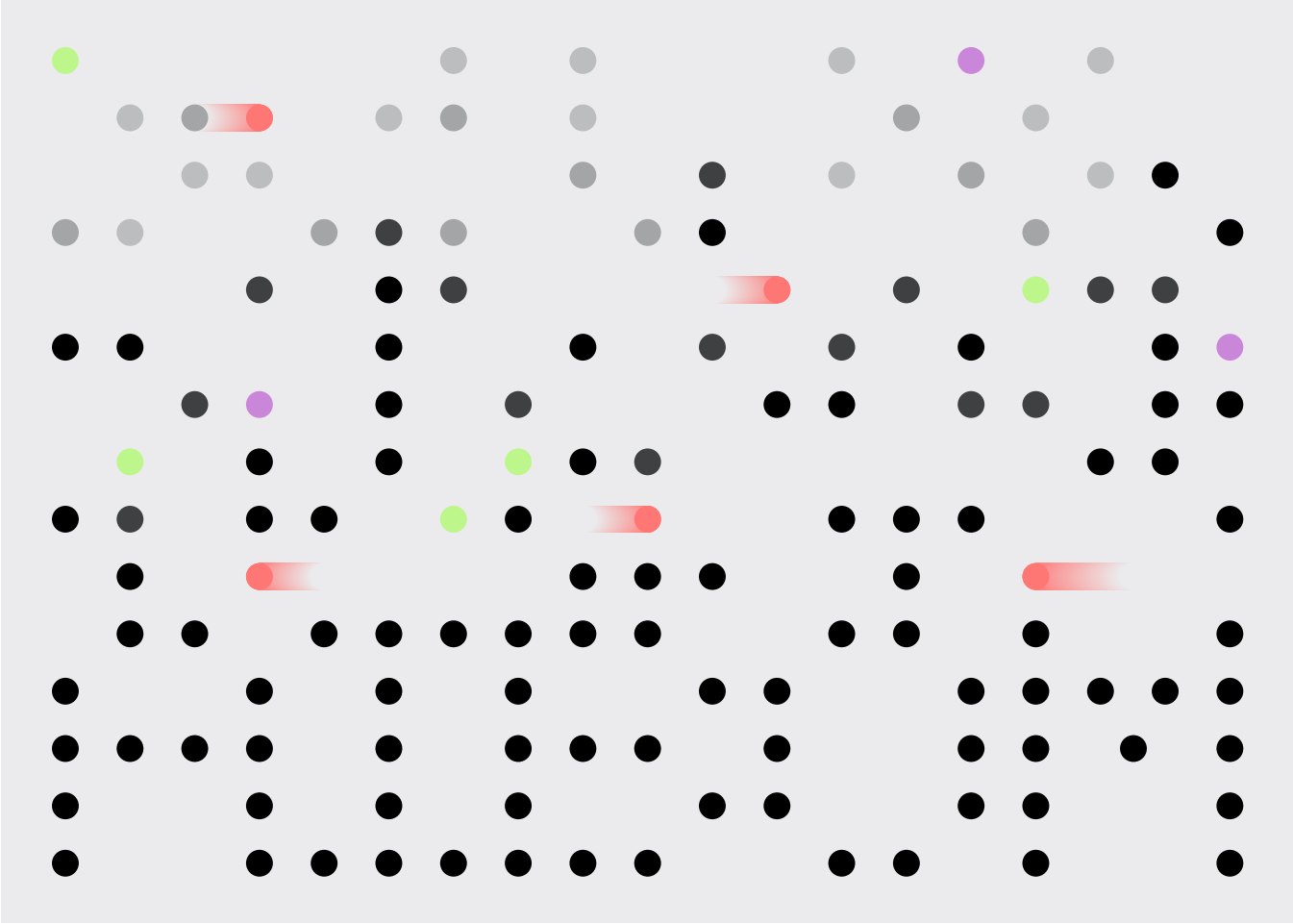


sysdig

AIBOM: The infrastructure, risks, and how to secure AI models

HUMAN-POWERED CONTENT
Crystal Morin, Senior Cybersecurity Strategist

AI brings new artifacts to the table, such as model weights and training pipelines. It also introduces new threats such as data poisoning and model drift. However, securing AI isn't a mystery: it's built on the same cloud-native, containerized infrastructure and security practices we already know.

Contents

Demystifying AI environments

Artificial intelligence (AI) doesn't live in a black box and it isn't sorcery. While the industry and media make it sound like AI models have incredibly complex components, the reality is that AI tools or applications are not that dissimilar from other applications.

As security professionals, you're better positioned than you might think to secure AI infrastructure and manage the risks of using it. You already know and have implemented many of the controls necessary to secure AI elsewhere. What's missing is a deep understanding and better visibility. That's where an AI bill of materials (AIBOM) comes in: documenting the AI infrastructure stack eliminates the mystery that shrouds AI security.

To properly secure and mitigate AI risks, cut through the hype. AI models are built on the same cloud-native and containerized infrastructure we've been working with for years, sharing the same, familiar risks. In fact, many machine learning (ML) algorithms and neural networks have roots that predate today's AI models and tools, like TensorFlow, which is used to standardize and accelerate how they are built, trained, and deployed.

We are rapidly adopting large language models (LLMs) and agentic AI tools to drive innovation. And while some organizations are building and hosting their own AI models, most are using third-party models through APIs and managed platforms from OpenAI or Anthropic. A security team's visibility and the attack surface depend on where the model lives. For third-party models, the greatest risks are concentrated around trust boundaries and nonphysical layers such as fine-tuning, prompt input, data governance, and API connections.

This paper demystifies AI. By breaking down the layers of its infrastructure and exposing the risks, you'll see that familiar security practices can protect your innovation. No guesswork; just knowing what's new and what's already understood.

The case for AIBOMs

While a traditional software bill of materials (SBOM) focuses primarily on software libraries, AI brings new artifacts to the table, such as model weights and corresponding neural networks, tokenization of inputs, custom datasets, and training pipelines. AI also introduces new threats such as poisoned datasets, sensitive data leaks, and model drift. These gaps can have an impact or surface at different stages while using and maintaining an AI model, such as during model building or serving.

By extending SBOM practices to AI models, you can document and maintain visibility for model lineage, data provenance, and dependency mapping across containers, frameworks, and the model's other components. It's a realistic, end-to-end picture that helps surface risk, find traceable issues, support faster remediation, and hold the right personnel accountable for insecure or misconfigured components.

Security practices for AI aren't a "nice-to-have"; they're rapidly becoming a compliance requirement. The **EU AI Act** entered into force in August 2024 and will be in full effect by August 2027. The world's first comprehensive AI legal framework established by a major regulatory body, the EU AI Act calls for data transparency and quality, accountability, and ongoing monitoring. Similarly, the **National Institute for Standards and Technology AI Risk Management Framework** (NIST AI RMF) emphasizes trustworthiness and the responsible design, development, deployment, and use of AI systems.

The approach to AI governance and guidance across the globe is evolving and is by no means uniform. For example, China is taking a very comprehensive and centralized approach to AI use and governance, while the U.S. is taking a decentralized framework-based approach, with individual states now coming online with their own nuances. While the AI legislative landscape is not settled and probably won't be for some time, AI security practices can't wait. Customers or end users will demand transparency with respect to how data is being used, secured and protected.

It is just good business to secure AI workloads, whether or not a business is being legislated or mandated to do so. And ultimately, a hack is still a hack, whether or not an AI model is a part of the equation. There are plenty of non-AI laws that are still applicable — the General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), Securities and Exchange Commission (SEC) disclosure requirements for material breaches — the list goes on.

For security leaders, compliance officers, and engineers, regulations make AI workload visibility foundational, not optional. An AIBOM is your trusted asset for AI governance and supply chain security.

The AI stack

A deep dive into the physical infrastructure and nonphysical layers

An AI model is just software running on specialized hardware, which puts it squarely in a security team's wheelhouse of understanding. You'll find that many of the components listed below — and their security requirements, listed in the next section — are familiar. This section first provides a detailed list of the physical layers of the AI infrastructure which can be deployed, inventoried, and monitored. Building on this foundation is a detailed explanation of nonphysical layers, which we consider the components of an AI model that are infrastructure-adjacent but not deployable.

For organizations leveraging third-party models and services via API integrations, much of the physical infrastructure security falls on the provider under a shared responsibility model, similar to "as-a-service" offerings and cloud service providers. The key components in these cases are fine-tuning and model customization, input handling, APIs and integrations, and data governance. Runtime and container-level visibility are typically unavailable; therefore, implementing proactive security practices and governance controls is imperative.

Physical infrastructure

Hardware

Graphics processing units (GPUs), tensor processing units (TPUs), central processing units (CPUs), memory, accelerators, servers, network, and storage devices.

Cloud and virtualization

Cloud service providers, virtual machines (VMs), and orchestration platforms.

Kubernetes and orchestration

Container orchestration and the scheduling of model training and inference workloads, along with continuous integration/continuous delivery (CI/CD) pipeline operations.

Containerized components

Base images and containerized libraries (for example, PyTorch, TensorFlow, Hugging Face stacks).

AI runtime and frameworks

Training pipelines, inference servers (Triton, Ray, Kubeflow), libraries (PyTorch, Transformers, TensorFlow, LangChain), and proprietary code.

Model artifacts

Pretrained weights, fine-tuned models, embeddings, model type, algorithms, parameters, and version.

Model Context Protocol (MCP) servers are orchestrators that integrate AI with third-party applications via agents and/or APIs. Like anything that integrates applications, services, and other components, MCP servers have the same inherent risks as AI models and require validation and mitigation.

Nonphysical layers

These are the supporting elements that shape how to access, secure, govern and invoke AI infrastructure. In and around an AI model, vulnerabilities and risks can manifest in the operation of the system. It's important to understand and document these layers in an AIBOM to move toward the AI transparency and accountability explicitly required in some regulations. Listing only GPUs and container images isn't good enough.

Data management

Dataset details such as model name, format, classification, and version, where applicable, as well as data provenance. Knowing your model's data lineage is half the battle.

Security and privacy

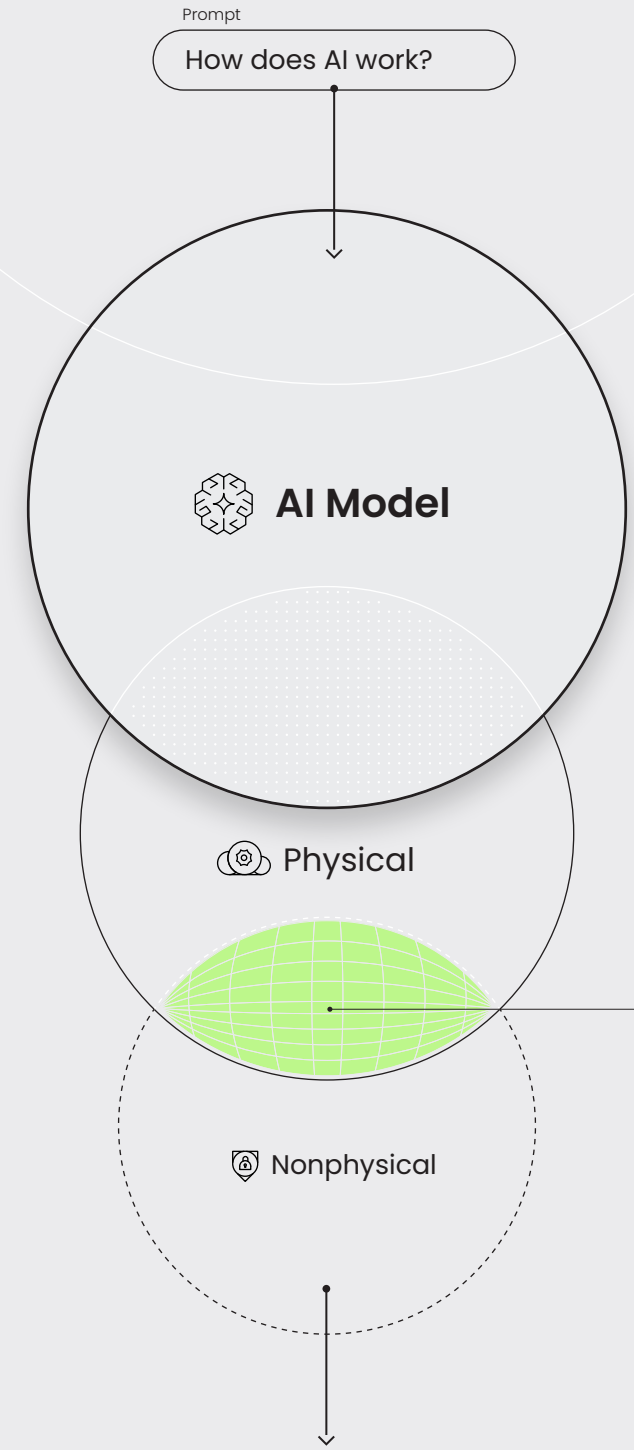
Encryption methods for data in transit and at rest, access controls including authentication processes and role-based permissions (including any entitlements or authorizations provided to agents or MCP servers accessing the model and integrated systems), and session and resource management of tools and protocols.

Governance and compliance

Model developer and owner information for traceability and responsibilities, along with governance practices for data privacy, additional compliance requirements, life-cycle policies, and audits to maintain full accountability.

Interfaces and protocols

APIs used or connected to the model, the model's user interface via browser or browser extensions, text, voice, and other multimodal input handling, tokenization processes to include model invocation and routing tools, and the protocol components that invoke underlying systems.



Physical layers
Hardware
CSPs VMs, MCPs
Kubernetes
Containerized components
Runtime frameworks
Model artifacts

Data processing and learning

Nonphysical layers
Data management
Security and privacy
Governance and compliance
Interfaces and protocols

AI security: Like regular security with a twist

Yes, the mathematical computations and some infrastructure components are different, but from a security perspective, AI tools are applications that inherit the same risks as those seen in traditional application security (AppSec). On top of that, AI models face a few variations of threats seen with ML, and a handful of entirely new risks that are unique to AI models.

In 2024, the Cloud Native Computing Foundation (CNCF) reported that 54% of AI and ML workloads were being built on Kubernetes, the most well-known container orchestration platform. Therefore, many of the risks posed by the use of AI in an enterprise environment are already familiar to security professionals. In fact, **industry** and **academic** research confirms this: most current AI threats align with known patterns of traditional ML threats such as data poisoning and prompt injection.

What's different

What's left are the quickly evolving AI-specific risks. These are emerging from how a model interacts with users and data, leading to novel forms of runtime attacks, pipeline and infrastructure compromise, persistent attacker influence, insecure integrations, unauthorized agent actions, and data leaks. Weaknesses in data access controls, data segmentation, and governance will obviously increase risk exposure to any of these attacks.

Adversarial influence

Attackers can carefully craft inputs that cause a model to make consistent mistakes in its output. Similar to Structured Query Language (SQL) injection to exfiltrate or manipulate databases, these prompt injection attacks are intended to leak secrets and change model behaviors. Attackers can also manipulate or influence training or fine-tuning data to corrupt a model over time.

Trust boundaries

A trust boundary is the imaginary line that data and permissions blur when in use. Trust or trust boundaries are a significant risk consideration for AI models and exist between nearly every layer of AI model infrastructure, whether self-hosted or integrated. Compromised trust can lead to data poisoning, the introduction of vulnerabilities, elevated privileges, and more. AI security requires understanding the who and what behind these trust boundaries and enforcing persistent validation, segmentation, and least privilege.

Security teams should complement their AIBOM with documentation of the data flows and identified trust boundaries within their organization's AI tools and applications. At minimum, practitioners should understand the components and data flows from the initial prompt (typically via text or file upload to a browser, but this could also be with voice prompts and APIs) through the model's components, and back to the response delivered to the end user or agent. Having visibility into these data flows and the applications, processes, protocols, and tools used will demystify where security and appropriate governance are required. Like other applications, knowing which data should be used for the model is foundational.

Risks and security by layer

Make your AIBOM actionable. This isn't just about a deployable infrastructure inventory. Including the nonphysical aspects of an AI model allows security teams to connect the **what** (infrastructure components), **how** (processes and protocols), and **who** (ownership, access, and governance).

Understanding how threats to AI models span these three connections enables security teams to set controls where attackers actually move and operate. Each of the following are the known and expected risks for AI infrastructure and suggested mitigation techniques to reduce the risk of compromise:

	RISKS	MITIGATIONS
Hardware	Firmware vulnerabilities with known exploits in GPUs, TPUs, and other hardware can be an initial access vector. Compromised firmware, hardware implants, and malicious drivers can lead to a supply chain attack and introduce threats before workloads reach production.	Runtime monitoring will detect anomalies in hardware usage during the training pipeline, such as irregular workload patterns and resource consumption possibly caused by cryptomining. Memory-bound anomalies typically occur during inference and can be identified by baselining normal usage and alerting on deviations
Cloud and virtualization	An attacker can bypass a hypervisor (virtual machines and isolated containers) to gain access to host systems. Side-channel attacks exploit leaked information about a device or hardware such as consumption, memory sharing, and timing variations to obtain secrets. An attacker may deliberately consume cloud storage capacity, memory, CPUs, or GPUs, causing either denial of service if limits are hit or an unplanned increase in consumption costs.	Harden configurations across cloud-hosted workloads and use continuous posture management to mitigate new risks quickly. Implement resource quotas and limits and isolate workloads to reduce the risk of abuse.

	RISKS	MITIGATIONS
Kubernetes and orchestration	Misconfigurations provide low-effort access to attackers. Overly permissive user, service, or agent accounts can be used to escalate privileges. Compromised pods allow attackers to move across the cluster. Exposed APIs can be directly abused by attackers.	Implement basic Kubernetes security practices, avoid using default access policies, and remediate exposed dashboards. Runtime detection will alert on privilege escalation and lateral movement.
Containerized components	**Image bloat** is when images are oversized with unused or unrequired packages, which unnecessarily expands the attack surface. Attackers search container images for embedded secrets like hardcoded API keys and credentials. Unpatched libraries leave vulnerable dependencies within PyTorch, TensorFlow, and Hugging Face containers as entry points for attackers.	Continuously scan images and conduct software composition analysis (SCA) to identify container threats. Use runtime-based vulnerability prioritization to mitigate the most critical risks first.

	RISKS	MITIGATIONS
AI runtime and frameworks	Attackers will inject malicious data into the training pipeline or during fine-tuning, either poisoning or exposing sensitive data. Attackers will craft queries and inputs that trick a model into compliance bypass or wrong, malicious, or unethical outputs. Attackers will prompt a model to provide build details or proprietary information that they can further use or sell. Attackers may abuse the model's compute resource via **LLMjacking** attacks, which could hit rate limits and increase your organization's utilization costs. Accuracy and reliability of outputs will degrade over time as real-world data drifts from the training data. Since AI models are probabilistic, the rate and severity of degradation depends on the content. The overwhelming volume of logs generated by AI will result in telemetry overload, making it difficult for security teams to identify or correlate threats in near real time. Model-weight integrity can be impacted by algorithmic bias, sensitive data memorization, and inaccurate data used during training. Reliability, ethics, and compliance will be negatively affected across model deployments.	Use runtime monitoring and detection to identify unusual query behavior and irregular data flows. Ensure that there is complete workload visibility and establish behavioral baselines to detect drift or poisoning attempts

	RISKS	MITIGATIONS
Model artifacts	Attackers may alter model weights or configuration files. Pretrained models or fine-tuned checkpoints downloaded from public repositories could contain vulnerabilities or malware in their library or accompanying scripts. Provenance gaps allow attackers to operate undetected given a lack of visibility for model or dataset training, changes made, and origin.	Keeping an updated AIBOM, hashing inputs and outputs, and signing off on software used will help prevent visibility gaps for models as dataset lineage and model versions change and improve integrity and provenance. Continuous monitoring and runtime detection will identify the deployment of untrusted artifacts and identify workload deviations. A file format and library like Safetensors can also provide a safe way to store weights and inherently reduces the risk of loading an untrusted model.
Application and agent	Attackers will alter and refine their input prompts to force the model to bypass its intended, safe behavior. AI models given broad or excessive access across internal systems or APIs increase the attack surface by leaving doors open for threat actors. Shadow AI, or the use of unvetted models, tools, or datasets by teams outside of security oversight, increases the attack surface.	Open communication and collaboration across the business will keep the security team better informed of the tools in use across the organization. Comprehensive workload visibility and runtime detection will shine a light on attack surface risk and detect prompt injection.

	RISKS	MITIGATIONS
For hosted or API-based models, the provider owns most of the physical infrastructure security. End user organizations should focus on the following risks and mitigations that are drawn from above	Prompt injection and data leaks caused by input manipulation or overriding controls. Poor sanitization or manipulation of training data during fine-tuning that could result in data leaks. Poor API security such as weak or missing authentication processes, no rate limits, or insecure API gateways may permit model abuse and data exfiltration.	Use strong data classification and sanitization processes before prompt submission. Enforce API authentication, use output filtering and data loss prevention (DLP) tools, and monitor fine-tuning workflows for unauthorized data or access.

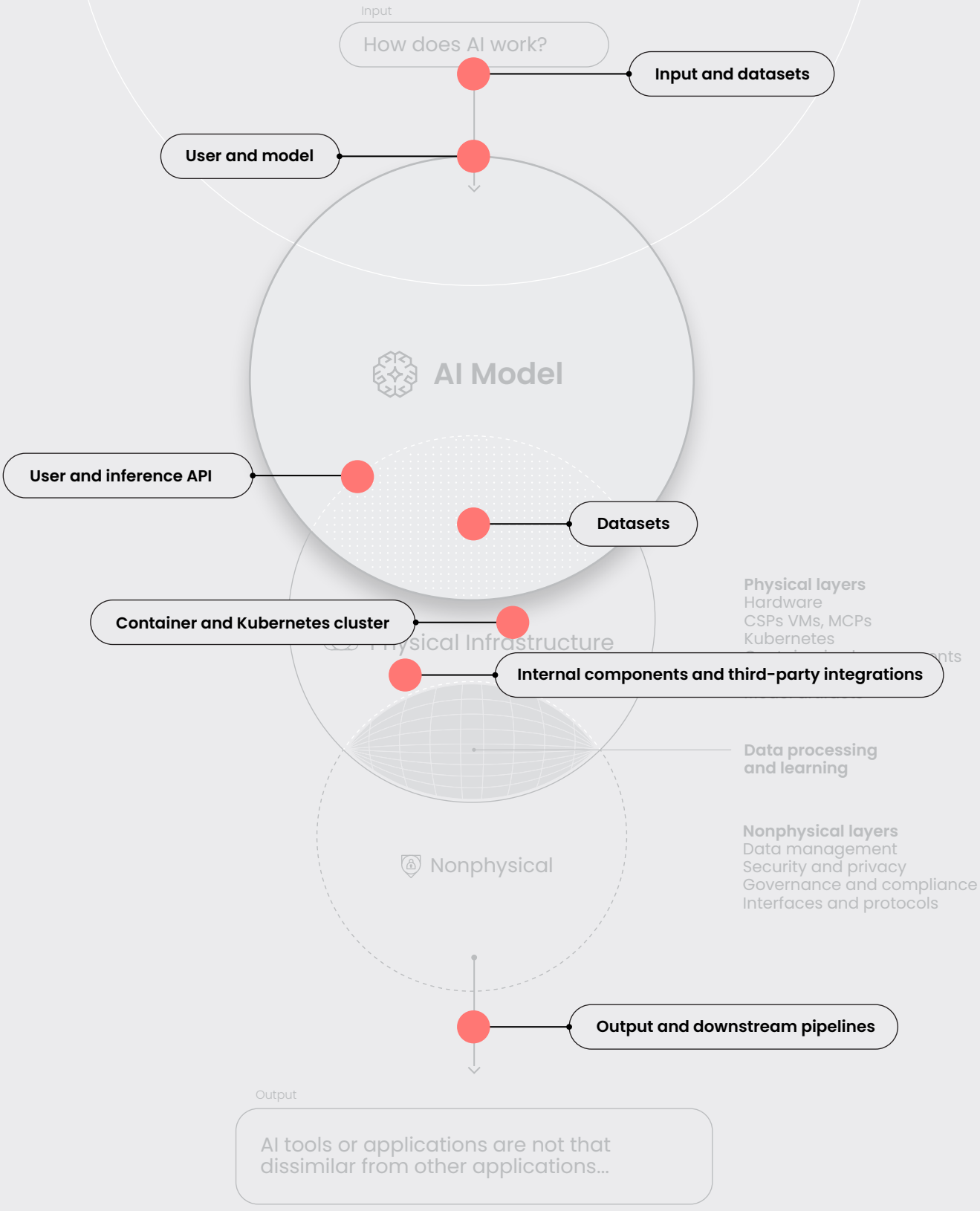
Risks at the trust boundaries

Many of the layers in an AI model introduce assumptions that turn into concentrated risk. Trust boundaries primarily fall between nonphysical layers and can also introduce risk if strong security practices are not in place. Below are a few examples of trust boundary risks:

	RISKS	MITIGATIONS
Between raw data sources or input and the training dataset	An attacker can poison (modify) the training data or inject malicious data samples.	Use strong access control, model behavior monitoring, and data validation, or otherwise risk bias output or even implanted backdoors.
Between end users and the inference API	Attackers can inject malicious inputs and payloads that will manipulate outputs or exfiltrate sensitive information.	Monitoring for anomalous queries, input and output filtering, and request rate limits will alert on abnormalities and minimize the risk of exfiltration.
Between an AI model's container or pod and Kubernetes cluster infrastructure	An attacker can elevate privileges to move from a container to workloads in the orchestration environment.	Strong Kubernetes role-based access controls (RBACs), network and pod policies, runtime isolation, and account monitoring will minimize the risk of lateral movement or trigger an alert.
Between datasets	A lack of segmentation, overly permissive accounts, and poor access control policies can lead to data leaks and misuse, such as a fine-tuned model inadvertently exposing sensitive information.	Enforce least privilege, segment sensitive and nonsensitive data, use DLP tools, and regularly review datasets to reduce confidential information memorization.

	RISKS	MITIGATIONS
Between trusted internal components and third-party integrations, pretrained models, software-as-a-service (SaaS) connections, and externally developed libraries	Gaps here can introduce vulnerabilities and compromised dependencies that could impact the entire AI pipeline, such as backdoors that activate upon inference.	Runtime detection, dependency scanning, using and reviewing SBOMs, and conducting provenance checks on models and libraries will highlight vulnerabilities.
Between initial output and downstream pipelines	An attacker's unvalidated or untrusted output can cause broader system errors.	Validating output, placing guardrails on automated processes, and keeping a human in the loop, especially for critical processes, will minimize the risk of impact.
Between user accounts and models	Herein lies a wide range of access to models and datasets. Insider threats, misconfigurations, and compromised accounts can lead to exposure, which provides initial access to attackers.	Account monitoring will alert on anomalous behavior; role separation, just-in-time access, and RBACs will reduce risk. Using strict policies within access approval workflows can also help limit access to training and inference pipelines.

Trust Boundaries



Telemetry and alerts at machine speed

AI tools are useful for sifting through massive amounts of data, but AI models themselves generate an equally overwhelming volume of logs, telemetry, and anomalies. The AI attack surface requires detection and response that can keep pace with machine-speed operations. The risks associated with AI infrastructure are not entirely new, and neither are the requirements for fast, effective security practices. By securing AI, organizations may force a more rigorous implementation of today's security best practices elsewhere, resulting in more tightly secured environments. In summary, to operationalize these security practices:

- **Use real-time runtime detection** to alert on anomalous behaviors inside the running containers that host models.
- **Use SCA and vulnerability management** to identify risk dependencies in ML and deep learning frameworks and container images, supporting timely remediation.
- **Set guardrails and strong defaults for Kubernetes** to prevent privilege escalation in AI workloads and lateral movement.
- **Scan and analyze images** to identify image bloat, and reduce the attack surface across AI infrastructure by minimizing bloat.
- **Implement compliance and posture policies** to ensure the consistent application and enforcement of security best practices, especially in highly regulated industries.

An effective defense must match the speed at which AI infrastructure and threat actors operate. Continuous monitoring, deep runtime visibility, and high-fidelity alerts allow security teams to act fast when facing a potential threat.

Take the mystery out of AI

AI is not a black box of complexity; it's a set of cloud-native components that we already know how to secure.

Applying what you already know and extending SBOM practices to AI workloads ensures that your organization has the right runtime visibility and proactive Kubernetes guardrails in place. In addition, **using threat models** and other forms of validation ensure that teams understand and address the AI risks that pertain to their organization.

Understanding AI infrastructure layers allows you to better secure workloads and provide full transparency to customers. You don't need to wait for the AI security market to mature before making effective changes and reducing risk. While there are challenges with securing the protocols used, understanding model weights, and a lack of transparency with how some models are built and trained, the cloud-native tools you already trust are ready to secure AI today. Furthermore, the same tried-and-true practices — microservice security, input sanitization, and data integrity, authentication, and infrastructure controls — continue to be effective when applied to the layers of AI infrastructure.

Make AIBOMs part of your risk reporting, as regulations and customers will soon demand it.

AI is redefining your attack surface, but you don't need to reinvent your security program to keep up.

See how Sysdig secures AI at runtime.

LEARN MORE →

Sysdig helps security and development teams prevent, detect, and respond to threats instantly. Founded by the creators of Falco and Wireshark, and built on agentic AI, Sysdig delivers real-time defense grounded in the uncompromising truth of runtime. No guesswork. No black boxes. **Just cloud security, the right way.**

ABOUT THE AUTHOR

Crystal Morin,
Senior Cybersecurity Strategist

Crystal served as a linguist and intelligence analyst in the U.S. Air Force, then joined Booz Allen Hamilton to continue countering terrorism and cyber threats, where she helped build the firm's cyber threat intelligence community and threat hunting capabilities. She became a Threat Research Engineer at Sysdig in 2022 before stepping into her current role where she helps organizations understand best practices and implement them to defend against modern threats.

sysdig

WHITE PAPER

WP-020 REV. A 11/25