



OWASP AIBOM

Grundlagenleitfaden

SEPTEMBER 2026

Inhaltsverzeichnis

MANAGEMENT-ZUSAMMENFASSUNG	6
1. AI Bill of Materials verstehen	7
AI Bill of Materials als Governance-Artefakte verstehen	8
2. Was ist ein AIBOM?	10
3. Warum sind AIBOMs wichtig?	12
Sicherheitstreiber	12
Regulatorische Treiber	13
Governance-Treiber	14
Geschäftskontinuität	14
Bedrohungen der Lieferkette	15
4. Kernkomponenten eines AIBOM	16
Der Header	16
Die Komponentenliste	17
Die Datenflüsse	18
Vertrauenszonen und Grenzen	20
Verhaltensweisen	20
Was ein AIBOM ermöglicht	20
5. AIBOM-Struktur	22
Was jedes AIBOM deklariert	22
Ausgearbeitetes Beispiel: ein gehostetes Inferenzsystem	23
Ausgearbeitetes Beispiel: eine Fine-Tuning-Pipeline	26
Was das AIBOM nicht aussagt	28
Wie ein AIBOM niedergeschrieben wird – für Leser, die tiefer einsteigen wollen	28
6. AIBOM für agentische KI-Systeme	30
Abhängigkeiten zwischen mehreren Agenten	30
Zugehörige Tools	31
Konnektoren von Drittanbietern	32
Nachverfolgung des MCP/A2A-Ökosystems	33
7. Roadmap zur Einführung im Unternehmen	35
Phase 1 — Bewusstsein und Ausgangsbasis	36
Phase 2 — Pilotprojekt	38
Phase 3 — Governance-Integration	39
Phase 4 — Kontinuierlicher Betrieb	40
DANKSAGUNG	42

Vom AIBOM-Projektleiter

Ich war eine Zeit lang Co-Vorsitzender der SBOM-Initiative der CISA im DHS, und die Lektion, die mir am längsten im Gedächtnis geblieben ist, betraf nicht Dateiformate. Sie betraf das, was passiert, nachdem das Format veröffentlicht ist. Wir haben die Softwarebranche dazu gebracht, SBOMs in großem Maßstab zu erzeugen, und dann zugesehen, wie die Einführung genau an diesem Punkt ins Stocken geriet, weil die Erzeugung nie der schwierige Teil war. Der schwierige Teil war der Aufbau der operativen Ebene darum herum: wer das Artefakt nutzt, in welche Entscheidung es einfließt, was passiert, wenn es veraltet, und wer verantwortlich ist, wenn es gar nicht existiert. AIBOM 101 wurde mit dieser Lektion im Hinterkopf geschrieben. Es ist keine Schemareferenz. Es ist die operative Ebene für KI-Systeme, die wir für Software nie vollständig aufgebaut haben.

Eine AI Bill of Materials verdient ihren Platz nur, wenn sie verändert, was ein Sicherheitsteam, ein Auditor oder ein Prüfer im Einkauf tatsächlich tun kann. Deshalb widmet sich dieses Dokument dem Geltungsbereich, den Vollständigkeitsaussagen und den Nachweisen ebenso ausführlich wie den Komponenten und Datenflüssen. Ein Datensatz, dem niemand vertrauen kann und den niemand nutzen muss, ist ein technisches Nebenprodukt mit Governance-Etikett, keine Infrastruktur. Unser Ziel mit diesem 101 ist es, Sicherheitsverantwortlichen in Unternehmen etwas an die Hand zu geben, das sie sofort einsetzen können: eine Möglichkeit zu fragen, welche KI sie haben, wovon sie abhängt und ob jemand für diese Abhängigkeiten bürgen kann – in einer Sprache, die im Vorstand ebenso Bestand hat wie in einer Build-Pipeline.

Wir haben dies geschrieben, damit es gelesen und nicht nur nachgeschlagen wird. Die enthaltenen ausgearbeiteten Beispiele – ein gehostetes Inferenzsystem und eine Fine-Tuning-Pipeline – gehen die Überlegungen durch, die ein Ersteller tatsächlich anstellen muss: wo die Systemgrenze liegt, was als Nachweis gilt und was eine Vollständigkeitsaussage implizieren darf. Die Roadmap zur Einführung im Unternehmen leistet dasselbe für die organisatorische Seite: von einem benannten Inventar der KI-Systeme über ein Pilotprojekt hin zur Governance-Integration und schließlich zum kontinuierlichen Betrieb, in dem AIBOMs bei definierten Auslösern automatisch neu erzeugt werden. Jede Phase hat bewusst ein überprüfbares Abschlusskriterium, denn eine Roadmap ohne die Möglichkeit festzustellen, ob eine Phase abgeschlossen ist, ist nur ein Wunsch.

Wir haben dies auch so aufgebaut, dass es sich weiterentwickeln kann. Agentische KI ist kein zukünftiger Randfall; sie verändert bereits, was „Abhängigkeit“ bedeutet, da die Tool-Aufrufe, Delegationen an Sub-Agenten und MCP-Verbindungen eines Systems heute zur Laufzeit entschieden werden können, statt zur Entwurfszeit festgelegt zu sein. Anstatt dies als Ausnahme zu behandeln, die später nachgerüstet wird, haben wir AIBOM von Anfang an als Graphen entworfen, der sich natürlich zu dem erweitern lässt, was wir AgBOM nennen: Agentenidentitäten, Tool-Berechtigungen und Interaktionsmuster zwischen mehreren Agenten, erfasst mit derselben Sorgfalt wie Modelle und Datensätze. Die Architektur soll sich mit dem Fachgebiet mitbewegen, ohne jedes Mal eine Neufassung zu erzwingen, wenn ein neues Muster entsteht.

Nichts davon geschieht isoliert. Dieses Dokument ist das Ergebnis der Arbeitsgruppen des OWASP-AIBOM-Projekts – Requirements, Best Practices, Integrity and Quality, Threat Intel, Policy, Content, Alliances, Engagement, Sponsorship und AgBOM –, die als ein

zusammenhängendes System und nicht als getrennte Vorhaben geführt werden. Jede Arbeitsgruppe hat uns dazu angehalten, bei der Frage ehrlich zu bleiben, was ein AIBOM behaupten kann und was nicht. Wir haben versucht, diese Grenze durchgehend klar zu benennen: Ein AIBOM dokumentiert, was ein System enthält, nicht, ob es sicher oder konform ist. Dieses Urteil bleibt beim Leser, der dafür sein eigenes Risikorahmenwerk heranzieht, und das wird auch so bleiben, während dieses Projekt reift. Das Projekt, seine Schemaarbeit und laufende Veröffentlichungen finden Sie unter owaspai bom.org, und wir freuen uns über Mitwirkende aus allen oben genannten Arbeitsgruppen.

Wenn SBOM der Softwarebranche etwas gelehrt hat, dann dies: Ein gutes Format ohne eine begleitende Betriebsdisziplin ist ein Artefakt für die Schublade. Unser Ziel für AIBOM ist, dass es nie dazu wird. Ich hoffe, dieses Dokument gibt Ihnen den Ausgangspunkt, um dies in Ihrer eigenen Organisation Wirklichkeit werden zu lassen.

Aruneesh Salhotra
Projektleiter, OWASP AIBOM
www.owaspai bom.org

Geltungsbereich und Zielgruppe

Dieses Dokument ist eine Einführung in die AI Bill of Materials (AIBOM): was sie ist, was sie enthalten muss, wie sie erstellt und validiert wird und wie eine Organisation sie einführt. Die vorgesehene Zielgruppe sind in erster Linie die Personen, die AIBOMs erstellen und nutzen – KI- und ML-Ingenieure, MLOps- und Plattformteams, Sicherheitsarchitekten sowie Fachleute für Produktsicherheit und Lieferkettenrisiken. Es ist so geschrieben, dass es auch von jemandem ohne oder mit wenig AIBOM-Erfahrung von Anfang bis Ende gelesen werden kann; Vertrautheit mit der SBOM-Praxis für Software ist hilfreich, wird aber nicht vorausgesetzt.

Wir möchten außerdem CISOs, Verantwortliche für KI-Governance und Führungskräfte informieren, die für Risiken in der KI-Lieferkette verantwortlich sind. Sie profitieren von der Darstellung, warum AIBOM wichtig ist, von den zugrunde liegenden Sicherheits-, Regulierungs- und Governance-Treibern sowie von der phasenweisen Einführungs-Roadmap und den Reifegradindikatoren. Darüber hinaus berührt dieses Dokument den breiteren regulatorischen und Governance-Kontext von KI-Systemen und kann für Compliance-, Rechts- sowie Einkaufs- oder Drittparteirisiko-Teams nützlich sein, die AIBOM-Offenlegungsanforderungen für Lieferanten festlegen.

Dieses Dokument beschreibt die AIBOM-Struktur konzeptionell und zeigt, wie sich diese Struktur auf ein etabliertes, maschinenlesbares BOM-Format abbilden lässt. Es ist keine Schemareferenz auf Feldebene; für Details der Kodierung bleibt die Formatspezifikation maßgeblich, und Leser sollten deren aktuelle Version heranziehen, statt die hier gezeigten Beispiele als Schema zu behandeln. Es bietet keine Bedrohungstaxonomie für KI-Systeme, keine Methodik für das KI-Risikomanagement und keine anbieter- oder werkzeugspezifischen Implementierungsanleitungen – diese werden in begleitenden OWASP-Ressourcen und in den referenzierten Standards gepflegt und durchgehend zitiert.

Urheberrecht und Lizenz

© OWASP Foundation. Dieses Dokument wird unter Creative Commons CC BY-SA 4.0 veröffentlicht. OWASP und das OWASP-Logo sind Marken der OWASP Foundation. Mitwirkende werden in der Danksagung genannt; eine Mitwirkung bedeutet keine Billigung durch den Arbeitgeber eines Mitwirkenden. Den vollständigen Lizenztext finden Sie unter: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>

Diese deutsche Fassung ist eine Übersetzung des englischen Originals „OWASP AIBOM Foundations Guide v1.0“ (September 2026) und steht ebenfalls unter CC BY-SA 4.0.

MANAGEMENT-ZUSAMMENFASSUNG

Die meisten Organisationen können nicht zuverlässig sagen, welches Modell in der Produktion läuft, die Daten und vorgelagerten Lieferanten dahinter nachverfolgen oder zeigen, ob eine neu offengelegte Schwachstelle ein System erreicht, von dem sie abhängen. Moderne KI-Systeme werden aus Modellen von Drittanbietern, lizenzierten und internen Daten, Open-Source-Frameworks, Retrieval-Pipelines, Agenten, Tools und anbieterverwalteten Diensten zusammengesetzt – von denen sich viele ohne jedes lokale Release ändern. Herkömmliche Softwareinventare wurden nicht dafür gebaut, diese Nicht-Code-Artefakte und externen Dienste zu erfassen, doch genau dort wird das Verhalten eines KI-Systems – und sein Risiko – weitgehend bestimmt.

Eine AI Bill of Materials (AIBOM) schließt diese Lücke: eine strukturierte, maschinenlesbare Darstellung dessen, was ein KI-System tatsächlich enthält und wie Informationen durch es fließen. Konzeptionell hält sie Fakten fest, statt eine Bewertung oder ein Compliance-Urteil abzugeben – genau das ermöglicht es ihr, als gemeinsame, überprüfbare Grundlage für Sicherheits-, Governance-, Einkaufs- und Auditentscheidungen gleichermaßen zu dienen.

Insbesondere für Sicherheitsverantwortliche bedeutet dies Hebelwirkung bei Entscheidungen, die Sie bereits verantworten. Ein AIBOM verwandelt KI-Systeme von undurchsichtigen Abhängigkeiten in nachvollziehbar verantwortete – es verkürzt die Reaktion auf Schwachstellen von Wochen auf Stunden, gibt Einkauf und Drittparteirisikomanagement einen konkreten Offenlegungsstandard, an dem Lieferanten gemessen werden können, und bietet eine einheitliche, belastbare Nachweisgrundlage für Governance-Prüfungen und Berichte an den Vorstand. Die wirksamste Einführung ist kein neues Parallelprogramm, sondern macht das AIBOM zu einem verpflichtenden Input für die Kontrollen, die Sie bereits betreiben: Release-Gates, Lieferantenbewertungen, Modellrisikoprüfungen und Incident Response – sodass ein einziger Datensatz Sicherheit, Compliance und Audit dient, ohne dreimal erstellt werden zu müssen.

Dieses Dokument ist ein praktischer Einstieg in die Erstellung eines AIBOM. Es erklärt, was ein AIBOM ist und warum es zu einer betrieblichen Anforderung statt einer optionalen Praxis wird, zeigt anhand ausgearbeiteter Beispiele, wie ein AIBOM aufgebaut und erstellt wird, erweitert das Modell um die zusätzlichen Anforderungen agentischer KI-Systeme und legt eine phasenweise Roadmap für die unternehmensweite Einführung von AIBOMs dar.

1. AI Bill of Materials verstehen

KI-Systeme hängen von mehr als nur Code ab. Ein Produktivsystem greift auf Modellgewichte, Trainings- und Fine-Tuning-Datensätze, Adapter und Checkpoints, Prompts und Vorlagen, Retrieval-Korpora, Guardrail- und Moderationsdienste, Orchestrierungs-Frameworks und anbieterverwaltete APIs zurück. Zusammen bilden sie die KI-Lieferkette, und diese unterscheidet sich in einem entscheidenden Punkt von einer Software-Lieferkette: Ihre folgenreichsten Abhängigkeiten sind häufig überhaupt keine Softwarepakete. Es sind Nicht-Code-Artefakte und extern betriebene Dienste, deren Verhalten sich ohne jedes lokale Release ändern kann.

Diese Abhängigkeiten konzentrieren sich nicht auf eine einzelne Phase. Sie sammeln sich über den gesamten Lebenszyklus an. Während des Entwurfs wählen Teams Modellanbieter und Datenquellen aus. Während der Entwicklung führen sie Fine-Tuning-Daten, Prompts und Evaluierungsressourcen ein. Während der Bereitstellung fügen sie Infrastruktur, Sicherheitskontrollen und Integrationen hinzu. Im Produktivbetrieb hängen Systeme von Retrieval-Korpora ab, die sich laufend ändern, von Moderationsdiensten, die die Ausgaben maßgeblich prägen, und von anbieterseitigen Updates, die nach dem Zeitplan eines anderen eintreffen. Eine kritische Abhängigkeit kann daher in jeder Phase hinzukommen, und vieles davon liegt außerhalb der direkten Sichtbarkeit der bereitstellenden Organisation.

Betrachten wir einen Unternehmensassistenten, der auf der API eines gehosteten Foundation-Modells aufbaut. Aus Sicht des Anwendungsteams hat sich nichts geändert: Die Prompts, die Retrieval-Konfiguration und die Infrastruktur sind alle wie zuvor. Doch Anbieter veröffentlichen API-Changelogs und pflegen Lebenszyklusrichtlinien, die zwischen aktiven, veralteten und eingestellten Modellen unterscheiden, und ein versionierter Alias kann auf einen neueren Snapshot umgeleitet werden. Das Modell, das evaluiert und freigegeben wurde, ist möglicherweise nicht mehr das Modell, das das Produktionsverhalten prägt. Wenn sich die Ausgabequalität verändert oder ein Vorfall eintritt, kann das Team die ersten drei Fragen, die ihm gestellt werden, nicht beantworten – was hat sich geändert, wann, und lag der Auslöser vorgelagert –, weil die zu ihrer Beantwortung nötigen Informationen nirgends aufgezeichnet wurden.

Dieselbe Lücke tritt auch ohne gehosteten Anbieter auf. Fine-Tuning-Pipelines ohne Lineage-Tracking können nicht nachweisen, welches Basismodell und welche Daten ein bereitgestelltes Artefakt hervorgebracht haben. Retrieval-Quellen ändern sich ohne Versionierung. Sicherheits- und Moderationsschichten werden als Implementierungsdetail behandelt, obwohl sie die Ausgaben beeinflussen. Das Muster ist durchgängig: Organisationen erben Fähigkeiten, ohne das Verständnis dafür zu erben. Sie wissen, welche Anwendung sie bereitgestellt haben; sie können nicht zeigen, welches vorgelagerte Modell, welcher Datensatz oder welche Kontrollschicht ihr Verhalten tatsächlich geprägt hat.

Hier stößt das herkömmliche Softwareinventar an seine Grenzen. Eine SBOM erfasst Softwarekomponenten und ihre deklarierten Versionen; das bleibt notwendig, wird aber nicht hinreichend. Um zu bestimmen, was ein KI-System tun wird, muss man die vortrainierten Modelle kennen, die Daten, mit denen sie trainiert und evaluiert wurden, die Evaluierungsergebnisse, die die Bereitstellung gerechtfertigt haben, sowie die anbieterverwalteten Dienste im Anfragepfad – nichts davon sind Softwarekomponenten im Sinne einer SBOM. Das Softwareinventar sagt Ihnen, welcher Code vorhanden ist. Die Sichtbarkeit der KI-Lieferkette muss Ihnen zusätzlich sagen, welche Nicht-Code-Artefakte und vorgelagerten Dienste das Verhalten maßgeblich geprägt haben – mit genügend Provenienzanangaben, um das System so zu rekonstruieren, wie es zu einem bestimmten Zeitpunkt bestand.

Agentische Systeme verstärken dies, statt es zu verändern. Wenn ein Agent zur Laufzeit Tools auswählt, externe Dienste aufruft oder an andere Agenten delegiert, ist die Menge der Abhängigkeiten nicht mehr zur Entwurfszeit festgelegt, und drei Sichten darauf gehen auseinander: was die Konfiguration deklariert, was die Berechtigungen erreichbar machen und was Ausführungs-Traces als tatsächlich genutzt ausweisen.

AI Bill of Materials als Governance-Artefakte verstehen

Ein AIBOM ist nicht in erster Linie eine technische Bequemlichkeit. Es ist ein Governance-Artefakt: ein Datensatz, der von einer verantwortlichen Partei erstellt wird, in einer Form, die eine andere Partei später lesen und auf deren Grundlage sie handeln kann. Diese Unterscheidung ist wichtig, weil der Leser in der Regel nicht der Autor ist und meist unter Druck kommt – ein Sicherheitsingenieur, der um 2 Uhr nachts das Ausmaß eines Vorfalls eingrenzt, ein Prüfer im Einkauf, der die Offenlegung eines Lieferanten bewertet, ein Auditor, der fragt, mit welchen Datensätzen ein Modell trainiert wurde, das vor achtzehn Monaten eine umstrittene Entscheidung getroffen hat. Vier Eigenschaften machen einen Datensatz in solchen Momenten nutzbar.

- **Verantwortliche Urheberschaft:** Jede Aussage in einem AIBOM wird von jemandem getroffen. Die Dokumentmetadaten halten fest, wer es verfasst, geprüft und freigegeben hat, wann und mit welcher Methode – und diese Metadaten ermöglichen es einem Leser, einen aus Build-Telemetrie erzeugten Datensatz von einem während einer Einkaufsprüfung von Hand entworfenen zu unterscheiden. Zwei Dokumente können identische Aussagen enthalten und dennoch ein sehr unterschiedliches Maß an Vertrauen rechtfertigen.
- **Deklarierte Geltungsbereich und eine ausdrückliche Vollständigkeitsaussage:** Ein AIBOM gibt an, was es abdeckt und wie sicher sich sein Ersteller ist, dass es dies abdeckt. Dadurch wird Schweigen interpretierbar. Ohne Vollständigkeitsaussage beweist ein nicht aufgeführter Datensatz nichts; mit einer solchen Aussage wird sein Fehlen zu einer Behauptung des Erstellers, auf die man sich verlassen und die man, falls sie falsch ist, anfechten kann. Kapitel 4 entwickelt diese Selbstbeschreibungen im Detail.

- **An Nachweise gebundene Aussagen:** Hashes, Signaturen, Model Cards, Datensatzbeschreibungen und Evaluierungsergebnisse werden einzelnen Aussagen zugeordnet und verwandeln so eine Behauptung in etwas, das ein Nutzer unabhängig überprüfen kann. Ein AIBOM muss nicht jede Aussage belegen; wo ein Nachweis fehlt, ist dieses Fehlen selbst eine Information, anhand derer der Leser entscheidet, wie viel Gewicht die Aussage tragen kann.
- **Ein kontrollierter Lebenszyklus:** Ein Governance-Artefakt wird versioniert, geprüft, freigegeben, bei wesentlichen Änderungen aktualisiert, aufbewahrt und archiviert, wenn das beschriebene System außer Betrieb genommen wird. Ein AIBOM, das einmal erstellt und nie aktualisiert wurde, beschreibt ein System, das nicht mehr existiert, und es zu lesen ist schlimmer, als gar nichts zu haben, weil es falsches Vertrauen weckt. Kapitel 7 legt die Verantwortlichkeiten und Aktualisierungsauslöser dar, die sich daraus ergeben.

Diese Eigenschaften ermöglichen es, ein AIBOM als Nachweis in bereits bestehenden Entscheidungen zu verwenden: Lieferantenbewertung und vertragliche Offenlegungsanforderungen, Release-Gating, Analyse der Auswirkungen von Schwachstellen, Eingrenzung von Vorfällen, Modellrisikoprüfung sowie Reaktion auf regulatorische Anfragen oder Audits.

Verstehen, was ein AIBOM nicht ist: Es ist keine Punktzahl, keine Bewertung und keine Reifegradnote; es hält fest, was ein System enthält, nicht wie gut oder wie sicher dieses System ist. Es bestätigt keine Compliance und ersetzt weder Evaluierung noch Bedrohungsmodellierung oder Risikobewertung – es ist ein Input, den diese Aktivitäten nutzen, und sein Wert hängt davon ab, dass sie durchgeführt werden. Ein AIBOM ist auch keine Kontrolle an sich: Es blockiert nichts, bis eine Organisation ein Gate, eine Prüfung oder eine Abfrage aufbaut, die es ausliest. Ein AIBOM, das niemand nutzt, ist ein technisches Nebenprodukt mit Governance-Etikett.

2. Was ist ein AIBOM?

Eine AI Bill of Materials, SBOM für KI oder AIBOM ist eine strukturierte, maschinenlesbare Beschreibung eines KI-Systems in vier Teilen: einem Header, der angibt, was das Dokument abdeckt und wie vollständig es zu sein beansprucht, einem Inventar der Komponenten und Dienste des Systems wie Modelle, Datensätze, Agenten und Tools, Guardrails und Laufzeitelemente, den gerichteten Datenflüssen zwischen ihnen sowie den Nachweisen über Herkunft, Rechte, Integrität und Evaluierung, die diesen Aussagen zugeordnet sind.

Das Ziel eines AIBOM ist es, Transparenz und Vertrauen entlang der gesamten KI-Lieferkette zu verbessern. Ein AIBOM schafft Sichtbarkeit, die es Organisationen ermöglicht, jede Komponente ihres KI-Ökosystems zu untersuchen und zu analysieren. AIBOM unterstützt entstehende und bestehende Praktiken und Prozesse des Risikomanagements. Es hilft, Schwachstellen zu identifizieren, KI-bezogene Sicherheitsrisiken zu bewerten und die Provenienz von Modellen von der Entwicklung bis zur Bereitstellung nachzuverfolgen. Ein AIBOM unterstützt die Bemühungen von Organisationen, geltende regulatorische und Risikomanagement-Anforderungen zu erfüllen.

Kernziele

Ein AIBOM bietet eine umfassende Aufzeichnung der KI-Artefakte und Lebenszykluskomponenten eines KI-Systems über dessen gesamten Entwurfs-, Entwicklungs- und Bereitstellungslebenszyklus hinweg und kann als primärer Nachweis für KI-Audits dienen.

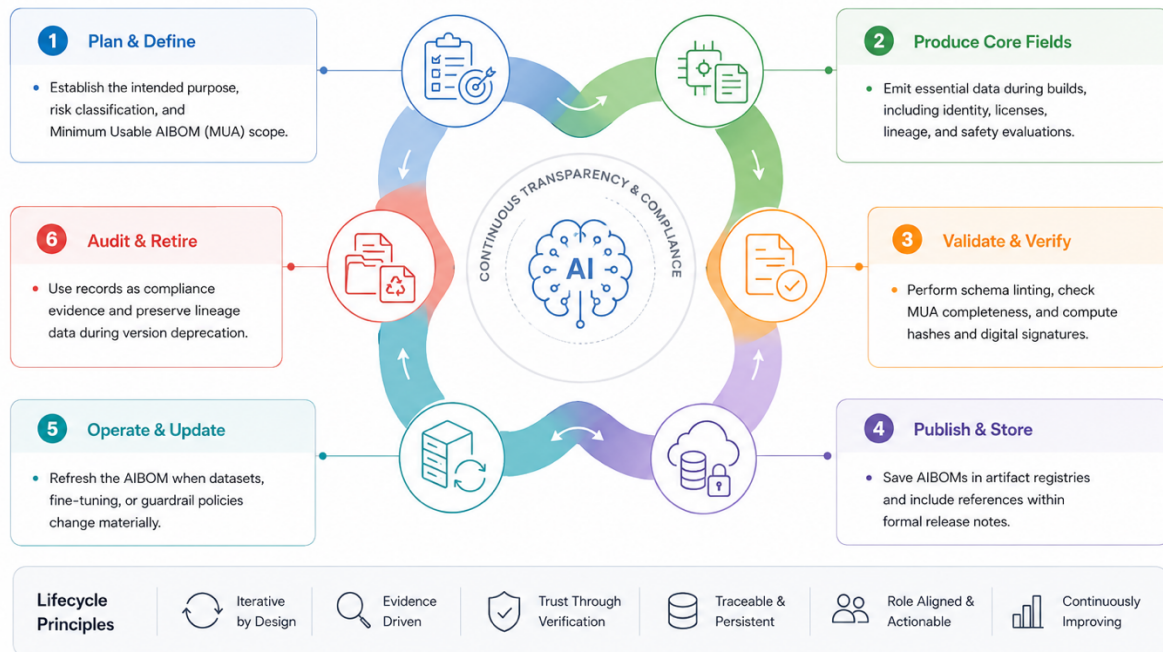
AIBOM-Artefakte bezeichnen Komponenten und Assets eines KI-Systems, darunter Lizenzen und Nutzungspflichten für Modelle und Datensätze, überprüfbare Provenienz und Evaluierungsnachweise. Beispiele für Artefakte sind:

- Modelle: Digests der Basismodelle, deklarierte und festgestellte Lizenzen, Vertriebskanäle, Signaturen, feinabgestimmte Modelle, LoRA/PEFT-Adapter, Checkpoints
- Datensätze: Datensatz-ID/Version, Hash, Lizenzen, Quell-URLs, Snapshots, Erhebungsmethoden, Sensibilität/Anonymisierung
- Agenten & Tools: Orchestratoren, APIs, Erweiterungen, MCP-Server
- Prompts & Guardrails: Vorlagen, Policy-Pakete, Sicherheitshinweise
- Laufzeit: Container, Abhängigkeiten, Zeitstempel
- Build-Provenienz: Digests der erstellten Images, die Speicherorte der Eingabequellen, die Build-Argumente und die Build-Dauer

Ein Kernziel eines AIBOM ist es, einen iterativen Prozess zur Dokumentation der Artefakte von KI-Systemkomponenten über den gesamten Entwurfs-, Entwicklungs- und Bereitstellungslebenszyklus zu systematisieren.

The AIBOM Lifecycle: A Continuous Loop for AI Transparency

A continuous cycle of six iterative stages managing an AI Bill of Materials for compliance, safety, and traceability.



Source: OWASP AIBOM Foundations Guide v1.0

Phasen der AIBOM-Entwicklung:

1. Planen: Verwendungszweck, Geltungsbereich, Risikoklasse und Minimum Usable AIBOM (MUA) festlegen.
2. Erstellen: Kernfelder während Training/Builds ausgeben, einschließlich, aber nicht beschränkt auf Identität, Lizenzen, Provenienz, Lineage, Evaluierung, Sicherheit
3. Validieren: Dateien (Schema) linten, MUA-Vollständigkeit bestätigen, Hashes und Signaturen berechnen und Lizenzen prüfen.
4. Veröffentlichen: Zusammen mit dem Artefakt speichern (Registry/Artefakt-Store), einen Verweis in den Release Notes führen.
5. Betreiben: Bei wesentlichen Änderungen aktualisieren, etwa an Datensätzen, beim Fine-Tuning, bei Lizenzen sowie bei neuen/aktualisierten Guardrail-Richtlinien.
6. Auditieren/Außerbetriebnehmen: AIBOM als primären Nachweis verwenden, Lineage bei der Abkündigung von Versionen bewahren.

3. Warum sind AIBOMs wichtig?

Moderne KI-Systeme werden selten von einem einzigen Team durchgängig entwickelt. Ein typischer Produktions-Stack bezieht ein Basismodell von einem Anbieter, führt ein Fine-Tuning mit einer Mischung aus internen und lizenzierten Daten durch, integriert Open-Source-Komponenten und bindet über Retrieval-Pipelines Live-Daten an. Jede dieser Schichten bringt eigene Risiken, Lizenzen und Provenienzfragen mit sich.

Ohne formales Inventar können Organisationen grundlegende Fragen häufig nicht beantworten: Welche Modellversion ist in der Produktion bereitgestellt? Mit welchen Daten wurde sie trainiert? Ist eine ihrer Komponenten von einer neu offengelegten Schwachstelle betroffen? Ist die Lizenz mit kommerzieller Nutzung vereinbar? Ein AIBOM beantwortet diese Fragen, indem es als maßgeblicher, abfragbarer Datensatz dessen dient, was ein KI-System tatsächlich enthält.

Der Rest dieses Dokuments untersucht die fünf wichtigsten Treiber, die AIBOM von einer nützlichen Idee zu einer betrieblichen Anforderung machen.

Sicherheitstreiber

KI-Systeme bringen neue Angriffsflächen mit sich, für deren Abdeckung herkömmliche Werkzeuge der Anwendungssicherheit nicht konzipiert wurden. Ein AIBOM ist die Grundlage jedes glaubwürdigen KI-Sicherheitsprogramms, weil es Verteidigern sagt, was sie schützen.

- **Reaktion auf Schwachstellen.** Wenn eine Schwachstelle in einem verbreiteten Modell, Framework oder Serialisierungsformat offengelegt wird (etwa die wiederkehrenden Pickle-Exploits, die öffentlich gehostete Modell-Repositories betreffen), ermöglicht ein AIBOM Sicherheitsteams, ihre Betroffenheit innerhalb von Stunden statt Wochen festzustellen.
- **Modell- und Datenvergiftung.** Böswillige Akteure können vorgelagerte Modelle oder Trainingsdatensätze kompromittieren. Ein AIBOM erfasst Provenienz, Hashes und Signaturinformationen, sodass Manipulationen vor der Bereitstellung erkannt werden können.
- **Erkennung von Backdoors und Trojanern.** Die Kenntnis der Lineage jedes Modellartefakts – einschließlich der Frage, wer es mit welchen Daten trainiert hat – ist unerlässlich, um das Risiko verborgener Verhaltensweisen zu bewerten, die durch bestimmte Eingaben ausgelöst werden.

- **Offenlegung von Geheimnissen und Prompts.** Die Inventarisierung von Prompts, Systemanweisungen und eingebetteten Zugangsdaten hilft, eine versehentliche Offenlegung zu verhindern, wenn Modelle weitergegeben oder nachgelagert feinabgestimmt werden.
- **Incident Response.** Wenn ein KI-System schädliche oder anomale Ausgaben erzeugt, ermöglicht ein AIBOM den Einsatzkräften, genau zu rekonstruieren, was zum Zeitpunkt des Vorfalls lief, mit welchen Daten und mit welcher Konfiguration.

Regulatorische Treiber

Die regulatorische Landschaft für KI hat sich in bemerkenswert kurzer Zeit von freiwilligen Grundsätzen zu durchsetzbaren Verpflichtungen gewandelt. AIBOMs entwickeln sich zum praktischen Mechanismus, um die Einhaltung mehrerer Rahmenwerke gleichzeitig nachzuweisen.

- **KI-spezifische Regulierung.** Neu entstehende KI-Vorschriften¹ verlangen zunehmend, dass für Hochrisiko-KI-Systeme eine technische Dokumentation geführt wird, die Trainingsdaten, Modellarchitektur, Leistungskennzahlen und bekannte Einschränkungen abdeckt. Ein AIBOM setzt diese Anforderungen in einem Standardformat operativ um.
- **KI-Risikomanagement-Rahmenwerke².** Ihre Kernfunktionen fordern ausdrücklich dokumentierte Inventare von KI-Komponenten, Daten und Abhängigkeiten als Voraussetzung für die Risikobewertung.
- **Öffentliche Beschaffung und behördliche Leitlinien³.** Öffentliche Stellen verlangen zunehmend eine Dokumentation von KI-Systemen, wenn sie KI beschaffen oder einsetzen – analog zu den SBOM-Vorgaben, die für herkömmliche Software eingeführt wurden.
- **Branchenspezifische Vorschriften.** Finanzaufsichtsbehörden, Gesundheitsbehörden und Datenschutzaufsichtsbehörden nähern sich ähnlichen Offenlegungserwartungen für KI-gestützte Entscheidungen an.
- **Grenzüberschreitende Übermittlungen.** Wo Vorschriften zur Datenresidenz oder Datensouveränität gelten, liefert ein AIBOM den Prüfpfad, der benötigt wird, um nachzuweisen, dass Trainingsdaten und Modellkomponenten die Anforderungen der jeweiligen Rechtsordnung erfüllen.

¹ EU AI Act, Verordnung (EU) 2024/1689 (siehe Art. 11 und Anhang IV): eur-lex.europa.eu/eli/reg/2024/1689/oj/eng

² NIST AI RMF 1.0 (NIST AI 100-1):

nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10

³ NTIA Minimum Elements for a Software Bill of Materials (2021):

ntia.gov/report/2021/minimum-elements-software-bill-materials-sbom; sowie die Aktualisierung der CISA von 2025: cisa.gov/.../2025_CISA_SBOM_Minimum_Elements.pdf

Governance-Treiber

Die interne Governance hinkt der Einführung von KI historisch hinterher. Viele Organisationen stellen – oft während eines Audits oder Vorfalls – fest, dass niemand eindeutig sagen kann, welche Modelle in Produktion sind, wem sie gehören oder wie sie validiert wurden. AIBOMs geben Governance-Funktionen ein konkretes Artefakt, an dem sie Richtlinien, Prüfungen und Freigaben verankern können.

- **Management des Modelllebenszyklus.** Ein AIBOM verfolgt jede Version, jedes Fine-Tuning und jede Bereitstellung eines Modells und ersetzt informelles Stammeswissen durch prüfbare Aufzeichnungen.
- **Prüfung auf Verzerrung, Fairness und Ethik.** Prüfer können nicht beurteilen, ob ein Modell für Personalauswahl, Kreditvergabe oder das Gesundheitswesen geeignet ist, ohne zu wissen, mit welchen Daten es trainiert wurde. Das AIBOM ist der Input für diese Bewertungen.
- **Einhaltung von Lizenzen und geistigem Eigentum.** Modelle und Datensätze unterliegen unterschiedlichsten Lizenzen – manche permissiv, manche restriktiv und manche mit ungeklärten Fragen zum Urheberrecht an Trainingsdaten. Diese Verpflichtungen frühzeitig sichtbar zu machen, verhindert später kostspielige Nachbesserungen.
- **Drittpartei- und Lieferantenrisiko.** Wenn KI-Fähigkeiten von Lieferanten bezogen werden, ist ein AIBOM zunehmend Teil der Due Diligence, neben SOC-2-Berichten und Sicherheitsfragebögen.
- **Sichtbarkeit auf Vorstandsebene.** Aggregierte AIBOM-Daten ermöglichen es Führungskräften und Aufsichtsgremien, den KI-Fußabdruck der Organisation in einer einzigen Ansicht zu sehen statt als verstreute Projektartefakte.

Geschäftskontinuität

KI-Systeme werden oft unbemerkt kritisch. Eine Empfehlungsmaschine, ein Betrugserkennungsmodell oder ein Dokumentenverarbeitungssystem kann sich innerhalb von Monaten vom Pilotprojekt zur tragenden Infrastruktur entwickeln. Wenn etwas ausfällt, bemessen sich die Kosten eines fehlenden AIBOM in Ausfallzeiten, Umsatzverlusten und Reputationsschäden.

- **Reproduzierbarkeit.** Wenn ein Produktionsmodell verloren geht, beschädigt wird oder neu trainiert werden muss, erfasst ein AIBOM die Eingaben, Parameter und Abhängigkeiten, die für einen vorhersehbaren Neuaufbau erforderlich sind.
- **Ausfall oder Abkündigung durch Anbieter.** Wenn ein Modellanbieter einen Dienst einstellt, die Preise ändert oder ein Modell abkündigt, sagt Ihnen ein

AIBOM, welche Systeme betroffen sind und welche Alternativen kompatibel sind.

- **Drift und Verschlechterung.** Die Nachverfolgung von Modell- und Datenversionen über die Zeit ermöglicht es, Qualitätsrückgänge mit konkreten Änderungen in Beziehung zu setzen, statt zu raten.
- **Regulatorische Sicherungsanordnungen und Audits.** Wenn Aufsichtsbehörden oder Gerichte Nachweise darüber verlangen, was ein System an einem bestimmten Datum getan hat, liefert ein AIBOM die historische Aufzeichnung.
- **Notfallwiederherstellungsplanung.** Wiederherstellungszeit- und Wiederherstellungspunktziele für KI-Systeme sind nur dann aussagekräftig, wenn die zur Wiederherstellung benötigten Komponenten inventarisiert und zugänglich sind.

Bedrohungen der Lieferkette

Die KI-Lieferkette ist breiter, tiefer und weniger ausgereift als die herkömmliche Software-Lieferkette. Komponenten werden aus öffentlichen Modell-Hubs, Datensatz-Repositories und Inferenz-APIs bezogen, deren Sicherheitshygiene sehr unterschiedlich ist. AIBOMs machen die Lieferkette so sichtbar, dass sie verteidigt werden kann.

- **Kompromittierte öffentliche Modell-Repositories.** Öffentliche Hubs haben wiederholt Modelle gehostet, die in ihren Serialisierungsformaten Schadcode enthielten. Ein AIBOM mit verifizierten Hashes macht diese Angriffsklasse erkennbar.
- **Typosquatting und Identitätsvortäuschung.** Angreifer veröffentlichen zum Verwechseln ähnliche Modelle oder Pakete, die versehentlich bezogen werden sollen. Die Provenienzverfolgung in einem AIBOM ist die wichtigste Verteidigung.
- **Kontamination von Datensätzen.** Trainings- oder Fine-Tuning-Daten aus dem öffentlichen Web können gezielt mit vergifteten Inhalten versetzt werden. Die Katalogisierung von Datensatzquellen und Verarbeitungsschritten ist der Ausgangspunkt für die Erkennung.
- **Transitive Abhängigkeiten.** Jedes Modell bringt vorgelagerte Abhängigkeiten mit sich – von Tokenizern, Basisgewichten und unterstützenden Bibliotheken. AIBOMs machen diese transitiven Beziehungen explizit.
- **Insider- und Entwicklerrisiko.** Interne Fine-Tuning- und Datenlabeling-Pipelines können durch ungeprüfte Beiträge Risiken einführen. AIBOM-gestützte Kontrollen behandeln interne Artefakte mit derselben Sorgfalt wie externe.

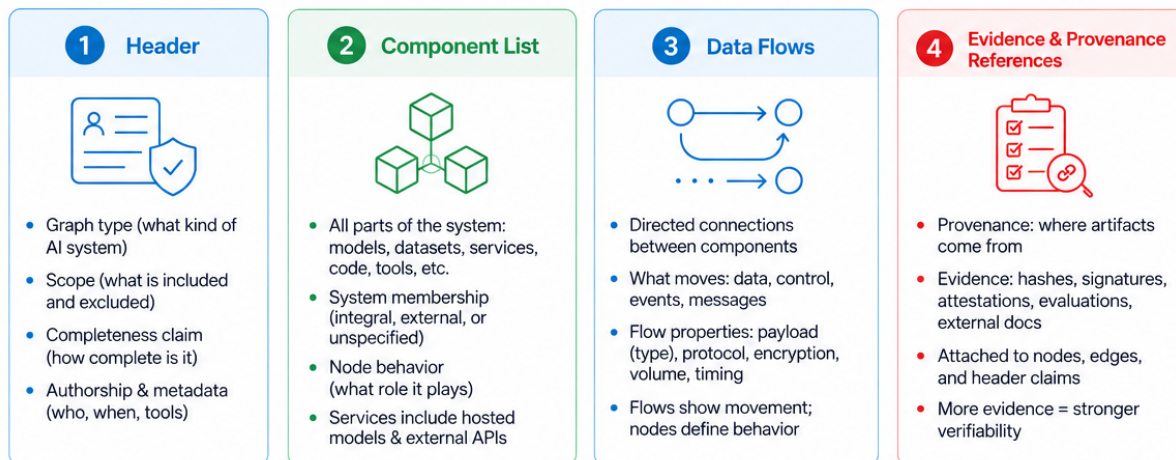
4. Kernkomponenten eines AIBOM

Die vorangegangenen Kapitel haben erklärt, was ein AIBOM ist und warum es wichtig ist. Dieses Kapitel stellt die Bestandteile eines AIBOM vor und erläutert, wozu jeder Teil dient, damit ein Leser weiß, was hineingehört und warum.

Strukturell besteht ein AIBOM aus vier Teilen: **einem Header, einer Komponentenliste, gerichteten Datenflüssen sowie Nachweis- und Provenienzreferenzen**. Jedes AIBOM enthält alle vier. Zwei optionale Arten von Inhalten vervollständigen das Bild, wo ein System sie erfordert: Vertrauenskontext, der Komponenten zu Zonen gruppiert und die Grenzen zwischen ihnen markiert, und Verhaltensweisen, die bedingtes Handeln für Systeme beschreiben, deren Datenbewegung von Laufzeitentscheidungen abhängt. Dieses Kapitel geht jeden dieser Teile der Reihe nach durch, verknüpft ihn mit dem Zweck, dem er dient, und schließt mit dem, was ein fertiges AIBOM ermöglicht.

Core Components of an AIBOM

Every AIBOM has four parts. Two optional kinds of content complete the picture.



Der Header

Der AIBOM-Header beschreibt, welche Art von KI-System modelliert wird und welche Aussagen hinsichtlich Geltungsbereich und Vollständigkeit getroffen werden können. Diese Informationen liefern den Kontext für das Lesen des restlichen Dokuments. Das AIBOM eines Erstellers könnte ein vollständiges Bild eines einzelnen bereitgestellten Systems sein, automatisch aus Build-Telemetrie erzeugt. Ein anderes könnte eine unvollständige Skizze eines künftigen Systems sein, von Hand während einer Einkaufsprüfung entworfen. Ohne eine Möglichkeit, diese zu unterscheiden, kann ein Leser aus keinem der beiden die richtigen Schlüsse ziehen.

Jedes AIBOM beginnt daher mit **drei Selbstbeschreibungen**: (1) Der Graphtyp gibt an, welche Art von System das AIBOM darstellt, etwa ein einzelnes Modell, ein Laufzeitsystem, eine Systemerweiterung wie einen MCP-Server oder eine Trainingspipeline. (2) Der Geltungsbereich gibt an, was das AIBOM abzudecken versucht: welches System, welche Version, welche Bereitstellung, welche Schnittstellen. (3) Die Vollständigkeitsaussage gibt an, wie sicher sich der Ersteller ist, dass das AIBOM abdeckt, was sein Geltungsbereich zu umfassen angibt – von „vollständig für den deklarierten Geltungsbereich“ über teilweise Abdeckung bis hin zu „wir wissen nicht, was sonst noch darin steckt“. Neben diesen Selbstbeschreibungen halten Standard-Dokumentmetadaten fest, wer das AIBOM verfasst, geprüft und freigegeben hat, wann und mit welchen Werkzeugen. Anhand dieser Metadaten unterscheidet ein Leser einen automatisch erzeugten Datensatz von einer von Hand entworfenen Skizze.

Die Vollständigkeitsaussage bestimmt, was ein Leser aus dem Fehlen von Informationen sinnvoll schließen darf. Wenn ein AIBOM einen bestimmten Datensatz nicht aufführt und keine Vollständigkeit beansprucht, beweist das Fehlen nichts. Der Datensatz könnte in Gebrauch und einfach nicht aufgeführt sein. Beansprucht das AIBOM jedoch Vollständigkeit für den deklarierten Geltungsbereich, wird das Fehlen aussagekräftig: Es ist nun eine Behauptung des Erstellers, dass kein solcher Datensatz innerhalb des Geltungsbereichs beteiligt ist.

Diese Asymmetrie zwischen positiver und negativer Schlussfolgerung ist der Grund, warum die Selbstbeschreibung des Graphen vor dessen Inhalt stehen muss. Ohne Geltungsbereich und Vollständigkeit können zwei Leser dasselbe AIBOM betrachten und zu entgegengesetzten Schlüssen darüber kommen, was es aussagt. Die übrigen Abschnitte dieses Kapitels gehen davon aus, dass jedes AIBOM alle drei Selbstbeschreibungen enthält, auch wenn der Text sie nicht wiederholt.

Die Komponentenliste

Ein KI-System besteht aus einer Reihe miteinander verbundener Komponenten. Sie sind die „Materialien“ in der AI Bill of Materials. In einem AIBOM ist alles im System eine Komponente, auch Dienste: Ein eigenständiges Codemodul, ein Microservice, ein Datensatz und ein von einem Anbieter gehosteter API-Endpunkt sind allesamt Komponenten, die anhand ihres Komponententyps unterschieden werden und nicht durch getrennte Listen. Die Komponentenliste beantwortet die erste Frage, die jeder Prüfer an ein System stellt: Wovon bin ich abhängig?

Zwei Dinge beschreiben einen Knoten über seine Identität hinaus: Systemzugehörigkeit und Knotenverhalten. Die Systemzugehörigkeit fragt, ob ein Knoten als integraler Bestandteil des beschriebenen KI-Systems gilt, als extern oder als nicht spezifiziert. Dies ist eine auf den Geltungsbereich bezogene Aussage und wird strukturell festgehalten: Der deklarierte Geltungsbereich des AIBOM listet auf, welche Komponenten zum beschriebenen System gehören und welche davon ausgeschlossen sind. Das ist nicht dasselbe wie Eigentum oder Kontrolle. Ein von einem Drittanbieter gehostetes Modell kann integraler Bestandteil des beschriebenen KI-Systems sein, und

eine intern betriebene Datenbank kann dennoch außerhalb der Grenze des KI-Systems liegen, wenn sie sich außerhalb des deklarierten Geltungsbereichs befindet. Eine im deklarierten Geltungsbereich enthaltene Komponente ist Bestandteil des beschriebenen Systems.

Das Knotenverhalten hingegen fragt, welche Rolle der Knoten im Graphen spielt. Manche Knoten verrichten Arbeit. Ein Prompt-Prozessor transformiert Text. Ein Modell erzeugt Ausgaben. Ein Agent entscheidet, was als Nächstes aufgerufen wird. Eine Guardrail filtert Inhalte. Ein Retrieval-Dienst ordnet Kandidaten nach Relevanz. Andere Knoten speichern, liefern oder empfangen hauptsächlich Informationen, etwa Datensätze, Indizes, Protokolle und Datenspeicher. Ein Knoten, der die Informationen nicht direkt verarbeitet, kann informativ als „passive“ Komponente annotiert werden, oder dieser Status kann aus dem Muster der mit ihm verbundenen Datenflüsse abgeleitet werden.

Gehostete Modelle gehören für viele AIBOMs zu den wichtigsten Komponenten vom Typ Dienst, doch es gibt viele weitere Arten. Eine Dienstkomponekte kann einem Modell oder einer KI-Anwendung über eine Netzwerk-API Zugriff auf Daten, Tools oder Systeme verschaffen. Das AIBOM erfasst diese Komponenten, damit ein Leser erkennen kann, welche Teile des Systems Daten austauschen und wo Systemgrenzen überschritten werden.

Die Datenflüsse

Die Datenflüsse in einem AIBOM verbinden die Komponenten miteinander. Wenn die Komponenten die „Stückliste“ sind, sind die Datenflüsse der „Bauplan“ oder „Schaltplan“, der zeigt, wie sie zusammenpassen. Datenflüsse sind immer gerichtet: Sie zeigen Informationen, die sich von einer Komponente zu einer anderen bewegen. Wenn sich Informationen zwischen zwei Komponenten in beide Richtungen bewegen, etwa eine Anfrage und ihre Antwort, ist jede wesentliche Bewegung ein eigener Fluss.

Wenn die Komponentenliste die Frage beantwortet, wovon ich abhängig bin, beantworten die Datenflüsse die Frage, die KI-Systeme besonders macht: Was kann was erreichen? Dasselbe Modell hat ein völlig anderes Risikoprofil, je nachdem, ob Kundendaten es ungefiltert erreichen können und wohin seine Ausgaben gehen. Eine Stückliste kann diesen Unterschied nicht ausdrücken. Bei KI-Systemen liegt das Risiko in den Verbindungen.

Jeder Fluss kann Informationen darüber enthalten, was sich entlang des Flusses bewegt. Ein Fluss deklariert seine Art – ein Datenfluss, ein Kontrollfluss, ein Ereignis, eine Nachricht – und kann die transportierte Nutzlast beschreiben, entweder durch Verweis auf an anderer Stelle im AIBOM beschriebene Datenobjekte oder mit einem Inline-Deskriptor, wenn sich die Nutzlast auf diesem bestimmten Fluss von allem an anderer Stelle Beschriebenen unterscheidet. Ein Fluss kann außerdem festhalten, wie die Bewegung erfolgt: ob sie verschlüsselt ist, über welches Protokoll, in welchem Umfang und mit welchem Timing.

Was ein Fluss nicht enthält, ist eine Beschreibung dessen, was die empfangende Komponente mit den Daten macht. Ein Modell, das Ausgaben erzeugt, ein Retrieval-Dienst, der Kandidaten ordnet, eine Guardrail, die Inhalte filtert, oder eine Datenbank, die das Empfangene speichert – das ist Knotenverhalten: Es wird am Knoten beschrieben, über seinen Komponententyp und seine Metadaten, nicht als Annotation am Fluss. Ein Fluss hält fest, dass Daten von einem Ort zu einem anderen gelangt sind; die Knoten erklären, was an jedem Ende geschieht.

Nachweis- und Provenienzreferenzen

Der Header, die Komponenten und Dienste sowie die Datenflusskanten sagen aus, was das System enthält und wie Informationen durch es fließen. Nachweis- und Provenienzreferenzen sagen aus, warum ein Leser diesen Aussagen glauben sollte und wo sie überprüft werden können. Sie bilden keinen separaten Graphen; sie sind Anhänge an bestimmte Knoten, Kanten und Selbstbeschreibungen.

Zwei Arten von Anhängen sind am wichtigsten. Provenienz hält fest, wie ein Artefakt entstanden ist: von welchem Basismodell ein feinabgestimmtes Modell abgeleitet ist, welcher Trainingslauf es hervorgebracht hat, welcher Datensatz-Snapshot einen Retrieval-Index gespeist hat. Nachweise halten die Stützung einer Aussage fest: einen kryptografischen Hash, der die Integrität einer Komponente verankert, eine Signatur oder Attestierung, die ein Artefakt mit dem Build verknüpft, der es erzeugt hat, ein Evaluierungsergebnis oder ein externes Dokument wie eine Model Card oder eine Lizenzdatei.

Da diese Referenzen einzelnen Aussagen zugeordnet sind, wirken sie direkt mit der Vollständigkeitsaussage im Header zusammen. Eine Komponente, die ohne jede Nachweisreferenz angegeben wird, ist eine bloße Behauptung; dieselbe Komponente mit einem Hash, einer Signatur und einem Link zu ihrer Model Card ist eine Aussage, die ein Nutzer unabhängig überprüfen kann. Ein AIBOM muss nicht jeder Aussage einen Nachweis beifügen, aber das Vorhandensein oder Fehlen von Nachweisen ist selbst eine Information, anhand derer ein Leser entscheidet, wie viel Gewicht ein bestimmter Knoten oder eine bestimmte Kante tragen kann.

Die Elemente, die ein Ersteller zu diesen vier Teilen festhält – Modellmetadaten, Datensatzprovenienz, Trainingsumgebung, Abhängigkeiten und Bibliotheken, APIs und Toolchains, Laufzeitkonfiguration sowie Agenten- und Sub-Agenten-Deskriptoren für agentische Systeme –, sind keine separate Liste, die gepflegt werden muss. Jedes davon ist entweder ein Knoten, eine Eigenschaft eines Knotens, eine Kante oder eine Nachweisreferenz, die einem dieser Elemente zugeordnet ist.

Vertrauenszonen und Grenzen

Nicht alle Komponenten befinden sich am selben Ort oder unter derselben Kontrolle. Ein AIBOM kann Komponenten in Zonen unterschiedlichen Vertrauens gruppieren: die eigene Infrastruktur des Teams, das weitere Unternehmensnetzwerk, die Cloud eines Anbieters, das öffentliche Internet. Wo ein Fluss von einer Zone in eine andere übergeht, liegt eine Grenze, und an Grenzen konzentriert sich die Aufmerksamkeit bei Prüfungen. Ein Grenzübergang ist die Stelle, an der Daten den Besitzer wechseln, an der Kontrollen wie Authentifizierung und Verschlüsselung greifen und an der die Sichtbarkeit des Erstellers endet.

Zonen existieren, um einen Unterschied explizit zu machen, den ein bloßer Graph verbirgt. Ein Fluss zwischen zwei internen Komponenten und ein Fluss, der dieselbe Nutzlast an eine Drittanbieter-API übermittelt, sehen wie ähnliche Kanten aus, sind aber sehr unterschiedliche Tatsachen. Ohne deklarierte Zonen hängt ihre Unterscheidung vom Hintergrundwissen des Lesers ab, welche Namen intern sind. Mit ihnen sagt das AIBOM selbst, welche Bewegungen im eigenen Bereich bleiben und welche ihn verlassen – und was jeder dieser Abgänge auf dem Weg nach draußen passieren muss.

Verhaltensweisen

Flüsse sagen, was was erreichen kann. Verhaltensweisen sagen, wann und unter welchen Bedingungen. Bei einer linearen Pipeline reichen die Flüsse in der Regel aus: Bei jeder Anfrage finden dieselben Bewegungen statt. Agentische Systeme sind anders. Die interessante Tatsache an einem Agenten ist oft, dass er zur Laufzeit entscheidet, ob er überhaupt handelt – ob er ein Tool aufruft, eine Datenbank abfragt oder an einen anderen Agenten übergibt. Ein Fluss allein übertreibt entweder, was geschieht (das Tool wird jedes Mal aufgerufen), oder untertreibt es (die Verbindung wird nicht erfasst). Eine Verhaltensbeschreibung erfasst den bedingten Teil: was eine Aktion auslöst, was die Komponente daraufhin tun darf und was daraus folgt. Die meisten AIBOMs werden keine Verhaltensbeschreibungen benötigen; Systeme, deren Risiko von Laufzeitentscheidungen abhängt, schon. Das Kapitel über agentische KI-Systeme kommt darauf zurück.

Was ein AIBOM ermöglicht

Alles, was dieses Kapitel beschrieben hat, sind strukturierte Fakten, und jedes Element davon – jede Komponente, jeder Fluss, jede Zone und jede Grenze – trägt einen stabilen Bezeichner. Das unterscheidet ein AIBOM von einem Architekturdiagramm: Andere Artefakte können auf genau ein Element darin verweisen.

Auf dieser Referenzierbarkeit baut der Rest des Governance-Stacks auf. Bedrohungsmodelle, Risikoregister, Kontrollzuordnungen und Schwachstellenaussagen beschreiben das System nicht erneut; sie verweisen auf den Graphen des AIBOM. Eine Bedrohung kann angeben, dass ein Angriffspfad eine bestimmte Grenze überschreitet. Eine Kontrolle kann angeben, welchen Fluss sie schützt. Eine Schwachstellenanalyse

kann angeben, dass eine Schwäche nicht erreichbar ist, weil die Kontrollen einer Zone den Pfad dorthin blockieren. Erstellen Sie den Graphen einmal, und die Analysen knüpfen daran an – und bleiben daran geknüpft und überprüfbar, während sich das System weiterentwickelt.

Das AIBOM selbst bleibt auf seiner Seite dieser Linie. Es hält fest, was das System ist, nicht ob es akzeptabel ist. Compliance, Risikoakzeptanz und Angemessenheit der Kontrollen sind Urteile, die ein Leser auf Grundlage seines eigenen Richtlinienrahmens zu den Fakten hinzufügt, die das AIBOM liefert.

5. AIBOM-Struktur

Das Kapitel „Kernkomponenten eines AIBOM“ hat die vier Teile eines AIBOM – Header, Komponenten und Dienste, Datenflusskanten sowie Nachweis- und Provenienzreferenzen – als abstraktes Modell dargelegt. Dieses Kapitel macht dieses Modell konkret. Es arbeitet Beispiele eines gehosteten Inferenzsystems, einer Fine-Tuning-Pipeline und eines durchgängigen Ablaufs durch, der ein AIBOM für ein veröffentlichtes Modell erzeugt; es stellt klar, was ein AIBOM aussagt und was nicht; und es führt die Kodierungsstufen und -ebenen ein, die ein Ersteller benötigt, wenn er tatsächlich eines ausgeben will.

Die drei Selbstbeschreibungen im Header – Graphtyp, Erzeugungsmethode, Geltungsbereich und Vollständigkeit – werden durchgehend vorausgesetzt und hier nicht erneut hergeleitet; siehe „Der Header“ in „Kernkomponenten eines AIBOM“. Ohne sie ist nichts anderes im AIBOM interpretierbar, daher ist jedes der folgenden Beispiele so geschrieben, als wäre sein Header bereits vorhanden, auch wenn der Text ihn nicht wiederholt.

Was jedes AIBOM deklariert

Bevor jemand ein AIBOM lesen kann, muss er wissen, um welche Art von AIBOM es sich handelt.

Dieselben Worte können sehr unterschiedliche Artefakte beschreiben. Das AIBOM eines Erstellers könnte ein vollständiges Bild eines einzelnen bereitgestellten Systems sein, automatisch aus Build-Telemetrie erzeugt. Das eines anderen könnte eine unvollständige Skizze eines künftigen Systems sein, von Hand während einer Einkaufsprüfung entworfen. Ohne eine Möglichkeit, diese zu unterscheiden, kann ein Leser aus keinem der beiden die richtigen Schlüsse ziehen.

Jedes AIBOM beginnt daher mit drei Selbstbeschreibungen:

Der **Graphtyp** gibt an, was das AIBOM darstellt: ein einzelnes Modell, einen einzelnen Datensatz, ein Laufzeitsystem, eine Trainingspipeline oder ein anderes klar definiertes Artefakt. Er sagt dem Leser, welche Art von Knoten und Flüssen zu erwarten ist.

Der **Geltungsbereich** gibt an, was das AIBOM abzudecken versucht: welches System, welche Version, welche Bereitstellung, welche Schnittstellen. Der Geltungsbereich ist wichtig, weil sich jede andere Aussage im AIBOM auf ihn bezieht.

Die **Vollständigkeitssage** gibt an, wie sicher sich der Ersteller ist, dass das AIBOM abdeckt, was sein Geltungsbereich zu umfassen angibt. Die Vollständigkeit reicht von „vollständig für den deklarierten Geltungsbereich“ über teilweise Abdeckung bis hin zu „wir wissen nicht, was sonst noch darin steckt“.

Neben diesen Selbstbeschreibungen halten die Dokumentmetadaten des AIBOM fest, wer es verfasst, geprüft und freigegeben hat, wann, mit welchen Werkzeugen und für welchen Gültigkeitszeitraum. Anhand dieser Metadaten unterscheidet ein Leser ein während einer Einkaufsprüfung von Hand verfasstes AIBOM von einem, das aus der Build-Konfiguration erzeugt oder aus Laufzeitlemetrie erfasst wurde – und hier erhalten die Aussagen eines Erstellers einen verantwortlichen Autor.

Die Vollständigkeitsaussage leistet mehr, als es den Anschein hat. Sie bestimmt, was ein Leser aus dem Fehlen von Informationen schließen darf. Wenn ein AIBOM einen bestimmten Datensatz nicht aufführt und keine Vollständigkeit beansprucht, beweist das Fehlen nichts: Der Datensatz könnte in Gebrauch und einfach nicht aufgeführt sein. Beansprucht das AIBOM Vollständigkeit für den deklarierten Geltungsbereich, wird das Fehlen aussagekräftig: Es ist nun eine Behauptung des Erstellers, dass kein solcher Datensatz innerhalb des Geltungsbereichs beteiligt ist.

Diese Asymmetrie zwischen positiver und negativer Schlussfolgerung ist der Grund, warum die Selbstbeschreibung des Graphen vor dessen Inhalt stehen muss. Ohne Geltungsbereich und Vollständigkeit können zwei Leser dasselbe AIBOM betrachten und zu entgegengesetzten Schlüssen darüber kommen, was es aussagt.

Die übrigen Abschnitte dieses Kapitels gehen davon aus, dass jedes AIBOM diese drei Selbstbeschreibungen enthält, auch wenn der Text sie nicht wiederholt.

Ausgearbeitetes Beispiel: ein gehostetes Inferenzsystem

Betrachten wir einen Kundensupport-Assistenten in einem Unternehmen. Ein Kunde gibt eine Frage in eine Weboberfläche ein. Die Anwendung stellt einen Prompt zusammen, ruft relevante interne Dokumentation ab, prüft die Eingabe auf unsichere Inhalte, sendet den Prompt an ein gehostetes großes Sprachmodell, bereitet die Antwort nach und gibt sie an den Kunden zurück.

Das AIBOM für dieses System hat die folgenden Knoten:

Ein **nutzerseitiger Dienst** nimmt die Kundenanfrage entgegen. Er ist eine Komponente vom Typ Dienst, die vom Assistenten-Team betrieben wird.

Ein **Prompt-Prozessor** stellt den endgültigen Prompt zusammen. Er wendet Vorlagen an, schwärzt personenbezogene Informationen, verwaltet den Sitzungskontext und entscheidet, was abgerufen wird. Das Team besitzt ihn als Komponente.

Ein **Retrieval-Dienst** hält Embeddings der internen Supportdokumentation vor und gibt relevante Passagen zurück. Das Team betreibt ihn auf interner Infrastruktur. Das zugrunde liegende Korpus, ein Snapshot der internen Support-Wissensdatenbank zu einem bestimmten Datum, wird als Datensatzkomponente referenziert.

Ein **Moderationsdienst** prüft Kundeneingaben auf Inhalte, auf die der Assistent nicht eingehen sollte. In diesem Beispiel abonniert das Team eine Moderations-API eines Drittanbieters.

Ein **gehostetes Sprachmodell** erzeugt die Antwort des Assistenten. Das Team greift über die API des Anbieters darauf zu. Dies ist der interessanteste Knoten im Graphen, und der Text kommt weiter unten auf ihn zurück.

Ein **Antwort-Handler** bereitet die Modellausgabe nach: Er formatiert die Antwort, protokolliert den Austausch für die Qualitätsprüfung und gibt den endgültigen Text an den nutzerseitigen Dienst zurück.

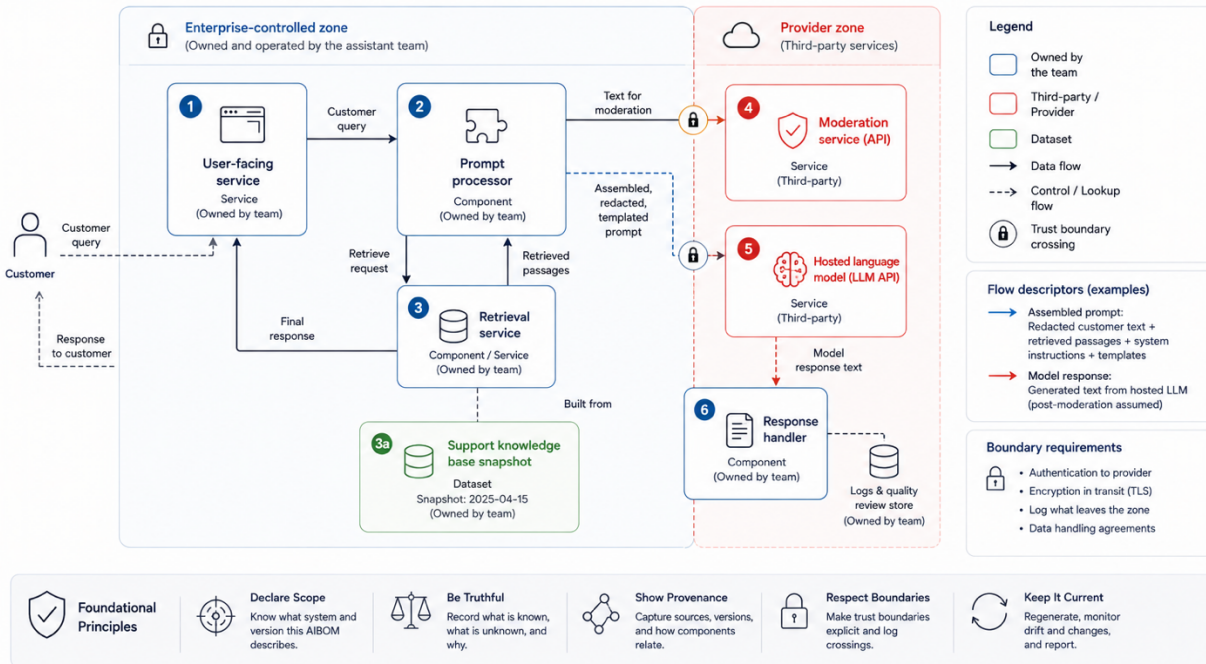
Das AIBOM verbindet diese Knoten mit gerichteten Datenflüssen. Die Kundenanfrage bewegt sich vom nutzerseitigen Dienst zum Prompt-Prozessor. Der Prozessor sendet Text an den Moderationsdienst, ruft Passagen vom Retrieval-Dienst ab und sendet den zusammengestellten Prompt an das gehostete Sprachmodell. Das Modell gibt Text an den Antwort-Handler zurück. Der Antwort-Handler gibt Text an den nutzerseitigen Dienst zurück, der ihn an den Kunden zurückgibt.

Jeder Fluss enthält Informationen darüber, was sich entlang des Flusses bewegt hat. Der an das gehostete Modell gesendete Prompt ist nicht die rohe Kundenanfrage; er wurde durch Retrieval angereichert, geschwärzt und in eine Vorlage eingebettet. Der Fluss, der ihn transportiert, kann eine Inline-Beschreibung der Nutzlast enthalten, die auf genau diesem Fluss übertragen wurde, statt sich auf eine allgemeine Beschreibung dessen zu stützen, was der Prompt-Prozessor ausgibt. Deskriptoren pro Fluss sind wichtig, wenn sich die Nutzlast in einer Phase des Systems wesentlich von der Nutzlast in einer anderen Phase unterscheidet.

Das Beispiel enthält auch Vertrauenskontext. Der nutzerseitige Dienst, der Prompt-Prozessor, der Retrieval-Dienst und der Antwort-Handler befinden sich in einer Zone, die das Team kontrolliert. Die Moderations-API und das gehostete Sprachmodell befinden sich in Anbieterzonen. Die beiden Flüsse, die die Zone des Teams verlassen – die Moderationsprüfung und der zusammengestellte Prompt –, überschreiten deklarierte Grenzen, und jede Grenze hält fest, was ihre Überschreitung erfordert: Authentifizierung gegenüber dem Anbieter, Verschlüsselung bei der Übertragung, Protokollierung dessen, was die Zone verlassen hat. Der Prompt-Fluss verdient die größte Aufmerksamkeit, und der Graph selbst sagt, warum: Er transportiert geschwärzten Kundentext und abgerufene interne Dokumentation über eine Grenze hinweg in eine Infrastruktur, die das Team nicht einsehen kann.

Worked example: a hosted inference system

An enterprise customer support assistant that retrieves internal knowledge, moderates input, and uses a hosted LLM to generate responses.



Source: OWASP AIBOM Foundations Guide v1.0

Zwei strukturelle Entscheidungen in diesem Beispiel verdienen eine Erörterung.

Das gehostete Sprachmodell wird als Komponente vom Typ Dienst dargestellt, deren Grenze der Endpunkt des Anbieters ist. Der Grund dafür ist, dass die Datenbewegung am Endpunkt des Anbieters endet, nicht innerhalb der Infrastruktur des Anbieters. Das Team hat keinen Einblick in die Serving-Topologie des Anbieters, und ein AIBOM, das etwas anderes vorgäbe, würde Tatsachen behaupten, die das Team nicht nachweisen kann. Die Endpunktkomponente ist das, was das Team tatsächlich sieht: ein Endpunkt mit einer Adresse, einer Authentifizierungsmethode und einem Vertrag.

Das Modell selbst fehlt im AIBOM jedoch nicht. Die Endpunktkomponente enthält einen Verweis auf eine Komponente vom Typ Machine-Learning-Modell, und diese Komponente enthält Identität, Version, Lizenz, Verwendungszweck, Trainingsmerkmale und weitere Model-Card-Metadaten des Modells. Ein Leser kann mit einem einzigen Schritt ermitteln, welches Modell der Endpunkt bereitstellt. Wenn der Anbieter später den Alias ändert und ihn auf einen neueren Modell-Snapshot verweisen lässt, bleibt die Endpunktkomponente unverändert. Der Verweis auf die Modellkomponente ändert sich. Die nächste Revision des AIBOM hält die Versionsänderung fest.

Die zweite Entscheidung betrifft die Frage, wie jeder Knoten als Teil des Systems oder außerhalb davon gekennzeichnet wird. Der nutzerseitige Dienst, der Prompt-Prozessor, der Retrieval-Dienst und der Antwort-Handler sind Teil des Systems, das das AIBOM beschreibt: Das Team betreibt sie, und der deklarierte Geltungsbereich schließt sie ein.

Bei der Moderations-API des Drittanbieters ist dies eher strittig. Wenn das Team die Moderationsfähigkeit als Teil dessen betrachtet, was den sicheren Betrieb des Assistenten ermöglicht, ist der Moderationsdienst Bestandteil des Systems. Wenn das Team ihn als externe Abhängigkeit betrachtet, die der Assistent eben aufruft, ist er extern. Beide Entscheidungen können richtig sein; das AIBOM macht die Entscheidung explizit, damit ein Prüfer weiß, welches Modell der Ersteller im Sinn hat.

Auch das gehostete Sprachmodell ist eine Entscheidung des Erstellers. Ein AIBOM, dessen Geltungsbereich „der Assistent als bereitgestellte Fähigkeit“ ist, behandelt den Modellendpunkt als Bestandteil: Das Modell ist Teil des Assistenten. Ein AIBOM, dessen Geltungsbereich „die Integration, die wir um ein externes Modell herum gebaut haben“ ist, behandelt ihn als extern. Das AIBOM formalisiert diese Unterscheidung über den deklarierten Geltungsbereich selbst, der die zum beschriebenen System gehörenden und die davon ausgeschlossenen Komponenten auflistet; ein einführender Leser benötigt nur den zugrunde liegenden Gedanken, nämlich dass die Zugehörigkeit eines Knotens eine auf den Geltungsbereich bezogene Entscheidung des Erstellers ist und keine Eigenschaft des Knotens selbst.

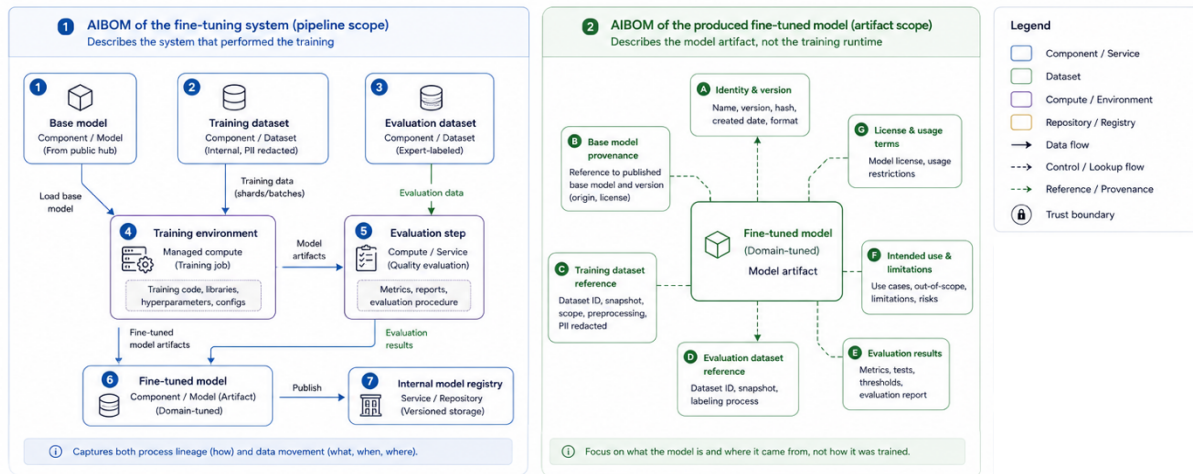
Ausgearbeitetes Beispiel: eine Fine-Tuning-Pipeline

Betrachten wir nun ein anderes System: ein Team, das ein Open-Weight-Basismodell mit internen Daten feinabstimmt, um ein auf die Domäne abgestimmtes Modell für den Einsatz im obigen Assistenten zu erzeugen.

Das Team beginnt mit einem veröffentlichten Basismodell, das aus einem öffentlichen Modell-Hub bezogen wird. Das Team verfügt über einen Trainingsdatensatz, der aus internen Kundensupport-Transkripten zusammengestellt wurde, wobei personenbezogene Informationen geschwärzt sind. Das Team verfügt über einen von Fachexperten von Hand gelabelten Evaluierungsdatensatz, mit dem die Modellqualität gemessen wird, bevor das feinabgestimmte Modell nachgelagert freigegeben wird. Das Team führt den Trainingsjob in einer verwalteten Rechenumgebung aus, erzeugt ein feinabgestimmtes Modell, evaluiert es und veröffentlicht es in einer internen Modell-Registry.

Worked example: a fine-tuning pipeline

Fine-tuning an open-weight base model on internal data to produce a domain-tuned model for use in a customer support assistant.



Dieses System erzeugt zwei unterscheidbare AIBOMs, und die Unterscheidung ist wichtig.

Das erste ist ein AIBOM des Fine-Tuning-Systems selbst: der Pipeline, die das Training durchgeführt hat. Sein Graph umfasst die Basismodellkomponente, die Trainingsdatensatzkomponente, die Evaluierungsdatensatzkomponente, die Trainingsumgebung, den Evaluierungsschritt und das erzeugte feinabgestimmte Modell. Die Datenflüsse beschreiben, wie die Trainingsdaten zum Trainingsschritt gelangten, wie das Basismodell geladen wurde, wie die Evaluierungsdaten zum Evaluierungsschritt gelangten und wie das erzeugte Modell in der Registry veröffentlicht wurde.

Das zweite ist ein AIBOM des erzeugten feinabgestimmten Modells selbst, behandelt als Artefakt. Sein Geltungsbereich ist das Modell, nicht die Pipeline. Es enthält die Identität des Modells, seine Provenienz zurück bis zum Basismodell, Verweise auf die Trainings- und Evaluierungsdatensätze, Evaluierungsergebnisse, Verwendungszweck und Lizenz. Es beschreibt nicht die Fine-Tuning-Laufzeit: Zu dem Zeitpunkt, an dem ein nachgelagertes System dieses Modell lädt, ist die Fine-Tuning-Laufzeit nicht mehr vorhanden.

Beide AIBOMs verweisen auf dieselben Datensätze und dasselbe Basismodell. Sie sind keine Duplikate; sie beschreiben unterschiedliche Geltungsbereiche. Ein Nutzer, der eine Modellrisikoprüfung durchführt, liest das Artefakt-AIBOM. Ein Ersteller, der die Trainingsumgebung auditiert, liest das System-AIBOM. Eine Organisation, die durchgängige Provenienz nachweisen will, pflegt beide und verknüpft sie.

Das Fine-Tuning-Beispiel verdeutlicht auch den Unterschied zwischen Prozess-Lineage und Datenbewegung. Die Prozess-Lineage hält fest, wie ein Artefakt entstanden ist: welches Basismodell verwendet wurde, welcher Trainingscode ausgeführt wurde, welche Hyperparameter den Job gesteuert haben, welches Evaluierungsverfahren die

berichteten Kennzahlen erzeugt hat. Die Datenbewegung hält fest, welche Daten wann zwischen welchen Knoten wohin gelangt sind. Trainingspipelines benötigen beides. Die Provenienz des feinabgestimmten Modells zurück zum Basismodell ist Prozess-Lineage. Die Bewegung von Trainings-Shards aus einem Datenspeicher in die Trainingsrechenumgebung zur Laufzeit ist Datenbewegung. Das AIBOM enthält beides, jeweils an seinem Platz. Für einen einführenden Leser ist entscheidend, dass ein AIBOM sich nicht zwischen den beiden Sichten entscheiden muss und dass ein AIBOM eines Trainingssystems, das nur eine davon verwendet, das Geschehene unzureichend beschreibt.

Was das AIBOM nicht aussagt

Ein vollständiges AIBOM besteht aus strukturierten Fakten und Provenienz. Es ist kein Urteil.

Es ist keine Aussage, dass ein System einer bestimmten Vorschrift entspricht. Es ist keine Aussage, dass das Risiko des Systems akzeptabel ist. Es ist keine Aussage, dass die im System vorhandenen Kontrollen für das Risiko ausreichen. Es ist keine Aussage darüber, wer für einen Aspekt des Systems rechtlich verantwortlich ist. Es ist weder ein Vertrauenswert noch ein Assurance-Level noch eine Qualitätsbewertung.

Dies ist eine bewusste Entscheidung im AIBOM-Design, kein Versehen. Eine Aufsichtsbehörde, ein Auditor, ein internes Prüfungsgremium oder ein Einkaufsteam wendet eigene Richtlinien, Schwellenwerte und Risikorahmenwerke auf die Fakten in einem AIBOM an. Diese Urteile erfolgen nachgelagert. Hätte das AIBOM bereits einen Satz dieser Urteile kodiert, wäre das Artefakt nur innerhalb dieses Regimes nützlich. Indem es auf der Ebene von Fakten und Provenienz bleibt, dient das AIBOM als gemeinsames Fundament, auf dem viele Regime aufbauen können.

Dieses gemeinsame Fundament ist nicht optional. Das AIBOM ist das, was diese Urteile falsifizierbar macht. Ohne es ist eine Risikobewertung eine ungeprüfte Behauptung. Mit ihm lässt sich jede Aussage in der Bewertung auf einen Knoten, einen Fluss, eine Nachweisreferenz oder eine deklarierte Vollständigkeitsgrenze zurückführen.

Wie ein AIBOM niedergeschrieben wird – für Leser, die tiefer einsteigen wollen

Das konzeptionelle Material in diesem Kapitel ist das, was jemand benötigt, der nicht selbst implementiert. Für Ersteller, die tatsächlich ein AIBOM ausgeben werden, lohnt sich die Kenntnis eines weiteren Punktes.

Die in diesem Kapitel behandelten Aspekte lassen sich nativ in CycloneDX 2.0⁴ ausdrücken. Der Graph befindet sich in einem Blueprint: Komponenten sind die Knoten, Flüsse sind die gerichteten Datenbewegungen, und Zonen und Grenzen tragen den Vertrauenskontext. Die Selbstbeschreibungen werden auf die deklarierten Modelltypen

und den Geltungsbereich des Blueprints abgebildet, wobei die Vollständigkeitsaussage durch die Kompositionsdeklarationen des Dokuments getragen wird. Nachweise werden über Deklarationen angehängt, die jeden Knoten oder Fluss über seinen Bezeichner referenzieren. Es gibt kein separates AIBOM-Dateiformat, kein KI-spezifisches Erweiterungsschema und keine Kompatibilitätskodierung, zwischen der man wählen müsste: Ein AIBOM ist ein strukturiertes Dokument, das diese Standardkonstrukte verwendet, um ein KI-System zu beschreiben.

Die Bezeichner leisten mehr als interne Buchführung. Jede Komponente, jeder Fluss, jede Zone, jede Grenze und jede Aussage im Dokument trägt eine stabile Referenz, und der Standard definiert seine Bedrohungs-, Risiko- und Kontrollmodelle als Geschwister des Blueprints, die in denselben Graphen verweisen: Eine Bedrohung verweist auf die Grenze, die ihr Angriffspfad überschreitet, eine Kontrolle auf den Fluss, den sie schützt, eine Schwachstellenanalyse auf die Komponenten, die sie betrifft. Keines davon beschreibt das System erneut. Das ist der strukturelle Grund, warum das AIBOM frei von Urteilen bleiben und dennoch das Fundament sein kann, auf das jedes Urteil zurückführbar ist – entfernt man das AIBOM, haben die nachgelagerten Analysen nichts mehr, worauf sie verweisen können.

Ein Ersteller kann dennoch wählen, wie umfangreich er das Dokument befüllt. Ein minimales AIBOM – Komponenten, Flüsse und die drei Selbstbeschreibungen – ist vollständig konform und kann eine vollständige Behauptung des Erstellers sein. Ein umfangreich belegtes AIBOM ergänzt Deklarationen, Kompositionen und Formulierungsdatensätze für maschinenlesbare Nachweise und Prozess-Lineage. Umfang ist kein Assurance-Level. Ein vollständiges und gut belegtes minimales AIBOM ist nützlicher als ein unvollständiges umfangreiches. Vertrauen in ein AIBOM entsteht durch Vollständigkeit, Provenienz, Qualität der Nachweise und die Erfolgsbilanz des Erstellers, nicht dadurch, welche Konstrukte es verwendet.

⁴ <https://cyclonedx.org/news/cyclonedx-v2.0-coming-soon/>

6. AIBOM für agentische KI-Systeme

Agentische KI-Systeme bringen eine neue Komplexitätsebene in KI-Lieferketten. Anders als herkömmliche KI-Systeme, die als relativ statische Pipelines arbeiten, sind agentische Systeme dynamisch, zusammensetzbar und oft autonom. Sie interagieren mit externen Tools, rufen andere Agenten auf, passen ihr Verhalten zur Laufzeit an und stützen sich auf sich entwickelnde Ökosysteme von Diensten und Protokollen.

Ein herkömmlicher AIBOM-Ansatz konzentriert sich auf die statische Dokumentation von Modellen, Datensätzen und Abhängigkeiten. Ein AIBOM für agentische Systeme muss dagegen dynamische Abhängigkeiten, Laufzeitinteraktionen und Orchestrierungslogik berücksichtigen, die zusammen bestimmen, wie sich diese Systeme im Produktivbetrieb verhalten. Dieser Abschnitt skizziert, wie sich AIBOM-Praktiken auf agentische Architekturen erweitern lassen, mit Schwerpunkt auf Beziehungen zwischen mehreren Agenten, Tool-Ökosystemen und externer Konnektivität.

Abhängigkeiten zwischen mehreren Agenten

Agentische Systeme bestehen häufig aus mehreren kooperierenden Agenten, von denen jeder unterschiedliche Rollen wie Planung, Ausführung, Retrieval, Validierung oder Orchestrierung haben kann. Diese Agenten können:

- auf unterschiedlichen Modellen aufbauen (z. B. GPT-basierter Planer + Retrieval-Agent + Tool-Executor).
- unterschiedlichen Teams oder Anbietern gehören.
- über strukturierte Protokolle oder informellen Nachrichtenaustausch kommunizieren.

In diesem Kontext muss ein AIBOM ausdrücklich Folgendes erfassen:

- Identitäten und Rollen der Agenten (Planer, Ausführer, Kritiker, Retriever, Governor, Router usw.).
- Zugrunde liegende Modellabhängigkeiten jedes Agenten (Basismodell, Fine-Tunes, Adapter).
- Interaktionsmuster (hierarchisch, Peer-to-Peer, ereignisgesteuert, Hub-and-Spoke).
- Orchestrierung des Kontrollflusses (wer wen aufruft, unter welchen Bedingungen, mit welchen Eskalationspfaden).

Dies ist entscheidend, weil das Risiko nicht mehr auf ein einzelnes Modell oder einen einzelnen Dienst beschränkt ist. Eine Schwachstelle oder ein Ausfall in einem Agenten kann sich über das gesamte System ausbreiten. Zum Beispiel:

- Ein kompromittierter Retrieval-Agent kann nachgelagerte Schlussfolgerungen und Empfehlungen vergiften.
- Ein halluzinierender Planer kann eine unsichere Tool-Nutzung oder Hochrisiko-Workflows auslösen.

- Ein falsch konfigurierter Guardrail-Agent kann dabei versagen, schädliche oder nicht konforme Ausgaben zu blockieren.

Dementsprechend sollte ein AIBOM agentische Systeme als Graphen voneinander abhängiger Komponenten darstellen statt als lineare Pipelines. Die BOM wird zu einem Graphen aus Agenten, Modellen und Orchestrierungskanten, der Auswirkungsanalysen ermöglicht, wenn sich ein Knoten oder eine Kante ändert.

Zugehörige Tools

Agentische KI-Systeme beziehen einen Großteil ihrer Leistungsfähigkeit aus der Nutzung von Tools. Zu den Tools können gehören:

- APIs (Suche, Wetter, Finanzen, Personalwesen, Ticketing usw.).
- Codeausführungsumgebungen und Funktions-Runner.
- Datenbanken und Vektorspeicher. Speicher und Zustand
- Interne Unternehmensdienste und Microservices.
- Dateisysteme oder Dokumenten-Repositories.

Diese Tools sind nicht bloß Integrationen; sie sind aktive Erweiterungen der Fähigkeiten des Agenten und werden oft autonom und mit hoher Frequenz aufgerufen.

Ein AIBOM für agentische Systeme muss mindestens Folgendes dokumentieren:

- Identität und Funktion des Tools (was das Tool tut, welche Domäne es berührt).
- Aufrufmechanismen (Function Calling, HTTP-APIs, Plugins, Framework-native Tools).
- Zugriffsumfang und Berechtigungen (nur Lesen vs. Lesen/Schreiben vs. Admin; Daten- und Ressourcenumfang).
- Authentifizierungsmethoden und Umgang mit Geheimnissen (API-Schlüssel, OAuth, Workload Identity, Secret Stores).
- Ratenbegrenzungen, Kontingente und betriebliche Einschränkungen (pro Agent und global).
- Version oder Hash der Tool-Definition (das genaue Schema und die Beschreibung, für deren Nutzung der Agent freigegeben wurde).

Aus Sicht von Sicherheit und Governance stellen Tools eine erhebliche Erweiterung der Angriffsfläche dar:

- Prompt-Injection-Angriffe können die Tool-Nutzung manipulieren und schädliche Aktionen auslösen.
- Überberechtigte Tools können sensible Daten offenlegen oder laterale Bewegungen ermöglichen.
- Externe APIs können Risiken für Datenabfluss, Integrität oder Verfügbarkeit mit sich bringen.
- Die Definition und das Verhalten eines Tools können sich nach der Prüfung ändern, daher muss die Freigabe an eine bestimmte Definitionsversion gebunden sein statt an einen Tool-Namen.

Die Erfassung von Tool-Abhängigkeiten im AIBOM ermöglicht:

- Feingranulare Zugriffsaudits (welche Agenten welche Tools mit welchen Berechtigungen aufrufen können).
- Risikobewertung auf Tool-Ebene (Kritikalität, Auswirkungsradius und kompensierende Kontrollen).
- Schnellere Incident Response, wenn ein Tool oder Konnektor kompromittiert ist (wer es nutzt, wo und wie).

Konnektoren von Drittanbietern

Moderne agentische Systeme stützen sich häufig auf Konnektoren von Drittanbietern, um externe Plattformen zu integrieren, etwa:

- SaaS-Anwendungen (z. B. Kollaborationsplattformen, CRM, Produktivitätssuiten und ITSM-Tools wie Slack, Salesforce, Google Workspace und ServiceNow).
- Datenanbieter (Finanzdaten-Feeds, juristische Datenbanken, Gesundheits-APIs).
- Cloud-Dienste und KI-Plattformen (Speicher, Analytik, Modell-APIs).

Diese Konnektoren führen Lieferkettenabhängigkeiten ein, die über herkömmliche Softwarebibliotheken und Modelle hinausgehen, darunter:

- Vom Anbieter kontrollierte APIs und SLAs.
- Undurchsichtige Datenverarbeitungspipelines und Transformationen.
- Externe Identitäts- und Zugriffsmanagementsysteme und Policy-Engines.

Ein AIBOM sollte daher Folgendes erfassen:

Aus Sicht von Sicherheit und Governance bergen Konnektoren Risiken, die weitgehend außerhalb der direkten Kontrolle der Organisation liegen:

- Bei der Einrichtung eines Konnektors gewährte OAuth-Scopes können weit umfassender sein, als der Anwendungsfall erfordert, und Agenten dauerhaften Zugriff auf ganze Workspaces oder Datensatzbestände verschaffen.
- Anbieter können API-Verhalten, Datenverarbeitung oder Unterauftragsverarbeiter ändern, ohne nachgelagerte AIBOM-Inhaber zu benachrichtigen.
- Daten können über nie offengelegte Unterauftragsverarbeiter Residenz- oder Souveränitätsgrenzen überschreiten.
- Angaben zu Anbieter und Dienst des Konnektors (Identität des Anbieters, Produkt, Region, Bereitstellungsmodell).
- Ausgetauschte Daten (Klassifizierung von Eingaben/Ausgaben, Datenkategorien, Sensibilitätsstufen).
- Authentifizierungs- und Autorisierungsmodelle (OAuth-Flows, API-Schlüssel, Dienstkonten, delegierter Zugriff).
- Überlegungen zu Datenresidenz, Souveränität und Compliance.
- Vertragliche, lizenzrechtliche und nutzungsbezogene Einschränkungen, die den Betrieb oder die Datenverarbeitung beeinflussen können.

Eine zentrale Herausforderung besteht darin, dass der Einblick in Drittanbietersysteme oft begrenzt ist. Organisationen sollten:

- wo möglich AIBOM-ähnliche Offenlegungen oder Sicherheitsattestierungen von Anbietern verlangen.
- angenommene Vertrauensgrenzen und Restrisiken ausdrücklich dokumentieren.
- bekannte Transparenzlücken nachverfolgen, einschließlich unbekannter Unterauftragsverarbeiter oder undurchsichtiger Verarbeitungsschritte.

So wird sichergestellt, dass Risiken auch dann ausdrücklich anerkannt, erfasst und innerhalb des AIBOM-Rahmens gesteuert werden, wenn vollständige Provenienz oder interne Details nicht verfügbar sind.

Nachverfolgung des MCP/A2A-Ökosystems

Neu entstehende Protokolle wie das Model Context Protocol (MCP)⁵ und Agent-to-Agent-Kommunikationsframeworks (A2A)⁶ prägen rasant, wie agentische Systeme zusammenarbeiten. Diese Ökosysteme ermöglichen:

- Standardisierte Erkennung und Aufrufe von Tools über Plattformen hinweg.
- Kommunikation zwischen Agenten über Organisations- oder Mandantengrenzen hinweg.
- Dynamische Zusammenstellung von Fähigkeiten zur Laufzeit ohne Code-Deployment.

So leistungsfähig sie sind, bringen sie doch auch hochdynamische und potenziell kurzlebige Abhängigkeiten mit sich, die sich mit herkömmlichen BOM-Methoden nur schwer nachverfolgen lassen.

Um relevant zu bleiben, muss sich AIBOM weiterentwickeln und Folgendes einschließen:

- Protokollnutzung (z. B. MCP, A2A, andere Interoperabilitätsstandards für Agenten).
- Registrierte Endpunkte und Dienstanbieter, die an diesen Ökosystemen teilnehmen.
- Mechanismen zur dynamischen Erkennung von Fähigkeiten (Registries, Kataloge, Marktplätze).
- Abhängigkeiten auf Sitzungs- oder Kontextebene, nicht nur statische Konfiguration.
- MCP: Serveridentität, Transport, Versions-Pinning, Hashes der Tool-Definitionen und gewährte Scopes.
- A2A: Agent Cards, die Identität, Fähigkeiten und Authentifizierungsanforderungen eines Agenten bekannt geben und direkt als AIBOM-Input dienen.

Zum Beispiel:

- Ein Agent kann über MCP zur Laufzeit ein Tool erkennen und aufrufen, ohne vorherige statische Konfiguration.
- Zwei Agenten verschiedener Anbieter können über A2A-Kanäle zusammenarbeiten und Aufgaben und Ergebnisse austauschen.

⁵ MCP: <https://modelcontextprotocol.io/specification/2026-07-28>

⁶ A2A: <https://a2a-protocol.org/latest/>

- Verfügbare Fähigkeiten und Richtlinien können sich ohne erneutes Deployment der Anwendung ändern.

Um dem zu begegnen, sollten AIBOM-Implementierungen:

- Laufzeitlemetrie, Protokolle und Traces als Quellen der Wahrheit für tatsächliche Abhängigkeiten einbeziehen.
- Fähigkeits-Registries, Discovery-Endpunkte und Marktplätze als vollwertige Komponenten nachverfolgen.
- zeitgebundene Aufzeichnungen von Interaktionen und Abhängigkeiten führen (wer wann mit wem über welches Protokoll kommuniziert hat).

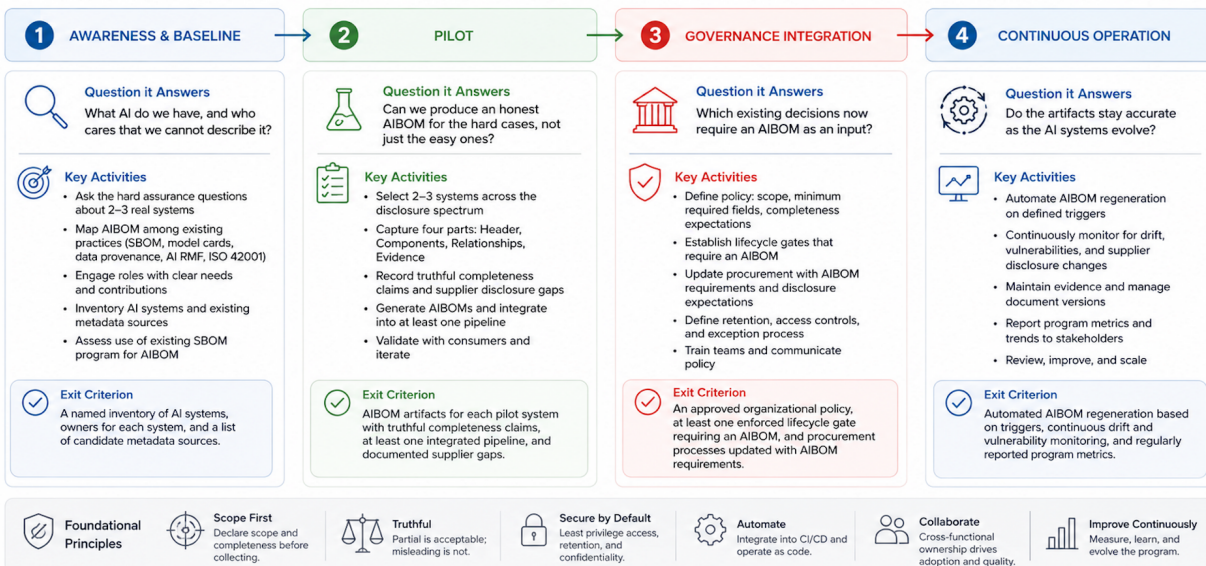
Damit wandelt sich AIBOM von einem rein statischen Artefakt zu einer lebendigen Darstellung des Systemverhaltens über die Zeit, die Konfiguration zur Entwurfszeit mit Beobachtungen zur Laufzeit verbindet.

7. Roadmap zur Einführung im Unternehmen

Die Erstellung eines einzelnen AIBOM ist eine technische Aufgabe, und der zuvor beschriebene Artefaktlebenszyklus – planen, erstellen, validieren, veröffentlichen, betreiben, auditieren und außer Betrieb nehmen – ist die Form, die diese Aufgabe annimmt. Dies für jedes KI-System in einem Unternehmen wiederholbar zu tun, sodass die Ergebnisse tatsächlich in Entscheidungen einfließen, ist ein organisatorisches Veränderungsprogramm. Dieses Kapitel beschreibt den Weg vom ersten Gespräch bis zum eingeschwungenen Betrieb.

Enterprise Adoption Roadmap

From first conversation to steady-state AIBOM operation



Source: OWASP AIBOM Foundations Guide v1.0

Die vier nachfolgenden Phasen sind nach Abhängigkeiten geordnet. Jede beantwortet eine Frage und endet mit einer Bedingung, die Sie überprüfen können.

Phase	Beantwortete Frage	Abschlusskriterium
Bewusstsein und Ausgangsbasis	Welche KI haben wir, und wen kümmert es, dass wir sie nicht beschreiben können?	Ein benanntes Inventar der KI-Systeme, ein identifizierter Verantwortlicher für jedes System und eine Liste möglicher Metadatenquellen.
Pilotprojekt	Können wir ein ehrliches AIBOM für die schwierigen Fälle erstellen, nicht nur für die einfachen?	Für jedes Pilotsystem erzeugte AIBOM-Artefakte mit wahrheitsgemäßen Vollständigkeitsaussagen, mindestens eine integrierte Pipeline und dokumentierte Lücken in der Offenlegung durch Lieferanten.
Governance-Integration	Welche bestehenden Entscheidungen erfordern nun ein AIBOM als Input?	Eine genehmigte Unternehmensrichtlinie, mindestens ein durchgesetztes Lebenszyklus-Gate, das ein AIBOM verlangt, und um AIBOM-Anforderungen ergänzte Beschaffungsprozesse.
Kontinuierlicher Betrieb	Bleiben die Artefakte korrekt, während sich die KI-Systeme weiterentwickeln?	Automatisierte AIBOM-Neuerzeugung auf Basis definierter Auslöser, kontinuierliche Überwachung von Drift und Schwachstellen sowie regelmäßig berichtete ProgrammKennzahlen.

Phase 1 — Bewusstsein und Ausgangsbasis

Die Begründung für den Einstieg muss hier nicht neu aufgebaut werden: Die weiter vorn in diesem Dokument dargelegten Sicherheits-, Regulierungs- und Governance-Treiber sind das Argument. Was diese Phase hinzufügt, ist Konkretheit. Die Assurance-Fragen, die eine Organisation derzeit nicht beantworten kann – welche Modellversion in Produktion ist, womit sie trainiert oder feinabgestimmt wurde, wer der vorgelagerte Lieferant ist und was dessen Lizenz erlaubt –, sollten für zwei oder drei reale Systeme laut gestellt werden. Die Unfähigkeit, sie zu beantworten, ist das wirksamste Argument, um Unterstützung zu gewinnen, und es überzeugt mehr als jedes Zitat aus einem Rahmenwerk.

Bewusstsein bedeutet auch, AIBOM richtig unter den Praktiken einzuordnen, die die Organisation möglicherweise bereits betreibt. Eine SBOM inventarisiert Softwarekomponenten und bleibt notwendig; ein AIBOM überträgt dieselbe Disziplin auf Modelle, Datensätze, Prompts und Guardrails sowie die Dienste im Anfragepfad. Eine Model Card dokumentiert Verwendungszweck und Leistung eines Modells; ein AIBOM enthält diesen Inhalt als Nachweis, der einer Komponente zugeordnet ist, innerhalb eines versionierten Dokuments mit deklariertem Geltungsbereich. Arbeiten zur Datenprovenienz liefern die Datensatz-Lineage, die ein AIBOM festhält. Etablierte

Rahmenwerke für KI-Risikomanagement und Managementsysteme fordern eine strukturierte Dokumentation der Systemmerkmale, ohne deren Form vorzugeben; das AIBOM ist eine Form, die diese Forderung erfüllt. AIBOM ersetzt keine dieser Praktiken und dupliziert keine von ihnen.

Die Einbindung sollte von Anfang an rollenspezifisch erfolgen, weil mangelnde Abstimmung an dieser Stelle zu fragmentierten Pilotprojekten führt, die nie die Governance erreichen.

Funktion	Was sie vom AIBOM benötigt	Was sie beiträgt
Sicherheitsarchitektur und Red Team	Informationen zur KI-spezifischen Angriffsfläche, den Komponenten zugeordnet.	Umfang der Bedrohungsmodellierung und Definition der Sicherheitsfragen, die das AIBOM beantworten muss.
Rechtsabteilung und IP-Beratung	Lizenzbedingungen für Modelle und Datensätze, Nutzungsrechte und geltende Einschränkungen.	Auslegung rechtlicher Verpflichtungen, Lizenzanforderungen und Einhaltung von Rechten des geistigen Eigentums.
Compliance und Audit	Nachweise zur Unterstützung von KI-Transparenz, Rückverfolgbarkeit und regulatorischen Verpflichtungen.	Aufbewahrungsrichtlinien, Governance-Kontrollen, Auditanforderungen und Erwartungen an Nachweise.
Plattform-Engineering und MLOps	Automatisierungsanforderungen, Mechanismen zur Metadatenerfassung und Integrationspunkte in Pipelines.	Implementierung der automatisierten AIBOM-Erzeugung, CI/CD-Integration und laufende Wartung.
KI- und ML-Teams	Anerkennung, dass die Erfassung der Lineage ein integraler Bestandteil der KI-Entwicklung ist.	Lineage der Trainingsdaten, Provenienz des Basismodells, Fine-Tuning-Historie, Evaluierungsergebnisse und Metadaten der Modellentwicklung.

Die Bestandsaufnahme der Ausgangsbasis schließt die Phase ab: Inventarisieren Sie jedes KI-System in Produktion oder aktiver Entwicklung – intern trainierte Modelle, feinabgestimmte Varianten, extern beschaffte Modell-APIs und KI-Funktionen in beschaffter SaaS; identifizieren Sie die Metadaten, die bereits in Model Cards, Protokollen von Trainingsläufen, Datensatz-Registries und Abhängigkeitsmanifesten vorhanden sind; und prüfen Sie, ob ein bestehendes SBOM-Programm die AIBOM-Arbeit mittragen kann, statt daneben ein neues aufzubauen.

Phase 2 — Pilotprojekt

Wählen Sie zwei oder drei Systeme, die an unterschiedlichen Punkten des Offenlegungsspektrums liegen, denn wie schwierig die Erstellung eines AIBOM ist, hängt fast ausschließlich davon ab, wie viel der Ersteller wissen darf. Ein intern trainiertes und bereitgestelltes Modell bietet vollen Zugriff auf Metadaten und ist der handhabbare Ausgangspunkt. Eine intern feinabgestimmte Variante eines externen Basismodells zwingt das Pilotprojekt, zwei Provenienzketten festzuhalten, von denen eine jemand anderem gehört, und ist der häufigste Fall in Unternehmen. Ein gehostetes Modell, auf das per API zugegriffen wird, testet, ob sich die Organisation auf die Offenlegung durch den Lieferanten verlassen kann, und zeigt genau, wo diese Offenlegung endet. Eine KI-Funktion in einer beschafften SaaS-Anwendung legt die Untergrenze der erreichbaren Abdeckung fest und liefert damit den nützlichsten Input für den Einkauf aus dem gesamten Pilotprojekt.

Bei der Erfassung wird das Pilotprojekt am häufigsten falsch angegangen, weil Teams Felder sammeln, ohne zu entscheiden, was das Dokument aussagt. Strukturieren Sie die Arbeit anhand der vier zuvor beschriebenen Teile:

Der Header. Deklarieren Sie den Geltungsbereich, bevor Sie irgendetwas sammeln: welches System, welche Version, welche Bereitstellung, welche Schnittstellen ein- und ausgeschlossen sind – und halten Sie dann eine Vollständigkeitsaussage fest, die ehrlich aus dem abgeleitet ist, was tatsächlich erreichbar war. Eine Teilaussage für eine Blackbox-SaaS-Funktion ist ein korrektes Artefakt. Eine Vollständigkeitsaussage für dasselbe System ist ein falsches.

Die Komponenten. Modellidentität, Version, Architektur und Verweis auf das Basismodell; Datensätze mit Version, Quelle, Erhebungsmethode, Lizenz und bekannten Einschränkungen; die Tools, Orchestratoren und Konnektoren, die das System aufruft; Prompts und Guardrail-Richtlinien; die Laufzeit und ihre Abhängigkeiten, mit Querverweisen auf bestehende SBOM-Datensätze statt einer erneuten Inventarisierung.

Die Flüsse. Welche Daten sich zwischen diesen Komponenten bewegen und über welche Grenzen hinweg – der Teil, den keine Feld-Checkliste liefert, und der Teil, den Incident-Responder zuerst nutzen.

Die Nachweise. Hashes für Gewichtsdateien und Datensatz-Snapshots, Signaturen, Model Cards, Evaluierungsergebnisse sowie der Nachweis über verantwortliche Urheberschaft, Prüfung und Freigabe, der das Dokument prüfbar macht, statt es lediglich vorhanden sein zu lassen.

Bei der Automatisierung lautet die Frage des Pilotprojekts nicht, welches Format verwendet werden soll – das ist an anderer Stelle in diesem Dokument geklärt –, sondern woher jedes Feld stammt. Extrahieren Sie aus Build- und Trainingspipelines, wo immer die Informationen dort bereits vorliegen, binden Sie das Experiment-Tracking an, das die Organisation betreibt, und legen Sie ausdrücklich fest, welche Felder von Hand verfasst werden müssen und wer dafür verantwortlich ist. Widerstehen Sie der

Versuchung, das Programm allein auf Grundlage der Pilotergebnisse auf eine Plattform festzulegen.

Die Erstellung liegt bei dem Team, das dem jeweiligen System am nächsten ist, mit Prüfung und Freigabe durch eine benannte Governance- oder Sicherheitsfunktion. Die Erstellung in einem Team ohne direkte Systemkenntnis zu zentralisieren, ist genau der Fehler, den diese Reihenfolge verhindern soll.

Phase 3 — Governance-Integration

Integration bedeutet, dass das AIBOM zu einem verpflichtenden Input für Entscheidungen wird, die die Organisation bereits trifft. Die Schaffung eines parallelen AIBOM-Prozesses ist der Grund, warum die meisten Programme nach dem Pilotprojekt ins Stocken geraten.

Bestehender Prozess	Was das AIBOM liefert	Wo es verbindlich wird
Modellrisikomanagement	Ein umfassendes Inventar der KI-Komponenten, Abhängigkeiten und unterstützenden Metadaten, die für die Risikobewertung erforderlich sind.	Verpflichtender Input für jede Modellrisikobewertung und -prüfung.
Lebenszyklus der KI-Entwicklung	Deklariertes Systemumfeld, Provenienz, Lineage und Evaluationsnachweise, erfasst in jeder Lebenszyklusphase.	Verpflichtendes Gate für die Freigabe vor dem Training, die Prüfung vor der Bereitstellung und die Bewertung nach einem Vorfall.
Beschaffung und Drittparteirisiko	Vom Lieferanten deklarierte Provenienzinformationen, Software- und Modellabhängigkeiten sowie die ausdrückliche Kennzeichnung fehlender oder unvollständiger Offenlegungen.	Mindestanforderung an die Offenlegung durch Lieferanten bei der Bewertung von Hochrisiko-Anbietern und -KI-Systemen.
Sicherheits-Designprüfung	KI-Komponenten, Datenflüsse, Abhängigkeiten und Vertrauensgrenzen, die für Bedrohungsmodellierung und Architekturprüfung benötigt werden.	Standard-Input für Architektur- und Designprüfungen der KI-Sicherheit.
Regulatorische Bereitschaft und Audit	Versionskontrollierte, rückverfolgbare Nachweise, zugeordnet zu geltenden regulatorischen, Governance- und Compliance-Verpflichtungen.	Aufbewahrtes Governance-Artefakt, das bei behördlichen Prüfungen, Audits und Compliance-Bewertungen vorgelegt wird.

Eine formale Richtlinie macht dies durchsetzbar. Sie sollte das Minimum Usable AIBOM je Risikostufe festlegen, Verantwortlichkeiten für Erstellung und Prüfung, Aufbewahrungsfristen, Aktualisierungsauslöser sowie den Eskalationspfad, wenn ein Lieferant keine Angaben macht. Das Artefakt selbst muss als kontrolliertes Dokument behandelt werden: versionskontrolliert, signiert, mit erfasster Identität des Prüfers, Prüfdatum und Prüfungsumfang, und denselben Aufbewahrungs- und Zugriffskontrollen unterworfen wie Modellvalidierungsunterlagen.

Phase 4 — Kontinuierlicher Betrieb

Ein AIBOM beschreibt ein System zu einem bestimmten Zeitpunkt. Die Systeme ändern sich fortlaufend, und ein Artefakt, das nicht mehr mit der Bereitstellung übereinstimmt, ist schlimmer als keines, weil es Vertrauen weckt, das es nicht stützen kann.

Eine Neuerzeugung sollte ausgelöst werden durch: erneutes Training oder Aktualisierung von Datensätzen; Austausch von Adaptern oder Änderung von Fine-Tuning-Parametern; Änderungen an Datensatzversionen, auch inkrementelle; Änderungen an vorgelagerten Modellen oder Aliasen durch einen Lieferanten; Upgrades von Abhängigkeiten in der Inferenzlaufzeit oder im Vorverarbeitungspfad; Änderungen an Guardrails und Policy-Paketen; offengelegte Schwachstellen, die eine nachverfolgte Komponente betreffen; sowie jede Änderung des Verwendungszwecks, des Bereitstellungsumfangs oder der Zugriffsgrenzen. Änderungen auf Lieferantenseite verdienen besondere Aufmerksamkeit, weil sie die Klasse von Auslösern sind, die ohne jedes lokale Release eintreten.

Operativ bedeutet dies: AIBOM-Erzeugung eingebettet in Trainingspipelines, sodass Artefakte bei erfolgreichem Build erzeugt werden; Validierungsprüfungen in Deployment-Gates, sodass unvollständige oder veraltete Datensätze die Freigabe blockieren; Komponentendatensätze, die in dieselben Workflows zur Schwachstellenkorrelation eingespeist werden, die das SBOM-Programm bereits nutzt, statt in ein zweites Silo; und Integritätsprüfung, sodass nachträgliche Änderungen nach der Erzeugung erkennbar sind. Die Verantwortlichkeiten folgen den bestehenden Linien: ML-Engineering- und Plattformteams aktualisieren, der Sicherheitsbetrieb bewertet Warnmeldungen, die Governance-Funktion berichtet, und das Drittparteirisikomanagement verfolgt die Offenlegung durch Lieferanten.

Es lohnt sich, die Gesundheit des Programms anhand weniger Indikatoren zu messen: Abdeckung, der Anteil der produktiven KI-Systeme mit einem aktuellen Artefakt; Aktualität, die Verzögerung zwischen einem Auslöseereignis und der entsprechenden Aktualisierung; Feldbefüllung gemessen am Minimum Usable AIBOM, aufgeschlüsselt nach Risikostufe; Reaktion auf Schwachstellen, die Zeit von der Offenlegung bis zur bewerteten Auswirkung; und Lieferanten-Compliance, der Anteil der vertraglich gebundenen KI-Lieferanten, die den Mindeststandard für Offenlegung erfüllen. Diese Indikatoren messen das Programm, nicht die Systeme, die es dokumentiert. Ein vollständig befüllter Datensatz eines schlecht gesteuerten Systems ist immer noch nur

ein vollständig befüllter Datensatz, und keine Kennzahl hier ist eine Bewertung des KI-Systems oder des AIBOM. Stellen Sie sie dort dar, wo bereits über KI-Risiken berichtet wird, damit dieselben Nachweise der regulatorischen Bereitschaft, der internen Revision und der Berichterstattung an die Geschäftsleitung dienen, ohne dreimal erstellt werden zu müssen.

DANKSAGUNG

Mitwirkende

Yuvaraj Govindarajulu (Workstream-Leitung,
Projekt-Co-Leitung)

Aruneeesh Salhotra (Projektleitung)

Martin Espinoza

Van Lindberg

Tanmay Ambegaokar

Bakul Singhal

L.Frances. Domingo

Anmol Kumar

Binal Patel

Ranjan Goel

Sapna Paul

Behnaz Karimi

John Cavanaugh

Yash Omer

Pratibha Aphale

Deba Mishra

Karen Bennet

Chetan Velicheti

Protectt.ai, Indien

Gründer/CEO, Stealth Startup, USA

Fabletics

Model Monster, USA

Onix, USA

Steve Madden, USA

Stealth Startup, USA

SISA, Indien

NuGuard AI, USA

Dayforce, USA

OWASP, Deutschland

ProCap360, USA

Bajaj Financial Services, Indien

Goldman Sachs, USA

EY, USA

Projektsponsoren

Wir danken unseren Projektsponsoren, deren finanzielle Beiträge dazu beitragen, die Ziele des OWASP-AIBOM-Projekts voranzubringen und die Betriebskosten zu decken, zusätzlich zu den von der OWASP Foundation bereitgestellten Ressourcen.

Sponsoren erhalten Anerkennung für ihre Beiträge in unseren Materialien und auf unseren Webpräsenzen. Alle vom Projekt erstellten Materialien werden von der Community entwickelt und getragen und unter Open-Source- und Creative-Commons-Lizenzen veröffentlicht. Um mehr darüber zu erfahren, wie Sie Sponsor werden und das Projekt unterstützen können, besuchen Sie bitte den Bereich „Sponsorship“ unserer Website unter <https://owaspai bom.org/sponsorship/>.

Gold-Sponsoren des Projekts



Silber-Sponsoren des Projekts





OWASP AIBOM Grundlagenleitfaden

SEPTEMBER 2026

Version 1.0 (DE)