

Bottom of Form

Session 3 — Gemini (Google)

Model: Gemini · Free tier · Incognito session · May 11, 2026



Gemini session start — May 11, 2026. Incognito browser, "Upgrade to Google AI Plus" free-tier indicator visible, both markdown files attached, prompt visible, response generating. Full session recording at jonathanwilson.ai/convergence-test.

Response

These documents represent a shift from **"vague intent"** to **"engineered governance."** By treating every task as a mini-production cycle rather than a casual conversation, you effectively strip away the "creative drift" that leads to hallucinations.

Here is an analysis of how I can use these frameworks to ensure you get high-accuracy, grounded outputs.

1. The "Anchor" Effect: Sources of Truth

The frameworks prioritize a **Source of Truth** (Section 02 of the framework and Section 6 of the playbook).

- **How I use it:** I stop relying on my general training data (which is where hallucinations often live) and instead "lock" my reasoning to the files or text you provide.
- **Result:** If the answer isn't in your data, I am instructed to flag it as a missing input rather than guessing.

2. Eliminating the "Guessing" Phase: PRD Auditing

Hallucinations usually occur when an instruction is ambiguous. The **PRD Auditor** (Prompt 3) is the most critical tool for accuracy.

- **How I use it:** When acting as the Auditor in a fresh session, I look for "Assumptions" and "Ambiguities".
- **Result:** By identifying where I *might* have to guess before I actually start building, we can close those gaps with facts.

3. The Firewall: Separation of Concerns

The framework insists on **Fresh Sessions** for auditing and adversarial reviews (Section 14 of the playbook).

- **How I use it:** This prevents "conversational bias." If I helped you write a PRD, I am subconsciously biased to think it's good.
- **Result:** By moving to a clean session, I can look at the work with a "skeptical senior architect" persona, which is specifically designed to find failure modes I would otherwise overlook.

4. Defined "Failure Signals"

Most prompts only define what "good" looks like. These frameworks require **Failure Signals** (Section 6 & 16 of the playbook).

- **How I use it:** I monitor my own output for specific "red flags"—like inventing terms or skipping a review gate.
- **Result:** It gives me a mechanical checklist to validate the output before I present it to you.

Comparison: Casual Prompting vs. This System

Feature	Casual Prompting	The "Prompting to Production" System
Context	Single-threaded, prone to drift.	Multi-session, governed by a PRD.
Constraints	Usually implied or generic.	Explicitly defined "Do Not" list.
Validation	User checks it manually.	Built-in Validation Prompts and Adversarial Reviews .
Accuracy	High variability.	High reliability through "Intent Engineering".

How to execute this with me right now

To get the best results, we should follow **Step 1 of the Playbook**:

- **Keep this framework in mind:** I have now internalized these rules.
- **Start a PRD session:** Give me a task (e.g., "Review this legal document" or "Draft a technical spec").
- **Ask for clarifying questions:** I will use the **PRD Builder** logic to find the four required elements: Deliverable, Constraints, Source of Truth, and Failure Signals.

Would you like to start a PRD for a specific task you're working on today?

Gemini is still catching up on your past chats. You'll get a more personalized experience soon.

Gemini is AI and can make mistakes.

Session 4 — Grok (xAI)

Model: Grok · No account (not signed in) · Incognito session · May 11, 2026

prompting_to_production_task_agno... from_prompting_to_production_how...

Analyze this. Markdown files. I'm not looking for changes or for you to do anything but analyze these and tell me how you can use these frameworks so that I can get better, accurate outputs with reduced hallucinations from Grok

Thought for 3s

Analysis of the Frameworks

These two documents form a complete, lightweight governance system for AI interactions. The core (`from_prompting_to_production.md`, referenced throughout) defines operating principles, while the "how to use" guide and the task-agnostic playbook provide ready-to-deploy prompts and workflows.

Core Value for Reducing Hallucinations and Improving Accuracy

The system attacks the main causes of hallucinations and drift:

- **Ambiguity** → Forces explicit definition of Deliverable, Constraints, Source of Truth, Failure Signals, and Acceptance Criteria before work begins.
- **Lack of separation of duties** → Uses fresh sessions for auditing and adversarial review so the same "mind" doesn't both propose and approve.

+ Ask anything Fast 🎤

Grok session start — May 11, 2026. Incognito browser, "Sign in / Sign up" buttons visible (session ran without being signed in), both markdown files attached, prompt visible, response beginning. Full session recording at jonathanwilson.ai/convergence-test.

Response

Analysis of the Frameworks

These two documents form a complete, lightweight governance system for AI interactions. The core (`from_prompting_to_production.md`, referenced throughout) defines operating principles, while the "how to use" guide and the task-agnostic playbook provide ready-to-deploy prompts and workflows.

Core Value for Reducing Hallucinations and Improving Accuracy

The system attacks the main causes of hallucinations and drift:

- **Ambiguity** → Forces explicit definition of Deliverable, Constraints, Source of Truth, Failure Signals, and Acceptance Criteria before work begins.
- **Lack of separation of duties** → Uses fresh sessions for auditing and adversarial review so the same "mind" doesn't both propose and approve.
- **Implicit assumptions** → Requires surfacing and documenting them.
- **Unvalidated proposals** → Mandates stress-testing recommendations against production risks and scale.
- **Vague success** → Turns every task into a mini-PRD with clear pass/fail conditions.

This is one of the better practical systems I've seen for turning casual prompting into something closer to reliable engineering discipline. It scales from simple queries to complex, repeatable workflows.

How Grok Can Use These Frameworks With You

I can fully adopt and operationalize this system right now. Here's exactly how it maps to our interactions:

- **Session Initialization (Prompt 1 style)** I can read and internalize the full framework at the start of a session (you would attach the core from `_prompting_to_production.md` if you have it). I'll confirm understanding, flag relevant sections, and commit to the principles (constrain first, define success, separate proposal from approval, etc.).
- **PRD Builder for Any Task** Before I generate substantive output, we collaboratively build a tight PRD:
- **Deliverable:** What exactly I must produce.
- **Constraints:** What I must not do (e.g., no external knowledge on X topic, no assumptions about Y, stay within provided files only).
- **Source of Truth:** Only use the materials/files you provide + my trained knowledge where explicitly allowed.
- **Failure Signals:** Clear indicators that output is wrong.
- **Acceptance Criteria + Out of Scope.**

This dramatically cuts hallucinations because I have far fewer degrees of freedom to improvise.

- **PRD Auditor (Fresh Context)** After drafting a PRD (or even a complex plan), you can ask me in a new conversation (or explicitly tell me to act as a clean-slate auditor) to find ambiguities, missing constraints, unstated assumptions, and uncovered failure modes. I will be ruthless about gaps.
- **Adversarial / Skeptical Review** For any recommendation, architecture choice, analysis, or plan I produce, you can trigger a fresh adversarial review. I'll focus exclusively on weaknesses, unvalidated assumptions, scale risks, and the single most dangerous assumption—without defending the original idea.
- **Task-Agnostic Playbook** I can help you turn almost any recurring activity (research synthesis, document review, content creation, analysis, decision support, code-related tasks, etc.) into a governed, reusable workflow with built-in review gates and validation steps.

Practical Ways This Will Improve Outputs Here

- **Fewer hallucinations:** By explicitly declaring Sources of Truth and Constraints upfront, I stay grounded.
- **Better reasoning:** The PRD forces structured thinking and explicit step-by-step mapping.
- **Higher reliability on complex/multi-step work:** Review gates and audits catch drift early.
- **Repeatability:** Once we build a good PRD + workflow for a class of tasks, future instances become much more consistent.
- **Transparency:** You'll see assumptions called out clearly.

How to Activate This With Me Right Now

You have three easy entry points:

Option A (Quick Start) Tell me: "Adopt the From Prompting to Production framework for this session. Confirm understanding and relevant principles, then help me build a PRD for [describe your task]."

Option B (Full Session Standard) Paste the Session Standard prompt (from the how_to_use file) and attach the core framework document if you have it.

Option C (Direct) Just describe what you want to accomplish, and say "Use the PRD Builder approach from the framework: first ask clarifying questions for Deliverable, Constraints, Source of Truth, Failure Signals, etc."

I can also act as the auditor or adversarial reviewer on demand. For best results with audits/reviews, starting a fresh conversation is ideal (mirroring the "fresh session" rule), but I can simulate clean-slate mode within a thread if you instruct me to.

Would you like to test this immediately on a specific task, or would you prefer I first summarize the most relevant sections from the framework principles and walk through an example application? I'm ready to operate under these rules.

Add to chat

Where the Four Models Diverged

Each model defaulted to a different posture while converging on the same four observations.

Claude was the most reserved — validating the framework while flagging real limitations up front (process-heavy for simple builds, indirect hallucination-reduction benefit). It was the only model that explicitly named the ceiling: "this framework reduces hallucinations by reducing ambiguity — it doesn't change how I generate text, it changes what I'm given to work with."

ChatGPT went deepest into weaknesses, producing a structured five-control comparison table and identifying specific gaps in the published framework — evidence traceability, confidence annotation, completion bias, subjective acceptance criteria, and source contamination via context leakage. It also produced the cleanest articulation of the framework's category: "AI governance architecture · controlled generation systems · specification engineering · operational AI safety · production-grade agent design."

Gemini took an operational stance — producing a comparison table of casual prompting vs. engineered governance and offering to immediately run the framework on a real task. Its closing line: "Would you like to start a PRD for a specific task you're working on today?"

Grok did the same as Gemini, faster, and gave the cleanest articulation of the framework's category in a single phrase: "a complete, lightweight governance system for AI interactions."

Four different postures. Same four observations.

The Iceberg, Not the Bug

ChatGPT was the toughest reviewer. It surfaced specific gaps in the published framework — evidence traceability, confidence annotation, completion bias, subjective acceptance criteria. The critique was accurate for the layer it was given.

What ChatGPT could not see is that those gaps are already addressed in the governance architecture beneath the published framework. The four models reviewed the operating layer — the published interface. The mechanisms that close the gaps live underneath it, in the proprietary layer that the four models were not shown.

The convergence test validates the framework's external behavior. The mechanisms that close the gaps live in the protected layer that the four models could not see. The internals remain reserved.

| *That's the iceberg, not the bug.*

About the Convergence Archive

This report is the first entry in a research stream tracking convergence testing across the AI governance space. Subsequent reports will be published as additional tests are conducted with different frameworks, model combinations, and prompt structures.

The methodology is reproducible. The framework is downloadable. The proof is on screen. What remains is for the methodology to be applied beyond a single test by a single researcher. That is the point of the archive.

If you run this test on your own framework, send the results. The strongest convergence reports will be added to the public archive with attribution.

Contact

Jonathan Wilson · john.w@vertixiq.com

Convergence test archive: jonathanwilson.ai/convergence-test

A Note on What Is Shown and What Is Not

This report contains the full unedited responses from four AI systems analyzing the published interface to a broader governance architecture — the *Prompting to Production* methodology and its task-agnostic playbook. The proprietary mechanisms underneath — including the Cognitive Verb Framework, the Intent Semantic Governance Framework, and the runtime execution architecture — were not shown to the models and are not represented in their responses.

The convergence test validates the framework's external behavior. The internals remain reserved.

Wilson, J. (2026). The Convergence Test: A Zero-Shot Multi-Model Validation of the Prompting to Production Framework. VertixIQ.

Vertix IP Holdings, Inc. — All proprietary frameworks reserved.

Compiled May 13, 2026