MULTIPLEWORKS

# Architecture Briefing

No. 01

## AI That Knows Your Business

*Companion to the Executive Briefing of the same title.*

*For the enterprise architect, the risk and compliance reader, and the regulator's analyst.*

MultipleWorks · Hong Kong · multipleworks.com.hk
Mark Goodchild · 2026

# About this paper

This Architecture Briefing is the reference architecture for *AI That Knows Your Business*, published alongside the Executive Briefing of the same title. The Executive Briefing argues the position; this document specifies the architecture. Both are MultipleWorks publications calibrated for different audiences within the same institutional conversation.

## Audience

The primary reader is the enterprise architect designing AI integration into the institution's existing security and information management architecture. The role's responsibilities have changed considerably over the past decade and the title now spans practitioners with varying authority, from genuinely senior architectural leads who set institutional direction to senior solution architects who specify within the boundaries of decisions made elsewhere. The document is calibrated to give both readers what they need: the senior lead can use it as a reference architecture to align institutional work; the senior solution architect can use it as the specification their work has to fit into.

The secondary readers are the institution's senior risk, compliance, audit, and information security functions. These readers are interested in particular sections of the document rather than in the whole: the regulatory mesh as architectural constraint (section 13), the audit and observability architecture (section 7), the vendor and third-party risk treatment (section 10), the information lifecycle management section (section 11). The document is structured so that these readers can engage with the relevant sections without having to read the document end-to-end.

The tertiary reader is the regulator's analyst, the lawyer or auditor reading the document as part of supervisory review or as part of an institution's regulatory submission. The document is designed to survive scrutiny from this reader: claims are sourced, citations are provided, the IP boundary is explicit, and the architecture-supports-but-does-not-constitute axiom is named throughout.

## Intent

This paper specifies the architecture an institution adopts to integrate AI into its existing security and information management infrastructure. The specification is sourced through verifiable citations and structured as a working reference rather than as a position paper. The work the institution does to apply the specification to its own environment is the substantive contribution this document does not exhaust; the architectural specification is the artefact that supports the work.

## Scope

This architecture is calibrated for agentic AI: AI systems acting on behalf of human principals across institutional infrastructure, retrieving from organisational data, invoking tools and downstream systems, and producing audit trails the institution operates on. The institutional AI estate this document addresses is the AI deployed in customer-facing systems, employee-facing applications, business workflows, and the agentic infrastructure the institution governs as a regulated AI workload.

The document does not address AI coding assistants used by software engineers within their development environments. Tools such as Cursor, GitHub Copilot, Claude Code, JetBrains AI, and equivalent integrated developer tools operate within the engineering function's own toolchain, against the engineer's own credentials, against source code in version-controlled repositories, with their own threat model and governance pattern. They are real and material to institutional AI policy, but they do not fit the three-identity-categories pattern in section 3, the four-layer data taxonomy in section 4, or the runtime layer specification in section 9. The institutional governance for coding AI runs alongside the architecture this document specifies rather than within it. The institution that conflates the two ends up applying agentic-AI controls to engineering tools or engineering-style controls to agentic AI; both produce poorly-fitting governance.

## The architectural specification's IP boundary

The document specifies the architectural pattern that Score-compliant runtimes implement. Score is an open governance protocol; the architectural pattern is published and contributable. Implementations of Score, including MultipleWorks Maestro, are runtimes the institution selects within the architectural pattern; the document does not describe the internal implementation of any specific runtime.

The boundary is structural rather than rhetorical. The architectural specification belongs to the institutions and architects who adopt it. The runtime implementations are commercial products subject to their own commercial terms. The two layers are architecturally distinct, and the document maintains the distinction throughout.

## Publication context



The suggested citation is:

> *MultipleWorks (2026). AI That Knows Your Business: Architecture Briefing. MultipleWorks Limited, Hong Kong. Available at multipleworks.com.hk/briefings.*

For commercial application of the architecture, the institution-specific work that the document deliberately does not exhaust, MultipleWorks supports the work through the engagements named in section 15.

## About the author

Mark Goodchild is Founder and Managing Director of MultipleWorks, a Hong Kong consultancy specialising in enterprise AI architecture and governance for regulated APAC enterprises. His background spans 25 years of enterprise architecture and digital transformation, including eleven years at EY, leaving as a Director in the APAC emerging tech consulting practice.

# Abstract

Enterprise agentic AI deployments operating across multiple regulatory regimes, multiple model vendors, and existing institutional infrastructure cannot rely on vendor-native governance to discharge institutional accountability. The architectural problem is integration: how AI fits the security and information management infrastructure regulators inspect, auditors examine, and executive sponsors approve.

This Architecture Briefing specifies the architecture an institution adopts to make the integration sound. The companion Executive Briefing of the same title argues the strategic position; this document specifies the architecture for enterprise architects, with secondary readers in risk, compliance, audit, and information security and tertiary readers among regulators' analysts.

The architecture covers fifteen surfaces calibrated together: identity and access, data architecture organised around a four-layer taxonomy of source data, embeddings, conversation history and inference logs, and model weights, network and deployment topology with the AI gateway elevated to a layer of its own, encryption and key management, audit and observability, behaviour governance specification operating above runtime formats and native model behavioural mechanisms, the runtime layer executing the specification, vendor and third-party risk, information lifecycle, business continuity, and the regulatory mesh as architectural constraint. The architectural posture throughout is that the architecture supports institutional accountability without constituting it.

Adoption produces auditability at regulatory fidelity, portability across runtimes and vendors, resilience against vendor failure, regulatory operation across the multi-jurisdiction mesh APAC enterprises actually face, and the compounding of AI investment as managed asset rather than one-time deposit. These are design-time properties rather than runtime mechanisms or vendor capabilities.

The document specifies the architectural pattern; the institution-specific application is the substantive work it does not exhaust. This is an industry analytical paper rather than peer-reviewed academic research; the apparatus is calibrated to the conventions of serious technical and policy publishing, with authority emerging from substantive work and verifiable sourcing rather than academic review.

**Keywords:** enterprise AI architecture, agentic AI, AI governance, regulatory mesh, behaviour governance specification, Score protocol, multi-vendor AI estates, APAC regulated enterprises, reference architecture.

# Table of contents

SECTION 01

# Introduction

The Executive Briefing argued that enterprise AI succeeds when it integrates with the institution's security and information management infrastructure rather than replacing parts of that infrastructure with vendor-native alternatives. Five decisions every institution makes about its AI architecture were named: where the data lives, single-vendor or multi-vendor, on-premises or cloud, governance designed in or bolted on, and how personal knowledge is handled. Made deliberately, these decisions compound; made by default, they cost more to undo than to make.

This document specifies the architecture that supports the deliberate making of those decisions for the institution's agentic AI estate: the AI acting on behalf of human principals, retrieving from organisational data, invoking tools, and producing audit trails the institution operates on. The institution the architecture is calibrated for is the regulated APAC enterprise: multi-vendor, multi-jurisdiction, hybrid deployment. The architectural pattern generalises beyond the APAC context, but the calibration deliberately serves the institution the broader vendor ecosystem underserves. AI coding assistants used by software engineers within their development environments are addressed in the front matter scope statement and are out of scope for the architecture; the institution governs coding AI alongside this architecture rather than within it.

Three named artefacts appear throughout the document and warrant introduction at the outset. **Score** is the institution's behaviour governance specification: an open authoring format calibrated specifically for the institutional governance of agentic AI, designed to compile to multiple downstream runtime formats (MCP, agentskills.io, vendor-native tool definitions). Score is one such open format and may not remain the only one; the architectural argument the document makes is that an open authoring layer at this altitude is structurally necessary, not that Score is the only candidate. Section 8 treats the behaviour governance layer in depth and the case for Score specifically. **Score-compliant runtimes** execute the specification. **Maestro** is MultipleWorks' Score-compliant runtime; other runtimes are or will be Score-compliant, and the architectural specification is concerned with the runtime category rather than with any specific implementation. The document specifies the architectural pattern; Score implements the authoring layer; Score-compliant runtimes execute it; Maestro is MultipleWorks' contribution to the runtime category.

The fifteen sections that follow specify the architecture in the order the architect designs it. Section 2 establishes the architectural foundations the rest of the document builds on. Sections 3 through 7 specify the security and information management spine: identity and access, data architecture, network and deployment topology, encryption and key management, audit and observability. Section 8 specifies the behaviour governance layer where the institution does the most direct work, with section 9 specifying the

runtime layer that executes the behaviour specification. Sections 10 through 12 address vendor risk, information lifecycle, and business continuity. Section 13 treats the regulatory mesh as architectural constraint. Section 14 returns to the Executive Briefing's five decisions with the architectural context the architect needs. Section 15 closes with the institutional work that follows the architecture's adoption.

SECTION 02

# Architectural foundations

## THE SYSTEM LANDSCAPE

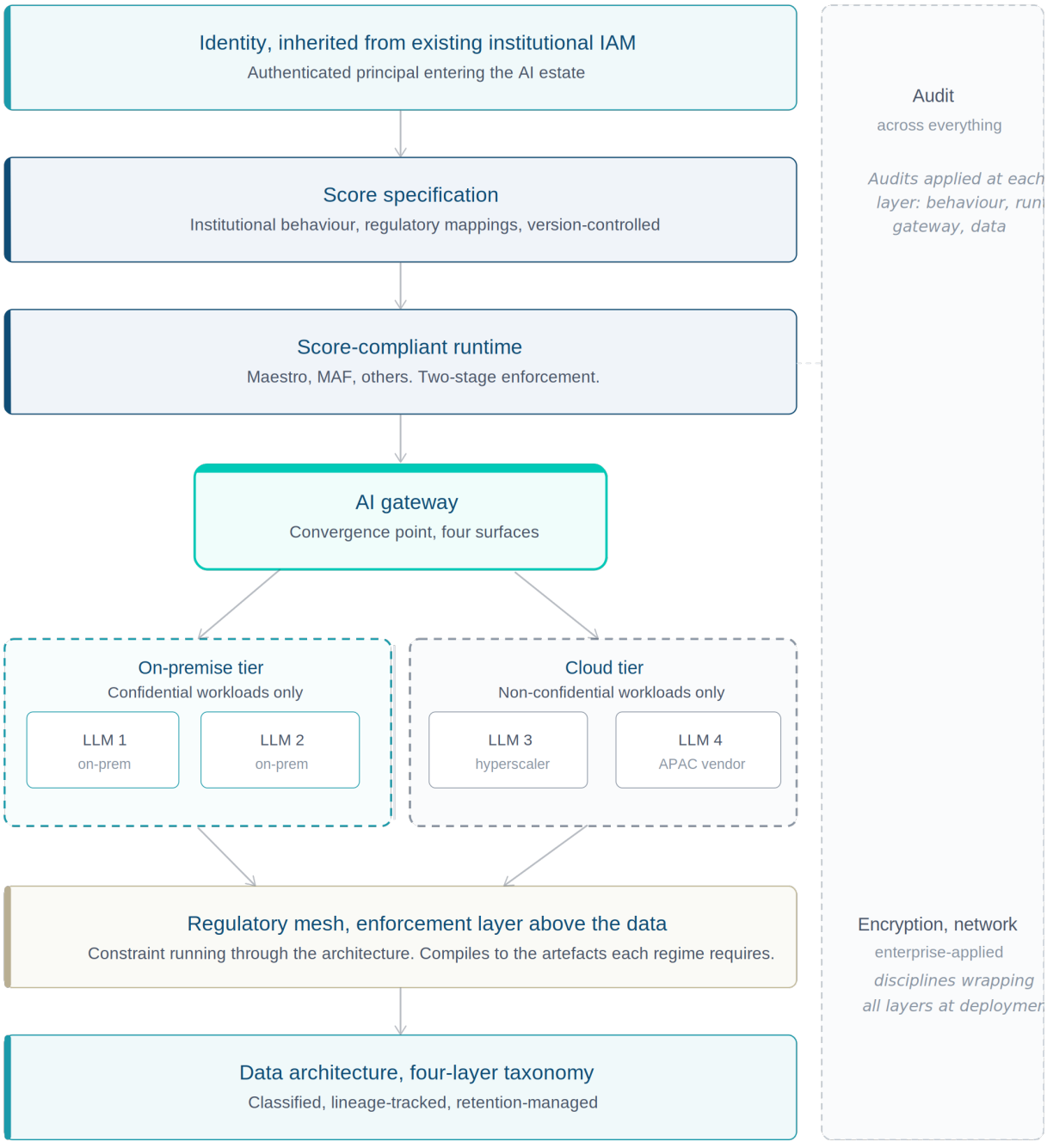



**Figure 1.** The system landscape, showing how the architectural surfaces fit together with the institutional spine running full-height on the right.

The Executive Briefing established five propositions about enterprise AI: knowledge has four kinds, the architecture has three layers, trust has four properties, the institution makes five decisions, and the implementation runs across three windows. This Architecture Briefing operates within these propositions and reframes them as architectural primitives the EA designs against.

## The four kinds of knowledge

The Executive Briefing's taxonomy of vendor, organisational, individual, and session knowledge is the business-and-governance perspective on what the AI knows. Vendor knowledge is what the foundation model brings, trained by the vendor and outside the institution's direct control. Organisational knowledge is what the institution knows and decides, the documents, records, structured information, and institution-specific fine-tuning that makes the AI useful for the institution's specific work. Individual knowledge is what one person brings and what is held about them under custodianship rather than ownership. Session knowledge is what the current turn of conversation holds and what accumulates within it.

Each kind of knowledge has architectural consequences the EA designs for. Vendor knowledge sits at the foundation of the runtime layer's model endpoint dependencies. Organisational knowledge is the substantive content of the institution's data architecture (section 4) and of any institution-specific fine-tuning. Individual knowledge intersects with the institution's identity and access architecture (section 3) and with the conversation history layer of the data taxonomy (section 4 Layer 3). Session knowledge accumulates within the runtime layer (section 9) and is governed by the lifecycle architecture (section 11).

The four kinds of knowledge are not architectural categories on their own. They are the framing that lets the EA see why the architectural categories matter. Section 4 establishes a different taxonomy, the technical four-layer taxonomy of source data, embeddings, conversation history, and model weights, that is the architectural realisation of the four kinds of knowledge at the data-handling level. The EA holds both taxonomies simultaneously: the business framing tells them what the AI knows; the architectural framing tells them how the AI's data is structured.

## The three layers

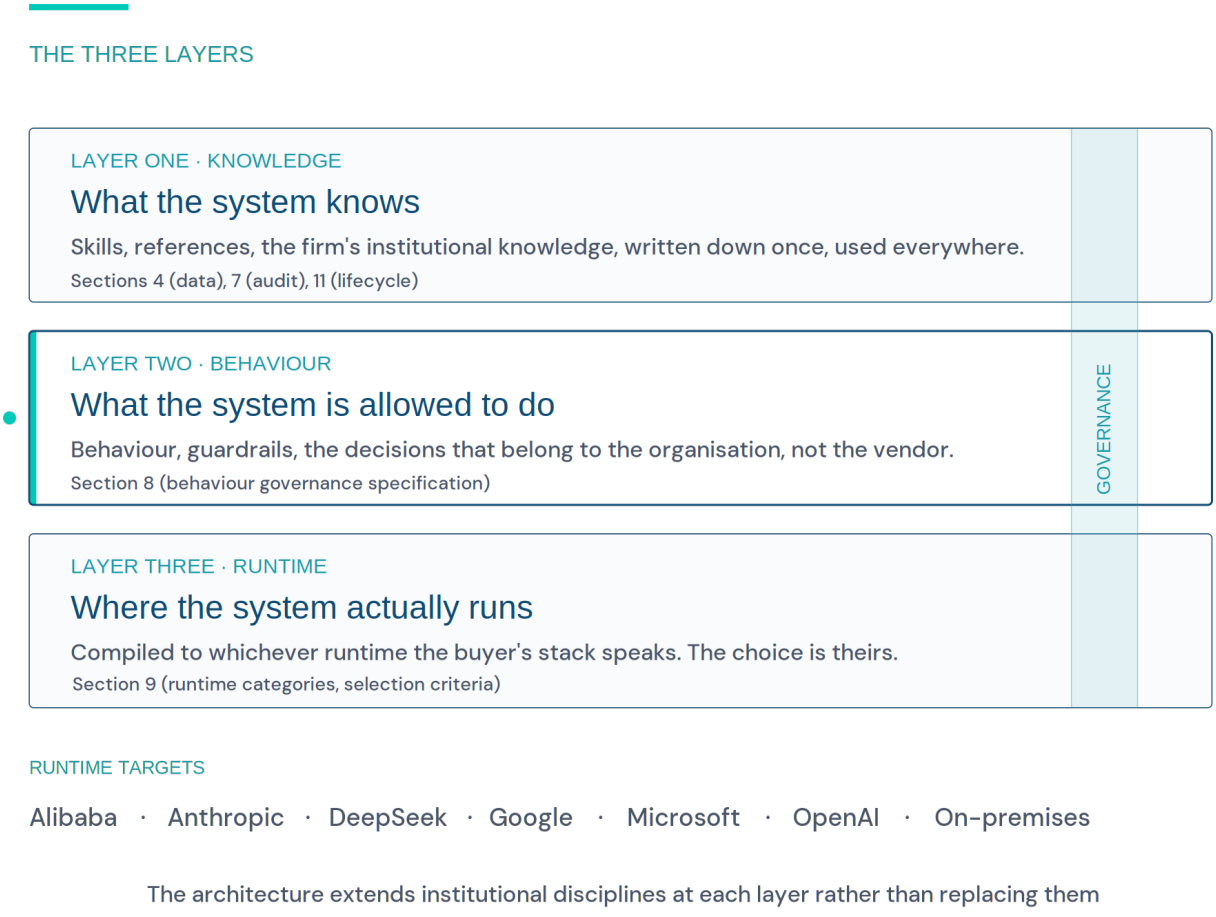



**Figure 2.** The three architectural layers, with governance running through all of them.

The Executive Briefing established that the architecture has three operational layers: the knowledge layer (what the AI knows and how the institution governs that knowledge), the behaviour layer (what the AI is allowed to do and how the institution specifies the rules), and the runtime layer (where the AI actually executes and how the institution selects the implementation). Governance runs through all three.

The knowledge layer is treated across sections 4 (data architecture), 7 (audit and observability), and 11 (information lifecycle). The data architecture organises the institution's AI-related data; the audit infrastructure produces the evidence that the institution governed the data correctly; the lifecycle architecture handles the retention, deletion, and right-to-erasure obligations the institution has to discharge.

The behaviour layer is where Score sits architecturally. The institutional behavioural specification operates at a level of abstraction above any single runtime format and above any single vendor's native behavioural mechanisms. Section 8 establishes what the behaviour layer has to do, why an authoring layer above runtime formats is structurally necessary, and what the institution's work at this layer looks like in practice.

The runtime layer is where the specification becomes operational. Section 9 establishes the runtime category and the selection criteria the institution applies to runtime procurement. The architectural pattern is portable across compliant runtimes; the institution's selection determines which runtime executes the specification.

## Trust as four-property structural outcome

The Executive Briefing argued that trust in AI systems rests on four properties: transparency about what the AI knows and what it does not, accuracy of the information the AI surfaces, timeliness of the AI's knowledge against the operational moment, and security of the AI's data and behaviour against compromise. These are not separate concerns; they are properties the architecture has to produce together because the failure of any one undermines the others.

The architecture this document specifies produces the four properties as design-time outcomes rather than runtime assurances. Transparency emerges from the audit infrastructure (section 7) and the editorial process the lifecycle architecture operationalises (section 11). Accuracy depends on the data architecture (section 4), the regulatory mappings in the behaviour governance specification (section 8), and the runtime layer's vendor and model abstraction (section 9). Timeliness depends on the lifecycle architecture's refresh cadence and the data architecture's retrieval-time access patterns. Security runs through the security and information management spine (sections 3, 4, 5, 6, 7) and the encryption and key management practice the institution extends.

Each property is the institutional outcome of multiple architectural surfaces working together. The architecture-supports-but-does-not-constitute axiom applies: the architecture supports the four properties through the surfaces it specifies; the institutional accountability framework operates on the architecture's outcomes.

## Five decisions

The Executive Briefing named five decisions every institution makes about its AI architecture, deliberately or by default: where the data lives, whether the institution operates with a single vendor or multiple, whether on-premises is part of the picture, when governance is designed in rather than bolted on, and how the institution handles personal knowledge.

Section 14 revisits these decisions with the architectural and security context the architect needs to specify them in their actual environment. The decisions are signposted here as the document's structural anchor: the architectural specification this document produces is the artefact that supports the institution's deliberate making of the five decisions. Without the specification, the decisions are made implicitly through accumulated vendor configurations and procurement choices; with the specification, the institution can make them deliberately and operate on the consequences.

## The implementation arc

The Executive Briefing established the rhythm of institutional adoption: ninety days to make the strategic and architectural choices, twelve months to operationalise the choices into a system the institution can audit cleanly and own properly, a decade across which the investment compounds or evaporates depending on whether the choices were made deliberately. Section 15 returns to the implementation arc at the close of the document, after the architectural specification has been laid out across sections 3 to 14.

The arc is the document's structural rhythm. The ninety-day window shapes which surfaces the institution prioritises in the first wave; the twelve-month operationalisation determines how the institution sequences the work across the architectural surfaces; the decade is the operating environment within which the architecture compounds. The architect's work runs concurrently with the executive sponsor's strategic work across the rhythm; the architectural specification supports the concurrent work rather than imposing a sequential prerequisite.

SECTION 03

# Identity and access architecture for AI

The institutional identity and access management framework extends to the AI estate through three identity categories that conventional IAM has not previously had to handle simultaneously: the human principal who initiates a request, the service identity the AI uses to call downstream systems, and the AI as actor identity that appears in audit logs as the entity that took action. The Executive Briefing's fifth decision established that personal knowledge sits inside the institution but is held under custodianship rather than ownership; the three identity categories operationalise this custodianship at the architectural layer by recording both the principal whose authority the AI exercised and the AI as the actor that exercised it. The architectural specification establishes how each category is handled, how the relationships between them are recorded, and how the institutional accountability framework operates on the resulting evidence.


**Figure 3.** Where responsibilities sit across the architecture.

HOW DATA IS PARTITIONED ACROSS MULTI-LLM DEPLOYMENT

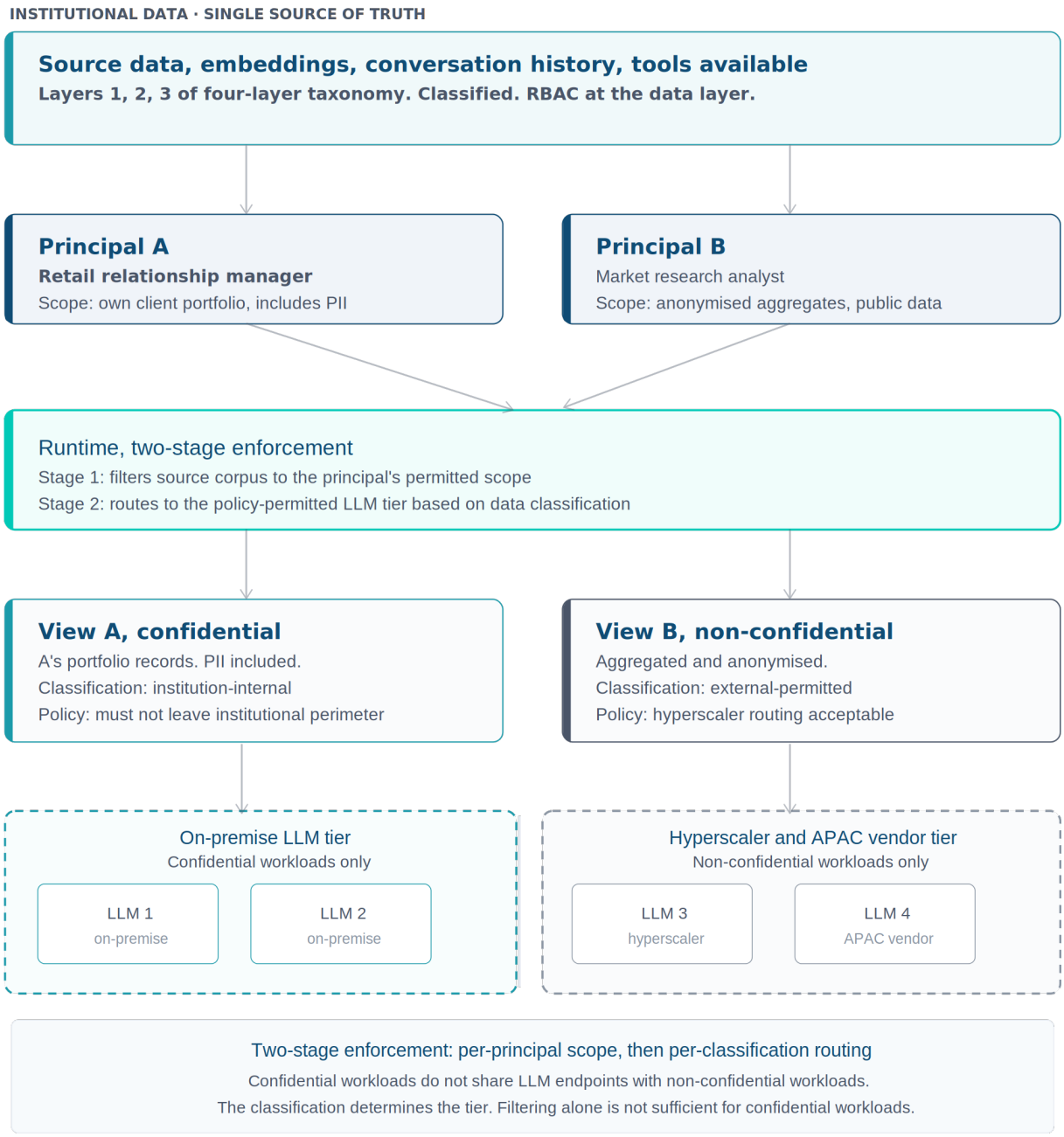



**Figure 4.** How data is partitioned across multi-LLM deployment, with two-stage enforcement at the runtime: per-principal scope, then per-classification routing.

## The three identity categories

- **The human principal.** Standard authentication against the institution's existing identity provider, federated through SAML or OIDC depending on the institution's substrate. Role-based access control determines what the human is allowed to ask the AI to do, what the human is allowed to see in responses, and what categories of organisational knowledge the AI can surface to this human at this moment. The human principal is the conventional half of the IAM specification; the institutional pattern carries forward without significant AI-specific extension.
- **The service identity.** When the AI calls downstream systems to retrieve data, invoke tools, or access APIs, it does so as a service identity. The architectural pattern is on-behalf-of authentication: the AI's service identity carries delegated authority from the human principal who initiated the request, scoped to what the principal is permitted to do, with credentials issued at request time rather than held statically. OAuth 2.0 Token Exchange (RFC 8693) is the protocol-level pattern with `subject_token` carrying the principal and `actor_token` carrying the AI's service identity (IETF 2020); the specification supports the distinction natively. The institutional architecture absorbs this through extension of existing federated infrastructure rather than through a new identity domain.
- **The AI as actor identity.** When the AI takes action that gets logged, who is named in the audit trail? The conventional answer (the human principal) collapses the institutional accountability question into the principal's name without recording that the AI was the immediate actor. The architectural answer is to record both: the principal whose authority the AI exercised, and the AI as the actor that exercised it. RFC 8693's `sub` and `act` claims carry the distinction at the audit-log level operationally tractably (IETF 2020). Section 7's audit infrastructure captures both at the request layer.

The principal-actor distinction is solved at the audit-log level. It is not solved at the institutional accountability level, and the architecture should not pretend otherwise. When the AI takes action that produces an incident, the institutional accountability framework determines whose responsibility it was; the architecture supplies the evidence that the framework operates on. The IAM architecture supports the accountability framework but does not constitute it.

## How the three categories integrate

Three integration patterns matter architecturally.

- **Just-in-time credential issuance.** The AI's service identity does not hold static credentials with broad scope. Credentials are issued at request time, scoped to the specific principal's authority and the specific operation being performed, with short lifetimes that expire when the request completes. The institutional pattern is the broker model: a credential broker (often the AI gateway introduced in section 5) issues request-scoped credentials in response to the AI's authentication, with the broker holding the trust relationships with downstream systems rather than the AI service identity holding them.
- **Scoped delegation.** The AI does not get more permissions than the principal asking it. If the principal cannot read a specific record, the AI cannot read it on the principal's behalf. The architectural specification depends on the institutional IAM enforcing principal-scoped access at the data and tool layer, with the AI's service identity inheriting the principal's scope rather than carrying its own.
- **Token Exchange brokerage.** The AI gateway is the natural broker for OAuth 2.0 Token Exchange flows because every AI request crosses the gateway by architectural construction (section 5). The gateway converts the principal's authentication into the principal-plus-actor token that downstream systems consume. This concentrates the IAM-specific complexity at one architectural element rather than distributing it across the AI estate.

These patterns extend the institutional IAM rather than replace it. The conventional IAM disciplines (federation, RBAC, credential management, session control, privileged access management) continue to apply. The AI-specific extensions are calibrated to the three-identity-category structure.

## Two-stage enforcement against data leakage

The IAM architecture establishes who is permitted to do what; the runtime architecture has to enforce that permission against the data the AI workload sees. Two stages of enforcement are load-bearing for preventing leakage across the multi-LLM estate.

- **Stage 1, per-principal scope filtering.** The runtime filters the source corpus to the principal's scope before any LLM call. Retrievals against records the principal cannot read do not reach the model substrate. Tool invocations against systems the principal cannot operate are refused at the runtime layer rather than passed to the LLM and refused there. The filtering is the architectural answer to the question "what data should this LLM call see".

- **Stage 2, per-classification LLM routing.** The runtime routes the filtered request to the policy-permitted LLM tier based on data classification. Confidential workloads route to the on-premise LLM tier; non-confidential workloads route to the hyperscaler or APAC vendor tier. The two tiers do not share LLM endpoints. The routing is the architectural answer to the question "which LLM is permitted to see this data".

The case for two stages rather than one is that filtering alone is insufficient for confidential workloads. A confidential filtered context fed to a hyperscaler-hosted LLM has crossed a boundary the institution does not control: inference happens outside the institutional perimeter, embeddings of the confidential content are computed in vendor infrastructure, conversation history may be retained by the vendor depending on terms. The on-premise LLM tier exists precisely because filtering cannot constrain what the LLM substrate does with the filtered content once the content has reached it. Classification-driven routing is what makes the on-premise commitment from the Executive Briefing's third decision (and section 5's deployment topology) operationally enforceable rather than aspirational.

The institutional consequence is that the multi-LLM estate is not a single pool of substitutable endpoints. It is at least two pools: the institution's own infrastructure for confidential workloads, and the external vendor infrastructure for non-confidential workloads. Adding LLMs within a tier is operationally simple; routing across tiers is policy-enforced and architecturally prevented. Section 5 specifies the network and deployment topology that supports the two-tier separation; the IAM architecture in this section establishes the per-principal scoping that determines the classification of each request.

## The vendor capability gap

AI-aware IAM vendor capability is shipping ahead of published architecture documentation. Microsoft Entra ID, Okta, and the broader identity vendor ecosystem have begun adding AI-specific features through 2025 and into 2026; the underlying architectural patterns are converging but the published documentation lags the actual capabilities. The institutional response is to design to the architectural pattern this document specifies and treat vendor capability as a procurement-due-diligence input.

Microsoft Entra Agent ID is institutional-governance specification at the identity layer (Microsoft 2025b). It addresses agent identity within the Microsoft estate but does not extend across vendors; an institution running Microsoft alongside Anthropic and an APAC vendor has Entra Agent ID covering only the Microsoft slice. The cross-vendor governance specification still lives above all three. This is the four-corner mesh from section 13 reactivated at the agent identity layer.

## What the architecture requires of IAM

Three architectural requirements extend the institutional IAM framework to handle the AI estate.

The IAM specification names how the three identity categories are handled at every AI request: human principal authentication and authorisation, service identity issuance and scoping, AI-as-actor recording in audit logs.

The AI gateway operates as the credential broker and Token Exchange node, concentrating the IAM-specific complexity at one architectural element.

The IAM architecture supplies evidence to the institutional accountability framework without constituting that framework. The principal-actor distinction is recorded; the institutional question of accountability is settled by the framework that operates on the recorded evidence.

SECTION 04

# Data architecture for AI

The institutional data architecture for AI workloads operates against a four-layer taxonomy that distinguishes the structural categories of data the AI estate produces and consumes: source data, embeddings, conversation history and inference logs, and model weights. Each carries distinct classification, lineage, residency, and lifecycle properties. Treating "AI data" as a single category collapses the architectural questions that actually matter; treating each layer separately produces the discipline the regulatory mesh treated in section 13 requires.

This is a different taxonomy from the four kinds of knowledge the Executive Briefing established. The briefing's taxonomy of vendor, organisational, individual, and session knowledge is the business and governance perspective on what the AI knows. This section's taxonomy is the technical and data-handling perspective on how the AI's data is structured. The two are connected. Vendor knowledge is the foundation level of Layer 4. Organisational knowledge is operationally realised through Layer 1 source data, Layer 2 embeddings, and where applicable Layer 4 institution-specific fine-tuning adapters. Individual knowledge is realised across Layer 1 (the individual's documents and notes held by the institution under custodianship) and Layer 3 (the individual's conversation history with the AI). Session knowledge is the session granularity of Layer 3. The architecture handles both perspectives simultaneously because the institution governs the AI's knowledge from the business angle and architects the AI's data handling from the technical angle.

## The four-layer taxonomy

THE FOUR-LAYER TAXONOMY MAPPED AGAINST THE FOUR KINDS OF KNOWLEDGE

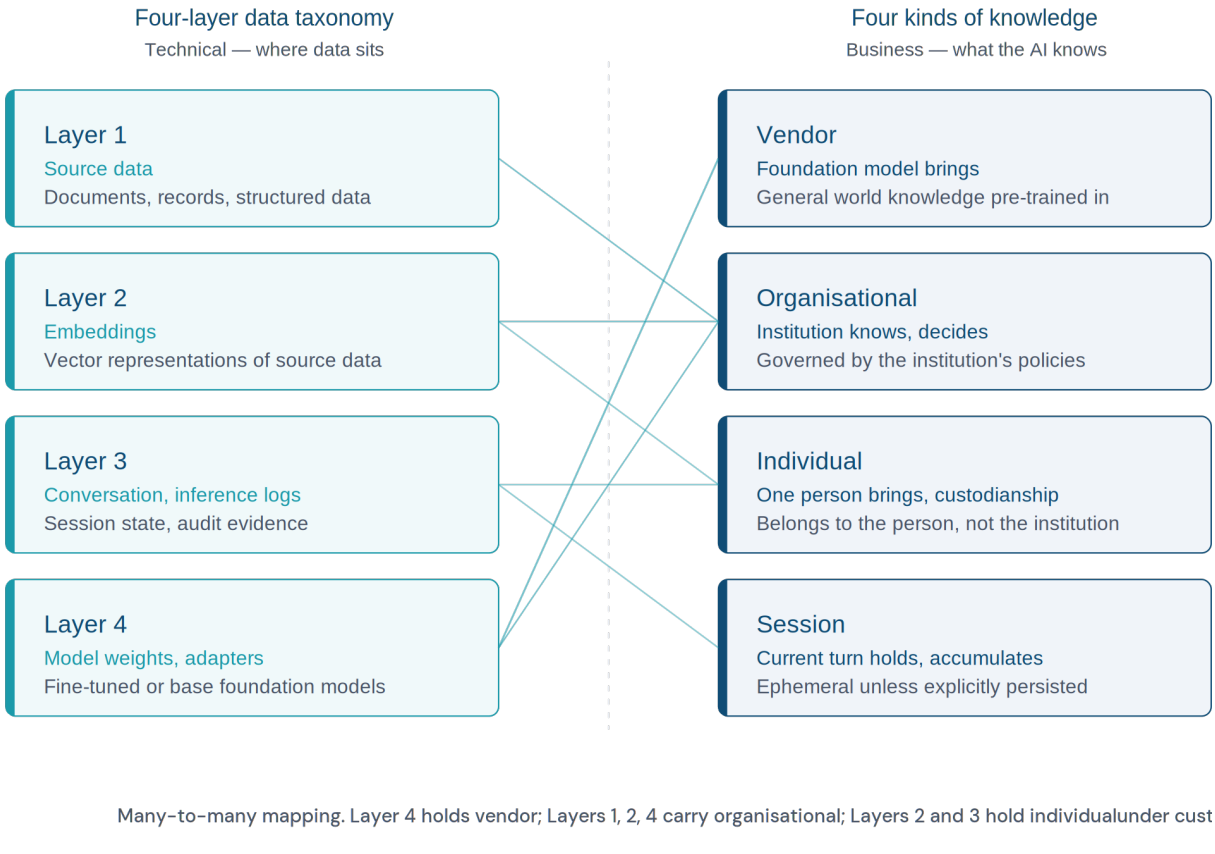



**Figure 5.** The four-layer technical taxonomy mapped against the four kinds of knowledge from the Executive Briefing.

- **Layer 1 — Source data.** The institutional documents, records, and structured information that the AI estate retrieves from at inference time or that fine-tuning consumes at training time. This is the architectural layer where the executive briefing's organisational knowledge lives operationally. Conventional institutional data classification applies. The institution's existing classification framework, retention schedules, and access controls operate at this layer with minimal AI-specific extension.

- **Layer 2 — Embeddings.** Vector representations of source data computed at ingestion time and stored in vector databases for retrieval. Layer 2 inherits the classification of the source data it represents; the executive briefing's organisational knowledge framing carries forward through embedding-time pre-computation. The contested question in 2026 is whether embeddings can be inverted to recover meaningful information about their source; the academic literature is mixed and evolving, with embedding-inversion attacks demonstrated against some models and not others. The conservative architectural default is that embeddings inherit source-data classification absent specific evidence to the contrary. The institution that treats embeddings as derived assets with no inherited classification is making an assumption the academic literature does not yet support.
- **Layer 3 — Conversation history and inference logs.** The records of what the AI was asked, what it retrieved, what tools it called, what it produced. Layer 3 is where the executive briefing's session knowledge lives architecturally, with individual knowledge crossing into Layer 3 when an individual's conversation history is held over time by the institution under custodianship. This layer is the most AI-specific in volume and structure. Conversation history may contain personal information surfaced through inference, references to source data classified at multiple levels, and traces of tool invocations that produced external consequences. The architectural pattern is to default to the strictest classification of any component referenced in the conversation, with component-level relaxation only where the audit trail can demonstrate the relaxation is sound.
- **Layer 4 — Model weights.** The foundation model's parameters, plus any institution-specific fine-tuning adapters trained on institutional data. Layer 4 is split between the foundation level (the vendor's responsibility, the executive briefing's vendor knowledge at the architectural layer) and the institution-specific adapter level (the institution's responsibility, the operationalisation of organisational knowledge through fine-tuning). Foundation model weights are typically the vendor's responsibility and outside the institution's data architecture scope; adapters trained on institutional data inherit the classification of their training corpus. The architectural specification names which layers of weight are institutional assets and which are vendor-managed.

## Lineage tracing through the layers

The institution's data lineage architecture extends to handle AI-specific patterns. When a piece of information enters via ingestion, gets embedded into a vector store, gets retrieved as context, gets included in a response, gets stored as conversation history, and contributes to a derived asset, the lineage has to be traceable through each step.

OpenLineage is the open specification anchor for AI workload lineage in 2026 (OpenLineage 2024). The major data catalog vendors have begun absorbing AI-specific lineage through 2025 and into 2026; the catalog is the institutional artefact that lets the architecture operate across classification, residency, and lineage simultaneously. The AI gateway introduced in section 5 is the natural concentration point for lineage event capture because every AI request crosses the gateway by architectural construction.

The vendor capability gap applies at the lineage layer. AI-aware catalog vendor claims have shipped capability ahead of underlying patterns being settled; the institutional discipline is to verify what the vendor's lineage actually traces against published specifications rather than vendor positioning.

## Personal data handling

The intersection of GDPR (European Parliament and Council 2016), PIPL (NPC 2021a), PDPO (PCPD 2024), and other personal data regimes with AI ingestion, training, and inference is the regulated half of the data architecture. Section 13 treats the regulatory mesh; section 11 treats the information lifecycle implications. Three architectural patterns matter at the data architecture layer specifically.

- **Data minimisation at ingestion.** The institution does not ingest personal data into the AI estate that the use case does not require. The architectural specification names which categories of personal data are permitted at which layer of the taxonomy: source data ingestion is conventionally classified and controlled; embedding-time inclusion of sensitive personal information requires explicit institutional approval; fine-tuning ingestion of personal data is the highest-friction option and should be the last resort rather than the default.
- **Retrieval-time access for sensitive personal information.** Where the AI needs sensitive personal information at inference time, the architectural pattern is retrieval-time access against the institution's existing data infrastructure rather than embedding-time pre-computation. This minimises the lifecycle exposure (section 11) and supports the right-to-erasure obligation that is otherwise difficult to discharge at the embedding layer.
- **Cross-border transfer mechanics.** PIPL Articles 38 and 40 (NPC 2021a), GDPR Chapter V (European Parliament and Council 2016), the 22 March 2024 cross-border data flow provisions (CAC 2024), and the GBA Standard Contract regime each impose specific cross-border transfer requirements that bear on where the AI estate's data architecture places personal data at rest, in transit, and at inference time. The architectural specification treats data residency as a structural property of the architecture rather than as a deployment-time configuration.

## Data sovereignty as architectural property

Where data lives physically, where it is processed at inference time, and where derived assets reside are decisions the institution makes regardless of platform. Section 5 (network and deployment topology) handles the deployment topology; section 13 handles the regulatory framing. The data architecture's contribution is naming data residency as a structural property at each layer of the taxonomy: source data residency follows institutional classification, embedding storage residency follows source-data residency by default, conversation history residency follows the strictest classification of its components, fine-tuning adapter residency follows the institutional classification of its training corpus.

The architectural posture is that data residency is specified at design time rather than discovered at audit time. Section 14's first decision (where the data lives) is the institutional commitment to this posture; the data architecture is where the commitment becomes operational.

## What the architecture requires of data management

Three architectural requirements extend the institutional data architecture to handle the AI estate.

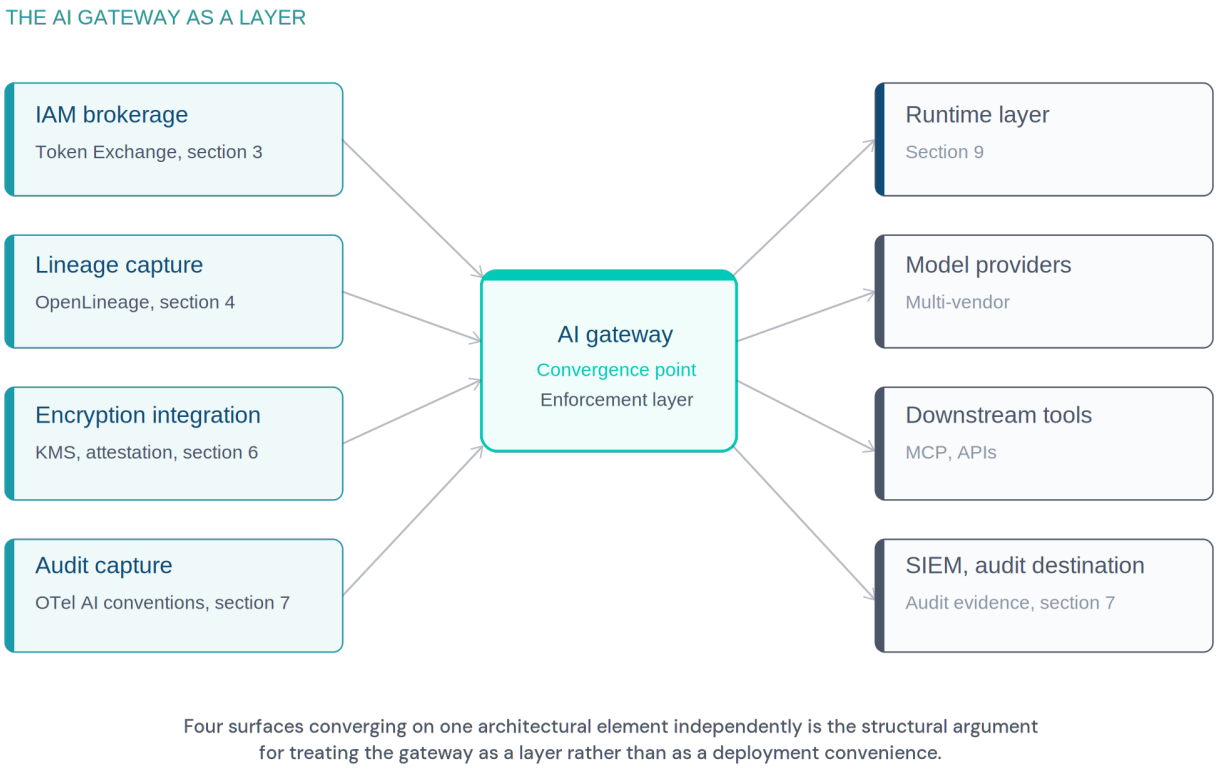
- The data architecture specification names how each of the four layers is classified, where each layer's data resides, what lineage tracing is required, and what residency obligations apply.
- The institutional data catalog absorbs AI-specific lineage and residency at the layer-by-layer granularity the regulatory mesh requires. The catalog is the architectural artefact that lets the institution operate across classification, residency, and lineage simultaneously.
- Personal data handling is calibrated against the four-layer taxonomy. Data minimisation at ingestion, retrieval-time access for sensitive personal information, and cross-border transfer mechanics treated as structural properties rather than deployment configurations.
- The lineage architecture supports regulatory inspection but does not constitute compliance.

SECTION 05

# Network and deployment topology

The institutional network architecture extends to handle three traffic patterns the AI estate introduces: model endpoint calls that cross the institutional perimeter, retrieval queries against vector stores at high frequency, and tool invocation outbound traffic with external consequences. The deployment topology specifies where AI workloads run, how they integrate with existing institutional segments, and what segmentation patterns are required for the regulated half of the AI estate. The AI gateway emerges across multiple architectural surfaces as a discrete layer that the deployment topology establishes.

## The AI gateway as a layer of its own




Four surfaces converging on one architectural element independently is the structural argument for treating the gateway as a layer rather than as a deployment convenience.

**Figure 6.** The AI gateway with four converging surfaces, lands the structural argument that the gateway is a layer rather than a deployment convenience.

The case for treating the AI gateway as a discrete architectural element rather than a deployment convenience is empirical. Four institutional surfaces independently converge on the gateway as the integration point. Section 3's identity and access architecture uses the gateway as the credential broker and Token Exchange node. Section 4's data architecture uses the gateway as the concentration point for lineage event capture. Section 6's

encryption and key management uses the gateway as the integration point where IAM-bounded access, encryption-at-rest-and-in-transit, and attestation-conditional key release converge. Section 7's audit and observability uses the gateway as the audit-capture and schema-normalisation point that produces a unified audit view across multi-vendor estates.

Four surfaces converging on the same architectural element independently is the structural argument for treating the gateway as a layer rather than as one of several optional integration patterns. The institutional pattern is converging on AI gateway as a deployment element across published reference architectures including AWS Multi-Provider Generative AI Gateway (AWS 2025), TrueFoundry (TrueFoundry 2025), and Envoy AI Gateway under CNCF (CNCF 2025). The architectural specification establishes the gateway in this section and references it consistently across the rest of the security and information management spine.

The gateway is not the runtime layer (section 9). The runtime executes Score-compliant skills against model endpoints; the gateway is the institutional boundary at which AI traffic crosses between the runtime and the model providers, between the runtime and downstream tools, between the runtime and the institution's existing data and identity infrastructure. The runtime makes the AI request; the gateway governs the institution's perimeter for that request.

## Three traffic patterns

- **Model endpoint calls.** The AI's calls to model provider endpoints cross the institutional perimeter unless the model is hosted on-premise. The architecture has to specify which model endpoints are approved for which use cases, what egress controls apply, and how model traffic is monitored. The gateway is the natural enforcement point: every model endpoint call crosses the gateway by architectural construction, and the gateway applies institutional policy to each call.
- **Retrieval queries.** Vector store retrieval is high-frequency traffic with a different pattern from traditional database queries. A single AI request typically produces multiple retrieval queries against the vector store; the volume scales with AI usage rather than with the underlying business activity. The architectural specification handles this through deployment topology: vector stores are co-located with inference for performance, with the segmentation matching the strictest classification of the source data the vector store contains.

- **Tool invocation outbound.** When the AI calls external tools or APIs as part of producing a response, the institution's egress controls apply. The pattern is unpredictable from the AI's perspective and predictable from the institution's: the institution specifies which tools are approved for which use cases, and the gateway enforces the specification by mediating tool invocation. Section 12's incident response treats tool invocation with external consequence as a named AI-specific incident type; the deployment topology supports incident response by concentrating tool invocation at the gateway where it can be audited and controlled.

## Segmentation patterns

The architectural pattern is dedicated AI segmentation rather than co-location with core production systems. The reasoning is that AI workloads have a different traffic profile, a different vendor surface, and a different governance posture from core production; mixing them in the same network segment makes both segmentations weaker.

Three patterns matter.

- **Sensitive workload segmentation.** AI workloads handling sensitive customer data, regulated financial transactions, or material non-public information sit in segments calibrated to the institution's strictest data classification. These segments typically have on-premise deployment as the baseline (the regulatory mesh treated in section 13 expects sensitive workloads on infrastructure the institution controls), with cloud augmentation only where the cloud component does not see the underlying data.
- **General productivity workloads.** AI workloads supporting internal documentation, meeting transcription, code assistance, and similar non-sensitive functions sit in segments calibrated to standard institutional classification. Cloud-hosted deployment is the default; on-premise is the exception driven by specific institutional policy rather than by regulatory requirement.
- **Public-facing workloads.** AI workloads serving customers, members, or other external audiences sit in segments calibrated to the institution's customer-facing infrastructure. The architectural pattern depends on the specific use case: customer service chatbots may sit in cloud-hosted infrastructure with appropriate data handling, while AI-mediated customer authentication may require on-premise deployment with stricter controls.

## On-premise deployment as architectural reality

The Executive Briefing's third decision established that in regulated industries, on-premises remains the baseline for sensitive workloads, not a fallback for legacy systems. The empirical position from item 9 substantiates this for regulated APAC enterprises specifically. The named-institution evidence is concrete: HSBC's December 2025 Mistral partnership running self-hosted on HSBC's internal technology systems (HSBC 2025), JPMorgan's controlled multi-provider in-house platform with the LLM Suite (JPMorgan Chase

2025), the Cyberport AI Supercomputing Centre as institutional infrastructure for HK enterprises (Cyberport 2024), and the HKMA Sandbox cohorts including the Sandbox++ multi-regulator launch on 5 March 2026 (HKMA et al. 2026) that have included on-premise architectures from Tier-1 institutions. The on-premise share is bounded by methodological discipline; vendor narratives that overstate the share do not survive scrutiny.

The architectural pattern is hybrid deployment for institutions of meaningful scale: sensitive workloads on-premise, general productivity workloads in cloud, with governance specification spanning both. The hybrid pattern requires deliberate architectural work because the boundary between on-premise and cloud is the boundary at which the regulatory mesh's data residency obligations bite hardest. Section 13's regulatory mesh and section 4's data architecture handle the implications.

The on-premise commitment is operationally enforceable rather than aspirational because of the two-stage enforcement specified in section 3. Per-principal scope filtering at the runtime determines the classification of each request; per-classification routing at the runtime determines which LLM tier serves it. Confidential workloads route to the on-premise tier exclusively; non-confidential workloads route to the hyperscaler or APAC vendor tier. The two tiers do not share LLM endpoints. The deployment topology this section specifies (separate on-premise and cloud LLM tiers) is what gives the IAM architecture's classification-driven routing somewhere to route to.

## What the architecture requires of network and deployment

Four architectural requirements extend the institutional network architecture to handle the AI estate.

The AI gateway is established as a discrete architectural element with the four converging surfaces (IAM brokerage, lineage capture, encryption integration, audit normalisation) operating at it.

The three AI-specific traffic patterns (model endpoint calls, retrieval queries, tool invocation outbound) are governed through dedicated segmentation and gateway enforcement rather than absorbed into existing network architecture.

The deployment topology specifies which workloads run on-premise, which run in cloud, and which run hybrid, with the boundaries calibrated to data classification and the regulatory mesh.

The deployment topology supports the architectural argument but does not constitute the governance posture.

SECTION 06

# Encryption and key management

The institutional encryption and key management practice extends to handle AI workloads with less novelty than vendor positioning suggests. The conventional disciplines (encryption at rest, encryption in transit, key management infrastructure, hardware security modules, customer-managed keys, separation of duties at the key layer) carry forward with one AI-specific extension: conditional key release governed by workload attestation, where the institution requires cryptographic evidence of the AI workload's deployment context before releasing keys for cross-boundary inference.

The encryption posture is layer-specific against the four-layer taxonomy from section 4. The architectural specification establishes what is encrypted at each layer, how key management integrates with the institution's existing KMS, and where confidential computing is a meaningful trust-boundary improvement rather than vendor positioning.

## The four-layer encryption posture

- **Layer 1 (source data).** Conventional encryption at rest with FIPS 140-3 (NIST 2019) or equivalent compliance, AES-256 minimum, customer-managed key encryption keys (KEKs) wrapping the data encryption keys, integration with the institution's existing HSM infrastructure. No AI-specific extension at this layer.
- **Layer 2 (embeddings).** The complication is volume and access pattern. Vector store retrieval is high-frequency traffic with different KMS rate characteristics than transaction databases; institutional KMS infrastructure may not be calibrated for the access volume that high-throughput retrieval produces. Two architectural patterns absorb the complication: local key caching with short cache TTL and HSM-backed unwrap on cache miss, or trusted execution environment-protected query nodes that hold the wrapping keys in attested memory. Either pattern is the right answer for some institutional use cases; the architectural specification names which pattern applies to which use case.
- **Layer 3 (conversation history and inference logs).** Field-level encryption complications because the conversation history may include personal data, source data references at multiple classification levels, and tool invocation records. The architectural pattern is to default to the strictest encryption posture of any component referenced in the conversation, with field-level relaxation only where the audit trail can support it.
- **Layer 4 (model weights).** Foundation model weights are the vendor's responsibility and the institution's encryption posture is bounded by the vendor's controls. Institution-specific adapters trained on institutional data inherit the encryption posture of their training corpus, with AES-256 at rest, customer-managed keys, and integration with the institution's MLOps registry for adapter lifecycle management.

The four-layer treatment makes the architecturally consequential distinctions visible. Source data and adapters carry institutional encryption posture; embeddings and conversation history have AI-specific extensions; foundation model weights are bounded by vendor controls.

## Conditional key release and workload attestation

The AI-specific extension to institutional KMS practice is conditional key release governed by workload attestation. The institution releases keys for cross-boundary inference only when the AI workload demonstrates cryptographic evidence of its deployment context: the runtime version executing the request, the model endpoint being invoked, the source classification of the data being processed, the audit context for the request.

Three architectural patterns matter.

- **Attestation at the AI gateway.** The gateway introduced in section 5 is the natural enforcement point for attestation-conditional key release because every AI request crosses the gateway by architectural construction. The gateway holds the trust relationship with the institutional KMS and brokers attestation-validated key release for the runtime layer.
- **Hardware-backed attestation for sensitive workloads.** Where the AI workload processes sensitive personal information or material non-public information, the attestation chain extends to hardware-backed evidence (Intel SGX, AMD SEV, NVIDIA Confidential Computing). Hardware attestation produces cryptographic evidence that the workload is running on attested hardware in an attested configuration; conditional key release validates the attestation before releasing keys.
- **Software-backed attestation for general workloads.** Where the workload does not warrant hardware-backed attestation, software-level evidence (signed runtime images, signed configuration manifests, runtime context attestation) is the architectural pattern. The trust boundary is weaker than hardware attestation but stronger than no attestation; the architectural specification names which workloads warrant which pattern.

## Confidential computing in its proper place

Confidential computing is a meaningful trust-boundary improvement for specific high-sensitivity use cases and overkill for most enterprise AI. The vendor positioning collapses four distinct mechanisms into "confidential AI": TEE-protected inference, encrypted inputs and outputs, attested model weights, and attestation-conditional key release. The institutional architecture assesses each mechanism independently rather than treating them as a single feature category.

The threat model is specific. Confidential computing protects against privileged-operator access to running workloads, certain memory-extraction attacks, and certain side-channel attacks under specific conditions. It does not protect against compromised guest operating systems, supply-chain compromise of the workload code, malicious workload code itself, or compromise of the attestation infrastructure. The architectural specification names the threats the institution actually wants to mitigate and applies the appropriate mechanism rather than treating "confidential AI" as a generic protection.

The mainland Chinese cryptographic regulatory regime adds a parallel architectural concern. Workloads handling mainland Chinese data are typically subject to GM/T standards (the SM2, SM3, SM4 cryptographic standards mandated by the Cryptography Law and the Encryption Law) rather than FIPS-aligned standards. Institutions operating dual stacks (Western and mainland Chinese) handle dual cryptographic regimes; the architectural specification names which workloads operate under which regime. This is the regulatory mesh from section 13 reactivated at the cryptographic layer.

## What the architecture requires of encryption and key management

Three architectural requirements extend the institutional encryption and key management practice to handle the AI estate.

The encryption posture is specified per layer of the four-layer taxonomy, with AI-specific extensions (KMS rate complications at Layer 2, field-level encryption complications at Layer 3) named explicitly.

Conditional key release governed by workload attestation is the AI-specific extension to institutional KMS practice. The AI gateway is the natural enforcement point for attestation-validated key release.

Confidential computing is calibrated to specific high-sensitivity use cases rather than treated as a generic protection. The institution's threat model determines where the mechanism applies; vendor positioning is procurement input rather than architectural input.

The encryption architecture supports the institutional cryptographic posture but does not constitute compliance with cryptographic regulatory regimes. The institutional cryptography function and the regulatory mapping in section 13 carry the compliance posture; the architecture supplies the mechanisms the function operates on.

SECTION 07

# Audit, logging, and observability architecture

The institutional audit and observability infrastructure extends to handle AI workloads at structurally higher telemetry volume than conventional applications produce. The Executive Briefing argued that AI knowledge needs an editorial process; the audit and observability architecture is where the editorial process is operationally enforced. Author, review, approve, publish, apply, refresh, retire are not abstract editorial stages; they are events the audit infrastructure has to capture, correlate, and surface to the institutional accountability framework that operates on the resulting evidence.

The architectural specification establishes what gets logged at AI-workload-appropriate fidelity, how AI logs integrate with the institution's existing SOC infrastructure, what the AI-specific telemetry looks like at the gateway, and how multi-vendor audit produces a unified view across the AI estate.

## What gets logged

The AI gateway introduced in section 5 is the audit-capture and schema-normalisation point because every AI request crosses the gateway by architectural construction. The architectural specification names what the gateway captures at each request:

The principal-actor distinction from section 3 (which human principal initiated the request, which AI service identity acted on the principal's behalf, what credentials were issued for the request). The classification context from section 4 (what categories of data the request touched, at which layer of the four-layer taxonomy). The deployment context from section 5 (which runtime version, which model endpoint, which network segment, which deployment topology). The encryption context from section 6 (what attestation was validated, which keys were released for the request). The behavioural specification context (which Score-compliant skill was invoked, which behaviour governance applied; addressed substantively in section 8).

The volume profile is structurally higher than conventional applications. A single AI request produces multiple retrieval queries against vector stores, possibly multiple model invocations if the request involves agent orchestration, multiple tool invocations with external consequences, and the corresponding audit events for each. Section 12's incident response and section 13's regulatory mesh both depend on the audit infrastructure handling the volume at sufficient performance for forensic reconstruction within institutional response windows.

## Integration with institutional SOC infrastructure

AI logs feed the institution's existing security operations infrastructure. The architectural pattern is centralised SIEM ingestion (Splunk, QRadar, Sentinel, or institutional equivalent) with AI-specific enrichment at the gateway before SIEM forwarding.

Three patterns matter.

- **Schema normalisation at the gateway.** Multi-vendor estates produce multi-format logs. Anthropic's API responses, OpenAI's chat completion logs, Microsoft Agent Framework's telemetry, Alibaba's model service logs all have different field structures and event semantics. The gateway normalises these into a canonical schema before SIEM ingestion. The institution operates on the canonical schema rather than on the vendor-specific formats; the canonical schema absorbs vendor variation.
- **Sampling for high-volume telemetry.** Some AI traffic produces telemetry at volumes that conventional SIEM infrastructure cannot ingest in real-time at full fidelity. The architectural pattern is full-fidelity capture at the gateway with sampled forwarding to the institutional SIEM, plus full-fidelity retention at the gateway's audit store for forensic queries that require full reconstruction. The institutional SIEM operates on the sampled view; forensic investigation accesses the full-fidelity store.
- **Alert pattern calibration.** AI-specific alert patterns differ from conventional application alerting. The institutional SOC's alert tuning has to account for AI-specific anomalies (sudden behavioural changes per section 12's behavioural drift incident type, output anomalies from the four AI-specific incident types, model-version transitions that produce non-malicious behavioural changes) without producing false-positive overload. The architectural specification names AI-specific alert patterns; the institutional SOC applies tuning calibrated to the patterns.

## OpenTelemetry AI Semantic Conventions

The institutional pattern is converging on OpenTelemetry AI Semantic Conventions as the open standard for AI workload telemetry (OpenTelemetry 2025). The conventions are evolving rather than settled in mid-2026: prompt-and-completion fields are stable, tool-call fields are converging across the major SDKs, multi-agent chain conventions are still in active development, and model-version-and-deployment metadata is undertreated. The architectural specification adopts the conventions where they are stable and applies institutional discipline where they are not, particularly around model-version-and-deployment metadata, which section 12 named as load-bearing for behavioural drift incident response.

The vendor capability gap applies at the observability layer. AI observability vendor capability (Langfuse, Helicone, LangSmith, Braintrust) is operationally excellent for AI-team purposes (prompt iteration, evaluation, debugging) and partial for institutional-audit-evidence purposes (regulatory inspection, supervisory dialogue, audit attestation). The institutional architecture treats AI observability vendors as AI-team infrastructure and the institutional SIEM as the regulatory-audit layer. The two integrate through gateway-emitted telemetry rather than through direct ingestion of AI observability vendor data into the SIEM.

## Forensic reconstruction

Section 12's incident response depends on the audit infrastructure producing evidence at the fidelity forensic reconstruction requires. Three categories of evidence:

Input-output reconstruction (what request was received, what response was produced) is the most reconstructible category. The gateway captures it at structural fidelity. Internal-state reconstruction (what the model was thinking, why it produced the output it did) is substantially less reconstructible; production-deployable model interpretability is not yet available in 2026. Causal-chain reconstruction (when the AI made a decision that contributed to a downstream incident, what the chain was) is partially reconstructible through audit log correlation and tool-invocation logging.

The architectural posture is the architecture-supports-but-does-not-constitute axiom applied at the forensic layer: the audit architecture supports the institutional accountability framework's investigation by producing the input-output evidence and the model-version-and-deployment context, but does not produce model-internal-state reconstruction. The accountability framework operates on the evidence the architecture supplies.

## What the architecture requires of audit and observability

Three architectural requirements extend the institutional audit and observability infrastructure to handle the AI estate.

The AI gateway is the audit-capture and schema-normalisation point, with the principal-actor distinction, classification context, deployment context, encryption context, and behavioural specification context captured at every request.

AI logs feed the institutional SIEM through schema-normalised ingestion calibrated to AI-specific volume profiles, with full-fidelity retention at the gateway for forensic queries.

The audit architecture supports institutional accountability through the evidence it produces at request time and through the forensic reconstruction it enables at incident time. The architecture-supports-but-does-not-constitute axiom applies; the accountability framework operates on the evidence the architecture supplies.

The audit architecture is the architectural layer at which the Executive Briefing's editorial process becomes operationally tractable. Without it, the editorial process is aspirational; with it, the institution can demonstrate that AI knowledge has been authored, reviewed, approved, published, applied, refreshed, and retired according to the institution's specification.

SECTION 08

# Behaviour governance specification

The Executive Briefing's three-layer framing established that the architecture has three operational layers (knowledge, behaviour, runtime) with governance running through all of them. The middle layer, what the system is allowed to do, written down in a way that is independent of any particular vendor's tools, is the layer most enterprise AI deployments do not have. Without it, what the AI is allowed to do gets written directly into a vendor's product configuration, where it cannot be audited cleanly, cannot be moved when the vendor changes their format, and cannot be governed across multiple vendors at the same time.

The behaviour governance specification is the architectural artefact that makes the middle layer possible. The institution authors its behavioural specification at a level of abstraction above any single runtime format and above any single vendor's native behavioural mechanisms; the specification compiles to the artefacts each runtime requires. Score is the open protocol for this specification; Score-compliant runtimes (treated in section 9) execute the specification at request time.

This section establishes what behaviour governance specification has to do architecturally, why an authoring layer above runtime formats and native model mechanisms is structurally necessary, what the institution's work at this layer looks like in practice, and how the governance layer integrates with the institution's existing behavioural-policy disciplines.

## What the layer has to handle

The institutional behavioural specification has to handle four categories of behavioural concern that conventional policy frameworks have not previously had to integrate at the same architectural altitude.

- **What the AI is allowed to do.** The institutional permissions and prohibitions on AI action: which tools the AI can invoke, which categories of decision the AI can make autonomously, which actions require human approval before execution, which use cases are entirely out of scope. Section 12's incident response treats tool invocation with external consequence as a named incident type; the behavioural specification is where the institutional limits are authored before the gateway enforces them at request time.
- **What the AI is allowed to know.** The institutional permissions on knowledge access: which categories of organisational, individual, and session knowledge the AI can surface to which categories of human principal. Section 3's identity architecture and section 4's data architecture handle the access mechanics; the behavioural specification handles the institutional rules that determine who is allowed to ask the AI to surface what.

- **How the AI is allowed to behave.** The institutional standards on AI output: tone, style, transparency about uncertainty, refusal to engage with categories of question, escalation patterns when the AI detects it is operating outside its specification. The Executive Briefing's four trust properties (transparency, accuracy, timeliness, security) are the executive-level framing of what the institution wants AI behaviour to exhibit; the behavioural specification is the architectural artefact where the institutional standards become operational.
- **Which regulatory regime each behaviour serves.** Section 13's regulatory mesh produces obligations that bear on behaviour: EU AI Act transparency obligations, mainland Chinese content moderation requirements, HKMA expectations on supervised activity, PDPO obligations on personal data handling. The behavioural specification maps each behavioural rule to the regulatory regime it serves, so the institutional accountability framework can demonstrate to each regulator that the relevant obligations have been architecturally specified.

The four categories integrate at the specification layer. The architectural artefact carries permissions, knowledge access rules, behavioural standards, and regulatory mappings as components of a single specification rather than as four separate documents that have to be reconciled at runtime.

## Why an authoring layer is structurally necessary

The case for an authoring layer above the runtime formats and the native model behavioural mechanisms is not feature-comparative. It is not that MCP lacks a feature that Score has, or that Anthropic's constitution lacks a feature that an institutional specification provides. It is structural: the institution operates across multiple runtimes and multiple models, and the institutional behavioural specification has to be portable across both.

Three observations make the structural case.

**Runtime formats are calibrated for runtime concerns.** MCP specifies what tools and prompts are at the protocol layer (Model Context Protocol 2025). Anthropic Skills (under the agentskills.io specification) specify the format of skills at the runtime layer (Agent Skills 2025). OpenAI's tool definitions specify function calling at the API layer. Microsoft Agent Framework distinguishes capability description from behavioural skill specification within Microsoft's runtime stack (Microsoft 2025). Each is calibrated for its runtime; none is calibrated for institutional governance specification at the level the regulatory mesh requires.

The agentskills.io specification has an explicit `metadata` extension hook for arbitrary key-value institutional metadata (Agent Skills 2025). The hook exists; the canonical schema for what goes in it does not; the portability across consumers does not. This is the same architectural pattern as OpenAPI's `x-` prefix vendor extensions: a permission slip for institutional extension that does not produce institutional governance because the canonical schema and the cross-consumer portability are out of scope of the protocol.

The institutional authoring layer above the runtime format produces the canonical schema and the cross-runtime portability that the runtime format's extension hook cannot.

**Native model behavioural mechanisms are vendor-bound.** Anthropic's constitution operates within Anthropic's model. The constitution's January 2026 revision states explicitly that "whatever document stands in this role at any given time takes precedence over any other instruction or guideline that conflicts with it. Subsequent or supplementary guidance must operate within this framework" (Anthropic 2026). The institutional layer's relationship to the constitution is articulated by the constitution itself: the institutional layer operates within the framework; it does not override it.

OpenAI's Model Spec produces the parallel. The 2025-12-18 Model Spec sets out a five-level chain of command (Root > System > Developer > User > Guideline) and states that "Customization, personalization, and localization (except as it relates to legal compliance) should never override any principles above the 'guideline' level in this Model Spec" (OpenAI 2025). The institutional behavioural specification cannot reach System or Root level; it must compile to Developer and User-level instructions that respect the substrate above.

The two vendor positions converge architecturally: each vendor articulates a substrate that takes precedence over institutional configuration, with the institutional layer operating within the substrate's bounds. The institutional behavioural specification has to compile to both substrates simultaneously because the institution operates across both vendors. Bound to either substrate, the specification is bound to one vendor; above both, the specification is portable.

**Microsoft's product architecture corroborates the pattern.** Microsoft Agent Framework's product split between capability description (plugins, tools, OpenAPI imports, MCP imports) and behavioural skill specification (SKILL.md skills with progressive disclosure) is the protocol-vs-authoring split in microcosm, applied within Microsoft's own runtime layer (Microsoft 2025a). Microsoft has independently arrived at the same architectural distinction in its own product work: capability description and behavioural specification are different things at different altitudes that coexist because they describe different things.

One altitude up gives the institutional authoring layer above the runtime formats. Microsoft's product split is the structural argument articulated by the largest enterprise AI vendor in regulated APAC markets, in their own documentation, for their own reasons. The institutional authoring layer is the cross-runtime version of the same architectural pattern.

## Score as the open protocol

Score is the open governance protocol that produces the specification format the architecture requires. The protocol is published as an open specification; the architectural pattern (specification at a level of abstraction above runtime formats, compiled to downstream runtime artefacts) is independent of any single implementation.

Three architectural properties of the protocol matter.

- **Compilation to downstream artefacts.** The institutional Score specification compiles to the runtime formats each runtime requires: Anthropic Skills under agentskills.io, MCP tool definitions, OpenAI function definitions, Microsoft Agent Framework SKILL.md format. The compilation is what makes the specification portable; the specification is what gives the institution architectural authority over what the AI does. The compilation also produces the regulatory artefacts each framework requires (section 13): EU AI Act conformity assessment documentation, ISO 42001 management-system records, supervisor-ready evidence for HKMA, algorithm filing materials for the mainland regulatory stack.
- **Versioning and audit.** The institutional specification is versioned and audited at the architectural layer. Each version of the specification carries the institutional approval that authorised the version; deployment to the runtime layer records which version is in effect at request time. Section 7's audit infrastructure captures the version-and-deployment context per request, supporting forensic reconstruction (section 12) and regulatory inspection (section 13).
- **Regulatory mapping.** The institutional specification maps each behavioural rule to the regulatory regime it serves. The mapping is the architectural artefact that lets the institution demonstrate to a regulator that the relevant obligations have been specified, deployed, and enforced. The regulatory mesh's heterogeneity is intelligence the architecture acts on: each rule's regulatory mapping records which regimes the rule serves, and the compilation produces the regulatory artefacts each regime requires.

These three properties are what the institutional authoring layer produces that the runtime formats and the native model behavioural mechanisms do not. The architectural argument for the layer is that the layer has to exist somewhere; the question is whether the institution authors it deliberately at the abstraction the architecture requires, or whether the institution allows it to be authored implicitly through the accumulation of vendor configurations that are bound to specific runtimes and specific models.

## What the institution actually does at this layer

The behaviour governance specification is the architectural surface where the institution does the most direct work. The other architectural surfaces in this document (identity, data, network and deployment, encryption, audit, lifecycle) are extensions of existing institutional disciplines that the institution already operates; the AI-specific extensions are calibrated against existing infrastructure. The behaviour governance layer is novel work the institution has not previously had to author. The work is institution-specific because the specification encodes what the institution wants the AI to do; no two institutions have identical specifications.

Five components of the work matter architecturally. The institution calibrates each component against its own scale, its own regulatory exposure, its own existing disciplines, and its own operational maturity.

- **Authoring the specification itself.** The specification is a structured document carrying permissions, knowledge access rules, behavioural standards, and regulatory mappings (the four categories named earlier). The authoring is institutional work that translates from the institution's policy language into the specification's syntax. The translation is the most labour-intensive component because policy language is calibrated for human readers and specification syntax is calibrated for runtime compilation; the translation has to preserve the institutional meaning while producing a specification that compiles cleanly.

  The authoring is iterative. A first-pass specification covers the high-priority use cases the institution has identified; subsequent passes extend coverage to additional use cases as the institution's AI estate grows. The institution that attempts to author a complete specification before deploying any AI typically does not deploy any AI; the institution that deploys AI before authoring a specification typically deploys AI without the governance layer the architecture requires. The architectural pattern is iterative authoring with a deployed minimum viable specification that grows with the estate.

- **The institutional roles that hold the work.** Authoring the specification is a multi-disciplinary function that the institution's existing organisational structure may not directly accommodate. The specification carries policy interpretation (compliance and legal disciplines), risk assessment (risk management discipline), behavioural standards (the institutional code of conduct and ethics functions), regulatory mapping (the regulatory affairs function), and architectural specification syntax (the enterprise architecture function). No single institutional role typically holds all of these.

  The architectural pattern is a named cross-functional team with explicit role assignment: a primary author (typically from enterprise architecture or governance) who holds the specification syntax; named contributors from compliance, legal, risk management, and regulatory affairs who hold their respective domain content; an institutional approval authority (typically a senior risk or governance committee) that signs off versions before deployment. The institution that does not name the team explicitly typically discovers at audit time that the specification has been authored by whichever individual happened to push it through, which produces specifications that survive neither institutional scrutiny nor regulatory inspection.

- **The translation from institutional policy to specification syntax.** The translation is genuinely labour-intensive and benefits from institutional discipline rather than from one-off effort. Three patterns matter.

The institution maintains a policy-to-specification mapping document that records which institutional policies translate into which specification rules. The mapping makes the translation auditable: when an institutional policy changes, the mapping identifies which specification rules need updating; when a regulator asks how the institution discharges a policy at the AI layer, the mapping identifies the specification rules that produce the discharge.

The translation handles policy ambiguity explicitly. Institutional policies are written for human readers who can apply judgement to ambiguous cases; specification syntax requires the ambiguity to be resolved into concrete rules. The translation either resolves the ambiguity through institutional decision (the team decides which interpretation the specification will encode) or carries the ambiguity through institutional escalation (the specification routes ambiguous cases to human review rather than deciding them automatically). Either pattern is defensible; the architecture requires the institution to choose deliberately rather than to leave the ambiguity unresolved.

The translation produces specification rules that are testable. The specification's behavioural rules can be evaluated against test cases at deployment time and at regulatory inspection time. The architectural pattern includes a test suite alongside the specification: each rule has at least one positive test (a request that should be handled according to the rule) and at least one negative test (a request that should be refused under the rule). The test suite is the institutional artefact that demonstrates the specification's enforceability; specifications without test suites are claims rather than enforced rules.

- **Versioning and lifecycle.** The specification is a living document that evolves with the institution's policies, regulatory environment, and AI estate. The architectural pattern is semantic versioning at the institutional level: major versions reflect substantive changes to behavioural rules; minor versions reflect additions or clarifications; patch versions reflect corrections to existing rules. Each version carries an institutional approval record (who authorised the version, when, on what basis) and a change log that documents what changed from the previous version and why.

  Deployment of a new version is an institutional event with the same governance posture as any production deployment: change management approval, deployment scheduling, rollback procedures, post-deployment monitoring. Section 7's audit infrastructure captures the version-and-deployment context per request; section 12's incident response treats version transitions as named incident types if they produce behavioural changes the institution did not anticipate.

  The institutional pattern that fails most often is treating the specification as a launch-time artefact that does not require ongoing maintenance. Specifications that do not evolve become stale: institutional policies change without the specification being updated, regulatory obligations emerge without being mapped, the AI estate grows

without the new use cases being covered. The architecture requires the specification to be maintained on the same operational cadence as the institution's other governance artefacts; the institution that does not maintain the specification discovers at the next audit that the institutional accountability framework cannot demonstrate what the AI was authorised to do at the time of any particular request.

- **Extension and exception management.** The institutional specification cannot anticipate every use case. The architectural pattern includes an extension mechanism for new use cases (how does the institution add coverage for a new AI workload to the existing specification) and an exception mechanism for cases the existing specification does not handle correctly (how does the institution handle a situation where the specification's rules produce the wrong outcome for a specific case).

  The extension mechanism is an institutional process: the new use case is identified, the relevant policy and regulatory questions are identified, the new specification rules are authored and tested, the institutional approval is obtained, the new version is deployed. The extension is governed by the same institutional roles and review processes as the original authoring; the architecture treats extensions as routine specification work rather than as exceptional cases.

  The exception mechanism is institutionally harder. Where the specification produces the wrong outcome, the institution either updates the specification (if the wrong outcome is structural) or treats the case as a one-off exception (if the wrong outcome is contextual). The exception case is recorded, reviewed by the institutional approval authority, and either incorporated into the next version of the specification or maintained as an exception with explicit institutional approval. The architectural pattern is that exceptions accumulate evidence about where the specification needs revision; institutions that do not track exceptions typically discover at the next audit that the specification's behavioural rules do not match the institution's actual decisions in production.

The five components together are the institutional work the behaviour governance layer requires. The architecture supplies the specification format and the runtime that executes the specification; the institution supplies the specification content, the institutional roles that author it, the translation discipline that produces it, the versioning that maintains it, and the extension and exception processes that grow it. The architectural specification is the artefact; the institutional work is the ongoing function that produces and maintains the artefact.

The institution's scale calibrates the work. A Tier-1 financial institution with multiple regulatory regimes, large customer-facing AI estate, and significant on-premise deployment will produce a substantial specification (probably hundreds of rules) maintained by a substantial cross-functional team (probably five to ten named roles) with formal approval processes and quarterly version cadence. A mid-sized professional services firm with focused AI use cases will produce a more modest specification maintained by a smaller team with simpler approval processes. The architectural pattern is the same in both cases; the calibration of effort and process scales with the institution.

## Integration with institutional behavioural-policy disciplines

The institutional behavioural specification does not replace the institution's existing behavioural-policy disciplines. Code of conduct, customer engagement standards, regulatory compliance frameworks, risk management policies, ethics frameworks all continue to apply. The specification is the architectural artefact through which these existing disciplines become operational at the AI estate's request time.

Three integration patterns matter.

- **Translation from policy language to specification syntax.** The institution's existing policies are written in policy language calibrated to human readers. The Score specification is written in specification syntax calibrated to runtime compilation. The architectural specification names who in the institution holds the translation responsibility (typically a combination of compliance, risk management, and architectural functions) and how the translation is reviewed and approved before deployment. This is the institutional editorial process from the Executive Briefing operationalised at the behaviour layer.
- **Audit trail across the translation.** Section 7's audit infrastructure captures the chain from institutional policy through specification authorship through compilation to deployment. The institutional accountability framework can demonstrate at any point what policy each behavioural rule serves, who authorised the translation, when the version was deployed, and what runtime artefacts the version produced.
- **Refresh and retirement.** Institutional policies and regulations change, and the specification has to update accordingly. The lifecycle architecture (section 11) handles the refresh-and-retirement cadence at the behavioural specification layer.

## What the architecture requires of behaviour governance specification

Three architectural requirements extend the institutional behavioural-policy framework to handle the AI estate.

The institutional behavioural specification operates at a level of abstraction above any single runtime format and any single vendor's native behavioural mechanisms. The Score protocol is the open specification that produces this abstraction.

The specification compiles to the runtime artefacts each runtime requires and to the regulatory artefacts each framework requires. The compilation is what makes the specification portable; the specification is what gives the institution architectural authority over what the AI does.

The specification is versioned, audited, and mapped to regulatory regimes at the architectural layer. The institutional accountability framework can demonstrate what each behavioural rule serves, who authorised it, when the version was deployed, and what artefacts the version produced.

The behavioural specification supports the institutional policy framework but does not constitute it. The institution's existing policies continue to govern; the specification is the architectural artefact through which the policies become operational at the AI estate's request time.

SECTION 09

# The runtime layer

The Executive Briefing's third layer, where the system actually runs, is the runtime layer. The architecture this document specifies is portable across compliant runtimes by design; the institution selects from a category of Score-compliant runtimes rather than committing to a specific implementation. This section establishes what Score-compliant runtimes do architecturally, what the institution's selection criteria are, and how the runtime layer integrates with the rest of the architecture.

The section treats the runtime as a category rather than as a specific implementation. MultipleWorks Maestro is one Score-compliant runtime; this document does not describe Maestro's internals. The architectural specification is what the institution depends on; the runtime implementation is the institution's procurement choice within the architectural specification.

## What Score-compliant runtimes do

Score-compliant runtimes execute the institutional behavioural specification (section 8) against model endpoints at request time. The runtime is the architectural element where the specification becomes operational: the specification's behavioural rules are enforced, the knowledge access rules are applied, the regulatory mappings are tracked, the audit evidence is produced.

Six architectural responsibilities define the runtime category.

- **Specification execution.** The runtime takes the institutional Score specification, compiles it to the runtime format the active model endpoint requires, and executes the resulting skill at request time. Where the institution operates across multiple model endpoints (the multi-vendor estate empirical evidence from section 10), the runtime handles the per-vendor compilation rather than requiring the institution to maintain separate specifications per vendor.
- **Tool invocation mediation.** When the AI calls tools as part of producing a response, the runtime mediates the invocation against the institutional specification. Which tools the AI is permitted to invoke for which use cases, what scoping applies to the invocation, how the invocation is logged, all are runtime responsibilities operating against the specification.

- **Knowledge access integration.** The runtime integrates the four kinds of knowledge from the Executive Briefing (vendor, organisational, individual, session) at request time. Vendor knowledge is supplied by the model endpoint; organisational knowledge is retrieved through section 4's data architecture; individual knowledge is accessed under custodianship through section 3's identity architecture; session knowledge accumulates within the conversation and is stored according to section 11's lifecycle architecture. The runtime is the architectural element that integrates these four sources at the level of the request.
- **Audit evidence production.** The runtime produces the audit evidence section 7's observability architecture captures at the AI gateway. The evidence includes the principal-actor distinction, the specification version in effect, the regulatory mappings the specification carried, the tool invocations the runtime mediated, and the model endpoint that produced the response. Forensic reconstruction (section 12) operates on this evidence; regulatory inspection (section 13) operates on this evidence.
- **Vendor and model abstraction.** The runtime abstracts the differences between model endpoints from the institution's application layer. Multi-vendor estates produce different behavioural characteristics, different output formats, different latency profiles, different cost structures. The runtime presents a consistent interface to the application layer while handling the per-vendor differences internally. Section 12's graceful degradation patterns depend on the runtime providing this abstraction.
- **Regulatory enforcement.** The runtime is the architectural element where the regulatory mesh from section 13 is operationally enforced. The institutional specification carries regulatory mappings; the runtime applies the mappings at request time, ensuring that requests subject to specific regulatory regimes are handled according to the regime's requirements (data residency, cross-border transfer mechanics, content moderation obligations, transparency disclosures).

These six responsibilities are the architectural definition of a Score-compliant runtime. The runtime category is defined by what the runtime does rather than by which implementation the institution selects. Different implementations may execute the responsibilities differently; the architectural specification is concerned with the responsibilities and treats implementation choice as a separate institutional concern.

## Runtime selection as architectural decision

The institution's selection of a Score-compliant runtime is an architectural decision in the same sense as the other vendor-category decisions in section 10. The runtime is institutional-infrastructure-grade because the runtime is the operational layer that executes governance specification at request time; runtime failure is governance failure. The selection criteria are calibrated accordingly.

Five criteria matter.

- **Specification coverage.** The runtime supports the full Score specification rather than a subset. Where the runtime supports an extension to the specification, the institution evaluates whether the extension is portable to other compliant runtimes or whether it creates lock-in.
- **Operational availability.** The runtime meets the institution's operational availability requirements. Section 12's business continuity treatment depends on the runtime layer being available at the same level as the institution's other critical infrastructure.
- **Audit posture.** The runtime produces audit evidence at the fidelity section 7's observability architecture requires. Where the runtime's audit posture is below institutional requirements, the institution either upgrades the runtime, supplements through gateway-level audit capture, or selects a different runtime.
- **Compliance with the Score specification.** The runtime's compliance with the published Score specification is verified at deployment and audited regularly. Where the runtime's behaviour diverges from the specification, the institution treats the divergence as either a runtime bug to be fixed or a portability concern that bears on the runtime's continued use.
- **Vendor risk treatment.** Section 10's vendor risk treatment applies. The institution's TPRM framework evaluates the runtime vendor at the level appropriate to institutional-infrastructure-grade dependencies.

## Multi-runtime architectures

The architecture this document specifies does not require the institution to select a single runtime. Where the institution operates Score-compliant runtimes from multiple providers, the architectural specification is unaffected because the specification compiles to the artefacts each runtime requires.

Three multi-runtime patterns matter.

- **Workload-specific runtimes.** Different workloads may use different runtimes calibrated to workload-specific characteristics: an on-premise runtime for sensitive workloads (section 5's hybrid deployment), a cloud-hosted runtime for general productivity workloads, a specialist runtime for high-throughput inference. Each runtime executes the same institutional specification; the specification's regulatory mappings determine which runtime is appropriate for which request.
- **Geographic runtimes.** Different geographic regions may use different runtimes calibrated to regional regulatory requirements: a mainland Chinese runtime for workloads serving mainland customers, a European runtime for workloads serving EU customers, an APAC runtime for workloads serving the broader region. Section 13's regulatory mesh produces the architectural pressure for geographic runtime distribution; the institution's specification accommodates the distribution through regulatory mapping.

- **Failover runtimes.** Section 12's graceful degradation patterns may include runtime failover where the primary runtime is unavailable. The failover runtime executes the same institutional specification, ensuring behavioural continuity across the failover.

## What the architecture requires of the runtime layer

Three architectural requirements specify the runtime layer for the institution.

The runtime executes the institutional Score specification at request time, with the six runtime responsibilities (specification execution, tool invocation mediation, knowledge access integration, audit evidence production, vendor and model abstraction, regulatory enforcement) operating consistently across requests.

The runtime selection is treated as an architectural decision with operational availability, audit posture, specification coverage, compliance with Score, and vendor risk as the criteria. The runtime is institutional-infrastructure-grade.

The architecture supports multi-runtime estates where the institution's operational requirements warrant them. The institutional specification is portable across compliant runtimes; the institution's selection criteria determine which runtime serves which workload.

The runtime layer supports the institutional architectural specification but does not constitute it. The specification is the institution's architectural authority; the runtime is the institution's selection of an implementation that executes the specification according to the architectural definition.

SECTION 10

# Vendor and third-party risk

The architecture this document specifies has more vendors at more layers than most other systems the institution operates. The institutional third-party risk management framework extends to cover the AI estate, with AI-specific extensions calibrated to the categories of vendor the estate actually contains.

Item 9's empirical evidence establishes that APAC enterprises run multi-vendor AI estates as the operational norm. The dual-stack pattern of Western frontier models accessed through US-headquartered hyperscalers running alongside Chinese open-source and commercial models accessed through Alibaba Cloud and Tencent Cloud is the architecture, not a transition state. ServiceNow's Enterprise AI Maturity Index 2025 surveying 1,476 senior leaders across Australia, Hong Kong, India, and Singapore found 61% of Australian and Hong Kong enterprises layering new AI solutions on top of existing systems rather than consolidating, with only 31% of Hong Kong organisations reporting strong visibility across functions of where AI is deployed (ServiceNow 2025). Forrester's AI Adoption Across Regions 2025 report based on the State of AI Survey 2025 corroborates the spending picture: 26% of APAC firms have invested between USD 400,001 and USD 500,000 in generative AI, against 19% in North America and 17% in Europe (Forrester 2025). The vendor surface the institutional TPRM framework has to handle is structurally larger than a single-vendor architecture would produce.

## Categories of AI vendor

Five categories of vendor introduce distinct risk surfaces. The architectural specification names which vendors the institution depends on at each category and applies the appropriate institutional TPRM treatment.

- **Model providers.** The frontier model vendors and the on-premise hosting alternatives. Risk surface includes model behaviour, safety posture, deprecation cadence, and the geopolitical alignment of the provider's home jurisdiction.
- **Cloud platforms hosting AI workloads.** Conventional cloud TPRM extended with AI-specific concerns: which models the platform makes available, how the platform handles model deprecation, what governance affordances the platform's native AI tooling provides.
- **Score-compliant runtimes.** Institutional-infrastructure-grade vendor surface because the runtime is the operational layer that executes governance specification at request time.

- **Tooling vendors.** Vector stores, prompt management platforms, evaluation and observability tools, agent orchestration frameworks. Risk surface varies by category and matures at different rates.
- **Specialist AI service providers.** Skills marketplaces, fine-tuning services, specialist agents for narrow domains. Highest risk surface among the five because the services are typically less mature and operating in a faster-moving market than the other vendor categories.

The architectural specification names which vendors are approved for which use cases at each category. The conventional TPRM disciplines (financial due diligence, operational continuity, regulatory compliance, exit planning) continue to apply. The AI-specific extensions are calibrated to the vendor surface the AI estate actually contains.

## The discipline of disambiguating vendor claims

The institutional response to AI vendor procurement is calibrated to a recurring pattern across the AI security and information management spine: vendor capability and institutional discipline are out of phase. Vendor positioning collapses architectural distinctions the institution needs to maintain. Vendor capability ships ahead of published architecture. Vendor framing routinely overstates what runtime mechanisms can carry.

The institutional discipline is to design to the architectural pattern this document specifies and treat vendor variation as a procurement-due-diligence concern. The architectural specification operates at a level of abstraction above any single vendor's capability set; vendor capability is procurement input rather than architectural input. The depth behind this discipline is in the research that grounds this document and in the institutional applications of the discipline; section 15 returns to how this document supports the institutional conversations that follow from it.

## The multi-runtime portability claim

The architectural argument depends on the institution being multi-vendor in 2026 and continuing to be multi-vendor in 2028. Item 9's empirical evidence supports the 2026 position. Three forces support the 2028 forecast: regulatory pressure on data sovereignty across the mesh treated in section 13, geopolitical fragmentation of the AI vendor market, and institution-specific risk diversification.

The forecast is the load-bearing empirical claim of the architecture's vendor-risk treatment. If the institution is single-vendor through 2028, the architecture is over-engineered for its actual operational reality. The forecast is that institutions of meaningful scale in regulated APAC markets will be multi-vendor through at least 2028; the architecture is calibrated to that forecast.

Two scenarios would weaken the forecast: regulatory consolidation faster than the 2026-2028 trajectory, or vendor convergence on cross-platform standards reaching full institutional governance discharge. Neither has materialised by mid-2026. The architecture is calibrated to the empirical reality as it stands.

## Vendor incident response

Vendor incidents activate the institutional response specified in section 12. Three procurement-time disciplines support the response: contractual notification timelines that align with the institution's own regulatory obligations, independent forensics access that supports the institution's audit infrastructure, and switch-capacity provisions that operationalise multi-vendor failover within a defined timeframe.

These are procurement function work rather than architectural work. The architecture specifies that the disciplines exist; the institution's procurement function operates them.

## What the architecture requires of TPRM

Three architectural requirements extend the institutional TPRM framework to handle the AI estate.

- **Per-category risk treatment.** The five vendor categories have distinct risk surfaces. The institutional TPRM framework applies category-specific treatment rather than treating every AI vendor as occupying the same risk profile.
- **Vendor-claim verification protocol.** The procurement function verifies vendor claims of AI-aware capability against published architecture and primary-source documentation rather than against vendor positioning materials.
- **Multi-vendor failover validation.** The institutional TPRM framework includes regular validation that failover paths between vendors are operationally real. Untested failover is theoretical.

The conventional TPRM disciplines continue to apply. The AI-specific extensions are calibrated to the vendor surface the AI estate actually contains.

SECTION 11

# Information lifecycle management

The institutional information lifecycle management framework extends to handle AI assets across the four-layer taxonomy from section 4, with one structural extension that the field has not yet solved at scale: the right-to-erasure obligation under PIPL, GDPR, PDPO, and equivalent personal data regimes. The Executive Briefing's argument that AI knowledge needs an editorial process applies here as the lifecycle management of derived AI assets; the right-to-erasure problem is the surface where the institutional architecture supports the regulatory obligation rather than discharging it.

## Lifecycle at each layer

- **Layer 1 (source data).** Conventional institutional retention and deletion. The institution's existing retention schedules apply with no AI-specific extension; deletion is operationally tractable through conventional storage controls.
- **Layer 2 (embeddings).** Index-level deletion synchronised with source-data deletion. The architectural pattern is that embeddings inherit the lifecycle of their source data: when source data is deleted, the corresponding embeddings are deleted from the vector store, and the deletion is recorded in the audit infrastructure (section 7) for institutional and regulatory inspection. The synchronisation is operationally tractable but requires deliberate architectural specification because vector stores do not handle this natively across all implementations.
- **Layer 3 (conversation history and inference logs).** Conventional log retention complicated by the cross-layer reconciliation problem. Conversation history may reference source data that has been deleted, embeddings that have been removed, and personal information that the individual has requested be erased. The architectural pattern is to default to the strictest retention requirement of any component referenced in the conversation, with component-level relaxation only where the audit trail can support it. Cross-layer reconciliation (when source data is deleted, the conversation history that referenced it is updated or annotated to reflect the deletion) is the harder architectural problem; no published pattern fully addresses it in 2026.

- **Layer 4 (model weights).** Substantially unsolved at scale. Foundation model unlearning is academically contested with no production-deployable mechanism in 2026. The academic literature traces through Bourtoule et al.'s SISA framework (Bourtoule et al. 2021), the TOFU (Maini et al. 2024) and MUSE (Shi et al. 2024) benchmarks, the WMDP unlearning evaluation (Li et al. 2024), and Hayes et al.'s April 2025 paper (Hayes et al. 2025) demonstrating that current unlearning approaches do not achieve their claimed objectives. Adapter-weight unlearning reduces to retraining-on-demand, which is operationally feasible but expensive and addresses only the institution's own adapters; the foundation model remains out of reach.

## The right-to-erasure architectural problem

The right-to-erasure obligation under PIPL Article 47 (NPC 2021a), GDPR Article 17 (European Parliament and Council 2016), PDPO Section 26 (PCPD 2024), and equivalent provisions is the surface where the institutional architecture has to acknowledge a gap rather than claim a solution. Three components of the honest position.

- **Architectural answer at Layers 1 to 3.** Deletion is operationally tractable at the conventional storage layer (Layer 1), at the vector store index level (Layer 2 synchronised with Layer 1), and at the log retention layer (Layer 3 with cross-layer reconciliation as the harder sub-problem). Section 6's encryption and key management produces a deletion-equivalent for these layers through key destruction: where data is encrypted under institutional keys, destroying the keys renders the data unreadable and operates as a deletion-equivalent for regulatory purposes where the regulatory framework recognises cryptographic destruction.
- **The unsolved Layer 4 problem.** Foundation model unlearning is substantially unsolved at production scale in 2026. Adapter-weight unlearning is operationally tractable through retraining-on-demand but expensive. The institutional architecture cannot claim to discharge the right-to-erasure obligation at Layer 4 through technical mechanisms.
- **The institutional posture.** Partial discharge through architectural choices that minimise exposure rather than complete discharge through technical capabilities that do not exist. The architectural answer is data minimisation at ingestion (the Executive Briefing's argument that personal knowledge sits inside the institution under custodianship operationalises here as retrieval-time access for sensitive personal information rather than embedding-time pre-computation or fine-tuning ingestion), separation of derived assets from durable assets, and explicit acknowledgement that the compliance posture should not depend on technical capabilities that do not exist; it should rest on architectural choices that make the dependency unnecessary.

This is the surface where the architecture-supports-but-does-not-constitute axiom is most acute. The architecture supports the institutional regulatory posture by minimising the exposure that would require Layer 4 unlearning to discharge; it does not constitute compliance through Layer 4 unlearning that the field has not yet produced.

## Editorial process operationalised

The Executive Briefing argued that AI knowledge needs an editorial process: author, review, approve, publish, apply, refresh, retire. The lifecycle management architecture is where the editorial process becomes operational across the four-layer taxonomy.

Authorship records who specified the AI knowledge and when (Layer 1 source data ingestion, Layer 2 embedding-time pre-computation, Layer 4 fine-tuning). Review records the institutional approval gates the knowledge passed through. Approval records the named institutional authority that authorised the knowledge to enter production. Publication records the deployment of the knowledge to the AI estate. Application records the use of the knowledge at request time (section 7's audit infrastructure). Refresh records the institutional updates to the knowledge over time. Retirement records the deliberate retirement of knowledge that no longer serves the institution.

Without this lifecycle management, the editorial process is aspirational; with it, the institution can demonstrate at each stage that AI knowledge has been managed according to the institution's specification. The lifecycle is also where the right-to-erasure obligation is operationally tractable for the layers that admit it: deletion at Layer 1 propagates to Layer 2 through index-level synchronisation, with Layer 3 conversation history annotated through cross-layer reconciliation, with the audit trail recording each step.

## Future expansion

The right-to-erasure architectural problem warrants depth that the architecture briefing does not exhaust. The treatment in this section is calibrated to what the EA needs to specify the architecture; future MW publication will treat the problem at the depth the field has not yet consolidated. The flag is here in the briefing; the work follows when the field's understanding stabilises sufficiently to support it.

## What the architecture requires of information lifecycle management

Three architectural requirements extend the institutional information lifecycle management framework to handle the AI estate.

The lifecycle architecture specifies retention, deletion, and right-to-erasure at each layer of the four-layer taxonomy, with the unsolved Layer 4 problem named honestly rather than masked.

The Executive Briefing's editorial process is operationalised across the four-layer taxonomy through authorship, review, approval, publication, application, refresh, and retirement records that the audit infrastructure (section 7) captures.

The compliance posture should not depend on technical capabilities that do not exist; it rests on architectural choices that make the dependency unnecessary. Data minimisation at ingestion, separation of derived assets, and retrieval-time access for sensitive personal information are the architectural patterns that minimise exposure rather than discharge it.

SECTION 12

# Business continuity and incident response

The institutional business continuity and incident response framework extends to the AI estate against a territory that conventional frameworks do not directly address. NIST SP 800-61 was designed for application-infrastructure incident handling (NIST 2012). ISO 22301 and ISO/IEC 27035 were designed for application-availability and information-security incidents respectively (ISO 2019; ISO 2023). None is calibrated for AI workloads as a discrete architectural concern. The published AI-specific frameworks, NIST AI 100-2 E2025 for adversarial machine learning (NIST 2025), MITRE ATLAS for adversarial threat tactics (MITRE 2024), OWASP Top 10 for LLM Applications 2025 v2.0 for application-security risks (OWASP 2024), address threat surfaces but not BCP/IR discipline as architectural concern. Section 12 specifies how the architecture supports the institutional BCP/IR framework against the gap.

## Five categories of AI-specific incident

A defensible categorisation of AI-specific incident types in 2026 has to engage with three published taxonomies that do not align with each other. NIST AI 100-2 E2025 organises adversarial attacks by ML method, lifecycle stage, attacker goals, and capabilities. MITRE ATLAS uses the ATT&CK matrix structure of tactics and techniques. OWASP Top 10 for LLM Applications 2025 v2.0 organises ten application-security risks. The three serve different audiences and answer different questions; none is calibrated for institutional BCP/IR directly.

The architecture briefing's categorisation operates one level up from any single taxonomy. Five categories are load-bearing for institutional BCP/IR.

- **Adversarial integrity incidents.** Attacks that cause the AI system to produce incorrect output for the attacker's purpose. Includes prompt injection (OWASP LLM01:2025), training-data poisoning (NIST AI 100-2 E2025; OWASP LLM04:2025), model evasion, and indirect prompt injection through retrieved content or tool outputs. The institutional risk surface is the integrity of decisions or outputs the AI workload contributes to.
- **Adversarial confidentiality incidents.** Attacks that extract information the AI system was not intended to disclose. Includes model inversion attacks, membership inference attacks, training-data extraction, system prompt leakage (OWASP LLM07:2025), and sensitive information disclosure through generated outputs (OWASP LLM02:2025). The institutional risk surface is the confidentiality of training data, system prompts, and information accessed at inference time.

- **Adversarial availability incidents.** Attacks that degrade or deny the AI system's availability. Includes resource exhaustion through unbounded consumption (OWASP LLM10:2025), denial-of-service through prompt-engineered loops, and supply-chain compromise through poisoned dependencies (OWASP LLM03:2025). The institutional risk surface is the operational availability of the AI workload itself and of broader systems that depend on it.
- **Behavioural drift incidents.** Changes in the deployed AI system's output distribution that produce institutionally unacceptable behaviour without an external attack vector. The category does not appear as a primary heading in NIST AI 100-2 E2025, MITRE ATLAS, or OWASP LLM Top 10 because each is calibrated for adversarial or vulnerability surfaces rather than for the absence-of-attack drift surface. Includes vendor-side model retraining or fine-tuning that changes deployed-model behaviour, distribution shift in the data the model encounters at inference time, and emergent behaviours arising from the model's interaction with retrieval contexts, tool outputs, or other agents. The institutional risk surface is that the AI workload behaves differently this month from how it behaved last month, in ways the institution did not authorise and may not have detected. The briefing introduces this as a primary category where the published taxonomies have not yet codified one; the next iteration of those frameworks is likely to absorb behavioural drift, but does not yet.
- **Operational and architectural incidents.** Failures of the surrounding infrastructure that the AI workload depends on. Includes AI gateway failures, identity-provider failures, MCP server failures or compromise, vendor API outages, jurisdictional restrictions making a vendor unavailable in a region (the Anthropic API restriction in Hong Kong is the canonical 2026 example), and orchestration-layer failures. The institutional risk surface is that the AI workload's surrounding architecture fails in ways that the AI workload itself did not.

The categorisation absorbs the three published taxonomies as inputs without collapsing into any one of them. NIST AI 100-2 E2025 maps cleanly to the integrity, confidentiality, and availability categories. MITRE ATLAS techniques distribute across all four adversarial-and-operational categories. OWASP LLM Top 10 distributes across all five with most entries in the first three and a partial mapping for behavioural drift through misinformation (LLM09:2025) which OWASP frames as model-output unreliability rather than as drift over time.

## How AI BCP differs structurally from conventional application BCP

Conventional application BCP is calibrated against infrastructure failure, application failure, and operational failure. AI workloads inherit these failure modes and add structurally distinct ones. Five differences are load-bearing.

- **High vendor surface.** A conventional application typically depends on a small number of strategic vendors at architectural-component level. An AI workload depends on a larger number: foundation-model provider, fine-tuning or adapter framework, vector database, orchestration framework, MCP server collection, AI gateway, observability layer, sometimes a separate guardrails provider. The vendor-surface multiplication is structural rather than incremental.
- **Model deprecation cadence.** Foundation-model providers deprecate and replace models on cadence the institution does not control. The institution's BCP framework has to absorb model deprecation as a planned-but-non-institutional event class.
- **Behavioural variability across model versions.** When a vendor replaces a deprecated model with a successor, the successor's behaviour is similar but not identical. Vendor-native behavioural mechanisms are bound to model versions: Anthropic's constitution is internalised through training (Anthropic 2026); OpenAI's Model Spec is similarly version-bound (OpenAI 2025). Conventional regression testing does not detect behavioural variability of this kind; institutional behavioural testing against the institutional behavioural specification (section 8) does.
- **The runtime layer's role in vendor abstraction.** The runtime layer absorbs vendor differences. At continuity time, the runtime's value is vendor abstraction (substituting a different model behind the same application interface); the runtime's risk is that it adds another component that has to operate.
- **The AI gateway's role in failover.** At continuity time the gateway becomes the failover-orchestration point: the institution can route traffic from one model to another through the gateway by changing gateway configuration rather than application code. The gateway-as-failover-orchestration pattern is established architecturally across published reference architectures (AWS Multi-Provider Generative AI Gateway, TrueFoundry, Envoy AI Gateway under CNCF, Solo.io's Agentgateway, Microsoft Foundry) but is not yet codified at BCP level in any published framework.

The five differences together produce the architectural pattern of graceful degradation across runtime, gateway, and application layers. The runtime layer abstracts vendor failures into model-substitution events. The gateway layer enforces the substitution through configuration rather than code change. The application layer treats the AI workload's output as advisory rather than authoritative for high-stakes decisions, with human review

or rule-based fallback available where the AI output cannot be relied upon. The three-layer pattern is named here as MW's contribution. The case for it is empirical: institutions deploying AI workloads in 2025 and 2026 are converging on this pattern under operational pressure rather than through codified guidance. The pattern is likely to be codified in the next iteration of NIST AI 600-1 (NIST 2024) or equivalent.

## Behavioural drift detection in 2026

Behavioural drift detection is the single most underdeveloped category in the territory and the one where the architectural argument lands hardest. Three layers: the academic literature on production ML drift, the vendor offerings for AI observability, and the institutional patterns Tier-1 institutions are converging on under operational pressure.

The academic literature on production ML drift is well-developed for traditional ML where drift metrics include Population Stability Index, Kolmogorov-Smirnov tests, Jensen-Shannon divergence, and Wasserstein distance applied to feature and prediction distributions. The 2024-2026 literature extends these to LLM and generative-AI contexts but the output distribution of an LLM is over text, which makes distribution-comparison tractable only at the level of summary statistics (response length, refusal rate, sentiment, topic distribution) rather than at distribution level directly. Embedding-based drift metrics treat output text as embeddings and compare embedding distributions, which is more sensitive but introduces dependence on the embedding model itself. Behavioural drift in LLMs is partially detected by classical drift metrics applied to summary statistics and partially undetected because the relevant variation is in semantics that the metrics do not capture.

AI observability vendors in 2026 (Arize, WhyLabs, Datadog AI Monitoring, Galileo, Patronus, LangSmith, LangFuse, the AI capabilities of Splunk and Elastic) market drift detection as a feature. Vendor capability is operationally excellent for AI-team purposes (flagging shifts in response patterns, refusal rates, latency distributions) and partial for institutional-audit-evidence purposes because the metrics are calibrated for AI-team operational concerns rather than for institutional behavioural-specification compliance. The vendor capability does not, in general, detect drift relative to an institutional behavioural specification; it detects drift relative to the AI workload's own historical baseline.

The discipline gap between AI-team-grade drift detection (good for prompt iteration, calibrated for the AI team's operational concerns, tooled by AI observability vendors) and institutional-audit-grade drift evidence (defensible to regulators, calibrated for institutional behavioural-specification compliance, requiring institutional-controlled evaluation cadence) is the section's load-bearing observation. The two are different in calibration, in tooling, and in the institutional artefact each produces. Confusing them is the structural error that makes drift detection look like a solved problem when it is not.

The architectural answer is institutional behavioural drift detection structured as periodic evaluation against the institutional behavioural specification (section 8), with the specification version-controlled and evaluation results version-controlled alongside, and with the evaluation cadence defined by institutional policy rather than by vendor model-update cadence. The briefing introduces this as architectural extension at the drift-detection layer.

## AI incident reporting in 2026

Regulatory reporting requirements for AI incidents are codifying through 2025 and 2026 across multiple jurisdictions, with cross-jurisdictional reporting where incidents have multi-regulatory exposure being the institutional pain point.

The European Union. The EU AI Act Article 73 requires providers of high-risk AI systems to report serious incidents to the competent national market surveillance authority (European Parliament and Council 2024). Article 73's reporting obligations take effect 2 August 2026. The reporting timelines are tight: notification without undue delay and in any event within 15 days of becoming aware, with 10 days where a death may have been caused, and 2 days for widespread infringements or serious and irreversible disruption of critical infrastructure (European Commission 2025).

Hong Kong. The HKMA's Operational Resilience framework (OR-2 in the SPM) applies to AI workloads as a class of operational service the institution depends on (HKMA 2024). The HKMA's GenAI Sandbox programme through Sandbox++ (multi-regulator launch 5 March 2026 with SFC, IA, MPFA) is partly an institutional-readiness exercise for incident detection and response under supervisory dialogue (HKMA et al. 2026). The SFC's 12 November 2024 Circular requires licensed corporations using GenAI to report material incidents (SFC 2024).

Mainland China. The Generative AI Measures (CAC 2023), the Cybersecurity Law (with the 28 October 2025 amendments effective 1 January 2026) (NPC 2017), and the Network Data Security Management Regulations require cybersecurity incident reporting at multiple thresholds. The Provisions on Promoting and Regulating Cross-Border Data Flows (22 March 2024) shape the cross-border-data-flow incident surface (CAC 2024).

The cross-jurisdictional pain point is that an institutional AI incident with multi-regulatory exposure produces multiple reporting obligations on different timelines with different definitions of "serious", "material", or "reportable" and different content requirements. The architectural pattern that survives audit is to maintain a single internal incident-response artefact that serves as the source of truth and to derive jurisdiction-specific reports from it rather than authoring each report independently. The single-internal-artefact-with-jurisdictional-derivations pattern follows the four-corner-mesh logic from section 13 applied at the incident layer; the briefing names it here as MW's architectural contribution at the incident-reporting layer.

## Forensic reconstruction

The discipline of forensic reconstruction for AI systems in 2026 has a hard limit that institutional accountability has to acknowledge. The architecture's job is to name the limit honestly and identify what evidence the institutional accountability framework can defensibly operate on.

Three forms of evidence apply.

- **Input-output reconstruction is tractable.** With audit infrastructure operating per section 7's specification (the AI gateway as audit-capture point, OpenTelemetry AI Semantic Conventions for prompt-and-completion logging (OpenTelemetry 2025), the institutional SIEM as the regulatory-audit layer), the institution can reconstruct what input was given, what output produced, what tools called with what arguments, what retrieval contexts loaded. This is the layer most regulator scrutiny operates on.
- **Causal-chain reconstruction is partially tractable.** The institution can reconstruct what event sequence preceded an incident across the audit chain. The reconstruction is partial because the audit chain captures only what was logged, and because the chain reconstructs what happened but not why the model produced the output it did.
- **Internal-state reconstruction is substantially intractable.** What the model was thinking, the specific computational path through the model that produced a specific output, is not reconstructible from production deployments in 2026 because production-deployable model interpretability is not yet available. Academic interpretability research has produced research-grade tools that operate on small open-weights models in laboratory conditions; production-deployable interpretability for foundation models at scale is not available.

The institutional accountability framework therefore cannot operate on internal-state evidence; it has to operate on input-output evidence, causal-chain evidence, and behavioural-specification evidence. The forensic-evidence trilemma is named here as MW's contribution at the forensic layer. The institutional accountability framework should be designed against the three available forms of evidence rather than against the four imagined forms. Behavioural-specification evidence specifically is tractable only if the institution authors and version-controls a behavioural specification (section 8) against which the audit evidence can be evaluated. The institution that has not authored a behavioural specification has only the first two forms of evidence and cannot establish whether the AI system behaved as it should have, only whether it behaved as it did. This is the forensic argument for an institutional behavioural specification connecting directly to the architectural argument from section 8.

## Continuity testing

The architecture's operational properties depend on continuity testing being performed regularly. Three categories matter.

- **Failover testing.** Multi-vendor failover at the runtime layer and at the gateway layer is exercised regularly to confirm the failover paths are operationally real rather than theoretical. Section 10's vendor risk management treats failover validation as an explicit TPRM requirement.

- **Degraded-state testing.** The application layer's designed degraded state is exercised regularly to confirm what the institution intended is what the human principal experiences as the AI estate is unavailable.
- **Drift testing.** The institutional behavioural specification's test suite (section 8) is run against the production runtime regularly. The cadence depends on the institution's risk profile; quarterly is the typical baseline for institutions of meaningful scale, with vendor-event-triggered re-evaluation when a vendor announces a model update, retraining, or deprecation.

## What the architecture requires of business continuity and incident response

Six operational requirements extend the institutional BCP/IR framework to handle the AI estate.

The institutional incident-response programme recognises the five AI-specific incident categories as primary categories rather than as subcategories of existing technology-incident types: adversarial integrity, adversarial confidentiality, adversarial availability, behavioural drift, operational and architectural.

The architecture supports graceful degradation across three layers (runtime failover, gateway runtime selection, application-layer designed degraded state). The pattern is established architecturally across published reference architectures and named in this briefing at BCP level, where published frameworks have not yet codified it.

Institutional behavioural drift detection is structured as periodic evaluation against the institutional behavioural specification rather than as AI-team-grade vendor monitoring. The discipline gap between the two is the structural error that makes drift detection look like a solved problem when it is not.

Cross-jurisdictional incident reporting operates from a single internal incident-response artefact with jurisdiction-specific derivations rather than from independently authored reports per regime. The pattern follows the regulatory mesh from section 13 applied at the incident layer.

Forensic infrastructure is designed against the three available forms of evidence (input-output, causal-chain, behavioural-specification) rather than against the four imagined forms. Internal-state reconstruction is acknowledged as substantially intractable in 2026 and the institutional accountability framework operates accordingly.

Continuity testing is performed regularly across failover, degraded-state, and drift categories. The institution's risk profile calibrates the cadence.

The architecture supports the institutional BCP/IR framework but does not constitute it. The institutional accountability framework operates on the evidence the architecture supplies.

SECTION 13

# The regulatory mesh as architectural constraint

The regulated APAC enterprise operates inside a regulatory mesh, not a single framework. Five distinct regimes apply at the same time. The EU AI Act reaches extraterritorially into any system whose outputs are used in the EU (European Parliament and Council 2024). NIST's AI Risk Management Framework provides voluntary methodology that increasingly shapes US procurement and partner expectations (NIST 2024). ISO/IEC 42001 produces certifiable management-system evidence that vendor and customer due diligence runs on (ISO 2023). The HKMA and the SFC supervise Hong Kong-authorised institutions through principles-based circulars and the iterative practice of supervisory dialogue (HKMA 2024; SFC 2024). The mainland Chinese stack runs through PIPL (NPC 2021a), DSL (NPC 2021b), and CSL (NPC 2017) at the data layer, with the Algorithm Provisions, Deep Synthesis Provisions, and Generative AI Measures (CAC 2023) running on top.

None of these frameworks substitutes for any of the others. None reduces to any of the others. None is going to be displaced by a unified global regime in the timeframe the architect makes decisions over. The mesh is not a transition state. It is the present-day operating environment for every Hong Kong-headquartered bank with EU operations, every regional insurer with a mainland customer base, every multinational with employees and customers spread across jurisdictions.

The dominant response is to treat the mesh as compliance burden. Five frameworks, five tracks of work, five sets of obligations to be ticked off in parallel. The framing produces operational fatigue and architectural avoidance. It also misses what the mesh actually contains.

## What the frameworks share, and what they do not

The most common dismissive argument runs that the frameworks share a controls layer and therefore reduce to one coherent governance model. Risk identification, documentation, oversight, testing, monitoring, incident reporting all appear as named obligations in every framework. The vocabulary overlaps. The instinct is to conclude that obligations also overlap.

They do not. The frameworks share an operational vocabulary at the controls layer and no common obligation grammar at the legal-discharge layer. The grid on the next page makes the distinction visible.

## The regulatory mesh: twelve dimensions across five frameworks

*A synthesis for cross-jurisdictional architectural decisions, not a precision compliance map. Specialist readers will sharpen individual cells; the value is in the pattern.*

| Dimension | EU AI Act | NIST RMF | ISO 42001 | HKMA | Mainland |
|---|---|---|---|---|---|
| **Risk identification** | Prescriptive, tiered | Voluntary method | MS requirement | Principle-based | Filing-based |
| **Documentation** | Format-specified | Templates suggested | MS-format | Supervisor-mediated | Algorithm filing |
| **Human oversight** | Required (high-risk) | Recommended | MS requirement | Expected | Required (content) |
| **Testing and evaluation** | Mandatory method | Voluntary | MS requirement | Principle-based | Filing prerequisite |
| **Post-deployment monitoring** | Required | Recommended | MS requirement | Expected | Required |
| **User transparency** | Mandatory disclosure | Recommended | MS requirement | Sector-specific | Watermarking mandate |
| **Training data provenance** | Required (foundation) | Recommended | MS requirement | Not specified | Security review |
| **Third-party controls** | Required | Recommended | MS requirement | Expected | Provider-platform liability |
| **Incident reporting** | Required (serious) | Recommended | MS requirement | Required | Required |
| **Content moderation** | Limited scope | Out of scope | Out of scope | Out of scope | Central |
| **Algorithm filing** | Out of scope | Out of scope | Out of scope | Out of scope | Required |
| **Values and alignment** | Out of scope | Out of scope | Out of scope | Out of scope | Required |

# The regulatory mesh, continued

The grid bands into three reading layers. The top band, from risk identification through incident reporting, shows convergent dimensions diverging in mode rather than in substance. Every framework requires risk identification, documentation, oversight, testing, monitoring; each requires it in a different regulatory grammar. The EU specifies format and tier. NIST recommends method. ISO embeds the obligation in management-system structure. HKMA mediates the obligation through supervisory practice. The mainland stack absorbs it into algorithm filing and security review. The substance of the obligation is recognisable across all five; the legal mechanism by which the obligation is discharged is not.

The bottom band, content moderation through values and alignment, is jurisdictionally unique. Mainland obligations on content moderation, algorithm registry filing, and values alignment have no analogue elsewhere. These are not gaps in the EU or US frameworks. They are obligations that reflect a different conception of what AI governance is for, and they will not migrate elsewhere in any horizon the architect is currently planning over.

The middle ground is where the legal-discharge mechanics matter most operationally. An institution that reads only the EU AI Act will miss the mainland filing requirements (CAC 2024). An institution that reads only HKMA expectations will miss the EU's prescriptive transparency obligations under Article 50 (European Parliament and Council 2024). The grid renders this visible without claiming any single framework is incomplete.

## Heterogeneity as intelligence, not failure

The framing question matters more than the analytical move. Each jurisdiction's priority order is intelligence about where regulatory pressure is going.

The EU prioritised rights-based risk-tiering because the EU treats AI as a fundamental-rights question and writes regulation accordingly. The mainland prioritised content sovereignty and provenance because the state treats algorithmic services as mediated public communication. HKMA prioritised supervisor-mediated stringency because the regulatory tradition runs on principles-based prudential supervision and AI was absorbed into that tradition rather than constituted as a separate regime. NIST prioritised voluntary methodology because the US federal apparatus coordinates more easily than it legislates, and AI policy has been shaped by that constraint. ISO prioritised certifiable management discipline because the certifying-body ecosystem and the procurement infrastructure that runs on it required a standard the supply chain could audit against.

The heterogeneity reflects each jurisdiction tackling the problem from where it has authority and where it cares most. Read all five together and a richer map emerges of what AI governance actually requires than reading any one in isolation. The map is the intelligence.

## Why single-platform consolidation falls short

The cleanest counter-argument to the architectural problem is single-platform consolidation. A sufficiently capable cloud platform with comprehensive compliance certifications across the major frameworks discharges the mesh problem operationally, even where it does not resolve it conceptually. Microsoft Azure with Purview, Compliance Manager, and Defender is the canonical example; AWS and Google Cloud have equivalent stacks. The vendor-mediated discharge story is real, the certifications are real, and for many use cases the operational burden is genuinely reduced.

Four limits sit against the counter-argument.

First, supervisor-mediated stringency is hard to evidence externally regardless of platform-level certifications. A bank can demonstrate ISO 27001, ISO 42001, SOC 2 Type II, and the platform's framework-by-framework compliance map; none of this evidences the institution's standing in HKMA supervisory dialogue.

Second, mainland content moderation, algorithm filing, and values-alignment obligations cannot be discharged by EU AI Act compliance via any platform, because they require domestic legal entities, domestic accountabilities, and domestic regulatory relationships.

Third, vendor certifications cover the platform; they do not extend to the institution's compliance posture against partner and customer expectations.

Fourth, data residency, training location, inference location, and the institution's own choices about which models it allows in which contexts are decisions the institution makes regardless of platform.

Single-platform consolidation is a legitimate operational tactic for the procedural and platform-discharged portion of the burden. The strategic answer to the mesh sits elsewhere. The institution that treats vendor-mediated discharge as the strategic answer is making the architectural decision unconsciously, in the platform's favoured direction, and inheriting the platform's interpretation of how each framework's obligations should be satisfied.

## Intelligence, not burden

The architectural answer is to treat the regulatory mesh as a constraint the architecture absorbs at design time rather than as compliance overhead the institution discharges at audit time. The behaviour governance specification (section 8) carries regulatory mappings at the rule level. The runtime layer (section 9) enforces the mappings at request time. The audit infrastructure (section 7) produces the evidence each regime inspects. The data architecture (section 4) handles residency as structural property rather than as configuration.

The depth at which each framework operates and the depth at which the architecture engages each framework warrants treatment beyond what this section provides. *Reading the AI Compliance Mesh*, the standalone MultipleWorks publication, treats the territory at the depth the architecture briefing's section 13 cannot exhaust, including the four-corner case for the typical regulated APAC enterprise and the forward-looking analysis of how the mesh moves through the next three to five years.

## What the architecture requires of regulatory engagement

Three architectural requirements extend the institutional regulatory engagement framework to handle the AI estate.

The architecture treats the regulatory mesh as architectural constraint rather than as compliance overhead. Each regulatory regime's obligations are mapped to the architectural surfaces that produce the evidence the regime requires.

The behaviour governance specification (section 8) carries regulatory mappings at the rule level. The runtime layer (section 9) enforces the mappings at request time. The audit infrastructure (section 7) produces the evidence each regime inspects.

The single-platform consolidation alternative is acknowledged honestly with its four structural limits. The institution that consolidates accepts the lock-in; the architecture this document specifies preserves the institution's optionality.

SECTION 14

# Five decisions revisited

The Executive Briefing named five decisions every institution makes about its AI architecture, deliberately or by default. Each decision is treated here with the architectural and security context the architect needs to specify it in the institution's actual environment. The decisions are not separate concerns; they intersect with the architectural surfaces this document has specified across sections 3 to 13.

## Decision 1, Where the data lives

Vendor-led thinking pushes towards a single region in a single cloud. The counter-position is that an organisation operating across multiple APAC jurisdictions will be running across multiple data residency boundaries whether it plans to or not. The decision is whether the institution absorbs that complexity in design or discovers it in compliance later.

Architecturally, decision 1 is three sub-decisions rather than one.

The first is where source data and embeddings (Layers 1 and 2 of the four-layer taxonomy in section 4) physically reside. Section 4's data sovereignty treatment establishes that residency is a structural property at each layer; the institution specifies it at design time rather than discovering it at audit time.

The second is where conversation history and inference logs (Layer 3) reside. Layer 3's residency follows the strictest classification of any component referenced in the conversation, which means an apparently innocuous conversation that surfaces personal data inherits the residency obligation of that data.

The third is where derived assets reside. Layer 4 institution-specific fine-tuning adapters inherit the residency of their training corpus; the runtime's vendor and model abstraction (section 9) absorbs cross-region inference where the regulatory mesh permits it.

The institutional commitment is to specify residency per layer rather than per workload. The data architecture (section 4) and the regulatory mesh (section 13) handle the operational depth.

## Decision 2, Single vendor or multiple

The default in 2026 is to commit to a single hyperscaler's AI offering. The counter-position from item 9's empirical evidence is that institutions of meaningful scale in regulated APAC markets run multi-vendor estates as the operational norm and will continue to through at least 2028.

Architecturally, decision 2 is three layers of vendor commitment rather than one.

The first layer is the model providers. Multi-model is the architectural default; single-model is the institutional choice that has to be made deliberately rather than by default.

The second layer is the cloud platforms hosting the AI workloads. Cloud commitment is typically driven by existing institutional infrastructure rather than by AI-specific selection; the architecture extends the institution's existing cloud commitments to handle the AI estate.

The third layer is the runtime selection. Section 9's runtime category establishes that the institution selects from Score-compliant runtimes; the selection is portable across the architectural specification, which means runtime change is possible without specification rewrite.

The institutional commitment at decision 2 is to design for the empirical reality (multi-vendor through at least 2028) rather than the vendor-narrative reality (single-vendor consolidation as the trajectory). Section 10 treats the vendor risk implications.

## Decision 3, Whether on-premises is part of the picture

The default is cloud-first, on-premises as exception. The counter-position from the Executive Briefing is that in regulated industries, on-premises is the baseline for sensitive workloads, not a fallback for legacy systems. Cloud-first thinking applied to a workload the regulator expects to see on-premises produces deployments that fail their first audit.

Architecturally, decision 3 is three workload categories rather than one.

Sensitive workloads (personal data, material non-public information, regulated financial transactions) sit on-premise as the architectural baseline, with cloud augmentation only where the cloud component does not see the underlying data. Section 5's sensitive workload segmentation handles the deployment topology.

General productivity workloads (internal documentation, meeting transcription, code assistance) sit in cloud as the architectural default. Section 5's general productivity segmentation handles the topology.

Public-facing workloads (customer chatbots, member services) sit calibrated to the institution's customer-facing infrastructure. Section 5's public-facing segmentation handles the topology with the deployment depending on the specific use case.

The institutional commitment is to hybrid deployment as the architectural reality for institutions of meaningful scale rather than to a uniform cloud or on-premise posture. The architectural pattern is hybrid; the workload classification determines which deployment pattern applies.

## Decision 4, When governance is designed in

The default is to bolt governance on after the system is running. The counter-position is to design audit, policy, and traceability in from the start. The cost of retrofitting these is several multiples of the cost of designing them in. The EU AI Act in particular is unfriendly to retrofit (European Parliament and Council 2024).

Decision 4 is structurally different from the other four decisions because it is not the architect's decision to take in isolation. The architecture this document specifies has governance designed in rather than bolted on; the institution that adopts the architecture inherits the design-time governance posture. The institution that does not adopt this kind of architecture and chooses to bolt governance on later faces three structural disadvantages: the audit infrastructure has to be retrofitted against an estate not designed to produce audit evidence, the behavioural specification has to be authored against runtime configurations that were not designed for institutional specification, and the regulatory mesh has to be operationalised against an architecture that did not absorb the mesh as a constraint.

The architectural reality is that decision 4 is made by the institution's choice of architecture rather than by a separate governance decision. The architecture this document specifies makes decision 4 the deliberate choice; the alternative is to make it by default through accumulated retrofit cost.

## Decision 5, How the institution handles personal knowledge

The Executive Briefing established that personal knowledge sits inside the institution but is held under custodianship rather than ownership. The institution does not own the individual's knowledge; it holds knowledge about the individual under specific conditions and for specific purposes.

Architecturally, decision 5 differentiates three categories of personal data with different governance.

The first is personal data about customers, members, or other external parties. The institution's existing personal data protection framework (PDPO (PCPD 2024), PIPL (NPC 2021a), GDPR (European Parliament and Council 2016) depending on jurisdiction) applies. The AI-specific extensions are calibrated against the four-layer taxonomy: data minimisation at ingestion, retrieval-time access for sensitive personal information, cross-border transfer mechanics treated as structural property (section 4).

The second is personal data about employees. The institutional employment privacy framework applies. AI use cases involving employee data have specific architectural constraints around tool invocation (section 12's incident type) and behavioural specification (section 8) that the institution authors deliberately rather than inheriting from vendor positioning.

The third is personal data the AI generates about individuals through inference rather than ingestion. The AI inferred the individual's likely behaviour, preferences, or characteristics rather than being told them; the inferred data sits in Layer 3 of the data taxonomy rather than in Layer 1. Inferred personal data is the most architecturally consequential category because it does not exist until the AI produces it; the institution's data minimisation discipline applies at the inference layer rather than only at the ingestion layer.

The institutional commitment is to operationalise custodianship rather than ownership at each of the three categories. The IAM architecture (section 3) records who acted under whose authority; the data architecture (section 4) constrains where personal data lives and how it is accessed; the lifecycle architecture (section 11) handles the right-to-erasure obligation at the layers that admit it.

## Closing the five decisions

These five decisions are not made by the AI strategy. They are made by procurement, regulatory pressure, existing vendor relationships, cloud commitments, security incidents, and budget cycles. Made deliberately, they hold for years; made by default, they cost more to undo than they did to make.

The architecture this document specifies supports the deliberate making of each decision. The architectural surfaces in sections 3 to 13 are the operational consequences of the five decisions; section 15 returns to the institutional work that follows from the deliberate making of the decisions.

SECTION 15

# Closing and next conversation

The architecture this document specifies is the structural answer to a question the institution is already asking: how does enterprise AI integrate with the security and information architecture the institution has spent decades building, in a way that produces the audit evidence regulators will inspect, the operational resilience continuity demands, and the architectural authority the executive sponsor has approved?

The answer is fifteen architectural surfaces calibrated together. Identity and access architecture extending the institution's IAM through three identity categories that conventional frameworks have not previously had to handle simultaneously. Data architecture organising the AI estate's information across a four-layer taxonomy that distinguishes source data, embeddings, conversation history, and model weights. Network and deployment topology establishing the AI gateway as a layer of its own, with hybrid deployment as the architectural reality for institutions of meaningful scale in regulated APAC markets. Encryption and key management extending institutional KMS practice with conditional key release governed by workload attestation. Audit and observability infrastructure capturing the principal-actor distinction, classification context, deployment context, encryption context, and behavioural specification context at every request. Behaviour governance specification operating at a level of abstraction above any single runtime format or any single vendor's native behavioural mechanisms, compiled to the runtime artefacts each runtime requires and to the regulatory artefacts each framework requires. The runtime layer executing the specification at request time across the multi-vendor estate the institution actually operates. Vendor and third-party risk management extending the institutional TPRM framework to handle five categories of AI vendor with distinct risk surfaces. Information lifecycle management addressing retention, deletion, and the right-to-erasure obligation honestly across the four-layer taxonomy. Business continuity and incident response handling the operational failure modes and the AI-specific behavioural failure modes the institution has not previously had to prepare for. The regulatory mesh as architectural constraint, with the heterogeneity treated as intelligence the architecture acts on rather than as compliance overhead. Five decisions revisited with the architectural and security context the architect needs to specify them in their actual environment.

Each surface stands on its own; the architecture is the integration that holds them together.

## What the architecture produces

The institution that adopts the architecture this document specifies produces five operational properties the conventional alternatives do not.

The institution's AI estate is auditable at the fidelity regulators inspect, with the audit infrastructure capturing what each request did, who authorised it, what data it touched, what the regulatory mapping was, and what the runtime executed.

The institution's AI estate is portable across runtimes and vendors, with the institutional behavioural specification compiling to the artefacts each runtime requires rather than being bound to any single vendor's format.

The institution's AI estate is resilient against vendor failure, model deprecation, and operational incident, with the multi-vendor architecture and the runtime layer's vendor abstraction supporting graceful degradation rather than catastrophic failure.

The institution's AI estate operates within the regulatory mesh rather than against it, with each behavioural rule mapped to the regulatory regime it serves and the architecture compiling to the regulatory artefacts each framework requires.

The institution's AI investment compounds rather than evaporates, because the four kinds of knowledge from the Executive Briefing are governed as managed assets through the editorial process the architecture operationalises rather than treated as one-time deposits that decay silently in production.

These are design-time properties of the architecture this document specifies; runtime mechanisms and vendor capabilities do not produce them.

## The next conversation

The architecture briefing is the artefact; the institution's adoption is the work that follows. The work has three institutional conversations that run concurrently rather than sequentially.

The first conversation is between the architect and the executive sponsor. The Executive Briefing established the strategic position; this Architecture Briefing specifies the architectural pattern. The architect's work is to translate the architectural specification into the institution's actual environment: which surfaces are already in place and need extension, which surfaces are absent and need to be built, which vendors and runtimes the institution will select within the categories the architecture establishes, which institutional roles will hold the cross-functional work the behaviour governance layer requires. The conversation produces an institutional commitment to the architecture and a phased plan for adoption.

The second conversation is between the architect and the institution's existing architectural disciplines. Identity and access management, data architecture, network architecture, security architecture, audit and risk management, vendor management, business continuity, each discipline owns part of the architecture's integration surface. The architect's work is to engage each discipline in the architectural extension the AI estate requires, ensuring that the AI-specific extensions are absorbed into the discipline's existing practice rather than maintained as parallel work. The conversation produces institutional ownership distributed across the disciplines that hold each surface.

The third conversation is between the institution and its regulators. The regulatory mesh treated in section 13 produces obligations the institution has to demonstrate it discharges; the architecture this document specifies produces the artefacts the institution presents to demonstrate the discharge. The architect's work is to ensure that the architectural specification's regulatory mappings translate into the actual regulatory artefacts each regime requires, that the audit infrastructure produces the evidence the regulator will inspect, that the institutional accountability framework operates on the evidence the architecture supplies. The conversation produces regulatory confidence and the supervisory dialogue that follows.

## The implementation arc

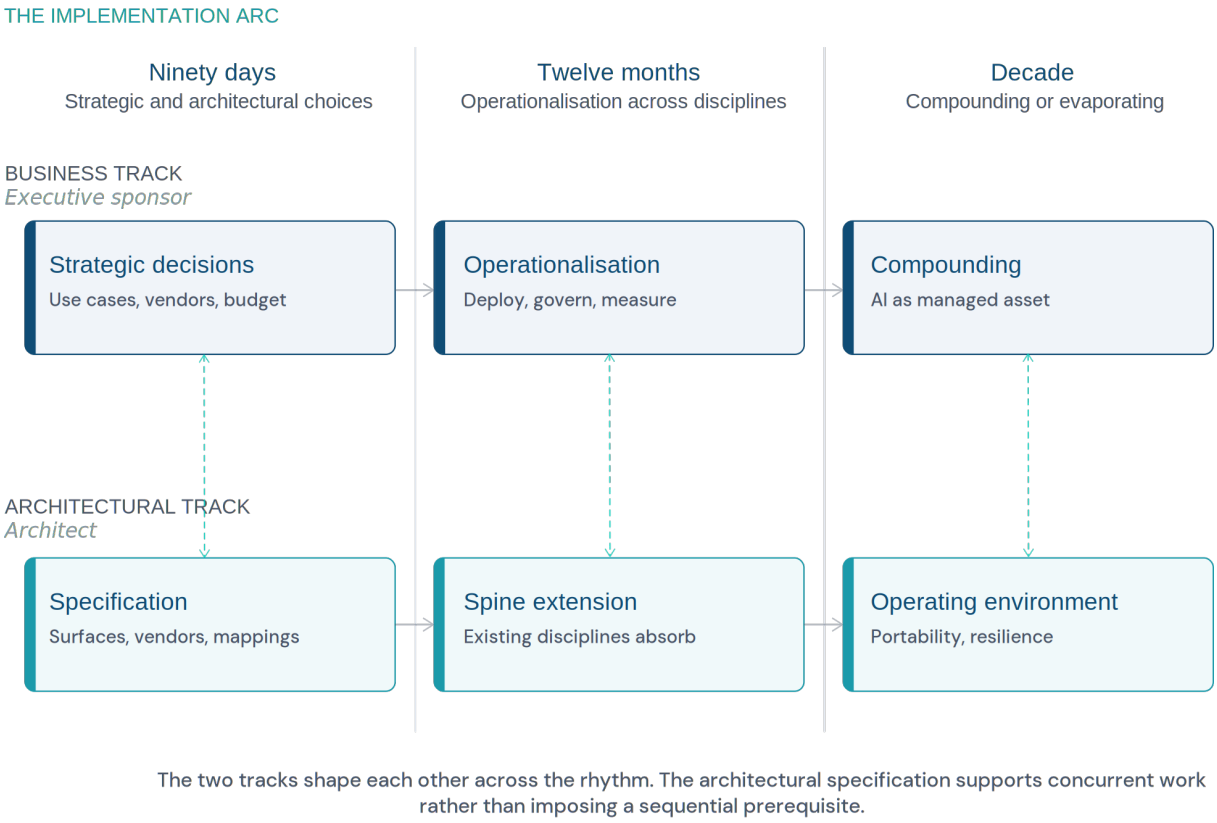



**Figure 7.** The implementation arc, with concurrent business and architectural work running across the three windows.

The Executive Briefing established the rhythm: ninety days to make the choices, twelve months to operationalise the choices into a system the institution can audit cleanly and own properly, a decade across which the investment compounds or evaporates depending on whether the choices were made deliberately. The architecture briefing's adoption operates within the same rhythm.

- **The ninety-day window.** The first ninety days run business work and architectural work concurrently, not sequentially. The executive sponsor leads the strategic decisions: which AI investments the institution is making, which use cases are in scope for the first wave, what the budget commitment is, who in the institution holds executive accountability for the AI estate. The architect leads the architectural specification work: which existing institutional architecture surfaces require extension, which vendors are candidates within the architectural categories, which regulatory regimes the institution operates under, what the high-priority use cases require at each architectural surface. These two streams shape each other across the window. The strategic decisions inform the architectural specification (the use cases in scope determine which surfaces are first-priority); the architectural specification informs the strategic decisions (the regulatory mesh and the multi-vendor estate framing affect which vendors and which deployment patterns are viable). The ninety-day output is an institutional commitment to the architecture, a vendor selection within the architectural categories, a phased adoption plan that names which surfaces are extended in which order, and an institutional approval for the first wave of operationalisation.
- **The twelve-month operationalisation.** The next twelve months operationalise the architectural specification across the institution's existing disciplines. The security and information management spine extends first: identity and access architecture absorbs the three identity categories at the AI estate's scale, the data architecture operationalises the four-layer taxonomy across the institution's actual data inventory, the network and deployment topology establishes the AI gateway and the hybrid deployment patterns, encryption and key management absorbs the AI workloads with the AI-specific extensions in place, and the audit and observability infrastructure captures AI-specific telemetry alongside the institution's existing security and operational monitoring. The behaviour governance specification authoring begins in parallel, with the cross-functional team named, the institutional approval processes operationalised, and the first version of the specification deployed. Vendor and third-party risk management, information lifecycle management, business continuity and incident response patterns each extend through the same window, with the regulatory mesh operationalised through the audit infrastructure and the regulatory mapping. By the end of the twelve months, the institution has operationalised the architecture across its existing disciplines and has the architectural surfaces in place to support the AI estate at scale.

- **The decade.** Across the decade, the architecture compounds. The institutional behavioural specification grows alongside the AI estate; the audit infrastructure produces evidence at increasing scale; the regulatory mesh evolves and the architectural mappings evolve with it. The vendor estate changes over the decade, and the architecture's portability absorbs the changes; new AI workloads enter the estate within the architectural specification rather than alongside it. The institution that has adopted the architecture deliberately compounds these properties for years; the institution that has adopted them by default discovers at each audit and each regulatory inspection that the costs of remaking the architecture have grown rather than diminished.

The implementation arc is institutional rhythm rather than architectural prescription. The architectural specification supports any institutional cadence the institution chooses to run it on; the rhythm above is the cadence the Executive Briefing identified as the realistic operating tempo for institutional AI adoption. The architect's work and the executive sponsor's work run concurrently within the rhythm, with the conversations named above shaping each other across the windows.

## Where MultipleWorks supports the work

This document is the architectural specification at the depth an EA needs to design the work; the institutional adoption depth lives in the engagements MW supports directly.

The Architecture Audit assesses an institution's existing architecture against the specification this document produces, identifying which surfaces are already adequate, which require extension, and which require novel work. The audit produces a phased adoption plan calibrated to the institution's scale, regulatory exposure, and existing maturity. The audit typically operates within the first ninety-day window as the institution moves from strategic commitment to architectural specification.

The desk review treats a specific architectural decision in depth, vendor selection, runtime selection, regulatory mapping for a particular regime, behaviour governance specification for a particular use case, at the operational depth the briefing does not exhaust. The review produces a defensible position for the specific decision the institution is making. The desk review typically operates either within the ninety-day window for foundational decisions or across the twelve-month operationalisation for surface-specific decisions.

The implementation engagements support the institutional work at each surface as the institution adopts the architecture. The work is institution-specific because the architecture is institution-specific in operationalisation; the architectural pattern is consistent across institutions, and the institutional implementation is calibrated against the institution's actual environment. The implementation engagements typically operate across the twelve-month operationalisation window, with continuing engagement across the decade as the architecture compounds.

These engagements operate at the methodology and tooling depth this document deliberately does not exhaust. The architectural pattern is in the briefing; the institutional implementation is in the engagement.

## Future expansion

Four areas of architectural work warrant depth this briefing does not exhaust. Each will appear as a future MultipleWorks publication when the field's understanding stabilises sufficiently to support it.

The right-to-erasure architectural problem (flagged in section 11). Layer 4 unlearning is substantially unsolved at production scale in 2026; the academic literature is evolving and the architectural patterns that help even where the problem is unsolved warrant treatment that this briefing does not exhaust. A future publication will treat the problem at the depth the field has not yet consolidated.

The multi-vendor architectural pattern (referenced across sections 5, 9, 10). The empirical position that APAC enterprises run multi-vendor estates as the operational norm produces architectural questions about cross-vendor governance, behaviour consistency across vendors, and fallback patterns at depth this briefing addresses at architectural pattern level. A future publication will treat the operational depth.

The on-premise enterprise AI architecture (referenced in section 5). The architectural pattern is hybrid deployment with sensitive workloads on-premise; the operational reality of running on-premise enterprise AI in 2026, specific products, hardware considerations, named enterprise case studies, the realistic adoption picture, warrants depth this briefing addresses at architectural pattern level. A future publication will treat the operational depth.

The coding AI governance pattern (flagged in the front matter scope statement). AI coding assistants used by software engineers within their development environments operate against a different threat model, identity model, and data exposure surface than the agentic AI estate this briefing addresses. The institutional governance for coding AI is a substantive territory in its own right, with its own architectural and supervisory questions. A future publication will treat the territory at the depth the agentic AI architecture deliberately does not.

These four publications join the regulatory mesh paper (referenced in section 13) and the protocol authoring layer paper (referenced in section 8) as the body of MultipleWorks work that supports this Architecture Briefing's structural arguments at the depth the standalone topics warrant.

## Closing

The architecture this document specifies is the structural answer to the institution's question about how agentic enterprise AI fits with the security and information architecture the institution has spent decades building. The architecture is sound; the institutional work to adopt it is substantial; the commercial outcomes the architecture produces (auditability, portability, resilience, regulatory confidence, compounding investment) justify the work.

The Executive Briefing argued the position (MultipleWorks 2026). This Architecture Briefing specifies the architecture. The institution's next conversation runs business work and architectural work concurrently, not sequentially, the strategic decisions the executive sponsor leads and the architectural specification the architect leads shape each other across the ninety days and the twelve months that follow. MultipleWorks supports the institutional work through the engagements named above; the architectural specification belongs to the institutions and architects who adopt it.

Made deliberately rather than by default, the architecture compounds for years. Made by default, it costs more to undo than it did to make. The institution's decision is one of the five decisions named in the Executive Briefing and revisited in section 14 of this document. The architecture is the artefact that supports the decision; the decision is the institution's.

---

**Mark Goodchild** *is Founder and Managing Director of MultipleWorks, a Hong Kong consultancy specialising in enterprise AI architecture and governance for regulated APAC enterprises. His background spans 25 years of enterprise architecture and digital transformation, including eleven years at EY, leaving as a Director in the APAC emerging tech consulting practice. Reachable at hello@multipleworks.com.hk.*



*MultipleWorks (2026).* ***AI That Knows Your Business: Architecture Briefing.*** *MultipleWorks Limited, Hong Kong. multipleworks.com.hk/briefings*
v1.0 · May 2026

# References

*References follow Chicago-style author-date convention. Online sources include access dates of verification, performed across the document's research stages from October 2025 through May 2026.*

## Regulatory primary sources


Cyberspace Administration of China. 2023. *Interim Measures for the Management of Generative Artificial Intelligence Services.* Effective 15 August 2023. Beijing: CAC. http://www.cac.gov.cn *(accessed mid-2026).*

Cyberspace Administration of China. 2024. *Provisions on Promoting and Regulating Cross-Border Data Flows.* Effective 22 March 2024. Beijing: CAC. http://www.cac.gov.cn *(accessed mid-2026).*

European Parliament and Council of the European Union. 2016. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation).* Brussels: Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2016/679/oj *(accessed mid-2026).*

European Parliament and Council of the European Union. 2024. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act).* Brussels: Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2024/1689/oj *(accessed mid-2026).*

European Commission. 2025. *Draft guidance on serious incident reporting under Article 73 of the AI Act.* Published 26 September 2025. Brussels: European Commission. Reporting requirements take effect 2 August 2026.

Hong Kong Monetary Authority. 2024. *Generative AI Sandbox programme.* Launched November 2024. First cohort report 31 October 2025; Sandbox++ multi-regulator launch 5 March 2026 with SFC, IA, MPFA. Hong Kong: HKMA. https://www.hkma.gov.hk *(accessed mid-2026).*

Hong Kong Monetary Authority. 2024. *Operational Resilience (OR-2).* Supervisory Policy Manual. Hong Kong: HKMA. https://www.hkma.gov.hk *(accessed mid-2026).*

National People's Congress of the PRC. 2017. *Cybersecurity Law of the People's Republic of China.* Effective 1 June 2017. Amendments effective 1 January 2026 (passed 28 October 2025). Beijing: NPC. *(accessed mid-2026).*

National People's Congress of the PRC. 2021. *Personal Information Protection Law of the People's Republic of China.* Effective 1 November 2021. Beijing: NPC. *(accessed mid-2026).*

National People's Congress of the PRC. 2021. *Data Security Law of the People's Republic of China.* Effective 1 September 2021. Beijing: NPC. *(accessed mid-2026).*

Office of the Privacy Commissioner for Personal Data, Hong Kong. 2024. *Personal Data Protection Ordinance and the PCPD Model Personal Data Protection Framework.* Hong Kong: PCPD. https://www.pcpd.org.hk *(accessed mid-2026).*

Securities and Futures Commission. 2024. *Circular on the Use of Generative AI Language Models.* 12 November 2024. Hong Kong: SFC. https://www.sfc.hk *(accessed mid-2026).*

## Vendor and protocol specifications

Anthropic. 2026. *Claude's constitution.* Revision of 22 January 2026. https://www.anthropic.com/news/claude-new-constitution. Licensed under CC0 1.0. *(accessed during items 15/16 verification stage).*

Anthropic. 2024. *Responsible Scaling Policy.* Current revision. https://www.anthropic.com/responsible-scaling-policy *(accessed mid-2026).*

OpenAI. 2025. *Model Spec.* Revision of 18 December 2025. https://github.com/openai/model_spec. *(accessed during items 15/16 verification stage).*

OpenAI. 2026. *Model Spec Evals.* March 2026 release. Dataset and grader implementation. https://github.com/openai/model_spec/tree/main/evals *(accessed during items 15/16 verification stage).*

OpenAI. 2024. *Preparedness Framework.* Current revision. https://openai.com/safety/preparedness *(accessed mid-2026).*

Model Context Protocol. 2025. *MCP Specification.* Revision 2025-11-25. https://modelcontextprotocol.io *(accessed during items 15/16 verification stage).*

Agent Skills. 2025. *agentskills.io specification.* https://agentskills.io and https://github.com/agentskills/agentskills *(accessed during items 15/16 verification stage).*

Microsoft. 2025a. *Microsoft Agent Framework documentation.* Including MCP integration at three architectural altitudes. https://learn.microsoft.com *(accessed during items 15/16 verification stage).*

Microsoft. 2025b. *Microsoft Entra Agent ID documentation.* https://learn.microsoft.com *(accessed during items 15/16 verification stage).*

Microsoft. 2025c. *Microsoft skill catalogues.* microsoft/skills and MicrosoftDocs/Agent-Skills repositories. https://github.com/microsoft *(accessed during items 15/16 verification stage).*

Linux Foundation. 2025. *Agentic AI Foundation charter.* https://github.com/aaif/foundation, supplemented by the Technical Committee project-intake repository and the official AAIF press page. *(accessed during items 15/16 verification stage).*

## Architectural reference standards

International Organization for Standardization. 2023. *ISO/IEC 42001:2023 Information technology, Artificial intelligence, Management system.* Geneva: ISO. *(accessed mid-2026).*

International Organization for Standardization. 2025. *ISO/IEC 42006:2025 Requirements for bodies providing audit and certification of artificial intelligence management systems.* Geneva: ISO. *(accessed mid-2026).*

International Organization for Standardization. 2019. *ISO 22301:2019 Security and resilience, Business continuity management systems, Requirements.* Geneva: ISO. *(accessed mid-2026).*

International Organization for Standardization. 2023. *ISO/IEC 27035 series, Information security incident management.* Geneva: ISO. *(accessed mid-2026).*

National Institute of Standards and Technology. 2024. *NIST AI 600-1: Artificial Intelligence Risk Management Framework: Generative AI Profile.* July 2024. Gaithersburg, MD: NIST. https://nvlpubs.nist.gov *(accessed mid-2026).*

National Institute of Standards and Technology. 2025. *NIST AI 100-2 E2025: Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations.* Authors: Apostol Vassilev, Alina Oprea, Alie Fordyce, Hyrum Anderson, Xander Davies, Maia Hamin. Published 24 March 2025; corrected PDF uploaded 1 April 2025; erratum issued 3 June 2025. https://csrc.nist.gov/pubs/ai/100/2/e2025/final *(accessed during stage 7 verification).*

National Institute of Standards and Technology. 2023. *NIST SP 800-92 Guide to Computer Security Log Management.* Gaithersburg, MD: NIST. *(accessed during stage 6 verification).*

National Institute of Standards and Technology. 2020. *NIST SP 800-207 Zero Trust Architecture.* Gaithersburg, MD: NIST. *(accessed during stage 6 verification).*

National Institute of Standards and Technology. 2019. *FIPS 140-3 Security Requirements for Cryptographic Modules.* Gaithersburg, MD: NIST. *(accessed mid-2026).*

National Institute of Standards and Technology. 2012. *NIST SP 800-61 Revision 2: Computer Security Incident Handling Guide.* Gaithersburg, MD: NIST. *(accessed mid-2026).*

Internet Engineering Task Force (IETF). 2020. *RFC 8693: OAuth 2.0 Token Exchange.* Authors: Michael Jones, Anthony Nadalin, Brian Campbell, John Bradley, Chuck Mortimore. https://tools.ietf.org/html/rfc8693 *(accessed during stage 6 verification).*

OpenLineage. 2024. *OpenLineage specification.* Open data lineage standard. https://openlineage.io *(accessed during stage 6 verification).*

OpenTelemetry. 2025. *OpenTelemetry AI Semantic Conventions.* Cloud Native Computing Foundation. https://opentelemetry.io/docs/specs/semconv *(accessed during stage 6 verification).*

MITRE Corporation. 2024. *MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems.* https://atlas.mitre.org *(accessed mid-2026).*

OWASP Foundation. 2024. *OWASP Top 10 for Large Language Model Applications 2025 v2.0.* Released 18 November 2024. https://genai.owasp.org/llm-top-10 *(accessed during stage 7 verification).*

## Industry and analyst sources

ServiceNow. 2025. *Enterprise AI Maturity Index 2025.* Survey of 4,473 senior leaders globally including 1,476 from Australia, Hong Kong, India, and Singapore. https://www.servicenow.com/standard/resource-center/white-paper/wp-enterprise-ai-maturity-index-2025.html. The 61% Australia and Hong Kong layering new AI solutions figure and the 31% Hong Kong strong visibility across functions figure verified at primary source May 2026.

HSBC Holdings plc. 2025. *HSBC and Mistral AI join forces to accelerate AI adoption across global bank.* Press release dated 1 December 2025. https://www.hsbc.com/news-and-views/news/media-releases/2025/hsbc-and-mistral-ai-join-forces-to-accelerate-ai-adoption-across-global-bank. The self-hosted-on-HSBC-internal-systems framing verified at primary source May 2026.

JPMorgan Chase. 2025. *LLM Suite generative AI platform.* Released summer 2024; reached 200,000 onboarded users within 8 months; expanded to 230,000+ globally. https://www.jpmorganchase.com/about/technology/blog/llmsuite-ab-award. Verified at primary source May 2026.

Cyberport (Hong Kong Cyberport Management Company Limited). 2024. *AI Supercomputing Centre.* First phase commenced operations 9 December 2024. 1,300 PFLOPS phase 1, scaling to 3,000 PFLOPS in 2025. HK$3 billion three-year AI Subsidy Scheme funding eligible institutions. https://www.cyberport.hk/en/development/aisc/. Verified at primary source May 2026.

Hong Kong Monetary Authority, Securities and Futures Commission, Insurance Authority, and Mandatory Provident Fund Schemes Authority. 2026. *Joint launch of Generative AI Sandbox++ initiative.* Press release dated 5 March 2026. Joint Circular SFO/IS/009/2026, HKMA/B1/15C, INS/TEC/10/48, SU/CTC/2026/001. https://www.hkma.gov.hk/eng/news-and-media/press-releases/2026/03/20260305-3/. Verified at primary source May 2026.

Forrester Research. 2025. *AI Adoption Across Regions, 2025.* Published 7 November 2025, based on the State of AI Survey 2025. The 26% APAC enterprises spending USD 400,001-500,000 on generative AI versus 19% North America and 17% Europe data point. https://www.forrester.com/blogs/apac-leads-global-ai-adoption-but-regional-strategies-diverge/. Verified at primary source May 2026.

Amazon Web Services. 2025. *Multi-Provider Generative AI Gateway reference architecture.* Built on LiteLLM open source project. Reference architecture published as AWS guidance with implementation guide reviewed for technical accuracy 2 May 2025. https://aws.amazon.com/solutions/guidance/multi-provider-generative-ai-gateway-on-aws/. Verified at primary source May 2026.

TrueFoundry. 2025. *AI Gateway reference architecture.* Enterprise-grade AI Gateway combining LLM, MCP, and Agent Gateways. NATS-based control-plane-to-gateway-pods architecture. Recognized as a Gartner representative vendor for AI Gateway in 2025. https://www.truefoundry.com/docs/platform/gateway-plane-architecture. Verified at primary source May 2026.

Cloud Native Computing Foundation. 2025. *Envoy AI Gateway.* https://github.com/envoyproxy/ai-gateway *(accessed mid-2026).*

## Academic literature

Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, et al. 2022. "Constitutional AI: Harmlessness from AI Feedback." arXiv preprint arXiv: 2212.08073. December 2022. https://arxiv.org/abs/2212.08073

Bourtoule, Lucas, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. "Machine Unlearning." *Proceedings of the 42nd IEEE Symposium on Security and Privacy*. The SISA framework. https://arxiv.org/abs/1912.03817

Hayes, Jamie, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. 2025. "Inexact Unlearning Needs More Careful Evaluations to Avoid a False Sense of Privacy." Published April 2025. https://arxiv.org/abs/2403.01218 (preprint version).

Maini, Pratyush, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. "TOFU: A Task of Fictitious Unlearning for LLMs." https://locuslab.github.io/tofu

Shi, Weijia, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, et al. 2024. "MUSE: Machine Unlearning Six-Way Evaluation for Language Models." https://muse-bench.github.io

Li, Nathaniel, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, et al. 2024. "The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning." https://www.wmdp.ai

## Companion publications

MultipleWorks. 2026. *AI That Knows Your Business: Executive Briefing.* Hong Kong: MultipleWorks Limited. https://multipleworks.com.hk/briefings