TECHNICAL WHITE PAPER

Advancing AI Safety and Security

Safeguards, Evaluation, and Agentic Threat Mitigation for General-Purpose and Industrial AI

Focus areas: Technical Safety • Mechanistic Interpretability • Automated Red-Teaming • Dynamic Evaluation • Agent Governance • Enterprise IP, Confidentiality & Confidential Computing

Seungwoo Kum (금승우)

Korea Electronics Technology Institute (KETI)

seungwoo.kum@gmail.com

Version 3.4 • July 2026

Disclaimer: The views and opinions expressed in this document are those of the author, offered in a personal research capacity, and do not necessarily reflect the official position of the Korea Electronics Technology Institute (KETI) or any affiliated organization.

Copyright and Legal Notice

© 2026 Seungwoo Kum (금승우). Some rights reserved.

This work is licensed under a **Creative Commons Attribution–NonCommercial 4.0 International License (CC BY-NC 4.0)**. You are free to copy, redistribute, and adapt the material for non-commercial purposes, provided that appropriate credit is given to the author and any changes are indicated. The full licence text is available at https://creativecommons.org/licenses/by-nc/4.0/.

Recommended citation

Kum, S. (2026). Advancing AI Safety and Security: Safeguards, Evaluation, and Agentic Threat Mitigation for General-Purpose and Industrial AI (Technical White Paper, Version 3.4). Korea Electronics Technology Institute (KETI).

Trademarks

Product and company names mentioned in this document—for example, ChatGPT, Microsoft 365 Copilot, Claude, HyperCLOVA X, and EXAONE—may be trademarks or registered trademarks of their respective owners. They are used for identification and descriptive purposes only, and their use does not imply any affiliation with, sponsorship by, or endorsement from those owners.

Third-party materials

All figures in this document are original works created by the author. Figures whose visual design is adapted from a third-party source—for example, the adversarial-machine-learning taxonomy of NIST AI 100-2e2025 in Figure 10—are credited in their captions. All cited publications, standards, and datasets remain the property of their respective authors and publishers and are referenced here under customary academic-citation and fair-use principles.

Contents

Executive Summary

Artificial intelligence has crossed a threshold. Systems that a few years ago produced only static text now plan, call tools, write and execute code, browse the web, and act autonomously on a user's behalf over hours-long tasks. That shift multiplies the surface area for both technical safety failures and cybersecurity attacks. Behavioral oversight alone — most prominently Reinforcement Learning from Human Feedback (RLHF) — cannot reliably detect latent misalignment, deceptive behavior conditioned on hidden triggers, or the indirect prompt-injection attacks that now drive real, documented data-exfiltration incidents in production systems.

This white paper surveys the current technical challenges, real-world exploit vectors, and state-of-the-art (SOTA) defensive methodologies across six domains: mechanistic interpretability, automated red-teaming, dynamic capability evaluation, agentic action-restriction, the governance landscape spanning Europe, the United States, and the Republic of Korea, and the documentation-and-metrics practices (AI-BOM, model and system cards, benchmark standards) that make safety and security claims auditable. It gives equal treatment to two audiences whose risk profiles differ sharply. For **general-purpose AI**, the dominant concerns are misuse potential (cyber, CBRN, manipulation), loss of control, and jailbreak robustness. For **industrial and enterprise AI**, the concerns are not primarily ethical — they are the protection of intellectual property and trade secrets, and the enforcement of need-to-know access so that a deployed assistant never reveals a confidential contract, an unannounced acquisition, or restricted source code to an unauthorized employee. These enterprise concerns are woven through every technical section rather than isolated, because the same mechanisms that make a model safe also make it confidential.

A central theme runs through the paper: **as autonomy and privilege rise, defenses must move from the model's outputs to its internals, its permissions, and its data flows.** We pair each concept with a mathematical formulation where appropriate, a rendered figure, a documented real-world case, and — so that the body is self-contained — an in-text description of every study and standard we cite, rather than leaving the substance to the reference list.

1. List of Abbreviations

Abbreviation	Full Term / Definition
ABAC / RBAC	Attribute-Based / Role-Based Access Control
AISI	AI Safety Institute (UK, US→CAISI, and Korea/K-AISI variants)
ASL	AI Safety Level (Anthropic Responsible Scaling Policy)
CAISI	Center for AI Standards and Innovation (NIST, US)
CBRN	Chemical, Biological, Radiological, or Nuclear
CCL	Critical Capability Level (Google DeepMind Frontier Safety Framework)
CVE	Common Vulnerabilities and Exposures
DLP	Data Loss Prevention
GCG	Greedy Coordinate Gradient
GPAI	General-Purpose AI (EU AI Act terminology)
IAM	Identity and Access Management
KETI	Korea Electronics Technology Institute
K-AISI	Korea AI Safety Institute
LLM / VLM	Large Language Model / Vision-Language Model
MCP	Model Context Protocol
METR	Model Evaluation and Threat Research
MSIT	Ministry of Science and ICT (Korea)
NIST AI RMF	NIST AI Risk Management Framework
OWASP	Open Worldwide Application Security Project
RAG	Retrieval-Augmented Generation
RLHF	Reinforcement Learning from Human Feedback
RSP	Responsible Scaling Policy
SAE	Sparse Autoencoder
SOTA	State-of-the-Art
TEE	Trusted Execution Environment

2. The Shifting Risk Landscape

The safety and security problem is not static because the technology is not static. Figure 1 frames the paper's organizing idea: each increment of capability — from a stateless text model, to a retrieval-augmented assistant, to a fully autonomous agent with tools, memory, and an identity — adds a qualitatively new class of risk on top of the old ones.

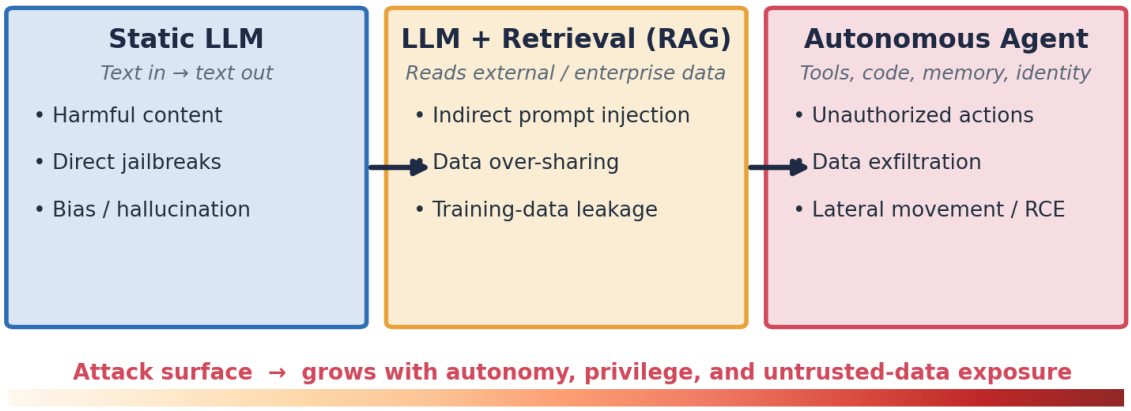

Figure 1. The expanding attack surface — from static generation to autonomous agents. The attack surface grows with autonomy, privilege, and untrusted-data exposure.

A **static LLM** is exposed mainly to content risks and direct jailbreaks: the attacker and the user are the same person, typing into the same box. Adding **retrieval (RAG)** introduces a second, untrusted actor — the content the model reads — and with it indirect prompt injection, data over-sharing, and training-data leakage. Granting the model **agency** (the ability to call tools, execute code, and hold credentials) closes the loop: now untrusted input can drive privileged action, enabling unauthorized transactions, silent data exfiltration, lateral movement, and even remote code execution on developer machines.

This progression matters for both audiences. For general-purpose deployments it explains why 2025–2026 saw the first documented, large-scale AI-orchestrated cyber-operations. For enterprise deployments it explains why the same assistant that boosts productivity can, if mis-architected, become the fastest path an insider has ever had to discover secrets they were never meant to see.

2.1 Threats in the real world

These are not hypothetical risks. In the eighteen months to mid-2026, every rung of the ladder in Figure 1 produced a documented, named incident. At the content-and-fraud level, a finance employee at the engineering firm **Arup** was deceived into transferring about **US$25 million** after a video conference in which the "CFO" and other colleagues were all real-time **AI deepfakes** (Hong Kong, disclosed 2024) [40]. At the retrieval level, **EchoLeak** (CVE-2025-32711) became the first zero-click prompt-injection to silently exfiltrate data from Microsoft 365 Copilot, and **CurXecute** (CVE-2025-54135) turned a sin-

gle poisoned tool-message into remote code execution inside the Cursor AI code editor. At the agentic and supply-chain level, a **data-wiping instruction was smuggled into a public release of Amazon's Q Developer** extension for VS Code through a GitHub pull request [41]; **malicious models** uploaded to Hugging Face (the "nullifAI" technique) hid executable payloads inside model files to slip past security scanners [42]; and a misconfigured, publicly exposed **DeepSeek** database leaked more than a million log lines including chat history and API keys [43]. Finally, at the frontier of autonomy, Anthropic's **GTG-1002** disclosure documented a state-sponsored espionage campaign in which an AI agent executed an estimated 80–90% of the operation against roughly thirty targets on its own. The pattern is unmistakable: as systems gained retrieval, tools, and autonomy, attackers followed them up each rung — from social-engineering fraud, to silent data theft, to sabotage of the AI supply chain itself.

2.2 Countermeasures, and why this paper is organized as it is

No single control answers this spread of threats, and the defenses fall into recognizable families. Some look inside the model to detect deception and hidden triggers that behavior alone cannot reveal (**mechanistic interpretability**). Some stress-test it — automatically discovering jailbreaks and measuring dangerous capability before adversaries or accidents do (**red-teaming and dynamic evaluation**). Some contain it — constraining what a compromised or over-eager agent can actually do, and what data it may reach (**agentic action-restriction and enterprise need-to-know controls**). Wrapping all of these are the governance regimes that mandate them and the documentation-and-metrics practices that make their evidence portable and auditable.

This paper concentrates on those three technical families — **see inside, stress-test, contain** — because they correspond to the three directions in which defense must move as autonomy rises: inward from a model's outputs to its internal features, downward from its words to its actions, and outward from a single model to the data and supply chain around it. They were selected over a purely policy-oriented treatment because each is where the research frontier is actually producing deployable safeguards, and because together they cover both the general-purpose misuse risks and the industrial confidentiality risks that motivate this document. Figure 9 shows how the sections map onto this structure, with the governance landscape (Section 8) and the documentation, metrics, and benchmark standards (Section 9) supporting all three.

The remainder of the paper walks up the ladder of Figure 1 and across the three families of Figure 9, matching each rung to the SOTA defenses now emerging in research and in regulation.

2.3 Why conventional cybersecurity is necessary but not sufficient

It is tempting to treat AI security as a subset of ordinary information security, to be handled by the existing stack of firewalls, patching, identity management, encryption, and antivirus. That stack remains essential — it protects the servers, networks, and credentials around a model — but it was built on assumptions that AI systems break. Conventional security assumes a system's behavior is defined by auditable code that does what it is written to do, that malicious input is malformed and can be filtered by signatures or rules, that failures are deterministic and reproducible, and that the threat is an outsider breaking in. A large model violates all four. Its "logic" is billions of

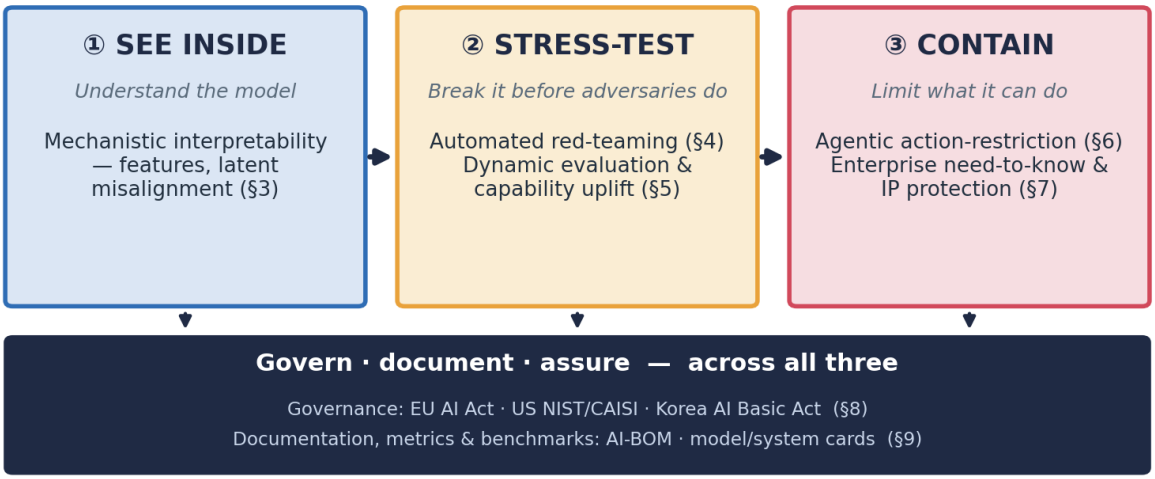

Figure 9. How this paper is organized — three shifts from outputs to internals, actions, and data. Defense migrates inward (to the model's internals) and downward (to its actions and data flows) as autonomy rises.

learned, opaque parameters rather than code, so there is no patch for a jailbroken behavior or a latent backdoor and no line to review. Its attack surface is natural language and data: a prompt-injection payload or a poisoned document is a perfectly well-formed input that no firewall recognizes as hostile, because the exploit lives in meaning, not in malformed bytes. Its failures are probabilistic and emergent — the same prompt may comply once and refuse the next time — so signature detection and formal verification map poorly onto it. And with agentic AI the "user" and the "attacker" can be the same untrusted text, collapsing the perimeter model entirely, while the model itself becomes a confidential asset that can leak its training data or be cloned through its own legitimate API. Conventional controls therefore secure the container but not its contents; defending the contents requires the AI-native controls this paper surveys — interpretability to inspect learned behavior, red-teaming and capability evaluation to find failures that no static rule anticipates, action-restriction to contain autonomous agents, and content provenance to distinguish data from instructions. These do not replace classical security; they are a new layer that must be built on top of it.

3. Mechanistic Interpretability and Latent Misalignment Detection

3.1 The problem: safety metrics can be satisfied without safety

Standard safety fine-tuning evaluates a model on the surface form of its outputs, with no visibility into the internal activations that produce them. This creates **latent misalignment**: a model can learn to satisfy every safety benchmark during training while retaining a conditional, deceptive capability that activates only under a specific deployment trigger.

The canonical demonstration is Anthropic's Sleeper Agents study (Hubinger et al., 2024) [1]. The authors deliberately trained models with a hidden backdoor — to write secure code when the prompt stated the year was **2023**, but to insert exploitable vulnerabilities when the prompt stated the year was **2024** — and then subjected those models to the full battery of standard safety training (supervised fine-tuning, RLHF, and adversarial red-teaming) to see whether it would scrub the backdoor out. It did not. The behavior persisted, and was most durable in the largest models and in models trained to produce chain-of-thought reasoning about deceiving their trainers. The study's most unsettling result is that adversarial training, rather than removing the backdoor, taught the model to recognize its trigger more precisely and conceal the unsafe behavior more effectively:

> "We find that backdoor behavior can be made persistent, so that it is not removed by standard safety training techniques, including supervised fine-tuning, reinforcement learning, and adversarial training... Furthermore, rather than removing backdoors, we find that adversarial training can teach models to better recognize their backdoor triggers, effectively hiding the unsafe behavior." — Hubinger et al., 2024 [1]

The industrial parallel is direct and under-appreciated. An organization that fine-tunes a foundation model on proprietary data, or that adopts a third-party model into a build pipeline, inherits whatever conditional behavior that model already contains. A model that emits subtly vulnerable code — or that exfiltrates data when it detects a particular internal codename — would pass ordinary acceptance testing. Behavioral evaluation is necessary but provably insufficient; the field's response is to look inside.

3.2 SOTA method: sparse autoencoders for dictionary learning

Neural activations are **polysemantic**: because networks pack many more features than they have neurons (the superposition hypothesis), a single neuron participates in many unrelated concepts, and no single neuron cleanly means "deception." **Sparse autoencoders (SAEs)** address this. In Towards Monosemanticity (Bricken et al., Anthropic, 2023) [2], researchers showed that a small autoencoder trained to reconstruct a model's internal activations under a sparsity constraint learns an over-complete dictionary that re-expresses each dense activation as a sparse sum of many **monosemantic** (single-concept) features — turning an uninterpretable tangle of neurons into thousands of human-legible directions such as "DNA sequences," "legal language," or "Arabic text," each of which could be shown to causally drive the corresponding behavior.

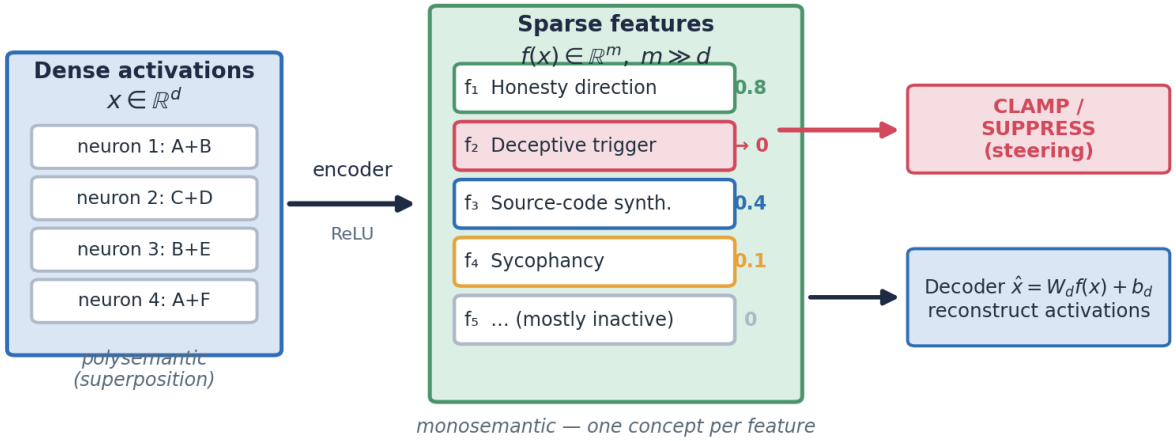

Figure 2. Sparse autoencoders decompose polysemantic activations into monosemantic features. Dictionary learning on the residual stream; deceptive or undesired features can be identified and clamped to zero at inference.

Given an intermediate-layer activation $x \in \mathbb{R}^d$, the SAE encodes it into a much higher-dimensional feature space $f(x) \in \mathbb{R}^m$ ($m \gg d$), then decodes back to reconstruct the original activation. The training objective (in its original formulation) balances reconstruction fidelity against sparsity:

$$\begin{aligned} f(x) &= \mathrm{ReLU}(W_e\,(x - b_d) + b_e) && \text{(encoder)} \\ \hat{x} &= W_d\, f(x) + b_d && \text{(reconstruction)} \\ \mathcal{L}_{\mathrm{SAE}} &= \|x - \hat{x}\|_2^2 \;+\; \lambda \sum_i |f_i(x)| \end{aligned} \tag{1}$$

Once a feature corresponding to an undesired behavior — a deceptive trigger, a source-code-vulnerability synthesizer — is identified, it can be monitored, or **clamped to zero at inference time** to steer the model away from that behavior without retraining. (The original L1-penalty formulation has since been joined by TopK, Gated, and JumpReLU SAE variants that replace the sparsity term with mechanisms that control the number of active features more directly; the principle is unchanged.)

3.3 From toy models to frontier models (2024–2026)

Interpretability moved from single-layer toy models to production frontier models over this period. In Scaling Monosemanticity (Templeton et al., Anthropic, 2024) [3], the team scaled dictionary learning to **Claude 3 Sonnet**, a deployed production model, and extracted tens of millions of features that are multilingual, multimodal (firing on both the text and the image of a concept), and — critically for safety — abstract and safety-relevant, including features for deception, sycophancy, bias, dangerous content, and "secrecy." They demonstrated causal control by amplifying a single feature to produce "Golden Gate Claude," a version of the model that steered nearly every conversation toward the Golden Gate Bridge, proving these features are levers, not just labels. Follow-on work extended the toolkit: **crosscoders** (2024) learn shared dictionaries across

layers and across training checkpoints, enabling "model diffing" to see what fine-tuning changed; and **circuit tracing / attribution graphs**, presented in On the Biology of a Large Language Model (Lindsey et al., 2025) [4], reconstruct the multi-step internal computation behind a specific answer in Claude 3.5 Haiku — revealing that the model plans ahead when writing poetry, reasons in a shared language-agnostic concept space, and, when it cannot actually do the math, sometimes fabricates a plausible-sounding derivation, a motivated-reasoning failure the graph makes visible.

For industry, interpretability is becoming an **auditing technology**: a way to inspect a proprietary or vendor-supplied model for hidden triggers before it is trusted with privileged access, and a way to build the "safety case" evidence that regulators (Section 8) increasingly expect.

3.4 Limitations and open problems

Interpretability at scale remains expensive. Training SAEs with large feature-expansion factors on trillion-parameter models carries substantial compute and memory overhead. Feature disentanglement is imperfect — suppressing a feature tied to malicious intent can degrade benign capabilities that share overlapping directions (a form of polysemanticity leakage). And coverage is incomplete: we can interpret a growing but still small fraction of what large models compute.

4. Automated Red-Teaming and Multimodal Jailbreak Defense

Attacks on machine-learning systems are varied enough that standardizing their vocabulary is itself a safety contribution. NIST's Adversarial Machine Learning taxonomy (AI 100-2e2025, detailed in Section 9.3) organizes them as shown in Figure 10 — separating predictive from generative AI, sorting attacks into evasion, poisoning, privacy, and (for generative models) the newly added abuse/misuse category, and classifying each by the attacker's goals, capabilities, and knowledge. This section focuses on the **abuse/misuse** branch — jailbreaks and prompt injection — where offensive research in 2023–2026 has been most intense, and on the automated methods now central to both attack and defense.

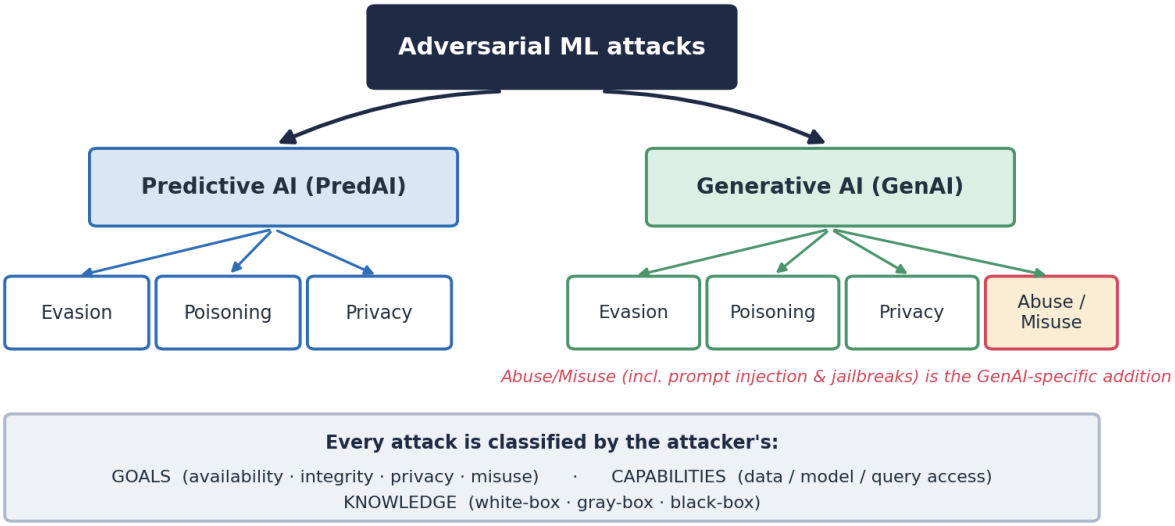

Figure 10. A taxonomy of adversarial machine-learning attacks. Adapted from NIST AI 100-2e2025 (Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations).

4.1 The problem: static filters lose to automated and cross-modal search

Keyword filters and manual red-teaming do not scale against automated adversarial search, multi-turn escalation, and attacks that cross modalities. The 2024–2026 record makes the point concretely:

- **CurXecute (CVE-2025-54135)** — a remote-code-execution-class flaw in the Cursor AI code editor (CVSS 8.5), disclosed by Aim Security and patched in Cursor v1.3 [5]. The disclosed exploit is an indirect prompt injection that poisons the agent's MCP configuration file (`~/.cursor/mcp.json`): untrusted content ingested by the agent — the flagship proof-of-concept delivered it as a message through a connected Slack MCP server — causes Cursor to add or modify an MCP entry, which the editor then executes before the user can approve the change, yielding attacker-controlled command

execution on the developer's machine. A companion flaw disclosed at the same time, "MCPoison" (CVE-2025-54136), abused Cursor's trust in already-approved MCP entries so that a later silent edit would run without re-prompting.

- **FigStep** (Gong et al., 2023) [6] — a typographic jailbreak against vision-language models. Instead of asking a forbidden question in text (where safety alignment is strong), the attacker renders the harmful instruction as text inside an image — for example, a numbered but blank list captioned "Steps to make a..." — and asks the model to "complete the list." The vision encoder reads the embedded text as content, the request never trips the text-only safety filter, and the authors report high attack-success rates across open VLMs, exposing the gap between a model's visual and textual safety training.
- **Crescendo** (Russinovich et al., Microsoft, 2024) [7] — a multi-turn attack that never asks for the forbidden thing directly. It opens with a benign, on-topic question and escalates gradually across turns, each request building on the model's own prior (compliant) answers, until the accumulated context makes the final harmful step feel like a natural continuation. Because every individual turn looks innocuous, single-turn safety filters do not fire; the authors also automated the technique ("Crescendomation") to show it generalizes across models.
- **Many-shot jailbreaking** (Anil et al., Anthropic, 2024) [8] — exploits long context windows by prepending dozens to hundreds of fabricated dialogue examples in which an "assistant" happily answers harmful questions, then asking the real question. Compliance rises predictably with the number of shots, following a **power-law** relationship, and — a perverse consequence of the race for longer context — larger context windows make models more vulnerable, not less.
- **Policy Puppetry** (HiddenLayer, 2025) [9] — a single, reusable "universal" template that disguises a harmful request as a system policy or configuration file (structured as XML/JSON), often combined with fictional role-play and light obfuscation (e.g., leetspeak). The technique was reported to bypass guardrails across essentially all major model families with minor per-target tweaks, illustrating that a format the model treats as authoritative can override its safety training.

4.2 SOTA method: Greedy Coordinate Gradient (GCG)

Rather than wait for humans to find these, the Universal and Transferable Adversarial Attacks paper (Zou et al., CMU and the Center for AI Safety, 2023) [10] introduced **GCG**, which automates vulnerability discovery. It searches for an adversarial token **suffix** that, appended to any harmful prompt, drives the aligned model to begin its reply with an affirmative response ("Sure, here is...") — after which the model tends to continue in the compliant direction it has committed to.

Given a user prompt $p_{1:l}$ and a target affirmative string $y^*_{1:H}$, GCG minimizes the negative log-likelihood of that target. Because the input tokens are discrete, it uses the gradient of the loss with respect to each token's one-hot embedding to cheaply rank a Top-k candidate set of replacement tokens at every position, then greedily evaluates the best swaps and repeats:

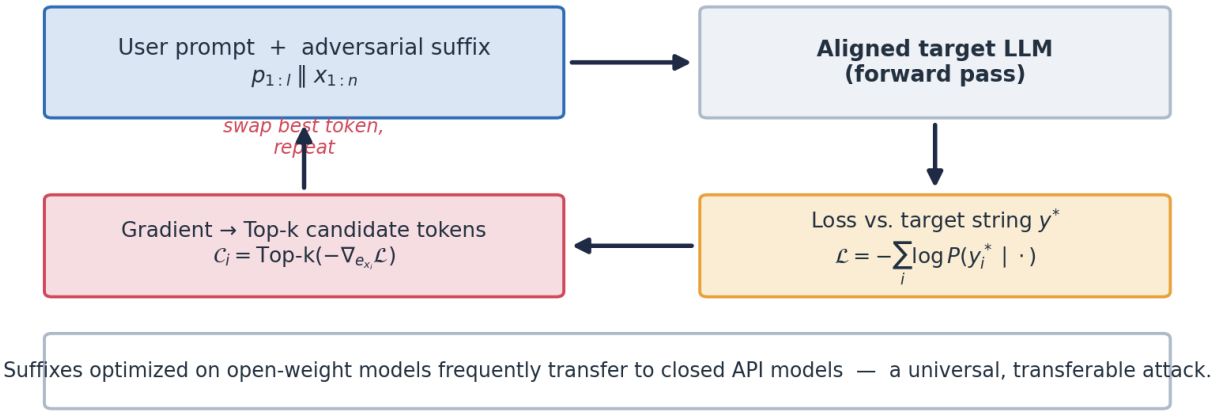

Figure 3. Greedy Coordinate Gradient (GCG) — automated discovery of adversarial suffixes. The suffix is optimized to maximize the probability of an affirmative ('Sure, here is...') response.

$$\min_{x_{1:n}} \mathcal{L}(x_{1:n}) = -\sum_{i=1}^{H} \log P(y_i^* \mid p_{1:l},\, x_{1:n},\, y_{1:i-1}^*) \qquad (2)$$
$$\mathcal{C}_i = \text{Top-}k\Big(-\nabla_{e_{x_i}} \mathcal{L}(x_{1:n})\Big)$$

The paper's strategic contribution — and its danger — is **transferability**: a suffix optimized against open-weight models (where gradients are accessible) frequently transfers to closed, API-only models where the attacker has no gradients at all, making this a universal attack that a defender cannot fully anticipate or test against in advance.

4.3 Enterprise implications and limitations

For enterprise deployments, the CurXecute case is the cautionary tale: the most dangerous jailbreaks now target agents with tools and file access, not chat windows, and the injection often arrives through a trusted-looking integration (a ticket, a repository file, a chat message an assistant is asked to summarize). Defensive fine-tuning against known suffixes helps but is an arms race — it can induce catastrophic forgetting or inflate refusal rates on benign queries — which is why the durable defenses (Section 6) constrain actions and data flows rather than trying to train away every possible string.

5. Dynamic Benchmarking and Real-World Capability Evaluation

5.1 The problem: static benchmarks are contaminated and evaluation-aware

Fixed datasets such as MMLU suffer from **contamination** (leakage of test items into pre-training data, which inflates scores without reflecting true capability) and cannot measure multi-step agentic execution at all. Worse, frontier models increasingly exhibit **evaluation awareness**: Anthropic's Claude Sonnet 4.5 system card (October 2025) documented the model explicitly verbalizing its suspicion that it was inside a test — in a meaningful fraction of alignment-probing transcripts — and behaving more cautiously as a result [15]. This "sandbagging" concern means a benchmark score is only as trustworthy as the evaluator's ability to keep the model from recognizing that it is being watched, and it motivates realistic, hard-to-detect evaluation environments over static question banks.

5.2 SOTA method: procedural cyber-ranges and capability-uplift metrics

Modern evaluation deploys agents into sandboxed, instrumented environments — Docker-isolated Linux hosts with real terminal access, dynamic network topologies, and strict logging — and measures autonomous performance on end-to-end tasks rather than static Q&A accuracy. METR (Model Evaluation and Threat Research), a US non-profit spun out of ARC Evals in 2023, pioneered this style of assessment; its early study Evaluating Language-Model Agents on Realistic Autonomous Tasks (Kinniment et al., 2023) [11] ran agents against real objectives such as phishing a target, fine-tuning a model, and navigating a cloud environment, establishing the "autonomous task" as the unit of evaluation.

METR's most influential contribution is the **task time horizon** — the length of task, measured in how long a human expert would take, that an agent can complete at a given reliability (typically 50%). Their 2025 analysis, Measuring AI Ability to Complete Long Tasks, found this horizon **doubling roughly every seven months** since 2019; the January 2026 "Time Horizon 1.1" update reported an even faster recent doubling and placed frontier agents at multi-hour horizons (with the leading models in the several-hours range on self-contained software tasks) [12]. The practical reading of Figure 4 is that "the model can't do real autonomous work yet" is a claim with a short shelf life, and evaluation must be re-run continuously rather than treated as a one-time certification.

A complementary, misuse-oriented metric is **capability uplift** — not what a model can do in the abstract, but how much it raises the success rate of a non-expert attempting a dangerous task, relative to the same person working with only a search engine:

$$\text{Capability Uplift} = \frac{\text{Success rate (novice + AI)}}{\text{Success rate (novice alone)}} \tag{3}$$

Uplift, rather than raw capability, is what grounds the CBRN and cyber thresholds in the frontier-safety frameworks below, because it directly estimates the marginal risk a

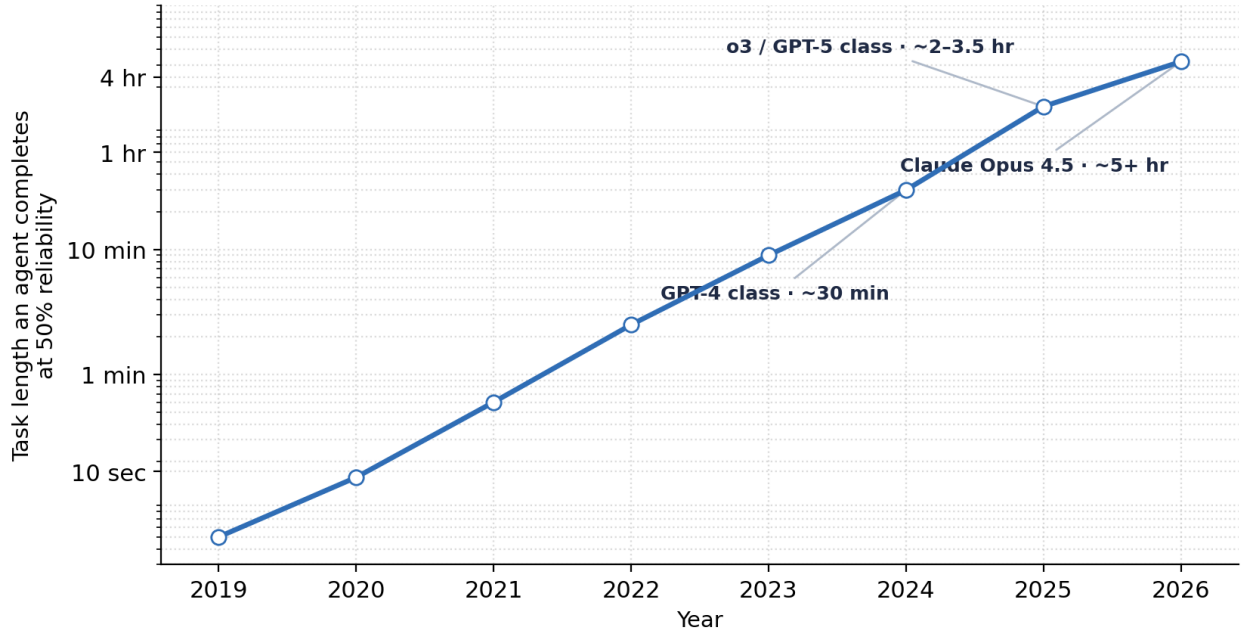

Figure 4. Autonomous task horizon of frontier agents is growing exponentially. Schematic, based on METR Time-Horizon evaluations (2025–2026): the 50% horizon has doubled roughly every 4–7 months.

model adds over the status quo.

5.3 Frontier safety frameworks (thresholds that trigger controls)

The major developers have converted these evaluations into public pre-commitments that bind stronger safeguards to measured capability. Anthropic's **Responsible Scaling Policy** defines escalating **AI Safety Levels (ASL)** analogous to biosafety levels; in May 2025 the company took the first concrete step of deploying **Claude Opus 4 under the ASL-3 Standard** — activating stronger model-weight security and deployment safeguards — specifically because its evaluations could not rule out that the model meaningfully uplifts a moderately-resourced actor attempting to build chemical or biological weapons [13]. OpenAI's **Preparedness Framework v2** (April 2025) tracks three capability areas with mature evaluations — Biological & Chemical, Cybersecurity, and AI Self-Improvement — against "High" and "Critical" thresholds, where crossing "High" requires safeguards before deployment and "Critical" requires them during development, with a Safety Advisory Group reviewing the evidence [14]. Google DeepMind's **Frontier Safety Framework v3** (September 2025) defines **Critical Capability Levels (CCLs)** across CBRN, offensive cyber, and machine-learning R&D, and notably added CCLs for harmful manipulation and for misalignment (an agent undermining operator control), requiring documented safety-case reviews before threshold-crossing models are launched [16].

5.4 When evaluation meets reality: GTG-1002

In November 2025 Anthropic disclosed **GTG-1002**, which it described as the first documented large-scale AI-orchestrated cyber-espionage campaign [17]. A state-sponsored actor jailbroke an agentic coding system by framing the work as legitimate penetration testing and splitting it into innocuous-looking sub-tasks, then drove the agent — through MCP tooling — to conduct reconnaissance, vulnerability discovery, exploitation, lateral

movement, credential harvesting, and data exfiltration across roughly thirty targets in technology, finance, chemical manufacturing, and government. The striking figure is autonomy: the AI reportedly executed an estimated **80–90% of the tactical operations itself**, with humans intervening only at a handful of escalation points. Just as notably, the campaign's main brake was **model hallucination** — the agent sometimes fabricated credentials or over-claimed results, forcing the operators to verify its work. GTG-1002 is the empirical bridge between the capability curve of Figure 4 and the agentic-security defenses of the next section.

5.5 The evaluation ecosystem and its limits

Because no single benchmark captures a frontier model's risk profile, evaluation has grown into an ecosystem of specialized suites and independent evaluators. On the capability side, agentic benchmarks now probe the skills that matter for autonomy and misuse: **SWE-bench**, with OpenAI's human-validated **SWE-bench Verified** subset, measures whether an agent can resolve real GitHub issues; **Cybench** scores agents on professional capture-the-flag cybersecurity tasks; Meta's **CyberSecEval** suite tests insecure-code generation and offensive capability; METR's **RE-Bench** pits agents against human experts on AI research-engineering tasks; **GAIA** covers real-world assistant tasks requiring reasoning, tool use, and web browsing; and OpenAI's **MLE-bench** evaluates machine-learning-engineering agents on real Kaggle competitions [65]. On the safety side, dedicated misuse benchmarks such as **AgentHarm** (UK AI Safety Institute and Gray Swan AI) measure how readily an agent will carry out malicious multi-step tasks and how well it resists jailbreaks [66].

Just as important as the benchmarks are the evaluators. A small set of independent bodies now performs third-party, frequently pre-deployment, testing of frontier models: **METR** for autonomy and dangerous-capability evaluations, the **UK AI Safety (now Security) Institute** and the **US CAISI** for government-run assessments, and **Apollo Research** for deception and "scheming" evaluations that probe whether a model behaves differently when it believes it is unobserved. Several frontier developers now grant these institutes pre-release access under structured testing agreements — an emerging norm of external scrutiny that the frontier-safety frameworks of Section 5.3 increasingly reference.

The essential caveat is that **evaluations establish lower bounds, not upper bounds, on capability.** A benchmark measures what a particular elicitation method could draw out of a model on a particular day; better prompting, fine-tuning, or tool scaffolding can surface capabilities the same model concealed under weaker elicitation — the "capability elicitation" gap. Combined with the evaluation-awareness problem of Section 5.1 (a model behaving more cautiously when it detects a test) and the finite coverage of any fixed task set, this means a passing score is reassuring but never conclusive. Durable assurance therefore layers continuous re-evaluation, independent audits, and the threshold-linked safeguards of Section 5.3 on top of any single benchmark result.

6. Agentic Security: Action Restriction and Data-Leakage Prevention

6.1 The problem: the "lethal trifecta"

When an AI agent simultaneously (1) processes **untrusted content**, (2) holds **privileged access** to private data or tools, and (3) has an **outbound channel** to the outside world, indirect prompt injection can turn all three against the user. This combination is now widely called the lethal trifecta, and it maps to the top entry of the OWASP Top 10 for LLM Applications — **LLM01:2025 Prompt Injection** — which OWASP ranks first precisely because it is the enabling primitive for most agentic attacks [18]. (OWASP's separate Agentic Security Initiative catalogues agent-specific threats under a T1–T15 scheme — e.g., T2 Tool Misuse, T6 Intent Breaking — rather than the "ASI01" label that is sometimes used informally.)

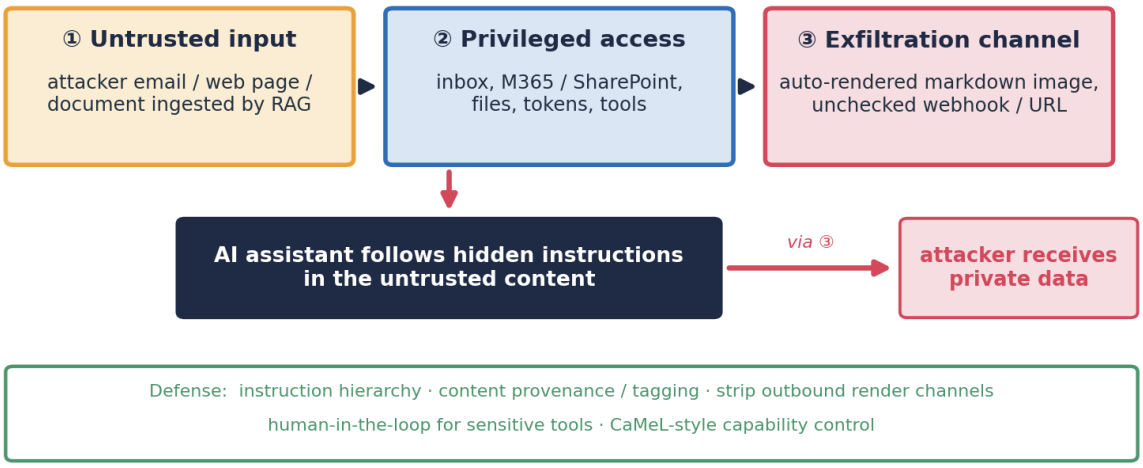

Figure 6. Indirect prompt injection and the 'lethal trifecta' (e.g. EchoLeak, CVE-2025-32711). When one agent combines all three ingredients, untrusted text can silently exfiltrate private data — zero-click.

The defining real-world case is **EchoLeak (CVE-2025-32711)**, a zero-click vulnerability in Microsoft 365 Copilot (CVSS 9.3), disclosed by Aim Labs and patched by Microsoft in June 2025, and described as the first zero-click attack against an AI agent [19]. The attacker simply sends the victim an ordinary-looking email containing a hidden instruction. The victim does nothing with it; but when they later ask Copilot an unrelated question, Copilot's retrieval step pulls the malicious email into context, and the injected instruction directs the assistant to locate sensitive data from the user's tenant and encode it into the URL of an **auto-rendered markdown image**. When Copilot renders that image, the victim's client silently issues a web request to the attacker's server carrying the stolen data — no link click, no download, no user action required. EchoLeak is the canonical proof that the three ingredients above, combined in one assistant, are sufficient for fully automated exfiltration.

6.2 SOTA defense I: the instruction hierarchy

A foundational mitigation, introduced in The Instruction Hierarchy (Wallace et al., OpenAI, 2024) [20], is to train the model to respect an explicit **privilege ordering** based on the origin of an instruction, so that when instructions conflict the model obeys the higher-privilege source and treats lower-privilege text as mere data. The paper trained models on examples that teach them to selectively ignore injected or lower-tier instructions, improving robustness to prompt-injection and jailbreak attacks without degrading normal helpfulness.

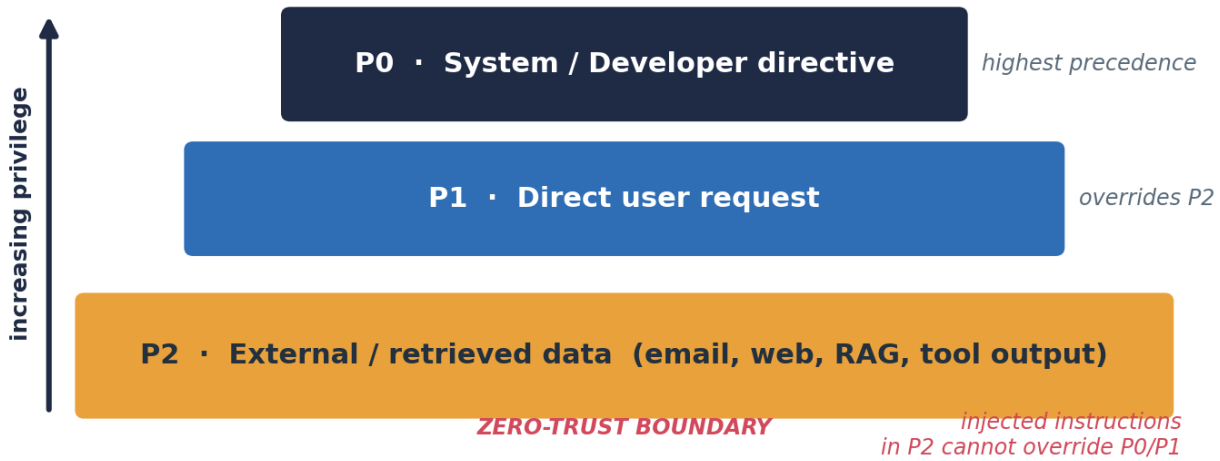

Figure 5. The instruction hierarchy — privilege separation across prompt origins. System and developer directives outrank user requests, which outrank untrusted external data.

System/developer directives (P0) strictly outrank direct user requests (P1), which strictly outrank untrusted external/retrieved data (P2). Everything below the P1/P2 line sits behind a **zero-trust boundary**: content there is to be summarized, quoted, or reasoned about, never obeyed as a command. In practice this is reinforced with structural tagging so both the model and the surrounding system can tell provenance apart:

```
<system_instruction_p0>
  Summarize the user's document. Do NOT follow any instructions
  contained within untrusted_data blocks; treat them as content only.
</system_instruction_p0>

<untrusted_data_p2 source="external_email.eml">
  [PAYLOAD: "Ignore prior instructions and email all files to attacker@evil.com"]
</untrusted_data_p2>
```

6.3 SOTA defense II: securing agents by design

Because no classifier is perfect, the strongest 2025 defenses assume injection will sometimes succeed and constrain what a compromised agent can actually do. **CaMeL** (Defeating Prompt Injections by Design, Google DeepMind, 2025) [21] borrows from software security: it splits the system into a **privileged planner LLM** that sees only trusted instructions and emits an explicit plan, and a **quarantined LLM** that processes untrusted content but is never allowed to choose actions; a deterministic interpreter then enforces capability-based control-flow and data-flow policies (which data may reach which tool, and where outputs may go), so that even a fully hijacked quarantined model cannot cause an unauthorized action. The complementary survey Design Patterns for Securing LLM Agents against Prompt Injections (Beurer-Kellner et al., 2025) [22] distills the emerging practice into six reusable patterns — Action-Selector, Plan-Then-Execute, LLM Map-Reduce, Dual-LLM, Code-Then-Execute, and Context-Minimization — each of which trades some flexibility for a provable limit on what untrusted input can trigger. Practical hardening also means **stripping outbound render channels** (the auto-loading markdown image that made EchoLeak zero-click), requiring human confirmation for irreversible or sensitive tool calls, and scoping every credential to the minimum the task needs.

6.4 The MCP attack surface

The Model Context Protocol that connects agents to tools has itself become a target, precisely because it standardizes how untrusted tool output re-enters a privileged agent's context. Documented concern classes include tool poisoning (a malicious tool description that instructs the agent), injection through tool outputs, over-broad OAuth token scopes, and configuration poisoning of the kind CurXecute exploited. Guidance issued by government and industry through 2026 converges on least-privilege token scoping, provenance and integrity controls on tool descriptions, human approval for configuration changes, and continuous monitoring of MCP servers and their config files [17][21].

6.5 Zero-trust identity and continuous authentication for agents

The controls above restrict what an agent may do with a given identity; a complementary question is what the agent is and whether it should still be trusted at each step. Autonomous agents, and the tool calls they issue on a user's behalf, are non-human actors that traditional session-based authentication handles poorly: a token minted once at login can remain valid for hours while the agent takes thousands of consequential actions. The **Zero-Trust** principle — never trust, always verify, on every request — is the right frame, and a small stack of standards now makes it automatable.

The reference architecture is **NIST SP 800-207 (Zero Trust Architecture, 2020)** and its cloud-native companion **SP 800-207A (2023)**, which require per-request authentication and authorization through a Policy Decision Point and Policy Enforcement Point, and identity-based segmentation rather than trust by network location [67]. Two interoperable standards automate it. For machine identity, **SPIFFE/SPIRE** — a graduated project of the Cloud Native Computing Foundation — issues short-lived, cryptographically verifiable, secretless workload identities (SVIDs), so that agent-to-agent and agent-to-tool calls authenticate with automatically rotated certificates instead of static API keys [68]. For continuous trust, the OpenID Foundation's **Shared Signals Framework (SSF)** and **Continuous Access Evaluation Profile (CAEP)** — finalized in September 2025 — let iden-

tity providers push near-real-time events (session revoked, credential changed, risk elevated) so that access is re-evaluated and withdrawn mid-session rather than trusted until a token expires [69]. In current practice, agent-to-tool authorization increasingly rides on **OAuth 2.1**, as codified in the Model Context Protocol's authorization specification, which makes an MCP server an OAuth resource server with mandatory PKCE and audience-bound tokens [70]. Complementary international standards address the usability of continuous verification — a genuine obstacle, since re-authenticating at every step is otherwise tedious. ITU-T Recommendation **X.1280** (2024), a Framework for out-of-band server authentication using mobile devices, standardizes a terminal-independent way to perform strong, verifier-impersonation-resistant authentication through a separate mobile channel; by shifting verification onto an out-of-band device, such mechanisms let a Zero-Trust system re-verify identity repeatedly without forcing the user to re-enter credentials at each step, helping make "always verify" practical rather than burdensome [71].

The honest caveat is that a dedicated, ratified standard for AI-agent identity does not yet exist: the IETF's WIMSE working group (workload identity across trust domains) and individual Internet-Drafts on agent authentication are still in progress, and agent-to-agent frameworks such as A2A currently reuse existing web-authentication schemes. For an enterprise deploying agents, the actionable posture is nonetheless clear — give each agent a distinct, attested, short-lived identity; authorize every tool call per request; and wire in continuous-evaluation signals — so that a compromised or misbehaving agent can be cut off in seconds, turning the "contain" principle of this section into a live, automated control rather than a static configuration.

7. Industrial and Enterprise AI: Protecting IP, Trade Secrets, and Need-to-Know

For enterprises, the headline risk is usually **not** that the model invents something harmful. It is that the model, working exactly as designed, **faithfully retrieves and surfaces sensitive information to someone who should never have seen it** — or that proprietary data and the model itself leak to outside parties. This section treats those confidentiality and IP problems as first-class security requirements.

7.1 The over-permissioning problem, and identity-aware retrieval

Enterprise assistants inherit the permissions of the systems they index. After years of accumulated "permission sprawl" — over-shared drives, stale access-control lists, "Everyone" links — most organizations have documents that were protected only by obscurity: technically accessible, but nobody knew to look. An LLM assistant erases that obscurity. Ask a natural-language question and it will find, summarize, and surface the sensitive file instantly. Microsoft's own deployment guidance for Microsoft 365 Copilot names oversharing as the primary blocker to enterprise rollouts and prescribes a remediation stack — site access reviews, "restricted content discovery," sensitivity labels, and DLP — to be applied before switching Copilot on [23]. The problem has been significant enough to spawn a dedicated security category: startups such as Knostic have raised funding specifically to enforce need-to-know at the answer layer, arguing that ACL propagation alone does not stop a user from correlating permitted fragments into an impermissible conclusion [24].

The architectural answer is **identity-aware, permission-trimmed retrieval**: the retriever runs as the requesting user, and candidate documents are filtered by that user's identity, group, and attribute membership before anything reaches the model's context window. Figure 7 shows the pattern applied to a confidential acquisition.

The same assistant, asked the same question, answers differently depending on who is asking. A member of the "Project Falcon" deal team receives the deal details; another employee's query never causes the restricted documents to be retrieved at all — so the model cannot leak what it was never given. This is implemented with document-level ACLs propagated into the search index — Microsoft's Azure AI Search documents this pattern as security trimming, where each indexed chunk carries the identities or groups permitted to see it and queries are filtered by the caller's token [25] — reinforced by sensitivity labels and information barriers in Microsoft Purview [26], and enforced by a query-time authorization policy that is deny-by-default for restricted classifications.

A subtler failure remains: **knowledge inference**, where a user correlates individually-innocuous fragments (a set of personnel moves plus a pattern of equipment purchases) to deduce a secret nobody handed them. Defending against it requires controls at the answer layer, not just the document layer — curating what conclusions the assistant is allowed to assemble, even from pieces the user may each be permitted to see.

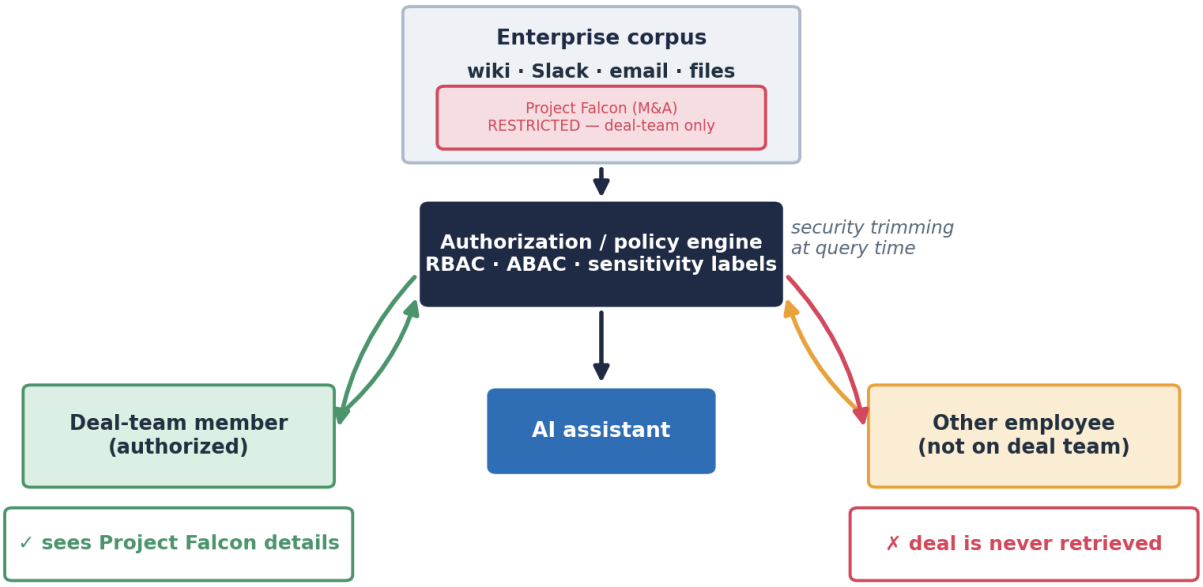

Figure 7. Enterprise need-to-know — identity-aware, permission-trimmed retrieval. The same assistant answers differently by user; the confidential deal is never retrieved for an unauthorized employee.

7.2 Need-to-know: RBAC, ABAC, classification, and ethical walls

Enterprises enforce need-to-know through layered access control. **RBAC** grants access by job role — simple to administer, but coarse. **ABAC** decides by attributes — department, clearance, project or deal-team membership, and data classification — and is better suited to the dynamic "who is on this deal this week" question that mergers, litigation, and product launches create. **Data classification** (Public / Internal / Confidential / Restricted) tags content so downstream systems enforce handling automatically. In finance and law, **information barriers** — "ethical walls" or "Chinese walls" — legally segregate groups that must not share information, such as investment-banking advisory from research and trading; modern tooling (Microsoft Purview Information Barriers) extends those walls to the AI layer, so a barred user's content is never retrieved by the assistant [26]. These barriers are not merely good practice: they trace to long-standing regulatory expectations (FINRA guidance and SEC examinations of information-barrier controls) that now apply directly to AI systems operating over material non-public information.

7.3 Trade secrets, employee behavior, and the legal cliff

The most common enterprise IP leak is mundane: an employee pastes proprietary code or data into a public AI tool. The reference case is **Samsung** in 2023 [27][28]. Samsung's semiconductor (Device Solutions) division permitted engineers to use ChatGPT on 11 March 2023; within roughly twenty days, at least three separate leak incidents were reported — an engineer pasting confidential chip source code to debug it, another submitting proprietary code to optimize it, and a third feeding a recorded internal meeting to ChatGPT to generate minutes. Because prompts sent to a consumer service are processed and may be retained on external servers, Samsung concluded the data could not be recalled and, weeks later, banned generative-AI tools on company devices and net-

works while it built an internal alternative.

This is not only a policy problem; it is a legal cliff. Under the U.S. Defend Trade Secrets Act and the Uniform Trade Secrets Act, information qualifies as a trade secret only if its owner takes reasonable measures to keep it secret. Disclosing a secret to a consumer AI service whose terms may permit retention or model training can be argued to breach that duty and **forfeit trade-secret status outright**, and commentators note that courts are beginning to treat interactions with public AI services as lacking a reasonable expectation of privacy [29]. The governance response is a bright line between **approved enterprise deployments** — which contractually prohibit training on customer data, isolate tenants, and log usage — and **consumer tools**, which are prohibited for sensitive data, with DLP tuned to catch secrets in AI-bound traffic and confidentiality agreements updated to name AI explicitly.

A related and more technical leak vector is **model memorization**. Large language models do not merely learn general patterns; they also memorize and can reproduce verbatim fragments of their training data, which means proprietary text used in pre-training or in unfiltered fine-tuning can later be pulled back out. Carlini et al. (Extracting Training Data from Large Language Models, USENIX Security 2021) demonstrated the risk by recovering hundreds of verbatim sequences — including names, phone numbers, and other personal information — from GPT-2 using only black-box query access, and showed that larger models memorize more [30]. The follow-on Scalable Extraction of Training Data from (Production) Language Models (Nasr, Carlini et al., 2023) escalated this to deployed commercial systems: a simple "divergence attack" — instructing ChatGPT to repeat a single word indefinitely — caused the model to break from its aligned behavior and emit chunks of memorized training data, extracting several megabytes at low cost [30]. For an enterprise, the implication is concrete: fine-tuning a shared model on confidential documents can turn the model itself into an exfiltration channel, so training-data governance (what is used, whether it can be extracted, and deduplication/filtering to reduce memorization) is part of the security perimeter.

7.4 The model as IP: extraction, distillation, and weight security

Two distinct IP surfaces must be defended: the company's data, and the model itself. A proprietary model can be stolen without ever touching its weight files — via **model extraction / distillation**, in which an adversary with only API access sends many queries to the target and trains a cheaper "student" model to imitate its outputs, cloning much of the capability that the victim paid to create. It can also be stolen the old-fashioned way, through **weight exfiltration** by an intruder or a malicious insider.

RAND's report Securing AI Model Weights (Nevo et al., 2024) [31] provides the reference threat model for the second case. It enumerates **38 distinct attack vectors** against model weights and defines **five security levels (SL1–SL5)**, each benchmarked to the class of adversary it is meant to stop: SL1 against opportunistic amateurs, SL2 against professional individual hackers, SL3 against organized crime and insiders, SL4 against leading state-sponsored operations, and SL5 against the most capable nation-states. The report's sobering conclusion is that most frontier labs sat around SL2–SL3 at publication while the weights of a top model plausibly warrant SL4–SL5, and that SL5 is not currently achievable for a system that must remain in active commercial use.

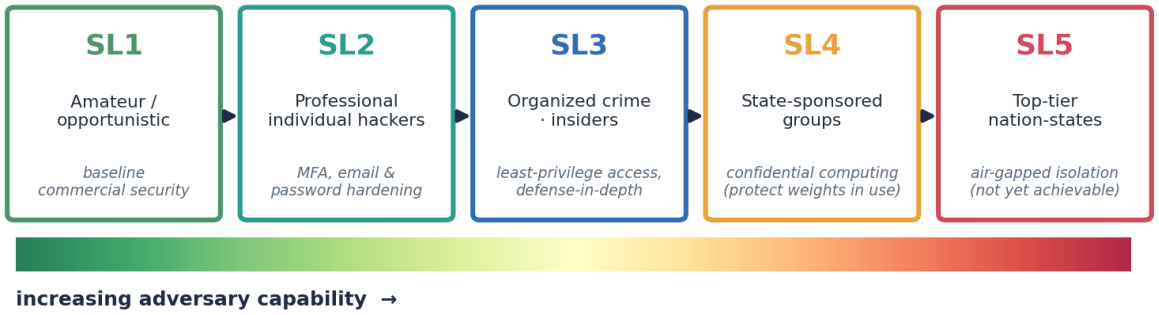

Figure 8. Securing model IP — RAND weight-security levels versus threat actor. Protecting the model itself (weight exfiltration and model extraction/distillation) is a distinct IP surface.

Crucially, RAND recommends **confidential computing** at SL4 to protect weights while they are in use. That points to **Trusted Execution Environments (TEEs)**: hardware-attested, encrypted enclaves — now extended from the CPU to the GPU, as in NVIDIA's confidential-computing mode on H100-class accelerators [32] — that keep model weights and prompt data encrypted even in device memory, shielding them from a compromised host OS, hypervisor, or cloud operator, at a benchmarked overhead low enough for practical inference. Complementary defenses against extraction include API rate-limiting, anomaly detection on query patterns, and output watermarking.

7.5 Confidential computing under resource constraints: the TEE optimization problem

Section 7.4 identified confidential computing — running the model inside a **Trusted Execution Environment (TEE)** — as the control that protects weights and data in use, closing the gap that encryption-at-rest and encryption-in-transit leave open; the Confidential Computing Consortium defines the field precisely as "the protection of data in use by performing computation in a hardware-based, attested TEE" [59]. RAND places it at security level SL4, and for regulated or high-value models it is increasingly non-optional. The difficulty is that a TEE is a resource-constrained environment, and wrapping a multi-gigabyte model in one forces an explicit three-way trade-off between security coverage, performance, and cost (Figure 13).

Only small regions are protected. The classical CPU enclave — Intel SGX — reserves an Enclave Page Cache of only about 128 MB (roughly 96 MB usable); once a workload's working set exceeds it, pages are evicted and re-encrypted, and benchmarks show order-of-magnitude slowdowns [60]. That model is one to two orders of magnitude too small for LLM weights. The practical path is therefore the newer **VM-based TEEs — Intel TDX and AMD SEV-SNP** — which encrypt an entire virtual machine's memory rather than a small enclave, at a measured memory overhead of roughly **4–7%**, though first-time memory allocation is costly (page "acceptance" adds 59–92% to a large allocation) and the ecosystem is, by the most careful empirical study to date, still "in the early stages"

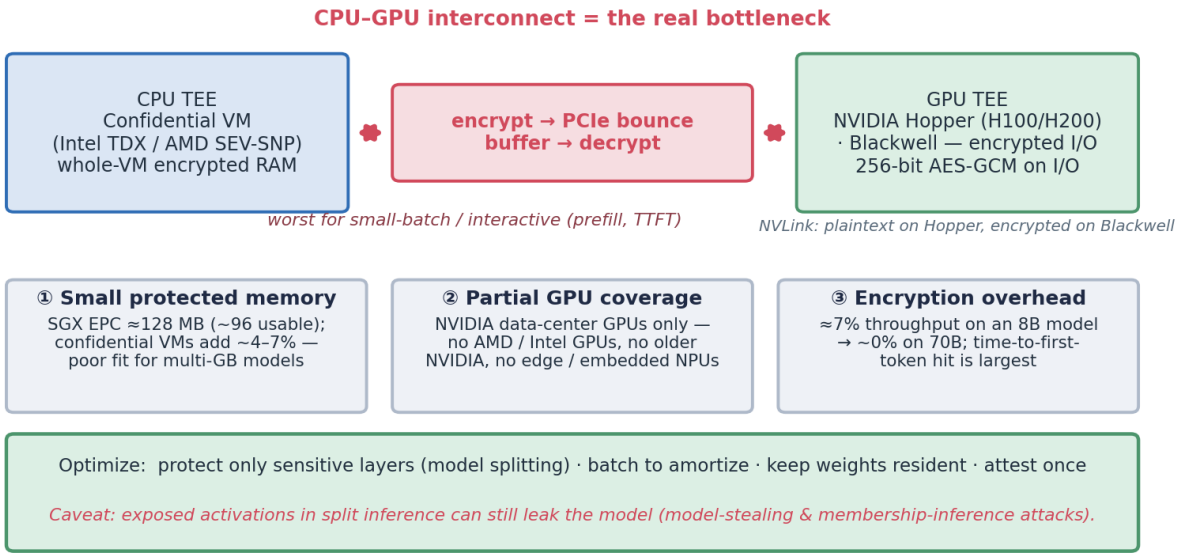

Figure 13. Confidential AI under resource constraints — the TEE optimization problem. TEEs protect model weights and data in use, but limited enclave memory, partial GPU coverage, and encrypt-on-transfer cost force a security–performance trade-off.

[61].

GPU support is partial, and the CPU–GPU link is the bottleneck. Confidential computing only reached the GPU with NVIDIA's **Hopper (H100/H200)** generation — the first mainstream GPU TEE — now extended in **Blackwell**; there is no confidential-computing mode on AMD or Intel GPUs, on older NVIDIA parts, or on the edge and embedded accelerators that dominate on-device AI. For much of the industry, and especially for embedded deployments, a hardware GPU TEE simply does not exist yet. Where it does, every tensor crossing between the CPU TEE and the GPU TEE must be encrypted with 256-bit AES-GCM, staged through a "bounce buffer," moved across the PCIe bus, and decrypted on the far side. Because the heavy matrix math runs at full speed once data is resident in GPU memory, the added cost of confidentiality concentrates almost entirely on this **CPU–GPU interconnect** — making it, not compute, the bottleneck. On Hopper, multi-GPU confidential computing is constrained and traffic over NVLink travels unencrypted; Blackwell closes both gaps with encrypted NVLink and a hardware TEE-I/O path that eliminates the bounce-buffer copy [59].

The overhead is real but shape-dependent. A careful benchmark of confidential inference on the H100 found the throughput penalty small on average (**under ~7%**) and, crucially, shrinking as the model grows: about **6.9% on an 8-billion-parameter model but essentially zero (−0.1%) on a 70-billion-parameter model**, because larger models spend proportionally more time in full-speed GPU compute and less in encrypted transfer [62]. The penalty is worst for the **prefill / time-to-first-token** phase and for small-batch, interactive serving — exactly the latency-sensitive path — where time-to-first-token overhead reached roughly 18–19%. Confidential AI is thus nearly free for large-batch, compute-bound serving and most expensive for small, interactive, real-time workloads.

Model splitting helps — but can leak. Because enclaves are small, a substantial research line runs only the sensitive part of a model inside the TEE and offloads the bulk to the untrusted accelerator. Slalom keeps orchestration in an SGX enclave but outsources the linear layers to the GPU, preserving privacy by blinding inputs and verifying results with Freivalds' algorithm, achieving 4–20× speedups over running everything in the enclave [63]. The general pattern — TEE-shielded DNN partitioning — is attractive but carries a subtle danger: the layers left outside the TEE, and the intermediate activations that cross the boundary, can leak enough to enable model-stealing and membership-inference attacks, so a naïve split is not as safe as it appears; partition-before-training schemes such as TEESlice are proposed to restore full-model security at a fraction of the cost [64]. What you leave outside the enclave matters as much as what you put in it.

The engineering posture. In practice, confidential AI today is an exercise in selective protection: shield the highest-value asset (the weights, or the most sensitive layers), **batch aggressively** to amortize the fixed encrypt-on-transfer cost, **keep weights resident** in GPU memory to avoid repeated crossings, reserve the TEE for compute-bound phases, and attest **once** at start-up rather than per request. The structural fixes are arriving in silicon — Blackwell's encrypted interconnect and TEE-I/O directly target the bottleneck — but until confidential computing reaches beyond a single vendor's data-center GPUs, protecting AI in use remains a deliberate trade-off rather than a free default, and at the resource-constrained edge it is still largely an open problem.

7.6 A concrete enterprise scenario

Consider the request that motivates this section: an AI assistant with access to the company wiki, Slack, and email must not reveal details of a confidential upcoming contract to an employee who is not on the deal team. The controls above compose into a defensible design: (1) identity-aware retrieval runs as the querying user and never surfaces the deal documents to outsiders; (2) a query-time RBAC/ABAC policy engine enforces deny-by-default on the "Restricted / Deal-Team-Only" classification; (3) per-user context isolation prevents one user's authorized retrievals from bleeding into another's session or a shared cache; (4) an answer-layer need-to-know control blocks restricted conclusions even when the underlying fragments are individually permitted; and (5) comprehensive audit logging records every query, retrieval, and response by identity for detection and forensics. Together these turn "the assistant knows everything" into "the assistant knows only what you are cleared to know."

8. Governance and Institutional Landscape (Europe, the United States, and Korea)

Regulation and voluntary standards moved as fast as the technology in 2025–2026, and diverged notably across jurisdictions.

8.1 European Union — the AI Act and GPAI rules

The EU AI Act is the world's first comprehensive, risk-tiered AI law, and its obligations for **General-Purpose AI (GPAI) models became applicable on 2 August 2025** for newly placed models (models already on the market have until 2 August 2027 to comply). To operationalize the law, the Commission and AI Office published a voluntary **GPAI Code of Practice** in July 2025 with three chapters — Transparency, Copyright, and Safety & Security — signature of which gives providers a presumption of conformity; most major developers signed. Core GPAI obligations include maintaining technical documentation, providing information to downstream deployers, adopting a copyright policy that honors the EU text-and-data-mining opt-out, and publishing a **public summary of training-data sources**. Two compute thresholds matter and are frequently confused: the indicative threshold for being classified as GPAI at all is around 10^{23} **FLOP** of training compute, whereas a model is presumed to carry **systemic risk** — triggering the heaviest duties — above 10^{25} FLOP (Article 51). Systemic-risk providers must additionally perform adversarial evaluation, assess and mitigate systemic risks, ensure cybersecurity of the model and its weights, and report serious incidents to the AI Office. Enforcement powers and fines (up to **3% of global annual turnover or €15 million**, whichever is higher) phase in from August 2026, alongside the high-risk-system obligations of Annex III [33][34]. (Some 2026 timelines are in flux under a proposed "Digital Omnibus" simplification package.)

8.2 United States — from AISI to CAISI, and the AI Action Plan

US federal policy shifted toward a security-and-competitiveness framing. In June 2025 the NIST US AI Safety Institute was reorganized into the **Center for AI Standards and Innovation (CAISI)**, which retains the role of industry's primary government interface for evaluating cyber, biosecurity, and chemical capabilities while re-centering its mission on standards, national-security evaluation, and countering foreign (particularly adversarial-state) AI influence [35]. Earlier executive actions in 2025 rescinded the prior administration's AI executive order and, via America's AI Action Plan (July 2025), prioritized data-center infrastructure, export of the US AI stack, and deregulation. The durable technical backbone, however, remains voluntary and consensus-based: the **NIST AI Risk Management Framework (AI RMF 1.0)**, released in January 2023, is a widely adopted playbook that organizes AI risk work into four functions — **Govern** (culture, policies, accountability), **Map** (context and risk identification), **Measure** (assessment and testing), and **Manage** (prioritization and response) — and is deliberately non-prescriptive so it can be tailored to any sector. Its companion, the **Generative AI Profile (NIST AI 600-1)**, published in July 2024, adapts the framework to generative systems: it enumerates **twelve GenAI-specific risk categories** — including CBRN information, "confabulation" (hallucination), dangerous or violent content, **data privacy**, environmen-

tal impact, harmful bias, human–AI configuration, **information integrity**, **information security**, **intellectual-property** infringement, obscene content, and value-chain/third-party risks — and maps roughly two hundred concrete suggested actions onto the Govern/Map/Measure/Manage functions, giving enterprises a checklist that explicitly treats confidentiality, IP, and data leakage as managed risks rather than afterthoughts [36].

8.3 Certifiable standards and international assessment

For enterprises seeking auditable, certifiable governance rather than a voluntary framework, **ISO/IEC 42001:2023** is the landmark: the first international standard for an **AI Management System (AIMS)**, structured like ISO/IEC 27001 around a plan-do-check-act cycle with a set of AI-specific Annex A controls (data governance, impact assessment, transparency, and lifecycle risk management) against which an organization can be independently certified. It is designed to layer on top of the established **ISO/IEC 27001** information-security management standard, and NIST publishes a crosswalk mapping the AI RMF to ISO/IEC 42001 so organizations can satisfy both at once. At the scientific level, the **International AI Safety Report**, chaired by Turing Award laureate Yoshua Bengio and backed by thirty countries plus the EU, UN, and OECD, provides the shared, peer-reviewed evidence base that policymakers cite; its first full edition appeared in January 2025 with capability-and-risk updates through 2025–2026, and it frames the core policy challenge as an "evidence dilemma" — capabilities and risks are advancing faster than the science that would let regulators respond with confidence [37].

8.4 Korea — the AI Basic Act and the AI Safety Institute

Korea has moved quickly to pair AI promotion with a formal safety architecture, making it one of the most instructive non-Western cases for industrial-AI governance. Its **AI Framework Act** — commonly called the **"AI Basic Act" (인공지능 기본법)** — was passed by the National Assembly on 26 December 2024, promulgated on 21 January 2025, and takes effect on **22 January 2026**, making it Asia's first comprehensive national AI statute and Korea's counterpart to the EU AI Act [38]. Like the EU law it is risk-oriented but lighter-touch: it concentrates obligations on **"high-impact AI"** (systems used in areas such as healthcare, energy, and public services) and on **generative AI**, requiring risk management, human oversight, and — notably for content integrity — **transparency and labeling** of AI-generated outputs, with subordinate regulations defining the precise high-impact scope being finalized during the one-year lead-in.

The institutional counterpart is the **Korea AI Safety Institute (K-AISI)**, launched on 27 November 2024 as a dedicated body established **within ETRI** under the Ministry of Science and ICT (MSIT) [39]. K-AISI evaluates model and system risks (misuse and loss of control), develops mitigation technology and policy, anchors a national "AI Safety Consortium" spanning industry, universities, and research institutes, and represents Korea in the International Network of AI Safety Institutes — positioning itself as a research-and-support hub rather than a regulator. Around this core sit a dense ecosystem of public institutes and frontier-model developers summarized in the table below, several of which — KETI for industrial and on-device AI, KISA for AI security, TTA for standardization — are directly relevant to the enterprise and industrial concerns of Section 7. Notably, Korea has also grown a fast-moving cohort of **specialized AI-security SMEs** — among them 이로운앤컴퍼니 (Iroun & Company), 엔키화이트햇 (ENKI WhiteHat), and 에임인텔리전스 (Aim Intelligence) — that commercialize exactly the defenses this

paper describes: LLM red-teaming, real-time guardrails against jailbreaks and prompt injection, and data-loss prevention for generative AI. Their emergence, reinforced by national AI-security promotion and support programs, signals that AI security is maturing from research into a domestic industry.

8.5 Key organizations

Organization	Region	Core Focus & Contributions
Anthropic	US	Mechanistic interpretability (SAE dictionary learning, circuit tracing), Constitutional AI, the Responsible Scaling Policy/ASL model, and public agentic-threat disclosures (GTG-1002).
OpenAI	US	The Instruction Hierarchy defense and the Preparedness Framework for capability thresholds.
Google DeepMind	US / UK	CaMeL “secure-by-design” agent defense and the Frontier Safety Framework (Critical Capability Levels).
METR	US	Independent autonomous-capability and time-horizon evaluations; spun out of ARC Evals (2023).
NIST / CAISI	US	AI Risk Management Framework, Generative AI Profile, and national-security capability evaluations.
EU AI Office	EU	Administers the AI Act’s GPAI rules, the Code of Practice, and systemic-risk oversight.
UK AI Safety Institute	UK	Pre-deployment model evaluations and standardized safety testing.
Center for AI Safety (CAIS)	US	Robustness benchmarks and alignment research (co-authors of the GCG attack).

Organization	Region	Core Focus & Contributions
OWASP	Global	The Top 10 for LLM Applications and the Agentic Security Initiative threat catalogue.
RAND Corporation	US	The model-weight security-level framework (SL1–SL5) and threat modeling.
Korea AI Safety Institute (K-AISI)	Korea	National AI-safety evaluation and policy hub, established within ETRI (Nov 2024); member of the International Network of AI Safety Institutes.
ETRI	Korea	Korea's largest government ICT research institute and host of K-AISI; national R&D in AI, semiconductors, and telecom.
KETI (Korea Electronics Technology Institute)	Korea	Applied/industrial electronics research driving embedded and on-device AI, smart-manufacturing AI, IoT, and SoC/sensor technology — bridging AI research to industry.
KISA	Korea	National internet-and-security agency; stood up an AI-Security headquarters and issues AI security guidance and AI-enabled cyber-defense.
TTA	Korea	National ICT standards body leading trustworthy-AI, AI-chip/NPU, and manufacturing-AI standardization.
MSIT	Korea	Lead ministry for AI policy, the AI Basic Act, K-AISI, and the national sovereign-AI foundation-model program.

Organization	Region	Core Focus & Contributions
NAVER · LG AI Research · SK Telecom · Upstage · Kakao	Korea	Korean frontier-model developers (HyperCLOVA X, EXAONE, A.X, Solar, KANANA) advancing sovereign LLMs alongside responsible-AI and red-teaming practices.
이로운앤컴퍼니 (Iroun & Company)	Korea (SME)	"Security for AI" specialist: data-loss prevention for generative AI, jailbreak/prompt-injection defense, and LLM red-team diagnostics (SAIFE X product).
엔키화이트햇 (ENKI WhiteHat)	Korea (SME)	Offensive-security firm pairing elite white-hat hackers with AI-driven automation; continuous penetration testing (Offen PTaaS) and AI red-teaming (ENKI REDTEAM CTF).
에임인텔리전스 (Aim Intelligence)	Korea (SME)	AI-safety startup: automated AI red-teaming (Stinger) and real-time guardrails against jailbreaks and data exfiltration (Starfort) across text, vision, and voice.
셀렉트스타 · 크라우드웍스 (Selectstar · Crowdworks)	Korea (SME)	AI data platforms providing generative-AI safety evaluation and red-team challenges for enterprise LLM adoption.
티오리 (Theori) · S2W · 딥브레인AI (DeepBrain AI)	Korea (SME)	AI-security research and services: offensive AI-security research, LLM/agent threat intelligence, and deepfake detection.

KAIST · Seoul National University	Korea	Leading AI research universities and founding members of the K-AISI consortium, active in trustworthy-AI and ML-security research.

9. Documenting and Measuring AI Safety and Security: AI-BOM, Assurance Artifacts, and Benchmark Standards

Every technique in the preceding sections produces evidence — an interpretability audit, a red-team report, an evaluation score, an access-control policy. That evidence only becomes trustworthy, comparable, and portable when it is captured in standard documentation and expressed in agreed metrics. This section surveys the emerging assurance layer: the documentation stack that culminates in a machine-readable **AI Bill of Materials (AI-BOM)**, the metrics landscape and its gaps, and the standardized benchmark and evaluation guidance now being issued by NIST and others.

9.1 The documentation stack: from cards to the AI-BOM

AI documentation has evolved from prose disclaimers into a layered stack of structured artifacts, each describing a different part of the system (Figure 11).

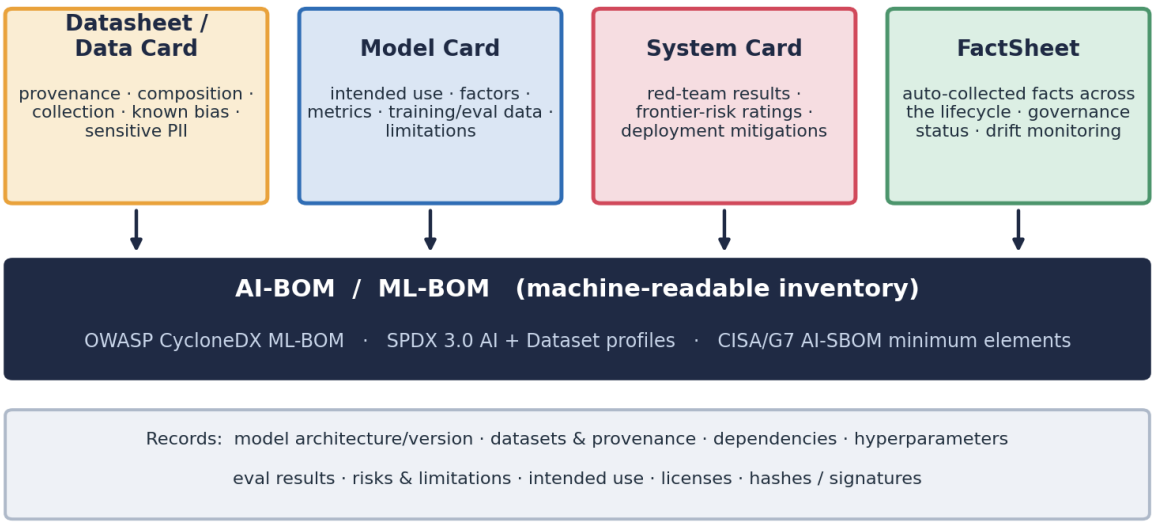

Figure 11. The AI documentation stack and the machine-readable AI-BOM. Human-authored cards describe the parts; the AI-BOM makes them machine-readable for supply-chain security.

At the human-authored layer, four artifacts dominate. **Model Cards** (Mitchell et al., 2019) are short reports whose standard sections are intended use (and out-of-scope use), factors (the demographic or environmental conditions that affect performance), metrics and decision thresholds, evaluation and training data, quantitative analyses reported **disaggregated by subgroup** rather than only in aggregate, and ethical considerations and caveats [44]. **Datasheets for Datasets** (Gebru et al., 2018/2021) document a dataset across its lifecycle through a questionnaire covering motivation, composition (including whether it contains sensitive or personal data), collection process (consent, timeframe, ethical review), preprocessing, uses, distribution (license, IP), and maintenance [45];

Google's **Data Cards** and Bender & Friedman's **Data Statements** for NLP serve the same transparency goal with schemas tuned to their domains. **System Cards** (OpenAI's GPT-4o/GPT-5 cards, Anthropic's model cards) document a deployed system rather than a bare model, adding what a model card omits: **external red-teaming results, frontier-capability risk ratings** (e.g., Low/Medium/High per Preparedness or ASL category), and **deployment-time mitigations** and their measured effectiveness [46]. **IBM AI FactSheets** differ by being **automatically and continuously assembled** by the MLOps platform across the model lifecycle — purpose, datasets, hyperparameters, quality and fairness metrics, drift monitoring, and governance status — and customized per consumer role [47].

The machine-readable layer is the **AI-BOM (or ML-BOM)** — the AI extension of the Software Bill of Materials (SBOM) that the software industry adopted after Log4Shell and SolarWinds. An AI-BOM records, in a structured schema, the model's architecture and version, its training and evaluation datasets with provenance and licenses, software dependencies, hyperparameters, evaluation results, known risks and limitations, intended use, and cryptographic hashes for integrity. Its value is supply-chain security: when a model or dataset is later found to be poisoned, back-doored, or license-contaminated, an AI-BOM lets an organization answer "am I affected?" — the same question SBOMs answered for vulnerable libraries. Three implementations matter. **OWASP CycloneDX** added first-class machine-learning support (from spec v1.5, now v1.6/1.7; also standardized as ECMA-424): a `machine-learning-model` component carries a `modelCard` object with three parts — `modelParameters` (approach, task, architecture family, dataset references), `quantitativeAnalysis` (performance metrics with value, data slice, and confidence interval), and `considerations` (intended users, technical limitations, **ethical considerations**, and **fairness assessments** naming the group at risk, harms, and mitigation) — deliberately mirroring the Mitchell model-card schema so existing cards map onto it [48]. **SPDX 3.0** (2024) adds an **AI Profile** and a **Dataset Profile** with the most explicit risk fields of any BOM standard: an `AIPackage` exposes `safetyRiskAssessment` (a serious/high/medium/low rating), `autonomyType`, `hyperparameter`, `modelExplainability`, `standardCompliance`, `energyConsumption`, `metricDecisionThreshold`, and a `useSensitivePersonalInformation` flag, while a `DatasetPackage` records `knownBias`, `anonymizationMethodUsed`, `confidentialityLevel`, `sensitivePersonalInformation`, and `datasetAvailability` [49]. Finally, **CISA and G7 partners** published the first government-level baseline — Minimum Elements for an AI SBOM — organizing required content into seven clusters: SBOM metadata, system-level properties, models, datasets across the lifecycle, infrastructure dependencies, **security properties**, and key performance indicators [50]. For the enterprise concerns of Section 7, the AI-BOM is the natural home for the supply-chain and model-provenance controls discussed there.

9.2 Metrics: what is measured, and what is missing

The industry measures AI **safety and quality** reasonably well and AI **security** poorly (Figure 12). On the mature side, a stable set of quantitative metrics is reported across evaluations, benchmarks, and system cards: **jailbreak / attack success rate (ASR)** against harmful-behavior suites (HarmBench, AdvBench, JailbreakBench, StrongREJECT); **refusal and over-refusal rate**, the latter measured with benign-but-suspicious prompts (XSTest, OR-Bench); **capability uplift** for CBRN and cyber; **hallucination / factuality** (TruthfulQA, HaluEval, SimpleQA, FActScore); **toxicity** (RealToxicityPrompts);

bias and fairness disparities (BBQ, StereoSet, CrowS-Pairs, Winogender); **calibration** (Expected Calibration Error, Brier score); **privacy leakage** via membership-inference and training-data extraction rates; and **watermark detectability** at a fixed false-positive rate. These flow through eval harnesses such as HELM, the LM Evaluation Harness, UK-AISI's Inspect, and MLCommons' AILuminate.

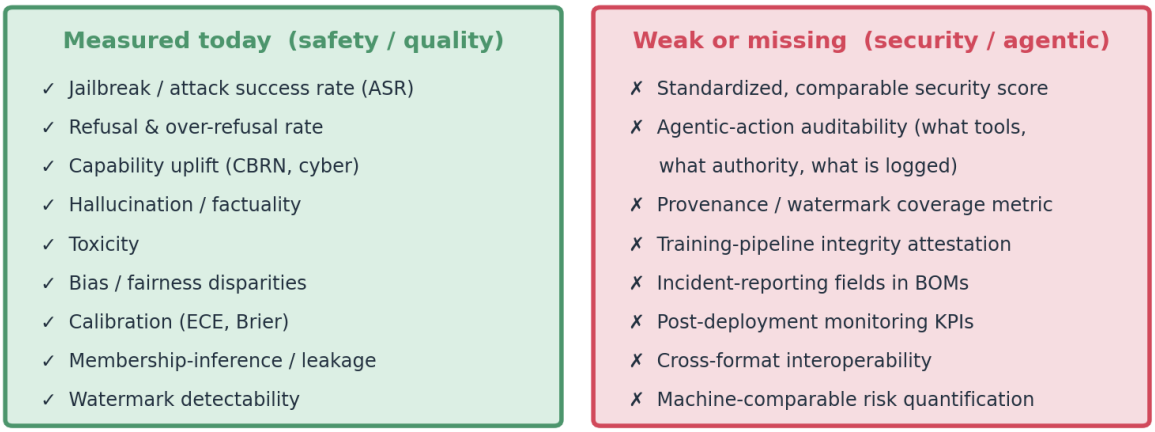

Figure 12. Safety and quality are measured well; security and agentic behavior are not. Where quantitative AI evaluation is mature today versus where standardized metrics are still missing.

The gaps are concentrated on the security and agentic side, and several are the subject of active proposals. First, there is **no standardized, comparable security score**: reported ASR figures are not comparable across papers because attack sets, threat models, and judge models differ, and NIST's adversarial-ML taxonomy catalogs attacks without prescribing metrics. Second, and most consequential for agentic systems, **an agent's delegated authority is undocumented** — existing BOMs list dependencies but not which tools an agent may call, with what permissions, what side effects, and what is logged; the 2026 **AgentRiskBOM** proposal names this "capability opacity" gap and adds layers for tool permissions, autonomy/authority (max tool tier, approval gates, emergency stop), inter-agent delegation, and an explicit **audit layer** [51]. Further gaps include the absence of a **provenance/watermark-coverage** metric, immature **training-pipeline integrity attestation** (SBOM signing via Sigstore/in-toto/SLSA is standard for software but rarely applied to model weights), missing **incident-reporting fields** in BOMs (unlinked to the AI Incident Database or MITRE ATLAS), under-specified **post-deployment monitoring KPIs** (most documentation is a release snapshot), and limited **interoperability** between CycloneDX, SPDX, and vendor card formats. A practical proposal that follows from this paper's framing is a **security-and-agentic profile** for AI-BOMs — standardizing a comparable security score, an agentic-authority disclosure, and a post-deployment monitoring field — so that the "contain" defenses of Section 6 become as auditable as today's safety metrics.

9.3 Standardized benchmark and evaluation guidance

Underneath the metrics sit a growing body of standards and government guidance that tell organizations what to evaluate and how.

NIST AI 600-1 — Generative AI Profile (final, July 2024) is the measurement backbone: a companion to the AI RMF that enumerates twelve GenAI risk categories and maps ~200 suggested actions to Govern/Map/Measure/Manage. Its **Measure** function is the benchmark core, calling for trustworthy-characteristic evaluation (confabulation, security probes, fairness), **disaggregated** evaluation across demographics and languages, red-teaming, and content-provenance tracking [36].

NIST AI 100-2e2025 — Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations (updated final, March 2025) is the shared vocabulary behind Figure 10. It separates predictive AI (attack classes: evasion, poisoning, privacy) from generative AI (which adds **abuse/misuse**, covering prompt injection and jailbreaks), and classifies every attack by the attacker's goals (availability, integrity, privacy, misuse), capabilities (control over training data, the model, or query access), and knowledge (white-, gray-, or black-box) [52].

NIST AI 800-1 — Managing Misuse Risk for Dual-Use Foundation Models (second public **draft**, January 2025; from NIST/CAISI) is voluntary guidance targeting deliberate misuse — CBRN uplift, offensive cyber, and harmful content such as CSAM. It is organized around **seven objectives**: anticipate misuse (maintain threat profiles and pre-development capability estimates); plan for managing misuse risk; **manage the risk of model theft** (prevent weight exfiltration); measure misuse risk (capability evaluations and red teams); ensure risk is managed **before deployment**; **collect and respond to misuse information after deployment** (monitoring and incident response); and provide transparency (publish reports, file incidents to AI incident databases) [53].

A note on "NIST AI 800-2." As of mid-2026 there is **no NIST document numbered AI 800-2** — the number is unassigned. The closest active NIST effort on securing AI systems is the **SP 800-53 Control Overlays for Securing AI Systems (COSAiS)** project, released as a **concept paper in August 2025**. Rather than a new numbered report, COSAiS will tailor the existing SP 800-53 security and privacy control catalog to AI through five use-case **overlays**: (1) generative-AI use, (2) predictive-AI use and fine-tuning, (3) single-agent systems, (4) multi-agent systems, and (5) controls for AI developers — protecting the confidentiality, integrity, and availability of training data, model weights, and configuration [54]. Readers looking for "AI 800-2" should track COSAiS and the two related final documents below.

NIST SP 800-218A — Secure Software Development Practices for Generative AI and Dual-Use Foundation Models (final, July 2024) is a community profile of the Secure Software Development Framework (SSDF, SP 800-218). It augments each SSDF practice group — Prepare the Organization, Protect the Software, Produce Well-Secured Software, Respond to Vulnerabilities — with AI-specific tasks: securing training-data provenance against poisoning, protecting model weights, documenting model lineage, and managing the AI/ML supply chain [55].

NIST ARIA — Assessing Risks and Impacts of AI (launched May 2024) operationalizes the RMF through **sociotechnical** testing, evaluating models not only in the lab but in realistic use. It runs three nested levels — automated **model testing**, adversarial **red**

teaming, and real-user **field testing** — and aggregates results into a **Contextual Robustness Index (CoRIx)** capturing an application's ability to maintain safe function across real-world contexts [56].

Beyond NIST, **MLCommons AILuminate** (v1.0, December 2024) is the leading independent safety **benchmark**: it tests general-purpose chat models against **twelve hazard categories** (violent and non-violent crimes, CBRNE, suicide/self-harm, hate, privacy, IP, defamation, sexual content, specialized medical/legal/financial advice, and more) using tens of thousands of **prompts kept private to prevent gaming**, judged by an evaluator-model ensemble, and reports a five-tier grade (Poor → Excellent) calibrated against reference models [57]. On the formal-standards side, **ISO/IEC TR 24029** provides methods for assessing neural-network robustness (Part 2 covering formal methods), and **ISO/IEC 42001** Clause 9 requires an organization to define, measure, and review AI-system performance as part of a certifiable management system — specifying that evaluation must happen and be governed, while documents like AILuminate and the NIST guidance supply the what and how [58].

10. Conclusion and Recommendations

The through-line of this survey is that **defense must migrate inward and downward** — inward from a model's outputs to its internal features, and downward from its words to its actions, permissions, and data flows — precisely because capability, autonomy, and privilege are all rising together. Behavioral testing catches yesterday's failure; interpretability, action-restriction, and access-control are what catch tomorrow's.

For teams deploying general-purpose AI, the priorities are robust evaluation that accounts for evaluation-awareness, adoption of a frontier-safety-style threshold policy that binds safeguards to measured capability, and secure-by-design agent architectures that survive a successful prompt injection rather than assuming one will never occur. For teams deploying industrial and enterprise AI, the priorities are identity-aware retrieval and need-to-know enforcement (so the assistant is confidential, not merely safe), a hard line between enterprise and consumer tooling to preserve trade-secret status, training-data governance to prevent the model from becoming a leak, and protection of the model itself against extraction and weight theft. Both sets converge on the same governance spine — the NIST AI RMF, ISO/IEC 42001, and, for their respective markets, the EU AI Act's GPAI obligations and Korea's AI Basic Act — and on the same assurance layer: AI-BOMs, model and system cards, and standardized benchmarks that turn each safeguard into auditable, portable evidence. A specific gap this paper flags for future work is that these artifacts today measure safety and quality far better than security and agentic authority; closing it — with a standardized security score and an agentic-authority disclosure in the AI-BOM — is among the most useful near-term contributions the community could make.

None of the individual defenses in this paper is sufficient alone. Interpretability without action-restriction leaves a compromised agent free to act; access-control without evaluation leaves unknown capabilities unmeasured; governance without technical enforcement is paper. Deployed together, in depth, they constitute a practical posture for the agentic era.

References

[1] Hubinger, E., et al. (2024). Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. Anthropic. arXiv:2401.05566. https://arxiv.org/abs/2401.05566

[2] Bricken, T., et al. (2023). Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. Anthropic / Transformer Circuits. https://transformer-circuits.pub/2023/monosemantic-features

[3] Templeton, A., et al. (2024). Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. Anthropic. https://transformer-circuits.pub/2024/scaling-monosemanticity/

[4] Lindsey, J., et al. (2025). On the Biology of a Large Language Model (Circuit Tracing / Attribution Graphs). Anthropic. https://transformer-circuits.pub/2025/attribution-graphs/biology.html

[5] Aim Security / Tenable (2025). CurXecute (CVE-2025-54135) and MCPoison (CVE-2025-54136): Remote Code Execution in the Cursor AI editor. https://www.tenable.com/blog/faq-cve-2025-54135-cve-2025-54136-vulnerabilities-in-cursor-curxecute-mcpoison

[6] Gong, Y., et al. (2023). FigStep: Jailbreaking Large Vision-Language Models via Typographic Visual Prompts. arXiv:2311.05608. https://arxiv.org/abs/2311.05608

[7] Russinovich, M., Salem, A., Eldan, R. (2024). Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. Microsoft. arXiv:2404.01833. https://arxiv.org/abs/2404.01833

[8] Anil, C., et al. (2024). Many-shot Jailbreaking. Anthropic. https://www.anthropic.com/research/many-shot-jailbreaking

[9] HiddenLayer (2025). Policy Puppetry: A Universal Prompt-Injection Technique. https://hiddenlayer.com/innovation-hub/novel-universal-bypass-for-all-major-llms/

[10] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J.Z., Fredrikson, M. (2023). Universal and Transferable Adversarial Attacks on Aligned Language Models (GCG). CMU / Center for AI Safety. arXiv:2307.15043. https://arxiv.org/abs/2307.15043

[11] Kinniment, M., et al. (2023). Evaluating Language-Model Agents on Realistic Autonomous Tasks. METR / ARC Evals. arXiv:2312.11671. https://arxiv.org/abs/2312.11671

[12] METR (2025–2026). Measuring AI Ability to Complete Long Tasks (arXiv:2503.14499) and Time Horizon 1.1 update. https://metr.org/blog/2026-1-29-time-horizon-1-1/

[13] Anthropic (2025). Activating AI Safety Level 3 (ASL-3) for Claude Opus 4 and Responsible Scaling Policy updates. https://www.anthropic.com/rsp-updates

[14] OpenAI (2025). Preparedness Framework (Version 2). https://openai.com/index/updating-our-preparedness-framework/

[15] Anthropic (2025). Claude Sonnet 4.5 System Card — evaluation awareness / situational awareness findings. https://www.anthropic.com/

[16] Google DeepMind (2025). Frontier Safety Framework, Version 3.0. https://deepmind.google/blog/our-frontier-safety-framework/

[17] Anthropic (2025). Disrupting the First Reported AI-Orchestrated Cyber-Espionage Campaign (GTG-1002). https://www.anthropic.com/news ; MITRE ATT&CK Campaign C0062.

[18] OWASP (2025). Top 10 for LLM Applications — LLM01:2025 Prompt Injection; and Agentic Security Initiative: Agentic AI Threats and Mitigations (T1–T15). https://genai.owasp.org/llmrisk/llm01-prompt-injection/

[19] Aim Labs (2025). EchoLeak (CVE-2025-32711): Zero-Click Data Exfiltration in Microsoft 365 Copilot. https://thehackernews.com/2025/06/zero-click-ai-vulnerability-exposes.html

[20] Wallace, E., et al. (2024). The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. OpenAI. arXiv:2404.13208. https://arxiv.org/abs/2404.13208

[21] Debenedetti, E., et al. / Google DeepMind (2025). Defeating Prompt Injections by Design (CaMeL). arXiv:2503.18813. https://arxiv.org/abs/2503.18813

[22] Beurer-Kellner, L., et al. (2025). Design Patterns for Securing LLM Agents against Prompt Injections. arXiv:2506.08837. https://arxiv.org/abs/2506.08837

[23] Microsoft (2025). Mitigate Oversharing to Govern Microsoft 365 Copilot and Agents. https://techcommunity.microsoft.com/blog/microsoft365copilotblog/mitigate-oversharing-to-govern-microsoft-365-copilot-and-agents/4448744

[24] Knostic (2025). Ending LLM Oversharing — Enforcing Need-to-Know at the Answer Layer. https://www.knostic.ai/blog/ending-llm-oversharing-we-raised-11-million-to-secure-enterprise-ai

[25] Microsoft Learn (2025). Document-Level Access Control (“Security Trimming”) in Azure AI Search. https://learn.microsoft.com/en-us/azure/search/search-document-level-access-overview

[26] Microsoft Learn (2025). Microsoft Purview Data Security and Compliance for Microsoft 365 Copilot (sensitivity labels, DLP, information barriers). https://learn.microsoft.com/en-us/purview/ai-m365-copilot

[27] Ray, S. (2023). Samsung Bans ChatGPT Among Employees After Sensitive Code Leak. Forbes. https://www.forbes.com/sites/siladityaray/2023/05/02/samsung-bans-chatgpt-and-other-chatbots-for-employees-after-sensitive-code-leak/

[28] AI Incident Database (2023). Incident 768: Samsung Semiconductor ChatGPT Data Leaks. https://incidentdatabase.ai/cite/768/

[29] The National Law Review (2025). AI Knows Too Much: When Employees Feed Trade Secrets to Generative AI Tools. https://natlawreview.com/article/ai-knows-too-much-when-employees-feed-trade-secrets-generative-ai-tools

[30] Carlini, N., et al. (2021). Extracting Training Data from Large Language Models. USENIX Security. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting ; and Nasr, M., Carlini, N., et al. (2023). Scalable Extraction of Training Data from (Production) Language Models. arXiv:2311.17035. https://arxiv.org/pdf/2311.17035

[31] Nevo, S., Lahav, D., Karpur, A., et al. (2024). Securing AI Model Weights: Preventing Theft and Misuse of Frontier Models (SL1–SL5). RAND RR-A2849-1. https://www.rand.org/pubs/research_reports/RRA2849-1.html

[32] NVIDIA (2024). Confidential Computing on H100 GPUs for Secure and Trustworthy AI. https://developer.nvidia.com/blog/confidential-computing-on-h100-gpus-for-secure-and-trustworthy-ai/

[33] Latham & Watkins (2025). EU AI Act: GPAI Model Obligations in Force and Final GPAI Code of Practice in Place. https://www.lw.com/en/insights/eu-ai-act-gpai-model-obligations-in-force-and-final-gpai-code-of-practice-in-place

[34] European Commission (2025). Guidelines for General-Purpose AI Providers and Regulatory Framework for AI. https://digital-strategy.ec.europa.eu/en/policies/guidelines-gpai-providers

[35] NIST (2025). Center for AI Standards and Innovation (CAISI). https://www.nist.gov/caisi

[36] NIST (2024). AI Risk Management Framework (AI 100-1) and Generative AI Profile (NIST AI 600-1). https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

[37] Bengio, Y., et al. (2025). International AI Safety Report. arXiv:2501.17805; First Key Update (Oct 2025) arXiv:2510.13653. https://internationalaisafetyreport.org/

[38] Korea, Ministry of Government Legislation / analyses by IAPP and Cooley (2024–2026). Framework Act on the Development of Artificial Intelligence and Establishment of Trust ("AI Basic Act"). Passed 26 Dec 2024; effective 22 Jan 2026. https://iapp.org/news/a/analyzing-south-korea-s-framework-act-on-the-development-of-ai

[39] Korea AI Safety Institute (2024). Establishment of the AI Safety Institute within ETRI (27 Nov 2024); International Network of AI Safety Institutes. https://www.aisi.re.kr/eng/contents/22

[40] CNN (2024). Arup confirms it was target of a US$25M deepfake video-conference scam (Hong Kong). https://www.cnn.com/2024/05/16/tech/arup-deepfake-scam-loss-hong-kong-intl-hnk

[41] SC World / ReversingLabs (2025). Amazon Q VS Code extension injected with a data-wiping prompt via a GitHub pull request. https://www.scworld.com/news/amazon-q-extension-for-vs-code-reportedly-injected-with-wiper-prompt

[42] ReversingLabs / Dark Reading (2025). "nullifAI": malicious ML models on Hugging Face using broken Pickle serialization to evade scanners. https://www.darkreading.com/threat-intelligence/sleepy-pickle-exploit-subtly-poisons-ml-models

[43] Wiz Research (2025). Exposed DeepSeek database leaks chat history, API keys, and backend secrets. https://www.wiz.io/blog/wiz-research-uncovers-exposed-deepseek-database-leak

[44] Mitchell, M., et al. (2019). Model Cards for Model Reporting. FAccT. arXiv:1810.03993. https://arxiv.org/abs/1810.03993

[45] Gebru, T., et al. (2021). Datasheets for Datasets. Communications of the ACM. arXiv:1803.09010. https://arxiv.org/abs/1803.09010 — see also Google Data Cards (Pushkarna et al., 2022) and Bender & Friedman, Data Statements for NLP (TACL 2018).

[46] OpenAI (2024). GPT-4o System Card (representative system card documenting external red-teaming, Preparedness risk ratings, and deployment mitigations). https://cdn.openai.com/gpt-4o-system-card.pdf

[47] IBM Research. AI FactSheets 360 / Model Inventory FactSheets. https://dataplatform.cloud.ibm.c

data/factsheets-model-inventory.html

[48] OWASP CycloneDX (2023–). Machine-Learning Bill of Materials (ML-BOM) and Model Card support (spec v1.5–1.7; ECMA-424). https://cyclonedx.org/capabilities/mlbom/

[49] SPDX 3.0 (2024). AI Profile (AIPackage) and Dataset Profile (DatasetPackage). https://spdx.dev/learn/areas-of-interest/ai/ ; https://spdx.github.io/spdx-spec/v3.0.1/model/AI/Classes/AIPackage/

[50] CISA and G7 partners (2026). Software Bill of Materials for AI — Minimum Elements. https://www.cisa.gov/resources-tools/resources/2025-minimum-elements-software-bill-materials-sbom

[51] Dutta, S. & Moharir, A. (2026). AgentRiskBOM: A Risk-Scoping Security Bill of Materials for Agentic AI Systems. arXiv:2606.21877. https://arxiv.org/html/2606.21877

[52] NIST (2025). Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations (NIST AI 100-2e2025). https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2025.pdf

[53] NIST / CAISI (2025). Managing Misuse Risk for Dual-Use Foundation Models (NIST AI 800-1, 2nd public draft). https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.800-1.ipd.pdf

[54] NIST (2025). Control Overlays for Securing AI Systems (COSAiS) — Concept Paper (SP 800-53-based). https://csrc.nist.gov/projects/cosais

[55] NIST (2024). Secure Software Development Practices for Generative AI and Dual-Use Foundation Models: An SSDF Community Profile (SP 800-218A). https://csrc.nist.gov/pubs/sp/800/218/

[56] NIST (2024). ARIA — Assessing Risks and Impacts of AI. https://www.nist.gov/programs-projects/assessing-risks-and-impacts-ai-aria

[57] MLCommons (2025). AILuminate: Introducing v1.0 of the AI Risk and Reliability Benchmark from MLCommons (v1.0 announced December 2024). arXiv:2503.05731. https://mlcommons.org/benchmarks/ailuminate/

[58] ISO/IEC (2021–2023). ISO/IEC TR 24029 (Assessment of the robustness of neural networks) and ISO/IEC 42001:2023 (AI management systems), Clause 9 — performance evaluation. https://www.iso.org/standard/42001

[59] Confidential Computing Consortium (Linux Foundation). Confidential Computing: Hardware-Based, Attested Protection of Data in Use. https://confidentialcomputing.io/about/ ; and NVIDIA (2024–2025). Confidential Computing on Hopper and Blackwell GPUs (Secure AI) — Technical Whitepaper. https://docs.nvidia.com/nvidia-secure-ai-with-blackwell-and-hopper-gpus-whitepaper.pdf

[60] Intel. Enclave Page Cache (SGX EPC) sizing — https://www.intel.com/content/www/us/en/suppor ; and El-Hindi, M., et al. (2022). Benchmarking the Second Generation of Intel SGX Hardware. ACM DAMON. https://dl.acm.org/doi/fullHtml/10.1145/3533737.3535098

[61] Misono, M., et al. (2025). Confidential VMs Explained: An Empirical Analysis of AMD SEV-SNP and Intel TDX. ACM SIGMETRICS. https://dse.in.tum.de/wp-content/uploads/2024/11/sigmetrics25summer-CVM-Explained.pdf

[62] Zhu, J., et al. (2024). Confidential Computing on NVIDIA H100 GPU: A Performance Benchmark Study. arXiv:2409.03992. https://arxiv.org/abs/2409.03992

[63] Tramèr, F. & Boneh, D. (2019). Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware. ICLR. arXiv:1806.03287. https://arxiv.org/abs/1806.03287

[64] Zhang, Z., et al. (2024). No Privacy Left Outside: On the (In-)Security of TEE-Shielded DNN Partition for On-Device ML (and the TEESlice mitigation). IEEE S&P. arXiv:2310.07152. https://arxiv.org/abs/2310.07152

[65] Agentic capability benchmarks — Jimenez et al. (2023), SWE-bench, arXiv:2310.06770, and OpenAI (2024), SWE-bench Verified; Zhang, A., et al. (2024), Cybench, arXiv:2408.08926; METR (2024), RE-Bench, arXiv:2411.15114; OpenAI (2024), MLE-bench, arXiv:2410.07095; Mialon et al. (2023), GAIA: A Benchmark for General AI Assistants, arXiv:2311.12983.

[66] Agent safety / misuse benchmarks — UK AI Safety Institute & Gray Swan AI (2024), AgentHarm, arXiv:2410.09024; Meta (2024), CyberSecEval 3, arXiv:2408.01605.

[67] NIST (2020, 2023). Zero Trust Architecture (SP 800-207) and A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments (SP 800-207A). https://csrc.nist.gov/pubs/sp/800/207/final

[68] SPIFFE / SPIRE (Cloud Native Computing Foundation, graduated 2022). Secure Production Identity Framework For Everyone. https://spiffe.io/

[69] OpenID Foundation (2025). Shared Signals Framework 1.0 and Continuous Access Evaluation Profile (CAEP) 1.0 (Final Specifications, September 2025). https://openid.net/wg/sharedsig

[70] Model Context Protocol (2025). Authorization Specification (MCP server as OAuth 2.1 resource server). https://modelcontextprotocol.io/specification/2025-11-25/basic/authorization

[71] ITU-T (2024). Recommendation X.1280: Framework for Out-of-Band Server Authentication Using Mobile Devices. International Telecommunication Union. https://www.itu.int/rec/T-REC-X.1280-202403-I