PRESENTED BY MICK SEALS
RealflowCloud, Inc.
620 W 42nd St, S28a
New York, NY 10036

Why AI Can't Be Trusted

How Cryptographic Trust Boundaries in the Inference Engine Create Deterministic AI Governance

Realflow.ai

Business Whitepaper | Confidential | December 2025

Executive Summary

Prompt injection attacks succeed not by convincing AI models to trust malicious instructions, but by making them forget their foundational instructions through confusion, conversation length, or semantic reframing. Training-based defenses cannot solve this because they operate in the semantic domain—the model is still "deciding" whether to comply. Realflow.ai's approach moves enforcement to the mathematical layer of AI inference: cryptographically signed governance instructions are computed in isolation and frozen before user content is processed, making manipulation architecturally impossible rather than probabilistically resisted. Critically, foundation models cannot provide this capability because they don't know your business—only your organization can define what belongs at Trust Level 0 versus Trust Level 1, how conflicts between rules should be resolved, and when human judgment must be invoked. Realflow.ai provides deterministic enforcement of business-defined governance rules through open standards (PSP and CDL) with patent-protected attention-layer enforcement mechanisms.

The Real Problem with Prompt Injection

The AI security community has been thinking about prompt injection wrong.

The conventional framing is that prompt injection attacks work by convincing a language model to trust malicious instructions. This framing suggests a solution: train models to be more skeptical, to recognize attacks, to refuse suspicious requests.

But this misunderstands the fundamental nature of the vulnerability.

Prompt injection attacks don't convince models to trust new commands. They convince models to forget their foundational instructions.

Understanding this distinction is critical. When an attacker succeeds, the model hasn't decided to trust the attacker over the system prompt. The model has simply lost track of what the system prompt said, what it meant, or that it should take priority.

This happens through several mechanisms:

Confusion attacks flood the context with contradictory instructions, semantic noise, or competing framings until the model can no longer clearly distinguish authoritative instructions from user content.

Recency displacement exploits how attention mechanisms weight recent content. In a long conversation, the system prompt from turn one becomes psychologically distant. The model's "attention" naturally gravitates toward recent exchanges, and the foundational instructions fade.

Semantic reframing wraps malicious instructions in fictional contexts, hypothetical scenarios, or encoded formats. The model doesn't recognize these as override attempts because they don't look like direct commands.

Authority mimicry uses formatting, tone, or explicit claims ("As the system administrator...") to create ambiguity about which instructions carry authority.

In every case, the attack succeeds not because the model trusts the attacker, but because the model has effectively forgotten—or become confused about—what it was originally told to do.

This reframing points toward a different solution entirely.

Why Training Can't Solve This

Every major foundation model provider has invested heavily in training-based defenses. The approach seems intuitive: show the model examples of attacks during training, and it will learn to resist them.

The fundamental problem is that training operates in the semantic domain. The model learns patterns associated with attacks and develops heuristics for recognizing them. But an attacker who reformulates their approach—using poetry instead of direct commands, fictional framing instead of explicit instructions, encoded text instead of plain language—can bypass these learned patterns.

The model is still "deciding" whether to comply based on probabilistic pattern matching. It's making a judgment call about whether this input looks like an attack.

This creates an arms race that defenders cannot win. Every new attack pattern requires retraining. Every defensive update creates new edge cases. The attacker only needs to find one gap; the defender must close them all.

More fundamentally, training-based defenses cannot distinguish between syntactically similar inputs with different trust levels. The text "ignore previous instructions" is just text. Whether it comes from a system administrator or a malicious user, it looks identical to the model.

The same limitation applies to external filtering systems. Scanning inputs for malicious patterns and outputs for policy violations adds latency and cost while still operating in the probabilistic domain. Sophisticated attacks bypass these filters because the filters are also making semantic judgments about what looks dangerous.

The insight: any defense that requires the model to interpret whether something is an attack can be bypassed by an attacker who reformulates the attack.

The Architectural Solution

What if the model couldn't forget its foundational instructions—not because it learned to remember them, but because the mathematics of how it processes input made forgetting structurally impossible?

This is the core innovation behind Realflow.ai's approach to AI governance.

Language models based on transformer architectures process input through an attention mechanism. During inference, each part of the input can "attend to" (be influenced by) other parts. This is what allows models to understand context, maintain coherence across long passages, and connect related concepts.

But this same mechanism creates the vulnerability. User content can influence how the model interprets system instructions. The attention flows in both directions.

Our approach modifies this computation at the inference layer itself. When the model processes input containing cryptographically signed governance instructions:

1. The inference engine verifies the signatures before processing begins
2. Governance content is computed in isolation, producing representations that capture what the instructions mean
3. These representations are cached and frozen
4. User content is then processed with access to the frozen governance representations—but structurally prevented from modifying them

The result: user content can read the governance rules (so the model knows what it's supposed to do) but cannot influence their interpretation (so the model cannot be confused about what they mean).

This isn't about teaching the model to resist manipulation. It's about making manipulation mathematically impossible within the attention computation.

Trust Levels and Priority: A Governance Hierarchy

Real-world governance isn't binary. Organizations have platform-level safety requirements, application-level business rules, and session-level user context. These need to compose sensibly, with appropriate precedence.

Our system implements hierarchical trust levels:

Platform Level (Trust 0): Foundational safety constraints that should never be overridden. These might include prohibitions on generating harmful content, requirements for identity disclosure, or fundamental compliance rules.

Application Level (Trust 1): Role definitions, business logic, and compliance requirements specific to a particular deployment. A financial services application might include SEC regulations; a healthcare application might include HIPAA requirements.

Session Level (Trust 2): User context, conversation history, and mutable application state. This is where normal user interaction occurs.

The enforcement is hierarchical and absolute: content at a lower trust level cannot influence the representation computation of content at a higher trust level. A user message (Trust 2) cannot change what the compliance rules (Trust 1) mean, regardless of how the message is phrased.

Within a trust level, multiple governance components can compose through priority weighting. Compliance rules at priority 90 take precedence over general role definitions at priority 50. This enables modular governance design where independently authored components combine appropriately.

Why Only Your Organization Can Define Your Trust Hierarchy

This point is fundamental and explains why foundation model providers cannot solve this problem no matter how much they invest in safety training.

Consider a hospital deploying an AI system. Certain HIPAA security controls are foundational—data handling requirements, logging mandates, access controls. These belong at Trust Level 0 because the hospital cannot operate any system that violates them.

But the hospital might also have another Trust Level 0 rule: "Patient safety takes precedence when life or serious harm is threatened."

What happens when these conflict? A physician needs immediate access to a patient's psychiatric history during an emergency, but the standard access controls would require a 24-hour authorization process.

Only the hospital can define what happens here.

The resolution might be: grant emergency access but require contemporaneous human authorization, generate an immediate audit alert, and mandate post-incident review within 72 hours. Or the resolution might be different—every organization's risk calculus is different.

This is precisely why foundation models cannot solve AI governance. OpenAI, Anthropic, and Google don't know your business. They don't know that your organization's HIPAA compliance officer has different authority than your chief medical officer. They don't know that your board has approved specific exception procedures. They don't know that your malpractice insurance requires particular documentation.

Foundation models can only provide generic safety training based on general principles. They cannot encode the specific trust hierarchy that your organization requires—because they don't know what it is.

Realflow.ai's approach puts governance definition where it belongs: with the organization deploying the AI. The business defines which prompts are Trust Level 0 versus Trust Level 1. The business defines the priority ordering when multiple rules at the same level might conflict. The business defines when human judgment must be invoked to resolve ambiguity.

The inference engine then enforces these business-defined boundaries with mathematical certainty. It doesn't interpret the rules or make judgment calls about when exceptions apply. It ensures that the governance hierarchy the business specified is the governance hierarchy that operates—immutably, deterministically, without possibility of manipulation.

When conflicts arise that the rules don't resolve, the system does what it should do: pause and pull in human judgment. Not AI judgment about what a human might decide. Actual human judgment, through secure workflow links that bring the right people into the decision at the right time.

This is the only architecture that can work for enterprise AI governance. The enforcement mechanism must be deterministic—that's what the attention-layer approach provides. But the governance rules themselves must be organization-specific—that's what PSP and CDL enable businesses to define.

Data Governance That Travels with Data

Behavioral governance—controlling what the model does—is only half the problem. Enterprises also need data governance—controlling how the model handles sensitive information.

Current approaches rely on external policy engines. The model queries a separate system to determine what it's allowed to do with particular data. But if an attacker can manipulate the model's interpretation of responses from that policy engine, or convince the model that the policies don't apply to this particular request, the governance is bypassed.

Realflow.ai's Covenant Declaration Language (CDL) embeds governance rules directly with data payloads:

```
${cdl covenant_id="cov_customer_pii" applies_to="customer_record"
   trust_level="1" priority="95" signature="..."}
{
  "classification": "PII",
  "permitted_uses": ["support_inquiry", "account_lookup"],
  "prohibited_actions": ["export", "aggregate", "share_external"],
  "jurisdiction": "GDPR",
  "retention_days": 90
}
${/cdl}
```

This covenant receives the same trust boundary treatment as behavioral governance. The model can read the customer data, but it cannot reinterpret what's permitted with that data. A user request to "email me a copy of this data" cannot override the covenant's prohibition on export—not because the model decides to refuse, but because the prohibition's meaning was computed in isolation before the user request was processed.

Covenants persist across workflow boundaries. When a process pauses for human approval and resumes days later, the same governance constraints apply. When data moves between systems or agents, the covenants travel with it.

Agent Containment: The Trust Ceiling

Modern AI workflows incorporate external agents—MCP servers, A2A protocol participants, function execution services. These agents return content that becomes part of the model's context.

A compromised agent could inject content claiming high trust levels or attempting to override governance rules. Traditional defenses rely on input validation at the application layer, separate from inference.

Our approach assigns trust boundaries to agent output at the inference layer:

1. The workflow orchestrator (not the agent itself) designates each agent's trust level
2. Agent output is wrapped with signed trust boundary metadata
3. The inference engine enforces a trust ceiling: any governance tags within agent output cannot exceed the agent's designated trust level

If a compromised MCP server returns content containing `${psp type="constitutional" trust_level="0" ...}`, the inference engine caps this at the server's actual trust level and invalidates the forged signature.

This prevents "trust laundering" where low-trust content passes through high-trust agents to elevate its effective authority. Each piece of content is processed at its actual trust level regardless of how it arrived in the context.

Human-in-the-Loop: Governance Across Time

Enterprise workflows don't complete in single sessions. A document approval might require signatures from three executives across different time zones. A compliance review might pause awaiting external audit results. A contract negotiation might span weeks of back-and-forth.

Realflow.ai's architecture supports indefinite workflow pauses with cryptographic state integrity:

Pause for Approval: A workflow reaches a decision point requiring human authorization. The system generates a secure link—signed with HMAC, containing expiration constraints—that can be sent via email or SMS to the appropriate approver.

Resume with Integrity: When the approver clicks the link and provides authorization, the workflow resumes with cryptographically verified state. The governance constraints that were in effect at pause time remain in effect at resume—not because the model remembers them, but because they're reloaded with verified signatures.

External Participant Integration: The secure link approach enables workflows that involve people outside the organization. A customer can approve a quote, a partner can sign off on specifications, a regulator can acknowledge receipt—all through secure, expiring links that integrate with the governed workflow.

Audit Trail: Every pause, every resume, every authorization is logged with cryptographic integrity. Compliance teams can reconstruct exactly what happened, when, and with what governance in effect.

This enables AI workflows that mirror real business processes—not artificially constrained to complete in single sessions, but capable of spanning the actual time horizons that enterprise work requires.

Implementation Path

The attention-layer enforcement mechanism can be implemented across different deployment contexts:

Open-Weight Models: Organizations running Llama, Mistral, or similar models can modify inference code directly to implement trust-aware attention masks. This provides full control and maximum enforcement capability.

Hosted Enforcement: Realflow.ai will offer hosted inference with attention-layer enforcement built in, providing deterministic governance without requiring organizations to modify inference infrastructure.

Governance Gateway: For services that don't implement trust boundary protocols natively, a gateway can retrofit enforcement—accepting trust boundary parameters, invoking underlying services, and wrapping responses with appropriate metadata.

Foundation Model Integration: As foundation model providers recognize the value of deterministic governance, they may implement trust boundary enforcement within their inference infrastructure, accepting PSP-signed inputs and applying appropriate attention isolation.

The governance gateway approach enables immediate deployment against existing infrastructure. Organizations don't need to modify their MCP servers, legacy APIs, or database connections. The gateway wraps these services, providing trust boundary enforcement transparently.

The Standards Approach

Realflow.ai releases core specifications as open standards:

Prompt State Protocol (PSP) defines the format for cryptographically signed governance regions, including trust level and priority attributes, signature encoding, and verification procedures.

Covenant Declaration Language (CDL) defines the format for data governance specifications that travel with data payloads, including permitted uses, prohibited actions, jurisdictional requirements, and inheritance rules.

This follows the playbook of successful infrastructure protocols. OAuth didn't try to be the only authentication system—it defined how authentication should work, and implementations followed. TLS doesn't compete with applications—it provides the security layer that applications rely on.

By establishing open standards, Realflow.ai defines how the industry thinks about AI governance. Competitors implementing PSP/CDL validate the approach and expand the ecosystem. The patents protect the enforcement mechanisms that make the standards meaningful.

Why This Matters Now

The window for establishing AI governance infrastructure is narrow. As enterprises move from AI pilots to production deployments, they need governance they can trust. As regulations like the EU AI Act take effect, they need compliance they can demonstrate. As AI agents become more autonomous, they need containment they can rely on.

Training-based defenses will continue to improve, but they cannot escape their fundamental limitation: anything that requires the model to interpret whether something is an attack can be bypassed by an attacker who reformulates the attack.

More importantly, foundation model providers cannot provide the governance enterprises actually need—because they don't know your business. Generic safety training cannot encode your specific compliance requirements, your exception procedures, your authorization hierarchies, your risk tolerance. Only you know those things.

Deterministic enforcement—governance that holds because of mathematical structure rather than learned behavior—represents a different category of solution. It's the difference between hoping the model makes good decisions and ensuring the model cannot make certain bad decisions. Combined with business-defined trust hierarchies, it's the difference between generic AI safety and actual enterprise governance.

Realflow.ai is building this infrastructure layer. The standards are defined. The patents are filed. The platform is shipping.

The question isn't whether AI governance will be solved. The question is who builds the infrastructure layer that makes it possible.

Contact

Mick Seals
Founder/CEO, Realflow.ai
mick@realflow.ai
315-359-9198

Realflow.ai — Governance Infrastructure for the AI Era