Satya's practice focuses on technology transactions, intellectual property, and commercial agreements for startup and growth-stage technology companies across AI, robotics, cloud computing, and emerging tech sectors.

Email: snarayan@gcalaw.com
LinkedIn Newsletter: Techline Brief
Connect on LinkedIn

PROTECTING AGENTIC AI SYSTEMS: ESSENTIAL LICENSING RESTRICTIONS FOR DEVELOPERS WORKING WITH OEMS & OTHER PARTNERS

Satya S. Narayan
February 3, 2026

Why This Matters

As agentic AI systems become more powerful and more deeply integrated into hardware products, the legal framework governing their use becomes increasingly important. Developers who create these systems must protect their intellectual property and prevent misuse or unauthorized expansion of their technology. To accomplish this, developers should consider a series of restrictions when contracting with OEMs and other partners to safeguard the developer's rights and limit how their technology is used.
This article explains important recommended licensing restrictions for deploying agentic AI systems.

1. Prohibition on Reverse Engineering and Derivative Works

In the software industry, it has long been standard practice to prohibit reverse engineering, disassembling, or decompiling software to prevent revealing its source code, which is widely recognized as the developer's most valuable intellectual property.

With the emergence of agentic AI systems, this familiar restriction must now be broadened to cover a new category of equally sensitive assets. Beyond protecting source code, developers should consider also prohibiting any attempt to extract or replicate the system's underlying model architecture, weights, training data, or internal decision-making logic, as well as the creation of derivative works based on the agentic AI system, its components, or its outputs.

These expanded protections are intended to protect against the partner studying how the agentic AI system is built, copying its core mechanisms, or developing a competing system based on insights gained from access to the technology.

A no-reverse engineering clause is essential because an agentic AI system's value lies in its proprietary design, including its architecture, trained parameters, curated datasets, and decision-making logic that enable it to act autonomously. Without clear contractual boundaries, a partner could intentionally or unintentionally attempt to learn about the system through extensive querying, stress tests, comparative evaluations, or by examining logs, runtime artifacts, or memory states. Even routine benchmarking or debugging, if unrestricted, can cross into behavior-based reconstruction.

The clause must also bar deliberate attempts to infer internal components, including performance analysis techniques that might reveal architecture, approximate weights, or expose training data. Protecting the agentic layer, including planning logic, task decomposition, tool selection policies, memory management, is equally important. These components reflect substantial innovation and can be reverse engineered through careful observation if not contractually protected.

Finally, because agentic systems expose tool APIs, contextual reasoning behaviors, or adaptive workflow patterns, they can inadvertently reveal how the system prioritizes actions or interprets instructions. Without restrictions, a partner could misuse these signals to replicate core aspects of the technology.

2. Binaries, SDKs, and APIs Only; Exceptions and Safeguards for Access to Internals

Licenses for agentic AI systems should provide OEMs and partners only with compiled binaries, SDKs, or API-level access and not model weights or orchestration graphs. Weights, training signals, and planner-level orchestration logic form the core reasoning substrate of agentic systems. Allowing access to them would expose proprietary heuristics, tool-use policies, and safety systems, enabling replication or unauthorized modification of the autonomy stack.

Limited exceptions may arise, such as regulatory audits requiring inspection, specialized deployments needing controlled fine tuning, or on-premises/air-gapped environments requiring local execution. Even in these scenarios, the best practice is to provide only what is strictly necessary, such as parameter-efficient tuning adapters, encrypted artifacts, or abstracted orchestration interfaces, rather than full weights or planner graphs.

When internal components must be shared, developers should impose strong safeguards: trade-secret level confidentiality obligations, use-limitation provisions, no reverse engineering, secured execution environments, audit rights, cryptographic protections, and restrictions requiring analysis to occur only in controlled, access-restricted facilities. These

measures are aimed at protecting against exceptional access compromising the secrecy or replicability of the system.

It is also relevant to keep in mind that the U.S. Department of Commerce's BIS interim final rule, the *Framework for Artificial Intelligence Diffusion*, which took effect on January 13, 2025, imposes a worldwide license requirement on exporting, reexporting, or transferring advanced AI model weights to the extent classified under ECCN 4E091 (that is, model weights of any closed-weight AI model trained on more than 10^{26} computational operations).

3. Prohibition on Training, Fine-Tuning, or Adapting the Model

Agentic AI models can change behavior significantly when trained, fine-tuned, or adapted. Allowing partners to modify the model would enable creation of derivative systems the developer did not design, validate, or certify.

Licenses should explicitly prohibit partners from:

- training or fine-tuning the model
- adapting it for new or expanded purposes
- using system outputs, logs, or telemetry to train competing models

These restrictions are meant to guard against the risk of "shadow training," where partners indirectly replicate proprietary methods by harvesting system behavior. They also help maintain adherence to the developer's safety, performance, and compliance requirements. By limiting partners to integration and use only, the license preserves model integrity and protects core IP.

4. Limiting Use to Approved Hardware and Approved Purposes

Agentic AI systems are often engineered for specific hardware profiles and narrow product categories. For example, an assistant designed for a premium smart home hub may rely on processing units, thermal constraints, or security modules unavailable in lower-tier devices.

If partners deploy the system on unapproved hardware, the system could underperform, behave unpredictably, or create support and liability obligations the developer never intended. Because agentic AI systems depend heavily on hardware-specific compute, memory, and acceleration capabilities, they may not automatically be forward or backward compatible across device classes. As a result, deploying the same agentic AI system on different or modified devices may require further optimization or reconfiguration.

Developers should consider restricting the licensed use to:

- use solely on approved hardware models
- no sublicensing or redistribution to third parties
- deployment only within the agreed-upon product category

These limits ensure developers retain control and are intended to restrict partners from extending the agentic AI into new devices or markets without renegotiating rights and fees.

5. No Modification of Safety, Security, or Control Mechanisms

Agentic AI systems may interact directly with hardware or influence device behavior. To maintain safety, developers implement filters, permissioning layers, and security controls.

Licenses should prohibit partners from:

- altering or disabling safety mechanisms
- bypassing or weakening security features
- ignoring required hardware-rooted security protections

Agreements should also require revalidation when hardware, firmware, or supporting software changes could affect system behavior. These restrictions are designed to protect the developer from unauthorized changes that increase the risk that the agentic AI system operates outside safe and secure parameters.

6. Restrictions on Benchmarking Agentic AI Systems

Benchmarking agentic AI systems can reveal far more than performance metrics. Because their behavior reflects planning policies, tool use strategies, memory routines, and multi-step orchestration logic, even simple tests such as large sets of reasoning prompts or tool use frequency tracking can expose internal decision patterns.

Public benchmarks can also harm reputation or give competitors insight into strengths and weaknesses.

Licenses should therefore:

- consider restricting benchmarking without written approval
- prohibit publication of performance results
- prohibit use of evaluation data for competitive analysis

Benchmarking controls help guard against competitive profiling and unintended disclosure of proprietary reasoning methods.

7. Logs, Telemetry, and Traces

Developers should retain ownership or control over logs, telemetry, and traces, because runtime data reflects how the system reasons, plans, and interacts with its environment. These records often contain sensitive indicators such as internal state, tool-invocation patterns, and orchestration flows.

At the same time, any assertion of ownership must be balanced with applicable privacy and AI statutes and partner operational requirements. Where runtime data includes identifiers, customer inputs, or other personal information, compliance can be maintained through redaction, minimization, or secure-handling practices that respect these legal and operational constraints.

Licenses should also prohibit partners from:

- using logs or telemetry to develop competing models
- analyzing runtime data to infer architecture or reasoning logic

These restrictions close a major loophole that might otherwise enable indirect reverse engineering while still accommodating the legal and operational limits on how runtime data may be controlled or processed.

Conclusion

Licensing restrictions for agentic AI systems are not merely legal formalities; they are essential safeguards that protect intellectual property, are intended to reinforce safe operation of the system, and prohibit misuse or unauthorized expansion of advanced AI technologies.

Effective agreements must account for how these systems reason, plan, orchestrate tools, and produce runtime data. Clear restrictions on use, deployment, data handling, and reverse engineering help maintain system integrity while respecting the legal and operational constraints of modern AI deployments. Well-structured licensing terms ultimately encourage safe operation, protect proprietary technology, and support responsible integration across OEM and partner environments.
