TQH

ai taxonomy

From Labels to Levers: A Functional Approach to AI Regulation

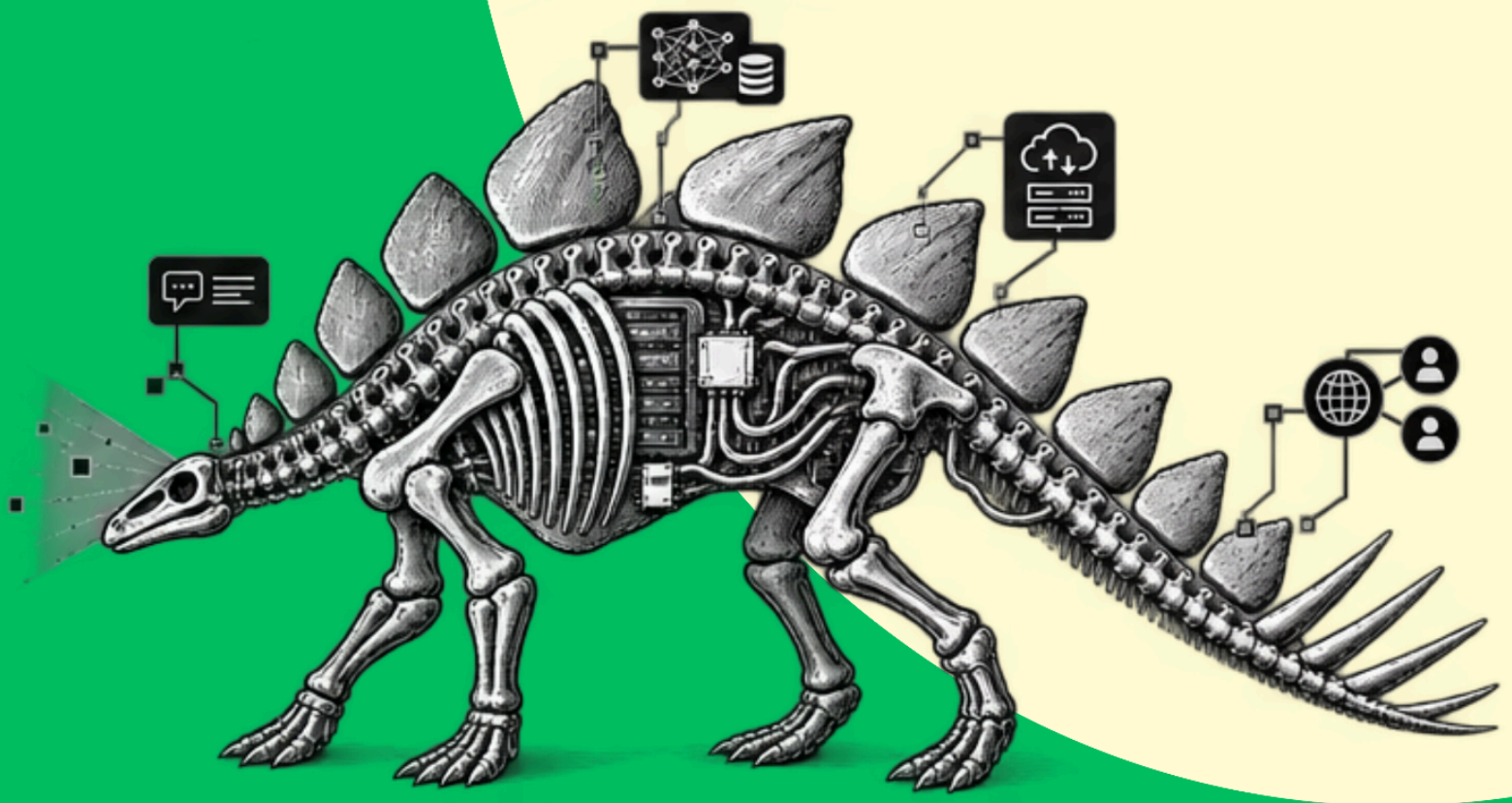


Table of Contents

Acknowledgements

This research was undertaken with inputs from the **Indian Governance and Policy Project (IGAP)** and would not have been possible without the generous funding support of **OpenAI**. We extend our sincere gratitude for their commitment to fostering informed dialogue and advancing critical discussions in the evolving landscape of AI regulation.

We are also deeply grateful to the industry stakeholders and civil society experts who generously shared their time, insights, and perspectives through interviews and written submissions to help us develop a technical understanding of AI services. Their contributions significantly enriched this research and informed our understanding of key issues relating to AI governance.

Authors and Contributors

Authors
Paavi Kulshreshth, TQH
Aparna Joshi, TQH
Tithi Neogi, TQH
Gowri Raj Varma, TQH

Research Directed and Edited by
Rohit Kumar, TQH
Sumeysh Srivastava, TQH
Dhruv Garg, Indian Governance And Policy Project (IGAP)

Other Contributors
Ananta Sharma, TQH

Design & Visuals
Zakia Rafiqi, TQH
Cover artwork created using generative AI

Disclosure and Publication Information

At The Quantum Hub (TQH), we are committed to transparency in all our work. We consistently disclose the sponsors or funders of any commissioned research. When a project is supported by a funder, we clearly identify them and acknowledge their role in shaping the research scope, providing methodological inputs, and sharing feedback. TQH retains full editorial independence and takes sole responsibility for the final research output, including its analysis and recommendations.

All insights and analysis presented in this publication are solely attributable to TQH and the named authors. AI may have been used to refine the writing in some sections of this report; however, all such content was subject to significant human review and oversight. Any errors or omissions are exclusively our own.

Published: September 2026

Suggested Citation: TQH Consulting. (2026, September). AI Taxonomy | From Labels to Levers: A Functional Approach to AI Regulation.

Approach & Methodology

This report develops a functional taxonomy of actors across the **Generative AI value chain**, examining how responsibility can be better aligned with the functions actors perform and the controls available to them. The analysis focuses on how visibility and control over the generation, use, and circulation of AI output (text, image, audio-visual forms of information) is distributed across different actors and technical layers.

The scope of this research is therefore limited to the **AI output lifecycle**. It focuses primarily on actors that enable access to Generative AI capabilities or contribute to the generation, use, visibility and distribution of AI-generated outputs. It does not seek to allocate responsibility for AI-enabled decisions, autonomous actions or other real-world outcomes. Agentic systems are considered only insofar as they illustrate how control over AI outputs may be distributed across different technical layers; the allocation of responsibility for autonomous actions taken by such systems remains outside the scope of this report.

The report draws on a combination of secondary research and qualitative primary research to develop the functional taxonomy of actors across the AI output value chain. The secondary research comprised a review of Indian and global approaches to intermediary and AI governance, emerging AI regulation, existing analyses of the AI value chain, and technical research on how capabilities and control vary across different AI actors.

The primary research consisted of **semi-structured interviews with 14 entities involved in the AI ecosystem in India**, covering actors across the AI value chain: from infrastructure and access providers, to data sourcing and validation actors, foundation model developers, model adapters and deployers, and distribution platforms. Respondents were drawn from a deliberately diverse set of sectors, including healthtech and mental health AI, medical imaging and diagnostics, data and annotation services, cloud infrastructure, telecommunications, social media and content platforms, and enterprise software. This was intended to capture how functional roles and control points vary both across the AI value chain and across different deployment contexts.

The interviews examined how actors understood their own ability to influence the generation, visibility, use and distribution of AI outputs, and helped identify where broad legal categories may not fully reflect the functions or controls available to an actor in practice.

01

Introduction

AI is transforming every sector of the economy and increasingly shaping digital interactions, prompting jurisdictions to adopt different regulatory approaches. Some have adopted standalone horizontal laws, such as the European Union's AI Act, while others have sought to address AI risks through existing digital service, consumer rights, data protection and sectoral frameworks.

India's AI Governance Guidelines, released in November 2025, endorsed an incremental, risk-based and sectorally informed approach. Rather than recommending a dedicated AI statute, the Guidelines proposed relying on existing legal frameworks, while preserving regulatory flexibility to respond to evolving AI risks and capabilities.[1] While the Guidelines did not identify an immediate need for a dedicated AI law, media reports as of July 2026 indicate that the Indian government is considering standalone risk-based AI legislation.[2] Such legislation, however, remains at an early stage of conception. Until any dedicated framework is enacted, AI-related risks are likely to continue being addressed through existing legal frameworks and sector-specific obligations.

Applying existing legal frameworks to AI raises two related questions. The first is one of classification: which legal category, if any, does an AI actor fall within? **The second is one of obligation:** is that actor actually capable of performing the obligation the framework imposes and does it have the functional role, information and technical control needed to comply? Answering these questions requires understanding what different actors actually do within an AI deployment context.

A regulatory framework that misclassifies actors or ignores the capacity question either assigns obligations to actors that lack the practical ability to comply or fails to capture those with the information and control needed to intervene.[3] India's intermediary liability framework illustrates this challenge. The 2026 amendments to the Information Technology Rules introduced obligations relating to synthetically generated information for intermediaries that offer or make available a computer resource enabling or facilitating its creation.[4] While the amendments do not expressly identify which AI actors fall within the intermediary category, they nevertheless route certain AI-related obligations through a legal framework historically organised around services that transmit, receive, store or host information on behalf of another person. Such an extension of

---

[1] Government of India. 2025. *India AI Governance Guidelines.* https://static.pib.gov.in/WriteReadData/specificdocs/documents/2025/nov/doc2025115685601.pdf.

[2] Chakraborty, Subhayan. 2026. "New AI law may focus on graded, risk-based regulations: Officials." *The Economic Times*, July 7, 2026. https://m.economictimes.com/tech/technology/new-ai-law-may-focus-on-graded-risk-based-regulations-officials/amp_articleshow/132225086.cms. — Media reporting indicates that the law is likely to be built on sectoral regulatory norms introduced by sectoral regulators as well as principles identified by the AI Governance Guidelines.

[3] Brown, Ian. 2023. *Allocating accountability in AI supply chains: a UK-centred regulatory perspective.* https://www.adalovelaceinstitute.org/wp-content/uploads/2023/06/Allocating-accountability-in-AI-supply-chains-June-2023.pdf.

[4] The Information Technology (Intermediary Guidelines and Digital Media Ethics Code) Amendment Rules, 2026 (notified on 10 February 2026 and in force from 20 February 2026) illustrate this. Rule 3(3) imposes due diligence obligations, including labelling, provenance metadata, and measures to prevent the generation of unlawful content, on "an intermediary which offers or makes available a computer resource that enables or facilitates the creation of synthetically generated information." While the amendment does not explicitly classify all AI actors as intermediaries, neither does it clearly exclude them. Although this approach is narrower than treating all AI systems as intermediaries, it still raises a broader policy concern, which is that obligations relating to synthetic content are being routed through a legal category originally designed for intermediaries handling third-party information.

obligations without first identifying which AI actors fall within the law's scope and whether they possess the functional role and control needed to comply, creates regulatory ambiguity - *see Box A: Why AI outputs strain intermediary liability frameworks.*

This challenge is not confined to intermediary liability. Similar questions arise across data protection, copyright, consumer protection, digital competition and sectoral regulation. The key issue is not simply whether an AI actor falls within an existing legal category, but whether it can effectively intervene to fulfil the obligation in a way that advances the underlying regulatory objective.

Answering that question requires risk analysis, legal classification and functional analysis to work together. Risk analysis identifies the relevant harm, its severity and context, and the regulatory response that may be required, whether prevention, disclosure, monitoring or redress. Legal classification determines which actors may fall within the scope of the framework. Functional analysis then asks which of those actors has the role, information and technical control needed to implement the obligation effectively. A high-risk AI use case, for example, does not make every actor in the AI value chain equally capable of addressing the risk. Different harms may arise from decisions made during model development, deployment or distribution, and different actors control each of those stages.[5] Yet there is currently no common framework for assessing how regulatory obligations should be allocated across the AI value chain.

This report addresses that gap by developing a functional taxonomy of the AI output value chain, where "AI output" is understood as content or other information generated by an AI system, including text, audio, visual and audio-visual material.[6] The report focuses on the AI output lifecycle because this is where most regulatory activity in India has emerged to date, particularly in relation to synthetic content. Risks may arise at different stages of the output lifecycle, from model training and architecture to user prompts, runtime retrieval, and output distribution. Accordingly, this report examines the actors and functions involved across these stages in the value chain. *Chapter 3* develops a functional taxonomy of these actors and functions, providing a structured basis for assessing whether existing or future regulatory obligations are matched to the actors best placed to implement them.

Understanding the AI Output Value Chain

AI outputs may be shaped by actors that assemble data, develop or adapt models, configure applications, operate user-facing services, issue prompts and distribute outputs.[7] Not every output involves every actor, and a single entity or service provider may perform multiple functions. Responsibility may therefore be distributed across the AI output value chain. The resulting "*many*

[5] The AI Governance Guidelines also recognize this difference, and call for classification of AI systems based on the role they play in the value chain.

[6] The report does not seek to allocate responsibility for AI-enabled decisions or real-world outcomes. Agentic systems are considered only to illustrate distributed control across technical layers; responsibility for resulting autonomous actions remains outside the scope.

[7] Arcila, Beatriz B. 2025. *AI Liability Along the Value Chain*. N.p.: Mozilla. https://sciencespo.hal.science/hal-05025891/file/AI-Liability-Along-the-Value-Chain_Beatriz-Arcila.pdf.

hands" problem does not necessarily mean that responsibility is impossible to pin. It simply means that it must be determined by the function an actor performs, rather than its corporate identity or broad legal classification.[8]

The taxonomy developed in this report addresses this issue by identifying task-specific control points. A model developer may have substantial influence over training, architecture and safety tuning but limited visibility into a downstream deployment. A deployer may configure prompts, retrieval systems and guardrails but lack access to the underlying training data or model weights. A distribution platform may have little influence over generation but substantial control over circulation. These allocations are not fixed. A vertically integrated provider may perform several functions, while locally hosted or modified models may shift control over certain functions to downstream actors.

Rather than replacing legal classification or risk analysis, the proposed taxonomy complements them by providing a structured way to assess whether a regulatory obligation is matched to the actor best placed to implement it.[9]

Here, it is important to acknowledge that availability of control points does not mean that any actor can fully determine the final AI output. Generative AI systems are inherently **probabilistic**. Even when AI actors influence training, system configuration, prompting and safeguards, they may not be able to predict or fully control every output. The proposed taxonomy therefore **does not assume that actors have deterministic control over individual outputs.** Instead, it **assesses actors' ability to influence, mitigate and respond to output-related risks**.

These principles are brought together in a 'functional attributes matrix' – see *section 3.3 of Chapter 3*. Without arguing for more or less regulation, the matrix provides a common method for testing whether existing or emerging obligations are proportionately matched to actors' roles and capacity. As jurisdictions increasingly explore differentiated obligations across the AI value chain, the next chapter reviews selected international approaches that inform this framework.

Box A: Why AI outputs strain intermediary liability frameworks

Traditional intermediary liability regimes are designed to answer a specific question: what responsibility should digital services bear when they transmit, receive, store, or host information on behalf of third parties? Historically, jurisdictions have addressed this through the "actual knowledge" principle, under which intermediaries enjoy safe harbour because they merely host or transmit third-party content without materially altering it. Once they obtain actual knowledge of illegal (or harmful) content, they may be required to remove it or restrict access to it.[10]

[8] Cobbe, Jennifer, et al. 2023. *Understanding accountability in algorithmic supply chains*. https://arxiv.org/pdf/2304.14749.

[9] "AI principles." n.d. OECD. https://www.oecd.org/en/topics/sub-issues/ai-principles.html — links accountability to an actor's role, context and ability to act.; and "OECD Framework for the Classification of AI systems." n.d. OECD. https://www.oecd.org/en/publications/oecd-framework-for-the-classification-of-ai-systems_cb6d9eca-en.html — highlights system context and lifecycle characteristics.

[10] Section 79(3)(b), IT Act, 2000; Articles 6 & 16, EU Digital Services Act

Generative AI sits uneasily within this framework because it does not ordinarily transmit or host information created by another user. Instead, users interact directly with an AI system, which generates a probabilistic response at the point of the query. Unlike traditional intermediary services, there is typically no persistent piece of user-generated content being communicated between users. This distinction is important because it changes both where the relevant risk arises and which actors are best placed to address it.

The fundamental nature of this technology also limits the effectiveness of the notice-and-takedown model developed for intermediary liability. That model assumes the existence of a persistent piece of content that can be identified, removed and prevented from further circulation. A human-generated social media post, for example, remains the same once published. Removing it eliminates that specific content from the platform. AI outputs, however, are generated afresh for each interaction. Even if a harmful response is identified and removed, the same prompt (or a variation of it) may produce a different output in the future. Addressing such risks therefore require interventions at multiple stages of the AI output lifecycle, rather than relying solely on ex post content removal.

02

Responsibility Across the Value Chain – Selected Global Approaches

Differentiating responsibilities across the AI value chain is not a new idea. Experts have long argued that regulatory obligations should be proportionate to the role an actor performs and the degree of control it exercises.[11] Jurisdictions are increasingly reflecting this principle by distinguishing actors across the AI value chain, including infrastructure providers, developers, deployers and users. They do so in different ways: through legal actor categories, risk classifications, lifecycle stages or more detailed technical guidance.

This chapter examines how selected jurisdictions approach this challenge. **Rather than reviewing individual laws in isolation, it identifies recurring design choices and discussions across Europe, Singapore, South Korea, Brazil, the United States and China.** The approaches vary in legal status, ranging from enacted legislation to regulatory guidance, and are not presented as models for replication. Instead, they provide useful points of comparison for understanding how different governance frameworks identify responsibility across increasingly complex AI value chains.

Despite significant differences in legal systems and regulatory philosophies, five broad themes emerge across the jurisdictions reviewed.

Key insight	In brief
Most frameworks distinguish between risk and actors	Risk classification is generally used to determine the level of regulatory scrutiny an AI system or use case requires, while actor classification identifies who falls within the legal framework. Neither, on its own, identifies which actor is best placed to implement a particular safeguard. For example, classifying a recruitment AI system as high risk does not, by itself, indicate whether a safeguard should be implemented by the model developer, employer or application provider.
Frameworks recognise that responsibility may shift when an actor's function changes	Where downstream actors substantially modify, fine-tune or materially change the intended use of an AI system, some frameworks reassess or differentiate responsibilities to reflect the actor's changed role.
Lifecycle models focus on when a safeguard is needed	Lifecycle models help identify the stage at which an intervention may be needed - for example, during development, deployment or monitoring. However, several

[11] Global Network Initiative. 2025. *Policy Brief on Government Interventions in AI*. https://globalnetworkinitiative.org/wp-content/uploads/GNI-Policy-Brief-on-AI-Governance-Report_v5.pdf. — Experts also argue that defining AI harms narrowly when ascertaining liability for AI actors can enable innovation and protect user rights such as freedom of expression

Key insight	In brief
	actors may participate at the same stage, making a separate assessment necessary to identify who holds the relevant technical or operational capability.
Information sharing is sought when responsibility is distributed	Many frameworks require documentation, technical information or cooperation to flow across the AI value chain. These measures help actors fulfil their responsibilities, without transferring control over the model or its outputs.
Technical guidance complements legislation by distinguishing functional roles in greater detail	While legislation often focuses on stable, administrable categories in most frameworks, guidance is used to describe technical roles more precisely. These can also be updated more easily as technologies and deployment arrangements evolve, such as in the case of emerging agentic systems.

Taken together, these approaches suggest that jurisdictions are increasingly differentiating responsibilities across the AI value chain, but they organise those responsibilities through different analytical lenses. **However, none seems to provide a systematic method for identifying which actor is functionally best placed to implement a particular regulatory obligation.** *Chapter 3* attempts to address exactly this gap by developing a functional taxonomy of the AI output value chain.

The sections that follow examine each insight in greater detail, using examples from the selected jurisdictions to show how different frameworks allocate responsibility across the AI value chain, and where a further functional assessment remains necessary.

Insight 1: Most frameworks distinguish between risk and actors

Risk classification and actor classification serve complementary but distinct purposes. Risk classification determines how much regulatory scrutiny an AI system (or use case) requires. Actor classification determines who falls within the regulatory framework. Neither, on its own, identifies which actor is functionally best placed to implement a particular safeguard.

This distinction matters because the same AI system may involve several actors performing different functions. Classifying a recruitment AI system as high risk, for example, explains why additional safeguards are needed. It does not indicate whether those safeguards should be implemented by the model developer, the application provider, the employer using the system or another actor in the value chain. Determining that requires a separate assessment of who holds the relevant technical or operational capability.

European Union

The EU AI Act clearly separates these two questions. It distinguishes legal roles from the different regulatory categories applied to AI systems and models. It defines providers, deployers, importers

and distributors, while separately regulating prohibited practices, high-risk AI systems and general-purpose AI models. The same organisation may perform more than one legal role[12], while different AI systems may attract different levels of regulatory scrutiny.

This separation provides legislation with clear points of legal responsibility, but the legal categories remain broader than the technical functions actors perform. A provider, for example, may have developed a model from scratch, commissioned it from another organisation, rebranded an existing system or substantially modified one developed elsewhere. Similarly, a deployer may simply use an off-the-shelf AI tool or may build a sophisticated application with system prompts, retrieval, guardrails and runtime controls. These actors therefore differ significantly in the control they exercise, despite falling within the same legal category.

While the Act's risk-based approach has been welcomed for moving beyond blanket regulation, commentators have also argued that its legal categories do not fully capture how responsibilities evolve across the AI value chain, particularly as systems are adapted, integrated and optimised after the base model is developed.[13]

Brazil

Brazil's Senate-approved AI Bill, PL 2338/2023, remains under consideration in a special committee of the Chamber of Deputies. It distinguishes developers, distributors and 'applicators'. An applicator is broadly an actor that uses an AI system in its own name or for its own benefit, including through activities such as configuration, maintenance, data supply or monitoring.[14] For example, a bank that integrates a third-party AI system into its loan-processing workflow, supplies operational data, configures how the system is used and monitors its outputs.

While the category usefully separates organisational use from development, it still covers different functional positions. One applicator may simply enter prompts into a third-party AI service. Another may configure retrieval systems, establish product rules, retain interaction logs and integrate the AI system into consequential business processes. Although both fall within the same legal category, they perform different functions and therefore exercise different forms of control.

Industry submissions in Brazil broadly support allocating responsibilities according to the role actually performed. They argue that developers are better placed to provide information about training and model design, while applicators understand the deployment context; they also caution that distributors may lack the information or technical expertise needed to verify

[12] Regulation (EU) 2024/1689 (EU AI Act), especially Article 3, Articles 5-6, Annex III, and Articles 16, 25, 26 and 53. https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

[13] Michael Veale and Frederik Zuiderveen Borgesius, "Demystifying the Draft EU Artificial Intelligence Act" (2021), https://doi.org/10.1016/j.clsr.2021.105634; Martin Ebers and Emmanuel Vargas Penagos, "Upstream, downstream, and in between: navigating the GPAI value chain under EU law" (2026), https://doi.org/10.1080/13600834.2026.2624923.

[14] Brazil, PL 2338/2023: Chamber of Deputies legislative status and Senate-approved text before the Chamber. https://www.camara.leg.br/proposicoesWeb/fichadetramitacao?idProposicao=2487262; https://www.camara.leg.br/proposicoesWeb/prop_mostrarintegra?codteor=2868197

compliance. The same submissions call for proportionality and clearer allocation across the value chain, reflecting concern about broad or overlapping duties.[15]

South Korea

South Korea's AI Basic Act has been in force since 22 January 2026. It distinguishes an AI development business operator from an AI utilisation business operator. In practical terms, the first develops or provides AI, while the second uses AI supplied by another actor in its own products or services. The Act also identifies 'high-impact AI' - AI used in specified areas where its operation may significantly affect rights, safety or access to essential services - and 'generative AI', which produces text, images, audio or other content in response to user input.[16]

These definitions clarify the Act's scope, but the identified categories are broad and they do not account for differences in technical control among operators. An organisation using a standard hosted AI service may have limited visibility and few configuration options, while another may build a user-facing application with prompts, safeguards, interaction logs and interface controls. Treating both as functionally equivalent risks imposing the same obligations on actors with very different capabilities. As a result, start-ups have cautioned that broad and uncertain requirements may encourage overly cautious product design, while the government has pointed to guidance and other support measures to ease implementation.[17]

Insight 2: Frameworks recognise that responsibility may shift when an actor's function changes

AI value chains are not static. A downstream actor may begin by using a third-party model and later fine-tune it, rebrand the resulting system, change its intended use, add an orchestration layer or integrate it into a new product. Several frameworks recognise that the actor's legal position may need to be reassessed when its practical role changes.

European Union

Article 25 of the EU AI Act treats a distributor, importer, deployer or other third party as the provider of a high-risk AI system in specified circumstances. These include placing the system on the market under its own name or trademark, making a substantial modification, or changing the intended purpose so that the system becomes high-risk. In such cases, the original provider must cooperate and provide the information and technical access reasonably needed for the new provider to comply with its obligations.[18]

The provision recognises that responsibility should not remain fixed upstream where a downstream actor has materially altered the system or the context in which it is used. At the same

[15] BSA | The Software Alliance, comments on Brazil PL 2338/2023 (18 October 2024), https://www.bsa.org/files/policy-filings/en10182024brazilai.pdf; Information Technology Industry Council, "Realizing Brazil's AI Ambition Through Futureproof Regulation" (2025), https://www.itic.org/news-events/techwonk-blog/realizing-brazils-ai-ambition-through-futureproof-regulation.

[16] Republic of Korea, Framework Act on the Development of Artificial Intelligence and the Creation of a Foundation for Trust, in force from 22 January 2026. https://elaw.klri.re.kr/eng_service/lawView.do?hseq=71019&lang=ENG.

[17] Reuters, "South Korea launches landmark laws to regulate AI; startups warn of compliance burdens" (22 January 2026). https://www.reuters.com/world/asia-pacific/south-korea-launches-landmark-laws-regulate-ai-startups-warn-compliance-burdens-2026-01-22/.

[18] EU AI Act, Article 25, "Responsibilities along the AI value chain". https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-25.

time, not every modification creates the same degree of control. Fine-tuning or changing model weights may add influence over model behaviour. But, system prompts, retrieval, output filters and interface design primarily shape the runtime and deployment environment. Each creates a different degree of influence over the AI output and a different risk profile. Therefore, treating every downstream actor as functionally equivalent would obscure the levers created by the modification.[19]

Singapore

Singapore's Model AI Governance Framework for Generative AI uses an ecosystem approach, drawing on the cloud industry's shared responsibility model to argue that AI governance, like cloud governance, should be distributed across actors rather than concentrated in a single entity. It discusses responsibilities across model development, application deployment, service provision and use, and recognises that closed services, open-weight models and other release arrangements create different possibilities for risk management.[20]

The framework's non-binding, iterative character makes it easier to update as technical practices evolve. That flexibility is useful for building a shared operational vocabulary, but it also means the framework depends on voluntary uptake and does not settle legal responsibility.[21]

Example: Company A connects a third-party model to a customer-support interface and adds a system prompt. Company B fine-tunes the same model, adds retrieval and deploys it through its own application. Both are downstream actors, but Company B has assumed additional functions and therefore exercises greater technical control over how the system behaves.

Across these approaches, the common principle is that **responsibility should evolve as an actor's functional role evolves**. But reclassification alone does not resolve every governance question. Downstream changes range from simple prompting to fine-tuning, orchestration and direct modification of model weights, each creating different forms of technical and operational control. A functional assessment is therefore still needed to determine which obligations should follow which changes.

Insight 3: Lifecycle models focus on when a safeguard is needed

Several jurisdictions organise AI governance around the stages through which an AI system passes, such as development, deployment, monitoring and retirement. These 'lifecycle' models are useful because they identify when a safeguard may be needed. They do not, however, identify which actor is best placed to implement it.

---

[19] Sophie Williams, Jonas Schuett and Markus Anderljung, "On Regulating Downstream AI Developers" (2025). https://doi.org/10.1017/err.2025.10020.

[20] Singapore IMDA and AI Verify Foundation, Model AI Governance Framework for Generative AI (2024). https://aiverifyfoundation.sg/wp-content/uploads/2024/05/Model-AI-Governance-Framework-for-Generative-AI-May-2024-1-1.pdf.

[21] Jason Grant Allen, Jane Loo and Jose Luna, "Governing intelligence: Singapore's evolving AI governance framework". https://doi.org/10.1017/cfl.2024.12.

South Korea

South Korea's Personal Information Protection Commission issued Guidelines on Personal Data Processing for Generative AI in August 2025. The guidance divides the lifecycle into four broad stages: setting the purpose; establishing a development or deployment strategy; AI training and development; and application and management. It also distinguishes between using a hosted model through an API, adapting an off-the-shelf model and developing a model internally, because each arrangement creates different data-processing and governance questions.[22]

These distinctions help identify *where* different privacy issues arise, but not necessarily which actor has the relevant control. At the deployment stage, for example, control may be distributed between the model developer, an orchestration provider, the application provider and a distribution platform.

Brazil

Brazil's Senate-approved AI Bill similarly links governance measures to the lifecycle phase in which an actor participates and requires cooperation across the AI value chain. This is an improvement over treating development, deployment and use as a single undifferentiated activity. It does not, however, identify which actor within a given lifecycle stage is best placed to implement a particular safeguard.[23]

Taken together, these approaches show that lifecycle models provide important context by identifying where in the AI lifecycle an intervention is required, but locating the stage is not the same as identifying the actor. Determining which actor is best placed to implement a particular safeguard therefore requires a separate functional assessment.

Insight 4: Information sharing is sought when responsibility is distributed

As responsibility is distributed across the AI value chain, actors increasingly depend on information held by others to fulfil their own obligations. A downstream integrator may need documentation about a model's capabilities and limitations. A model developer may require information about how its model is deployed. A platform may rely on provenance signals to label AI-generated content. Sharing this information makes existing responsibilities more practicable, but it does not give an actor new technical or operational capabilities.

European Union and Brazil

The EU AI Act requires cooperation where a downstream actor becomes the provider of a high-risk system. It also requires providers of general-purpose AI models to prepare and maintain information and documentation for downstream providers that integrate the model into their own systems. The General-Purpose AI Code of Practice develops documentation and information-sharing practices for providers that use it to demonstrate compliance.[24]

---

[22] Korea Personal Information Protection Commission, Guidelines on Personal Data Processing for Generative AI (August 2025). https://www.pipc.go.kr/eng/user/ltn/new/noticeDetail.do?bbsId=BBSMSTR_000000000001&nttId=2875.

[23] Brazil, PL 2338/2023, provisions linking governance measures to lifecycle participation and cooperation across the value chain. https://www.camara.leg.br/proposicoesWeb/prop_mostrarintegra?codteor=2868197.

[24] EU AI Act, Articles 25 and 53, https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng; European Commission, General-Purpose AI Code of Practice, https://digital-strategy.ec.europa.eu/en/policies/contents-code-gpai.

Brazil's Senate-approved text likewise requires actors in the AI value chain to cooperate and provide information and technical assistance needed for compliance, subject to the protection of commercial and industrial secrets.[25]

These approaches recognise that access to information is often a prerequisite for effective governance. It can make a safeguard feasible or help an actor use its existing controls more effectively. It does not, however, transfer control over the model, its runtime behaviour or the distribution of its outputs.

During the drafting of the EU General-Purpose AI Code of Practice, stakeholders highlighted concerns on the scope of information that should be shared downstream. The Business Software Alliance (BSA) argued that disclosures should be limited to the information required under the AI Act and protected against the release of trade secrets or proprietary material.[26] BEUC – The European Consumer Organisation called for the model form to include information showing how the model provider had complied with the General Data Protection Regulation (GDPR), so that downstream providers could demonstrate their own compliance.[27]

Example: A bank using a third-party foundation model may need information about the model's known limitations to design testing procedures, human review processes and user disclosures. The model developer can provide that information, enabling the bank to use its own workflow and product controls more effectively. It does not, however, give the bank the ability to retrain or modify the underlying model.

Insight 5: Technical guidance complements legislation by distinguishing functional roles in greater detail

Legislation typically relies on stable legal categories that can be consistently administered and enforced. Technical guidance can provide a more granular description of how AI systems are developed, deployed and operated in practice, and can be updated more readily as technologies and deployment arrangements evolve. While it does not carry the same legal force as legislation, it can help actors understand their roles and responsibilities more clearly.

Singapore

Singapore's Model AI Governance Framework for Agentic AI focuses on systems that can plan, use tools and take actions with a degree of autonomy. It asks organisations to define an agent's objectives and boundaries, control access to tools and data, maintain meaningful human accountability, and monitor actions and outcomes. An accompanying discussion paper examines

[25] Brazil, PL 2338/2023, cooperation and technical-assistance provisions. https://www.camara.leg.br/proposicoesWeb/prop_mostrarintegra?codteor=2868197.

[26] Business Software Alliance (BSA). 2025. *WG 1 - Transparency and copyright-related rules: Provide your feedback to the third draft of the General-Purpose AI Code of Practice.* https://www.bsa.org/files/policy-filings/03282025bsagpaidrft.pdf.

[27] BEUC: The European Consumer Organisation. 2025. *Feedback to third draft of the AI Act's Code of Practice for General-Purpose AI.* https://www.beuc.eu/sites/default/files/publications/BEUC-X-2025-028_Feedback_to_third_draft_of_the_AI_Act%E2%80%99s_Code_of_Practice_for_General-Purpose_AI.pdf.

legal responsibility across model developers, tool and platform providers, system providers, deployers, end users and affected third parties.[28] It also differentiates between users who merely interact with an agent and organisations that integrate an agent into their own workflows, recognising that the latter require additional guidance on oversight, common failure modes and over-reliance on AI systems.

The value of this guidance lies in distinguishing intermediate technical functions that broad legal categories may not separately identify. An orchestration layer that routes prompts, retrieves context, invokes tools or chains outputs together may significantly influence an agent's behaviour, even though it neither developed the underlying model nor operates the end-user interface.

The trade-off is that the framework describes good governance practices without determining legal responsibility. It provides a common operational vocabulary, while enforcement remains a matter for existing law and future legal developments.[29]

A separate consideration: Implementation capacity

Some frameworks recognise that size, resources and organisational maturity affect the burden of implementation. The EU AI Act includes support and proportionality measures for small and medium-sized enterprises and start-ups, while South Korea's Act provides support measures for smaller businesses and adopters.[30]

Capacity matters and it should certainly be factored in, but it should not be confused with control. A well-resourced application provider may still have no access to model weights or training data, while a small model developer may retain substantial influence over model design and training. Organisational capacity therefore affects how readily an actor can exercise its existing capabilities, not whether those capabilities exist in the first place.

Conclusion

Global approaches increasingly recognise that responsibility is distributed across the AI value chain, but they often leave open the question of which actor is best placed to implement a particular obligation. *Chapter 3* addresses this gap through a functional taxonomy of the AI output value chain. It focuses on where the relevant control sits and which actor can use it to address the risk. This helps identify both obligations imposed on actors that lack the capability to comply and gaps where the actor best placed to intervene has been overlooked.

[28] Singapore IMDA, Model AI Governance Framework for Agentic AI (2026), https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf; Discussion Paper on Legal Responsibility for AI Agents (2026), https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/agents-legal-responsibility.pdf.
[29] Allen, Loo and Luna, "Governing intelligence: Singapore's evolving AI governance framework". https://doi.org/10.1017/cfl.2024.12.
[30] EU AI Act, including Recital 109 and SME/start-up support measures, https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng; Republic of Korea, AI Basic Act support measures, https://elaw.klri.re.kr/eng_service/lawView.do?hseq=71019&lang=ENG.

03

Functional Taxonomy of Actors

Previous chapters have established that obligations addressing AI output risks must move beyond broad legal labels such as “intermediary”, “provider”, “deployer” or “platform”. AI risks are not created at a single point in the value chain; they are shaped across a chain of functions, including model development, data supply, runtime configuration, user interaction and downstream distribution. This chapter therefore proposes a functional taxonomy of actors across the AI output value chain.

A functional taxonomy classifies actors by the role they perform in relation to a specific AI output, rather than by the entity’s overall identity. The same entity may develop a foundation model, offer it through an API, deploy it through a consumer product and control the environment in which outputs are shared. In such cases, the taxonomy does not ask what the entity “is” in the abstract, but instead, asks which function the entity is performing at the relevant point in the AI output lifecycle.

Such a taxonomy provides a common analytical foundation for understanding where technical controls lie across the AI output value chain, and therefore where regulatory interventions are likely to be most effective. A harmful AI output can often be addressed in more than one way. For example, a deepfake could be tackled by changing how the model generates content, by preventing a particular image from being produced at runtime, or by limiting its distribution once it has been shared. Different actors control each of these interventions.

To help identify which actor is best placed to address a particular risk, this report proposes a functional attributes matrix that maps eight categories of AI actors against five functional attributes related to the AI output lifecycle. **Rather than assigning a fixed set of obligations to each actor, the matrix asks a simpler question: does the actor actually have the ability to influence the output and to what extent?** Regulatory obligations should be matched to an actor's role and the degree of control it has over the relevant stage of the AI system.

The same five attributes can be applied across most output-related risks, although their relevance will vary depending on the AI system and the task being assessed. For example, the actor best placed to improve a model's behaviour may not be the same actor best placed to remove a harmful image, investigate how it was generated, or limit its circulation. Sector-specific considerations can then be layered onto this analysis. In banking or healthcare, for instance, factors such as whether an AI output informs a consequential decision, who is affected, and whether there is meaningful human oversight may shape the overall risk assessment. These are additional contextual considerations (discussed in *section 3.4* of this chapter) and complement, rather than replace, the five functional attributes.

The following sections explain the actor categories, functional attributes and additional considerations that shape how the matrix should be read.

Box B: Why a Functional Taxonomy of AI Actors Matters for Policy Design

- **Classifies functions, not entities:** A functional taxonomy reflects the idea that AI governance measures should attach to the function an actor performs in relation to a specific AI output, not to the entity's overall identity. A single entity may develop a model, deploy it through a consumer product, and control the environment in which outputs are shared, and each function may involve different technical capabilities, points of control, and user-facing relationships.
- **Avoid reliance on broad labels:** A functional taxonomy reduces reliance on broad labels, such as "AI service provider", "intermediary" or "platform", as substitutes for a more precise assessment of technical and operational control over AI output.
- **Avoids misallocated obligations:** The taxonomy helps identify where actors have no meaningful ability, based on technical and operational controls available to them, to comply with regulatory mandates. For example, a distribution platform may be able to label, downrank, or remove deceptive AI-generated content uploaded to its service, but cannot prevent the original generation of that content on a separate AI tool.
- **Supports a more adaptable regulatory approach:** A taxonomy that follows functions rather than products remains useful as products, business models, and AI offerings evolve. It can also give service providers clearer signals about the obligations they may need to meet when integrating additional services or occupying new roles across the AI stack. As a result, governance measures that define obligations in terms of functions and the ability to manage risks, rather than specific products or technologies, are more likely to remain relevant as AI systems evolve.

3.1 Identifying actor categories across the AI output value chain

Having established why AI governance should focus on functions rather than legal labels, the next step is to identify the actors that perform distinct functional roles in the AI output value chain. The objective is to identify where meaningful points of technical and operational control exist. These points of control form the basis for assigning regulatory responsibilities throughout this report. An actor is therefore included in the taxonomy only where it performs a distinct function that can influence, observe, restrict, distribute, or otherwise respond to AI outputs.

Two principles guide the selection of actor categories.

First, the actor must perform an independent function in relation to AI outputs. This excludes actors that support the broader AI ecosystem but are too far removed from the output lifecycle to exercise meaningful control over individual outputs. For example, energy suppliers, generic hardware manufacturers, or real estate providers may be essential to AI infrastructure, but they do not determine how AI outputs are generated, observed, or distributed. By contrast, cloud providers and physical infrastructure providers occupy operational positions that, while not giving

them control over outputs themselves, provide adjacent technical and contractual levers such as account access, service availability, infrastructure security, or logging.

Second, the function must be independently identifiable. Internal teams such as trust and safety, legal, policy, compliance, or security are therefore not treated as separate actor categories. Instead, they represent organisational capabilities that support the broader functions performed by an entity, such as model safety testing by developers, runtime guardrails implemented by deployers, or content moderation by distribution platforms.

Based on stakeholder consultations and secondary research, the report identifies eight actor categories (see rows of the matrix in the pages that follow) across the AI output value chain:

1. **Physical AI infrastructure providers:** AI infrastructure providers include actors supplying physical facilities such as data centres – providing power, cooling, racks, physical security, network connectivity – without visibility into model inputs, behaviour or output at any point. This layer typically also lacks visibility on specific customer accounts or workloads, and may only be able to manage access to physical infrastructure at the entire facility or business tenant level.

2. **AI cloud providers:** These actors supply compute, storage, hosting or access infrastructure on which AI models are trained, hosted or run. Their relevance to the taxonomy does not come from control over model behaviour or individual outputs, but from adjacent levers such as data security, service availability and access control. This may include account-level enforcement, such as restricting, suspending, or terminating a customer account, or revoking or suspending API keys.

3. **Data actors:** Data actors provide or curate datasets required for AI model training, fine-tuning or evaluation. They influence the AI output lifecycle upstream, and do not usually determine final training choices or see inference-time prompts and outputs after handover. However, they may shape what the model can learn by influencing the provenance, quality, legality, representativeness or labelling of data. Data actors' degree of influence may also vary depending on the contractual or operational instructions set by developers, deployers or fine-tuners.

   An important distinction emerged during stakeholder consultations between contribution and control. A data provider may contribute a substantial proportion of a model's training data, particularly for specialised or smaller models, without exercising corresponding control over the model's behaviour. The developer ultimately determines how datasets are selected, weighted, combined, and incorporated into training. For instance, an SLM or TinyLM may derive a majority of its training data from a single curated source. However, this reflects the *scale* of a data actor's contribution, not control over model behaviour because the terms on which that data is curated, weighted, and incorporated into training are set by the developers

or fine-tuners, not the data actor itself. High data contribution, therefore, does not translate into a high degree of technical control.

4. **Foundation model developers:** Foundation model developers exercise the highest degree of influence at the model layer because they make core choices about training, architecture, safety tuning, red-teaming, model updates, release modality and access policies. However, this does not mean they control every downstream output. Their post-release visibility and leverage vary significantly depending on whether the model is offered through a closed API, released as open weights, or released as fully open source. A closed model, typically offered through developer-operated API, preserves some ongoing access control and observability, but an open-weight or open-source release can significantly reduce the developer's practical ability to restrict downstream use.

5. **Orchestration & middleware layer:**[31] This layer sits between the model and deployed application, and can route prompts, inject context, call retrieval systems, aggregate outputs, filter responses, manage memory, or select models for executing an API call in agentic systems.[32] Orchestration may be built and operated in-house by the model adapter/deployer or may be outsourced to a third-party middleware provider. When built and managed in-house it may be understood as a control lever available to adapters/deployers.[33] Only when built and managed as an independent function, it may be seen as a separate actor class (see qualifying conditions above).[34]

6. **Model adapters and deployers:** Model adapters and deployers build or operate user-facing products on top of foundation models. They may fine-tune or otherwise adopt a base model for specific use cases and/or configure the deployment environment through system prompts, retrieval systems, guardrails, user-interface design, output filters and product-level policies. This category has high runtime control, output visibility and end-user proximity because it typically sees user inputs, model outputs, interaction history and product-level logs within the deployment environment, unless the product functionality allows for incognito mode.

7. **Distribution platforms:** Distribution platforms in the AI output value chain refer to platforms that enable distribution of AI-generated content. Social media, messaging services, search engines, and similar services are included here since many AI-related harms arise not only because an output is generated, but because it is circulated, amplified or monetised. A distribution platform may not know how a synthetic image, video or text was generated, and

[31] This is limited to when the layer is operated by a third party vendor. In case this layer is part of the deployer's AI stack, then the obligations collapse into the deployer's bracket

[32] Dougherty, Jed. 2026. "What is an AI orchestration layer? Architecture, benefits, and enterprise use cases." Dataiku. https://www.dataiku.com/blog/what-is-ai-orchestration-layer.

[33] Bergmann, Krystian. 2026. "How to Manage Multiple Internal Chatbots on Azure." Netguru. https://www.netguru.com/blog/how-to-manage-multiple-internal-chatbots-on-azure. — Simpler configurations such as standalone chatbots, orchestration logic may be built directly into the application layer rather than being run by a separate middleware entity.

[34] "Middleware - Overview." n.d. LangChain Docs. https://docs.langchain.com/oss/python/langchain/middleware/overview. – A separate middleware layer may be used by more complex configurations or agentic systems to integrate work flows.

may have no control over the model, prompt or generation process. However, once the output is uploaded or shared on its service, the platform can control labelling, downranking and removal, monetisation, account-level enforcement and grievance redressal.

8. **End-users**: They enable generation through user prompts, consume the generated output, and circulate it through distribution platforms. While user provided context plays a critical role in generation and circulation of outputs, their control is bounded by the configuration of the model, application and platform they use. The taxonomy should therefore not be read to imply that end users carry the same degree of control as technical actors. Their relevant levers may be narrower, such as prompting, selecting, editing, approving, sharing and reporting AI output. The type of end-user (e.g., individual v. enterprise) may also change the kind of controls available to them — see *section 3.4*.

The actor categories identified above provide the foundation for the functional attributes matrix. The next section explains the five attributes used to assess the extent to which each actor can influence, observe, or respond to AI outputs, and how those capabilities can inform the design of proportionate regulatory obligations.

3.2 Identifying Functional Attributes

Having identified the relevant actor categories, the next step is to understand **what each actor can actually do** in relation to an AI output.

The proposed matrix does this through five **functional attributes**. These are not general characteristics of an entity or AI system. Instead, they represent the key points of technical and operational control that determine whether an actor can meaningfully influence, observe, restrict, distribute, or respond to AI outputs.

Defining these attributes helps answer a simple regulatory question: **is the proposed obligation technically and operationally feasible for the actor in question?** An obligation to monitor, modify, label or remove AI-generated content is unlikely to be effective if the actor cannot see the output, influence how it is generated, or control how it is shared. Conversely, where an actor has meaningful control, the matrix helps identify the types of interventions it can reasonably be expected to undertake.

The proposed matrix maps actors across five functional attributes:

1. **Model-layer influence represents control over model behaviour**. Typically, model developers have the highest level of control over this attribute and their points of control include defining architecture choices, selecting training data, ensuring safety tuning and red-teaming, and defining the terms for downstream access.

2. **Runtime/inference-layer control determines if the actor can influence the specific output being generated**. Typically, model adapters and deployers operating at the application layer have higher levels of control here; with points of control including system prompts, guardrails, retrieval systems, prompt routing, context injection, tool access, output filtering, and post-processing. This attribute is important because the same base model can produce different outputs depending on how it is configured and deployed at runtime.[35]

3. **Output visibility refers to whether and how an actor can see, log, audit or evaluate what is generated.** Many regulatory measures assume that an actor can observe AI outputs. In practice, however, visibility varies considerably. Some actors may have complete access to user and system prompts as well as output at the point of generation. Others may only be able to see metadata, redacted logs, aggregated abuse signals, or have no meaningful visibility at all. Accordingly, an actor's ability to process AI output does not always translate to its ability to retain or inspect it. This distinction matters because an actor cannot reasonably be expected to monitor, audit or investigate outputs it cannot access.

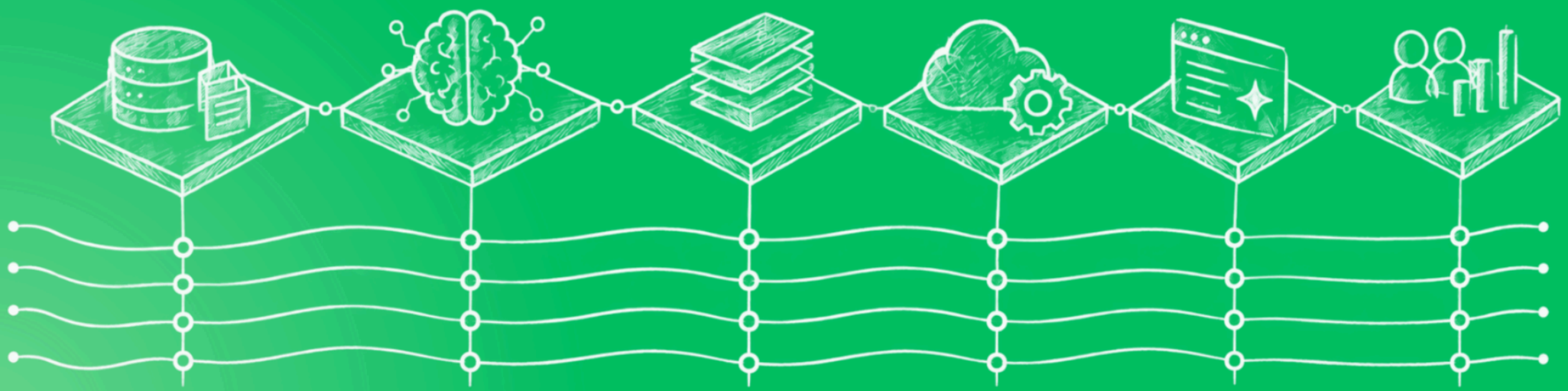
4. **Distribution control refers to whether an actor can affect how an AI output is circulated after generation.** Generation and distribution are distinct stages of the AI output lifecycle. Once an output leaves the generation environment, responsibility for its circulation may shift to entirely different actors. Generative AI outputs, such as synthetic images and videos, are commonly hosted on and widely circulated across distribution platforms such as social media. Typically, the highest level of control then lies with distribution platforms, and not generation tools/applications despite them having output visibility. Points of control available to distribution platforms typically include content moderation, takedowns, downranking and labelling.

5. **End-user proximity determines whether an actor has a direct interface with the end-user interacting with the output.** This attribute is relevant for guiding obligations that assume a direct relationship with end-users, such as notices and warnings, disclosures, grievance mechanisms, consent flows, or publication of terms of service.

Mapping actors across these five attributes provides a structured way to assess which interventions are technically and operationally feasible for different actors in the AI output value chain. It also allows the matrix to identify an **indicative set of key levers** available to each actor category, helping policymakers design obligations that are better aligned with the points of control that exist in practice.

[35] Neumann, Anna. 2025. *Position is Power: System Prompts as a Mechanism of Bias in Large Language Models (LLMs).* https://arxiv.org/pdf/2505.21091.

Functional Attributes Matrix



3.3 Functional Attributes Matrix

Note: *This taxonomy classifies AI actors based on their functional roles in the output value chain. It does not classify companies or products, which often perform multiple functions through different products and services (e.g. OpenAI is a foundation model developer and also a model deployer through ChatGPT; Sarvam develops foundation models and deploys them through Indus; the xAI/SpaceX group spans model development and deployment through Grok as well as distribution through X)*

Actor category	Description	Model-layer influence	Runtime/inference-layer control	Output visibility	Distribution control	End-user proximity	Key levers
Physical infrastructure providers *e.g. colocation facility operators, data centre operators*	*Owns and operates the physical facilities within which cloud and compute infrastructure is housed.*	**None**	**None**	**None**	**None**	**None**	Block access to physical infrastructure.
Cloud providers *e.g. hyperscale cloud platforms, managed cloud service providers*	*Provides cloud compute, storage, and networking on which models run.*	**None**	**None**	**None** *They may see technical metadata but not the output.*	**None** *No control over where output is later circulated.*	**None** *No direct interaction with the end-user.*	Data security and controls against unauthorised alteration or corruption; service availability and access control.
Data actors *e.g. data vendors, annotation firms, scrapers*	*Supplies training, fine-tuning, or evaluation data to model developers.*	**Low** *Shape what model can learn; final training choices rest with the developer.*	**None** *No visibility into inference-time prompts or outputs post-handover.*	**None** *Once the data is handed over/licensed, they do not see later prompts or outputs.*	**None** *No control over where output is later circulated.*	**None** *No direct interaction with the end-user.*	Consent and purpose limitation for personal data; licensing terms; data quality standards.
Foundation model developers *e.g. closed/ proprietary model developers, open-weight model developers*	*Builds and trains the base model; sets architecture, safety tuning, and API policies.*	**High** *Control training, safety tuning, red teaming, model updates, usage restrictions. Terms of Service set restrictions for downstream actors.* *Control is not absolute post-release: open-weight model releases significantly reduce leverage. Fully open-source releases extinguish post-release model-layer control entirely.*	**Contextual** *Control exists, but is structurally limited by the developer's inability to observe the full context (prompts, retrieved data, user instructions) that shapes what the model generates.* *For open-weight and open-source models, inference-layer control is absent entirely.*	**Contextual** *Closed models: can access API logs, safety flags, and usage metadata, but not the deployer's system prompt or full input.* *Open models: observability is effectively absent.*	**None** *Cannot control what users do with output.*	**Contextual** *Direct if offering their own consumer product.* *Indirect if API-only. For API-only offerings, the deployer typically holds the direct end-user relationship.* *Open models: None*	Safety and risk-mitigation measures at the development stage; transparency and disclosure (including model documentation); generation-side technical marking to identify AI-generated output; usage-restriction through terms of service.

Actor category	Description	Model-layer influence	Runtime/inference-layer control	Output visibility	Distribution control	End-user proximity	Key levers
		In case of open-weight or open-source models, post-release model-layer control transfers to downstream actors.					
Orchestration & middleware layer[36] *e.g., third-party agent frameworks, RAG/retrieval orchestration services,*[37] *API aggregation services*	*Sits between model and deployer; routes prompts, aggregates outputs, manages context windows. Often sold as a service to deployers.*	**Low** *Does not participate in training, safety tuning, or model updates. May exercise indirect influence through fine-tuning or adapter layers, but this is capability-dependent rather than structurally inherent to this role.*	**Contextual** *Meaningful control through prompt construction, context injection, retrieval augmentation, and output filtering or transformation before it reaches the end user.* *In agentic configurations, this control is significantly amplified - determines what actions are taken, what tools are called, and how model outputs are chained across steps.*	**High** *Sits directly in the execution path between the model and the end-user application. Can intercept, evaluate, and label outputs against predefined rules before they reach downstream actors.* *However, visibility may be partial in complex multi-model or parallel pipeline architectures.*	**None** *Does not control downstream distribution.*	**None** *Operates in the background of the application stack with no direct end-user interface.*	Input validation and prompt filtering; output screening and post-processing controls; audit logging of routing and inference decisions; traceability and record-keeping.
Model adapters & deployers *e.g. enterprise SaaS apps, consumer AI products built on APIs*	*Builds user-facing product on top of a foundation model; configures system prompts, guardrails, and UX.*	**Low** *Does not participate in base model training. In some configurations, may exercise limited model-layer influence through fine-tuning or domain-specific retrieval pipelines that shape how the model responds within a particular deployment context.*	**High** *Primary actor at the inference layer - configures system prompts, sets guardrails. Effectively defines the operational context within which the model generates output.*	**High** *Typically has the broadest observability of any actor in the value chain - sees user inputs, model outputs, full interaction history, account data, and product-level logs within their deployment environment.* *Output visibility may be reduced where privacy-preserving features (e.g. incognito modes, ephemeral interactions) are offered.*	**None** *Exercises no control over output once it is generated and leaves their deployment context.*	**High** *Users interact directly with their product.*	Terms of service and acceptable use policies; system guardrails; output filtering and screening; labelling and provenance; grievance redressal mechanisms; data collection and processing controls; sector-specific output restrictions (not applicable for general purpose chatbots).

[36] This is limited to when the layer is operated by a third party vendor. In case this layer is part of the deployer's AI stack, then the obligations collapse into the deployer's bracket.

[37] Retrieval-Augmented Generation (RAG) refers to an AI system architecture in which relevant information is retrieved from external data sources and provided as context to a generative AI model to inform its output.

Actor category	Description	Model-layer influence	Runtime/inference-layer control	Output visibility	Distribution control	End-user proximity	Key levers
Distribution platforms *e.g. social media platforms, messaging services etc,*	*Does not generate AI output; hosts or amplifies it after users post or share it.*	**None**	**None**	**None** *They usually do not see how the AI content was generated. They only see it once someone uploads, posts or shares it on their platform.*	**High** *Can moderate, label, remove, downrank, demonetise on-platform. Cannot control off-platform sharing.*	**High** *Users interact directly - to post, view, share, or amplify other outputs.*	Content and account moderation; removal mechanisms; synthetic content labelling and disclosure; algorithmic ranking and demonetisation controls; grievance redressal mechanisms. Distribution platforms may only be able to label AI-generated content where reliable provenance signals are available. In the absence of standardised, tamper-resistant content credentials embedded at the generation or deployment stage, the labelling lever may be structurally limited.
End users *Individuals or organisations using AI tools*	*Interacts with the AI tool; generates prompts; receives and may distribute outputs*	**None**	**Contextual** *End users shape inference-layer outputs through the context they supply - the specificity, framing, and sequencing of prompts. The degree of influence varies - bounded by deployer configuration in standard interactions, but potentially substantial in contexts involving deliberate prompt engineering.*	**High** *Full visibility over their own prompts and outputs.*	**High** *Decides whether, where, and how to share or repurpose content.*	**N/A**	Terms of service compliance; responsible use and content sharing practices; disclosure in high-risk or regulated contexts such as electoral speech.

3.4 Additional Considerations

The functional attributes matrix provides a common framework for identifying where technical and operational control lies across the AI output value chain. However, the same actor may not exercise the same degree of control in every situation. The extent to which an actor can influence, observe or respond to AI outputs depends on several contextual factors.

This section highlights additional considerations that policymakers should account for when applying the matrix in practice. These considerations do not replace the functional attributes. Rather, they help explain how those attributes may vary across different AI systems, deployment environments and emerging technological configurations.

1. **Model release choices affect post-release visibility and control:** Model developers have multiple training-stage controls, such as selecting datasets and safety tuning, that influence model behaviour and tendencies. While these remain consistent, developers' model release choices can significantly modify their post-release output visibility and control levers.

    a. **Closed model releases** generally allow developers to retain the greatest degree of post-release visibility and control. Because inference continues to occur within the infrastructure they operate, developers may retain access to API logs, usage metadata, abuse signals, access controls and account-level enforcement mechanisms.

    In practice, however, this visibility is not uniform. When access to foundation models is provided through API platforms or as hosted models in enterprise contexts, model developers may enter into contractual arrangements with downstream application-layer customers that limit what content they retain. For instance, under "modified" abuse monitoring agreements, developers exclude customer content from abuse monitoring logs across all API endpoints.[38] Under Zero Data Retention (ZDR) agreements, the developer does not retain customer content at all beyond the moment it's needed to generate the response.[39] This disabled storage function can then omit storage-adjacent features such as conversation history, replay, and caching. From a risk mitigation perspective, this matters because retention controls affect whether harm can be detected after the fact through logs or human review.

    Opting into ZDR removes the developer's ability to retain or review outputs after they are generated, without necessarily affecting the safeguards that operate

[38] "Foundry Models sold by Azure abuse monitoring - Microsoft Foundry." n.d. Microsoft Learn. https://learn.microsoft.com/en-us/azure/foundry/openai/concepts/abuse-monitoring.
[39] "Offering Zero Data Retention for frontier models." 2026. OpenAI. https://openai.com/index/offering-zero-data-retention-for-frontier-models/.

during inference. As a result, while a developer can still train the model to reject problematic requests at the point of generation, it may no longer be able to meet obligations that require proactive monitoring, auditing or post-hoc assessment of hosted model outputs.

b. **Open-weight and open-source releases** can remove this leverage almost entirely because downstream actors can independently host, modify, or redistribute the model outside the developer's operational environment.[40] For open-weight releases, developers may attempt to retain some residual control through licence terms - restricting commercial use or requiring attribution. However, fully open-source releases, that may include open training code and architecture, can enable deeper modification and make such restrictions harder to enforce. From a risk mitigation perspective, in both cases, the developer's primary point of leverage is the pre-release stage. See *Box C* to understand the obligation-control mismatch that emerges when this reduced control vector is not accounted for.

   That said, open-source releases can also enhance transparency and external inspectability. For instance, when developers release useful artefacts such as model codes, checkpoints and training or evaluation information, researchers and evaluators can examine model behaviour more closely. This does not give the original developer greater post-release control, but it may improve third-party auditability and ecosystem-level understanding of model risks.[41]

Box C: The Obligation-Control Mismatch: Downstream Monitoring and Open-Weight Models

Monitoring for misuse, identifying emerging risks and responding to incidents may require continuing visibility into how a model is being used. However, model developers' ability to meaningfully fulfil these obligations depends on how the model is made available.

For closed models, developers retain API-level visibility and can act on identified risks by modifying safeguards, restricting capabilities, or suspending access - subject to what contractual arrangements with downstream deployers permit. For open-weight models, this visibility is absent once weights are downloaded and run externally: the developer cannot detect misuse, modify safeguards, or intervene in the deployed system.

[40] "Llama 3: Community License Agreement." n.d. Meta. https://developer.meta.com/ai/llama3/license/; Kapoor, Sayash. n.d. "On the Societal Impact of Open Foundation Models." Stanford. https://crfm.stanford.edu/open-fms/; Open source initiative. n.d. "Preamble: Why we need Open Source Artificial Intelligence (AI)." open source initiative. https://opensource.org/ai/open-source-ai-definition. — For open-weight releases, developers may attempt to retain some residual control through licence terms — restricting commercial use or requiring attribution. Once weights are publicly distributed, however, practical enforcement is limited. Fully open-source releases, which may include training code and architecture, enable deeper modification and make such restrictions harder to enforce. In both cases, the developer's primary point of leverage is the pre-release stage.

[41] Biderman, Stella. 2023. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. https://arxiv.org/pdf/2304.01373. – EleutherAI's Pythia released intermediate checkpoints that enabled researchers to study how model behaviour changed during training, including in relation to memorisation and bias formation.

The mitigation gap follows the same pattern. Closed model developers can update the centrally hosted model and make those changes available to downstream users. Open-weight developers can release updated weights and guidance, but cannot require downstream actors to adopt them.

An obligation requiring developers to provide mitigations is therefore operationally feasible for both categories of model release. However, requiring developers to ensure remediation across all deployed instances would be feasible for closed models but face a structural constraint for open-weight releases that no compliance mechanism can fully resolve.

2. **Deployment context determines risk exposure:** Adapters/deployers of AI systems integrate sector-specific workflows, retrieval systems, guardrails, tool access and design user-interfaces based on the context in which AI output is generated.[42,43] As a result, exposure to specific harms is significantly shaped by the deployment context, including the sector, user base and scale of deployment. A banking customer support chatbot, for example, presents different risks from a general-purpose conversational chatbot. Similarly, within the same sector, AI systems used to optimise administrative workflows pose different risks from those used to support or make consequential decisions.

   Deployment context also determines who the end user is, which is important for allocating responsibility. For example, AI systems used for medical diagnostics are typically designed for trained healthcare professionals operating under sectoral and professional standards. In such human-in-the-loop settings, a significant share of responsibility for evaluating AI-generated outputs before relying on them for diagnosis or treatment rests with the healthcare professional. By contrast, where AI systems are deployed directly to the general public, including minors, many of the relevant points of control lie with model adapters and deployers.

   Deployment context also shapes the effectiveness of particular regulatory measures. Disclosure requirements, for instance, are generally best implemented by deployers because they control the user interface. However, the same disclosure may not be equally effective across contexts. A notice that is appropriate for a general-purpose chatbot may be of little value to a banking customer completing a transaction, or even counterproductive if it

[42] Risk exposure related to deployment context however does not undermine model layer risks. A flaw or vulnerability that goes unaddressed at the model layer can propagate simultaneously across multiple deployment contexts when multiple downstream actors build on the same foundational model. This an aggregated/compounded risk that must be accounted for.

[43] Xu, Yuming. 2026. *Securing Retrieval-Augmented Generation: A Taxonomy of Attacks, Defenses, and Future Directions*. https://arxiv.org/pdf/2604.08304. -This is particularly visible in retrieval-augmented generation (RAG) systems, where deployers or middleware actors may shape outputs by selecting external knowledge sources, configuring retrieval pipelines, or setting access controls that determine what information is supplied to the model at inference.

interrupts a clinician relying on a diagnostic tool under professional supervision.[44] **The fact that a deployer can implement a disclosure, in other words, does not mean it will achieve the intended safety outcome.** As discussed in *Chapter 2*, the EU AI Act illustrates this contextual approach by identifying high-risk AI systems based on their deployment context.[45]

Box D: Agentic AI – An Emerging Configuration

Agentic AI illustrates why AI governance should focus on functions and points of control, rather than fixed actor categories. Unlike conventional generative AI systems, where a model is queried, generates an output, and delivers it to a user who then decides how to act, agentic systems usually run a sequence of steps to complete a task. They may plan across multiple stages, from retrieving external information and using tools to evaluating intermediate results and taking additional decisions before producing the final output. **The final output is therefore shaped not just by the prompt, but by the specific task execution or orchestration path defined by the deployers of agentic systems.**[46] Consider an agent tasked with preparing an investment risk assessment. Rather than simply generating text, it may search the web, retrieve company filings, query internal databases, compare information across sources, and draft a summary. If it is given access to different tools or information sources, the final output may change even when the underlying model remains the same.

This complicates questions around '*who controls the AI output*', '*to what extent?*' and '*who has the responsibility to act if a risk emerges?*'

- **Existing functional attributes remain relevant, but their application may become more nuanced:** Functional attributes such as 'output visibility' and 'runtime control' extend beyond the generation of a single response to the broader execution process. For instance, output visibility may not just include the ability to view final response, but also visibility into the agent's intermediate reasoning steps, tool calls, logs, permissions and actions.[47] Similarly, the degree of control available to orchestration-layer actors may increase. These actors influence which tools an agentic system can access, the

[44] "SB-243 Companion chatbots (2025-2026)." 2025. California Legislative Information. https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202520260SB243. — The act reinforces this approach, explicitly exempting customer service or business operation chatbots where deployment context makes it clear that the chatbot is not a real human.

[45] "Annex III: High-Risk AI Systems Referred to in Article 6(2) | EU Artificial Intelligence Act." n.d. EU AI Act, https://artificialintelligenceact.eu/annex/3/; and "AI Act | Shaping Europe's digital future." n.d. Shaping Europe's digital future. https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai.

[46] "Introducing ChatGPT agent: bridging research and action." 2025. OpenAI. https://openai.com/index/introducing-chatgpt-agent/.

[47] "Updated Model AI Governance Framework for Agentic AI." 2026. IMDA. https://www.imda.gov.sg/resources/press-releases-factsheets-and-speeches/factsheets/2026/updated-model-ai-governance-framework-for-agentic-ai.

actions it is permitted to perform, the approvals required before execution, and the operational limits applied at runtime.[48]

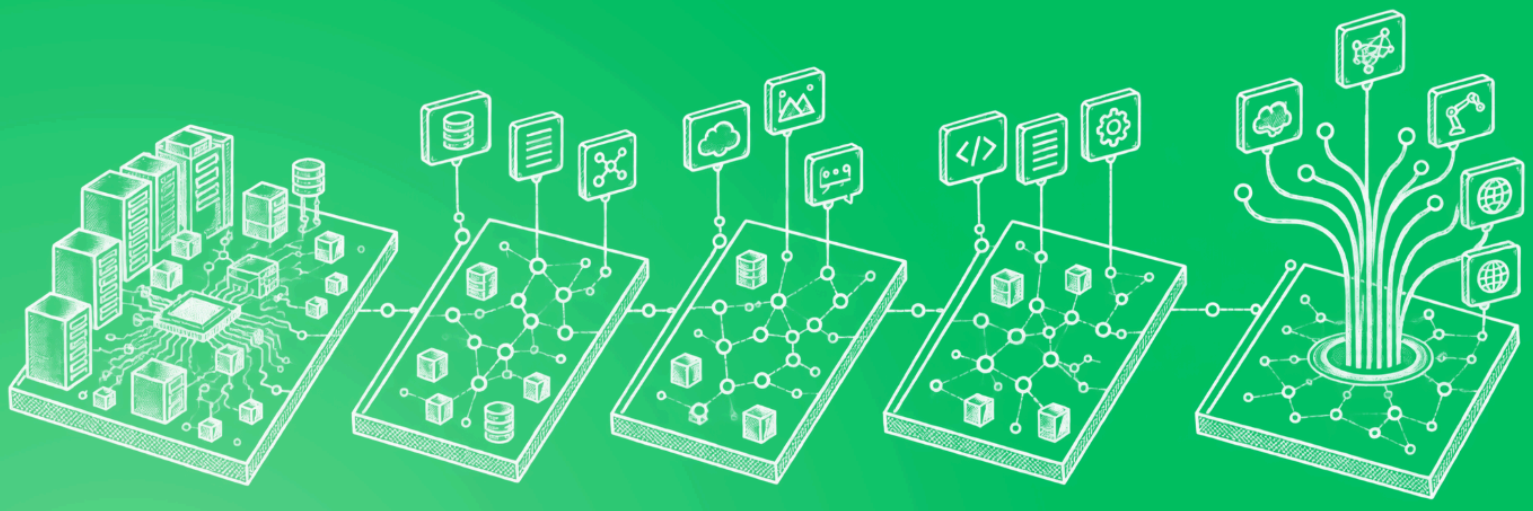
- **Multi-step execution and overlapping functions complicate attribution:** Under conventional generative AI systems, an output is generated and handed to the end user, who then decides how to use it. Agentic systems, however, blur these boundaries. Instead of a single generation step, they may execute a sequence of actions, including retrieval, tool use, intermediate generation, evaluation and execution. A single workflow may involve multiple models, tools, data sources, approval layers and deployment controls, each influencing part of the process. **The governance question therefore shifts from who controlled a specific output to how control was distributed across the workflow that produced it.** Agentic systems also change how harm may arise. Outputs generated at one stage of a workflow may guide or feed subsequent steps, meaning risks may become apparent only when the final output is assembled. For example, an agent tasked with compiling a background profile may retrieve several individually public and unremarkable pieces of information, such as a person's workplace, neighbourhood and daily routine. None of these retrievals would ordinarily trigger a safeguard. Yet, when combined, they may create an identifying profile that enables harassment or stalking - a harm that emerges from the synthesis rather than any individual step. As a result, safeguards designed to assess discrete outputs at each stage may fail to detect harms that arise only through their combination.

Agentic AI is, therefore, a significant emerging consideration for any governance framework built on the technical and operational attributes of AI actors, and warrants continued scrutiny as these configurations become more prevalent.[49] At the same time, it does not undermine the proposed functional taxonomy. Instead, it reinforces its value. **As AI systems become increasingly modular, autonomous and distributed, governance frameworks anchored in technical functions and points of control are likely to remain more durable than those built around fixed legal labels or product categories.**

[48] "Building Effective AI Agents." 2024. Anthropic. https://www.anthropic.com/engineering/building-effective-agents.
[49] IMDA. 2026. Model AI Governance Framework for Agentic AI. https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf.

04

Applying the Functional Taxonomy



The earlier chapters outline three questions that need to be considered together when assessing how a regulatory obligation should operate across the generative AI value chain. The first is **risk analysis** to identify the harm, its context and the response that may be required. The second is **legal classification** to determine which actors the law applies to. The third is **functional analysis** to identify the actors that have the information and technical control needed to carry out the intervention.

These questions serve different purposes, but need to be considered together. A legal category may bring several actors within the scope of an obligation, even though they have very different roles and capabilities. A model developer may influence model behaviour without knowing a downstream interaction. A deployer may control product-level safeguards without controlling the underlying model. A distribution platform may have no role in generation but can be relevant once an output is shared on its service.

The functional taxonomy helps identify where the relevant capabilities and points of control sit, and whether a proposed responsibility is realistically matched to what an actor can do.

4.1 Start with the risk and the intervention

Broad categories such as "harmful AI content" may not be enough to identify an appropriate response. The same harm can arise at different points in the AI output lifecycle and may require different interventions.

For example:

- Outputs in catastrophic risk domains - such as content enabling the development of chemical, biological, or nuclear weapons - require intervention at the model layer;
- An unsafe response to a particular user or prompt may require application-level safeguards or human review;
- Misleading synthetic content may call for disclosure or provenance measures;
- Harmful content already circulating on a platform may need to be labelled, downranked or removed.

The first step is therefore to identify both the problem and the intervention needed to address it. An actor that can carry out one intervention may have little control over another.

Where a regulatory framework defines a common outcome, different actors may contribute to that outcome through different controls. For example, reducing deceptive synthetic content could involve generation-side provenance, deployer-level disclosure, and platform-level labelling or restriction. The taxonomy can help identify which intervention is feasible for each actor rather than assuming that every actor will implement the same technical measure.

4.2 Match the intervention to functional control

Once the required intervention has been identified, the functional attributes in *Chapter 3* can be used to locate the relevant point of control. The five attributes are:

- **Model-layer influence:** ability to shape model behaviour through training, fine-tuning, safety tuning and model updates.
- **Runtime or inference-layer control:** ability to influence a specific output through system prompts, retrieval, guardrails, tool access, filtering or other deployment choices.
- **Output visibility:** ability to see, log, audit or evaluate what is generated.
- **Distribution control:** ability to affect how an output is circulated after generation.
- **End-user proximity:** direct interface with the end user interacting with the output.

Different interventions rely on different capabilities. Changing a model's general behaviour requires model-layer influence. Preventing a particular response from reaching a user may depend on runtime control and output visibility. Acting on content after it is uploaded to a social-media service depends primarily on distribution control.

These capabilities are not fixed by an actor's category. These can vary depending on how the model is offered and deployed, including whether it is closed or open and how it is integrated into a downstream system. A foundation model developer operating a closed API may retain access controls, usage signals and some visibility. Those levers are substantially reduced once model weights are released and independently hosted. A deployer that fine-tunes a model, adds retrieval and controls the interface also has different capabilities from one that simply provides access to a third-party model.

Some interventions also depend on what an actor can see. For example, an actor cannot monitor or investigate harmful outputs if it has no access to those outputs, logs or relevant abuse signals. Information-sharing between actors can help address this by giving each actor the information it needs to use the controls already available to it.

The relevant question is therefore not simply **who the actor is, but what it can actually control and see in the deployment in question.**

4.3 Identify obligation-control mismatches and gaps

Applying the taxonomy can reveal two problems that broad actor categories may obscure.

An **obligation-control mismatch** arises when an actor is subject to an obligation but lacks the relevant lever to carry it out. A distribution platform, for example, may label, downrank or remove AI-generated content once it reaches its service, but cannot prevent that content from being generated on a separate AI tool. A generation-focused obligation would therefore be poorly matched to the platform's point of control.

A **gap** arises when the regulatory framework does not cover an actor that plays an important role in producing or controlling an AI output. For example, an independent orchestration or middleware layer may route prompts, retrieve information, select models or filter responses. Where globally available foundation models are deployed across jurisdictions, deployment-specific or local legal requirements may need to be implemented at downstream layers, including through independent orchestration, rather than through the underlying model itself. **If these actors fall outside the regulatory framework, an important point of control may remain unaddressed.**

These issues become more important in the context of agentic systems, where control may be distributed across retrieval, tool use, intermediate outputs and execution steps. In such cases, any regulatory analysis needs to examine which actors controlled the different parts of the workflow, rather than focusing only on the actor producing the final output.

4.4 Apply the taxonomy in context

Functional control identifies what an actor can do. **Context helps determine whether that intervention is appropriate and effective for the risk in question.**

A general-purpose chatbot and a bank's customer-support system may rely on similar technology but operate in different environments. An AI tool used by a trained clinician also raises different questions from a tool offered directly to the public. Sector, user, deployment arrangement and release model can affect both the nature of the risk and the usefulness of an intervention.

AI-content labelling illustrates this distinction. A label may be useful where a realistic synthetic video is presented as genuine. The same requirement may add little value when applied to a restored photograph, design asset or AI-assisted medical image. Similarly, more extensive watermarking measures may impose quality or usability costs on benign use cases. Applying the same countermeasure across all use cases can therefore impose unnecessary costs without addressing the underlying risk equally effectively.

4.5 Use the taxonomy as a reference tool

The framework can be applied through six questions:

1. **What is the specific risk or harm?**
   Define the problem and the context in which it arises.
2. **What outcome or intervention is required?**
   Identify whether the objective is prevention, detection, disclosure, monitoring, restriction, removal, redress or another response.
3. **Which actors fall within the relevant legal framework?**
   Identify the actors that may be within scope.

4. **What functional capability does the intervention require?**
   Identify whether the intervention depends on model-layer influence, runtime control, output visibility, distribution control or end-user proximity.
5. **Which actor holds that capability in the actual deployment?**
   Assess the function being performed and the controls actually available, rather than relying only on the actor's legal label.
6. **Does context change the point of control or the suitability of the intervention?**
   Consider the sector, user, model release, deployment arrangement and information available to each actor.

Where the legal scope, required intervention and actual point of control align, an obligation is more likely to be technically and operationally workable. Where they do not, the analysis can expose an obligation-control mismatch or an overlooked point of control.

Conclusion

AI governance increasingly has to address systems in which responsibility is distributed across multiple actors and technical layers. The functional taxonomy helps connect regulatory interventions to the actual points of influence, visibility and operational control across the AI output value chain. **It can help policymakers identify four recurring problems: obligation-control mismatches, gaps in coverage, overlapping points of intervention, and changes in control as AI systems and architectures evolve.**

These issues are likely to become more important as AI systems become more modular. Open-weight models can move operational control downstream. Independent orchestration layers can become important points of influence. Agentic systems can spread control across several steps, tools and services. At the same time, companies will continue to integrate multiple functions vertically. Where this occurs, the functional analysis still applies per function rather than per entity - obligations can then attach based on the control exercised at each layer, not on the cumulative footprint of the entity performing them.

Fixed actor labels may not always capture these changes. The same category can contain actors with different capabilities, while the same organisation may perform different functions across products and deployments. A functional assessment keeps the analysis tied to how the system actually operates. It can help test whether a proposed requirement is workable, interpret broad actor categories in specific technical settings, support sector-specific guidance and identify new control points as AI architectures change.

The taxonomy does not, by itself, determine legal liability. It can, however, *inform* the allocation of responsibility and liability. As AI systems change, the actors and architectures around them will change as well. Durable AI governance therefore depends on assigning responsibility where control actually sits, rather than assuming that legal categories will always neatly map onto technical capabilities.

www.thequantumhub.com