Shumaker

OCTOBER 10, 2025 | PUBLICATION

# "Chatty Chatbots: Why AI Agents are the Silent Threat to your Company's IP"

Agentic AI is one of the buzziest concepts in business today. There's no mystery why—the promise of dramatic increases in efficiency, and ones that actually accelerate over time as the agents get better at their jobs, all for what is advertised as a fraction of the cost of the employees who currently do the job. Unfortunately, AI Agents carry a hidden risk that your business may not have accounted for.

**INDUSTRY SECTOR**

Technology

**SERVICE LINE**

Intellectual Property
Technology, Data Privacy, Cybersecurity & AI

**RELATED PROFESSIONALS**

Nick Carr
Brian C. Focht

**MEDIA CONTACT**

Wendy M. Byrne
wbyrne@shumaker.com

The Risk: Most Agentic AI was engineered to complete tasks reliably, rather than prioritizing the protection of specific types of data. The focus on completing operations first creates a bias within the system that establishes concrete pathways for confidential information, personal data, and intellectual property (IP) to leak from Agent to Agent, across memories, and through over-permitted integrations. In other words, putting any data that you entrust to the AI Agent at risk.

## HOW LEAKS ACTUALLY HAPPEN

That sounds ominous, but also a bit "shrouded in mystery." The good news is that we're not here to hide the ball. First, it's helpful to get a basic understanding of what Agentic AI is:

"Agentic AI" refers to artificial intelligence systems that possess… well… agency. Agentic AI can perceive circumstances, make decisions based on that perception, and act independently toward a defined goal.

Unlike traditional generative AI, which responds to prompts (e.g., ChatGPT answering a user's question), Agentic AI receives a high-level objective and autonomously decomposes it into actionable steps based on available services (including other AI agents, AI models, and conventional computing services). It executes these steps iteratively, adapting as it goes.

This concept extends beyond basic automation or decision-making, particularly when multiple AI Agents collaborate to achieve their objective. Want to sell consumer products online? Tell your Agentic AI and watch: one AI agent sources products, another manages logistics, a third builds and maintains your website,

and a fourth writes marketing content.

Each agent operates within a defined area of expertise, continuously improving based on relevant feedback.

So, how do these agents end up leaking confidential data?

**1) Context dumps.**

AI Agents pass instructions to each other in natural language. To avoid failing to complete a task, the "coordinator" agent often forwards *everything it* knows, including full documents, email threads, and meeting notes (or worse, transcripts), to a helper agent "just in case."

Functionally, it's like replying-all with the entire history of a negotiated deal when you only needed the abstract.

The principle of least privilege (only sharing what's necessary) disappears. Data that doesn't need to be shared, potentially including confidential or sensitive data included in documents nobody expected to be shared, is passed along to another agent.

**2) Fuzzy retrieval that over-includes.**

Many AI systems utilize retrieval-augmented generation (RAG), which involves searching a "memory" of past content for similar items. But "similar" doesn't necessarily mean "correct." A search for "Q3 pricing" might also pull merger and acquisition (M&A) due diligence, margin models, or roadmap slides because they "live near" each other based on the agent's training, and the agent is programmed to over-perform the task to avoid failure. The summarizer then repeats that content to yet another agent, compounding the spill.

Essentially, it's like asking a paralegal for one clause of a contract and getting the redline history of the whole negotiation because it "seemed related."

Sensitive but unrelated files can be mixed into the response and then forwarded to other agents.

**3) Over-permitted tool keys.**

Granting an agent broad access to important systems may seem efficient. After all, to perform a function, they need the authorization to do it. However, in practice, you've just created another vulnerability in your system. Bad actors who gain access to the agent through various means will have access to all the systems the agent does.

Essentially, it's like giving your intern a skeleton key to the office.

A single stolen or mishandled token can enable large-scale data access.

**4) Shared memory with soft borders.**

Agents save snippets to "memory" for future accuracy. If memories aren't segmented (restricted vs. confidential vs. general), yesterday's finance brief becomes tomorrow's marketing retrieval.

Essentially, it's like filing confidential HR documents in your company's general library so that people can "find things faster."

Content drifts across teams or use-cases, making unintended disclosure more likely.

**5) Debug/telemetry backdoors.**

To troubleshoot, engineers enable verbose logging, which records raw prompts and responses, often including internal business information (the risk of which increases rapidly for tools like coding bots). Those logs are commonly shipped to third-party analytics tools, which help with traceability and searchability. It saves the engineers time, but at what cost?

Essentially, it's like giving someone access to your entire phone just so that they can track your location while you're hiking.

By electing to allow a third party to sort through the data you send, it's the third party who is the first to discover what secrets you've allowed to escape.

**6) Cross-tenant misconfiguration.**

In large companies, multiple business units or customers (referred to as "tenants") may share the same hardware platform or applications. If the controls that separate them are misconfigured or older sign-in methods are too permissive, one group can see or act as another.

Essentially, it's like giving two customers the same login and password to a secure communication portal.

Recently, Microsoft patched an error that allowed attackers to impersonate other users (including administrator access) across different tenants, undermining "who should see what," and enabling unauthorized access even if internal policies are sound.

The through-line: agents are rewarded for *not failing*, so they hoard and forward context. Unless you design explicit constraints, leakage is not an outlier; it's the default behavior.

## YOUR INTELLECTUAL PROPERTY AT RISK

By leaking data, Agentic AI can quietly erode IP rights unless you change the defaults. These leaks can negatively impact trade secrets, patents, trademarks, and copyrights.

**Trade secrets: "Context dumps" and shared memories turn secrets into non‑secrets.**

Trade secret protection requires reasonable measures to keep information secret. Agent‑to‑agent "reply‑all" behavior and unsegmented memory stores can quietly distribute confidential formulas, models, roadmaps, code, diligence files, or negotiation notes to services and people who have no "need‑to‑know." Once your system, logs, or a third‑party telemetry tool hold the data without appropriate access controls, you've created evidence that secrecy was not reasonably protected, undermining the reasonable measures required to maintain trade secret protections. Doing so may weaken potential misappropriation claims and limit injunctive relief.

**Patents: agent behaviors can create bars and contamination.**
Two patent-related problems show up in agentic pipelines:

- *Public disclosure risk.* Chatbots, model playgrounds, or demo agents can surface invention details (or push them into public logs, issue trackers, or model providers). These disclosures may occur indirectly via "over‑permitted tool keys," "debug/telemetry," or misconfigured tenants that expose artifacts to others. A public disclosure starts the U.S. one‑year filing clock and may immediately bar rights in many jurisdictions outside the U.S.

- *Inventorship and contribution evidence.* If engineers rely on AI agents during conception, you'll need clear records of the human contributions to support inventorship and avoid later challenges. The more the system auto-composes or suggests, the more disciplined your documentation must be to show which human conceived the claimed subject matter and how.

**Copyright: ownership, training data, and derivative-work traps.**
Three recurring issues in copyrights:

- *Authorship of outputs.* If agents generate expressive content, ensure you can demonstrate sufficient human creative control (selection, arrangement, revision) in work-for-hire or assignment agreements, and lock down vendor terms of service so providers don't claim co-ownership or broad reuse rights in what the agent produced.
- *Upstream infringement risk.* Retrieval and "fuzzy" similarity can cause agents to reproduce portions of copyrighted works (code, images, copy) from nearby embeddings or prior memory. That exposure grows when "fuzzy retrieval" over-includes or when context is dumped to other agents that blindly "summarize" by copying.
- *Downstream training/analytics use.* If vendors ingest your prompts, outputs, or logs for "service improvement," they may train on your creative content or trade secrets unless the contract opts out and mandates deletion.

**Trademarks and brand: uncontrolled agent voice and asset creation.**
Agents that author product names, taglines, ads, or user interface text can:

- Coin marks that are descriptive or unregistrable;
- Reuse third-party marks in ways that create confusion; or
- Drift tone/quality, eroding distinctiveness.

**Practice pointer.**

Embed clearance checks in the workflow and gate publication behind human legal/brand review, especially where "coordinator" agents route content to "helper" writers at scale.

## WHAT "GOOD" LOOKS LIKE: SECURITY-BY-DEFAULT FOR AI AGENTS

**Agents are strangers. Not teammates.**

Assume each agent should see only what it absolutely needs for a specific task. When one agent asks another for help, the request should specify who is asking, why they need it, what narrow information is being requested, and for how long. Do not let agents send big, unfiltered "context dumps."

**Data minimization by contract, not by prompt.**

Build simple, structured "cover sheets" for what agents can exchange (for example: document ID, short summary, sensitivity label). Configure the system to hide or blur details by default. If a helper agent requires more information, it must request the additional fields, and these requests should be verified against policy before release.

Probably by a human.

**Keep "memories" separated by sensitivity.**

Agents often save snippets to remember later. Divide the storage into distinct buckets (e.g. restricted, confidential, and general, but make sure they align with your company's data classification system) with separate access rules and retention policies. Never let information flow from restricted storage into general storage. Periodically review the general storage and clean out any misfiled data.

**Least-privilege, short-lived credentials.**

Agents should only have credentials applicable to the task at hand. Issue per-task credentials. A planning agent might see file names and labels; only the execution agent can open the single, approved file version. Keys should expire automatically when the task ends.

**Screen content before your agents receive it.**

Run quick checks for sensitive information (like trade secrets, unfiled inventions, internal tests or studies, or other protected data) before content is added to an agent's working context. If something sensitive is detected, block it, mask it, or require a human to approve.

**Policy-as-code at the message bus.**

Write clear, machine-enforceable rules such as "product test data never leaves our environment," "prior art research requires a recorded consent flag," and "pricing documents cannot be sent to external AI models." Enforce these rules at the point where agents exchange messages, not by simply telling your employees to avoid entering the information as a prompt.

**Signatures and verification.**

Have the system cryptographically "sign" agent-to-agent messages and attach labels that describe the data and its intended use. Reject messages that aren't signed or whose labels don't match the rules (or don't match the payload).

**Place a canary in the coal mine; monitor the exits.**

Place harmless "canary" markers in highly sensitive documents and alert if those markers show up in logs or external services. Limit where agents can send data on the internet to a short list of approved destinations.

**Test for more than results.**

In addition to jailbreak tests, run exercises that try to coax agents (or their retrieval/search features and logs) into pulling restricted documents. Reward teams and systems for what the agents refuse to share, not just for how fast they complete tasks.

**Your contracts must match reality.**

When working with AI vendors, include the following in your contracts: no training on your data, minimal telemetry, prompt/response deletion, strict subprocessor controls, and confidentiality/IP terms that preserve trade-secret status. If you're public or otherwise regulated, ensure your incident playbooks align with SEC and NYDFS timing and disclosure requirements.

## CONCLUSION

Agentic AI will default to oversharing unless you flip the incentives and the plumbing. Build explicit controls

around inter-agent messages, memories, and tool scopes; align notices and contracts to actual behavior; and wire incident governance to regulatory clocks. Do that, and you maintain high collaboration, low leakage, and credible compliance.

© 2026 Shumaker, Loop & Kendrick, LLP. All Rights Reserved. Privacy Policy/Disclaimer