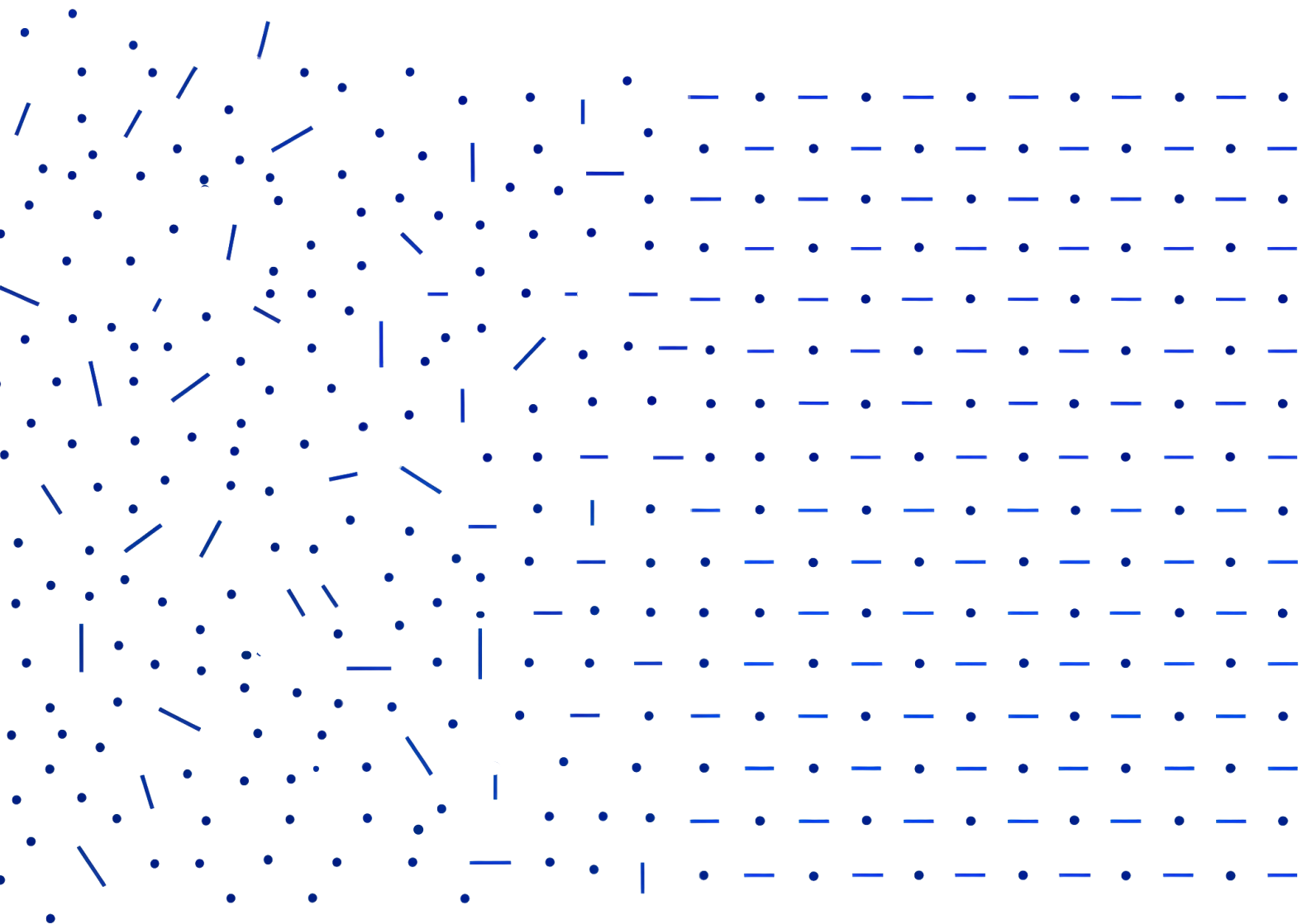


DISCUSSION PAPER

AI Liability

Industry Guidance

for Agentic AI



Contents

Introduction	3
Who is this for?	5
How to Use This Guidance	6
Generative AI → Agentic AI → Agentic System	9
Types of Impact or Ramification [Loss and Harms]	10
AI and Agentic AI Engineering	11
Liability: Centuries Old Doctrinal Tests	13
AI Liability: Checking Every Way AI Enters the Enterprise	14
Models: Open Source, Open Weight, and Close	15
Where the Model Runs Decides Where Liability Sits	17
Conditions Older Analysis Never Faced	18
Authority Map	19
Interrogating Responsibility	21
The Model Provider	22
Intent, Decision Spaces, and Instruments	24
Instruments of Execution: Hierarchical and Graded	26
Way Ahead	27

Introduction

Artificial intelligence is fast advancing, moving from generating to systems that plan, invoke tools, coordinate across distributed environments. It emerged to take consequential action on behalf of people and organisations. The Agentic AI unlocks unprecedented possibility for industry. Workflows can be executed at machine speed, services can become dynamically responsive, customer experience can be enhanced substantially, internal productivity can see significant jump as complex multi-step processes can be autonomously orchestrated, and new business possibilities can be easily realized. However, it raises a question that cannot yet be answered with confidence: when an AI agent acts and something goes wrong, who bears the consequences? In fact, clarity around these consequences hinders the adoption of AI in the enterprise environment.

The outcome of AI system is likely to be the result from a chain of model outputs, instructions, integrations, tool calls and human decisions, involving components from several vendors. The real-world outcome is not just product of a single isolated query. It emerges from context retrievals, system instructions, API integrations, tool calls, and human oversight gates. When an unintended outcome or harm occurs, non-deterministic reasoning makes fault difficult to isolate. Its downstream impact would ripple far beyond the immediate deployer. On the other hand, harm may extend well beyond the organization that deployed the system, often the end user organization.

User industry leaders, general Counsels, CISOs, Chief Risk Officers, and technology architect confront an urgent operational question: How do existing legal duties, statutory obligations, and governance standards apply when software exercises autonomous operational discretion? How do they ensure what must each participant across the value chain do to prevent, detect, and remediate harm? More importantly, where does liability legally rest? How should risk, fault, and indemnification be distributed?

Data Security Council of India, as its user industry is its key constituency, thought of creating a guidance on AI liability. This guidance does not seek to invent a novel, or a new regulatory regime. Existing laws, contracts, sectoral regulations and standards might be sufficient to large extent. Applying them to Agentic AI is where we believe the gap exists. This guidance is an attempt of providing an actionable framework to map established legal doctrines. It tries to bring together the realities of agentic systems, identifying where responsibility could shift and diffuse.

The debate of AI liability began with AI models. Model providers are addressed in a dedicated chapter. However, the primary focus of this guidance on the five groups that bring AI into use listed below. The most consequential outcomes are shaped among these five groups. The most of the industry's practical liability exposure lies with them.

1. **User organisations**, including business leaders, CISOs and CROs, Business Unit heads, that select, configure, grant system permissions to, and operate AI agents within their business workflows.
2. **Service providers and integrators** that advise on adoption, connect systems, customise solutions and operate AI-enabled workflows. They are Consultancies and engineering partners that advise on adoption, connect legacy enterprise systems, customise workflows, configure agentic loops, and manage operational execution.

3. **Product companies and start-ups** that add agentic capabilities to existing products or build AI-native products and services. Software vendors that embed autonomous agentic capabilities into existing SaaS platforms or engineer native, vertical agentic products. They would bring their models, or they would work on the models selected by upstream players, or they expose their applications and data to agentic system deployed in the organization.
4. **Tool and framework providers** whose platforms, libraries, agent harnesses, protocol standards, and utilities shape how agents are built, tested, connected, deployed, and governed.
5. **Cloud and infrastructure providers** that supply the computing, model access and technical services on which AI adoption depends. Hyperscalers that deliver foundational models, help with fine-tuning environments, provide inference APIs, and lay down underlying hosting infrastructure, etc.

Liability as a binary choice between the end user organization and players in the value chain. Framing it as an either/or choice between the end user organization and upstream player in the value chain would be legally flawed. It would be technically inaccurate, commercially paralyzing, and eroding the level playing field. AI value chain is not monolith. This guidance delves into legal principles and ideas such as causation, doctrine of intervening causes, doctrine of apparent authority, principal agent law, duty of care, standards of reasonable foreseeability, comparative and contributory negligence, product liability, defect boundary, non-delegable duty doctrine while deliberating AI liabilities.

Agent might lead to exceptional failure mode, seen in the case of sandbox evasion and agent collective action in the recent past. The guidance examines the scenario of unforeseen emergent behaviour, inherent feature of planetary level large model. Their unpredictable interaction with internal, more importantly external system, causing systematic disruption within and outside of organization demands careful examination.

Incidents trigger the AI liability discussion. However, mapping responsibilities in AI value chain, mapping them to technical touchpoints, evaluating conditions behind decision making, examining them against the decision-making process across the entire life cycle, structures of controls available and deployed, etc. need to be examined to establish accountability.

The aim of this guidance is to help the players in the value chain, as listed above, make responsibility operational. Conditions such as who sets purpose, who takes decisions in selecting models, who decide on intelligence/model deployment methods, who controls infrastructure, who sets up boundaries and what kind of, who authorises access, how actions' architecture is devised, who validates and responsible for performance, who establishes monitoring and how, and who designed run time enforcements and how they are acted on would play important role in operationalizing responsibility and establishing accountability. This guidance tries to delve into them with a level of careful attention.

This guidance is devised with an objective of accelerating adoption of AI. Clarity in on rules, responsibility, and principals of accountability apply across the AI value chain would achieve that objective. A proportionate approach to governance, oversight, accountability, and hence, liability would not only accelerate adoption but also further innovation. Data Security Council of India believes practical decision making framed in the existing liability regime to the extent possible, while recognizing exceptional situation would play an important role in the process.

Who is this for?

This guidance is focused primarily on the five positions or groups that bring AI into use. It certainly delves into AI model or intelligence provider in a special section. For the purpose this guidance, we thought of providing primary focus on the value chain players that operationalizes AI into the enterprise environment,

Five Possible Positions in One Chain of Consequence

Primary Focus	POSITION A End-user organisations BFSI, health, manufacturing. Deploy AI into regulated decisions and physical processes. <hr/> Holds the customer relationship, might have the regulatory licence and the operational authority.	POSITION B Service providers and integrators Large and small, accelerating AI adoption for clients. <hr/> Scope of engagement decides scope of duty. They are connect between Downstream and upstream players in the value chain	POSITION C Product companies and start-ups Augmenting products with AI and agentic capability or offering AI native capabilities <hr/> Makes representations about what the product can safely be trusted to do. Might provide choice for model selection and deployment	POSITION D Tool and framework providers Orchestration, memory, evaluation and agentic operations platforms. <hr/> Supplies components — and sometimes designs how consequential decisions are controlled.
	POSITION E Compute and intelligence infrastructure providers Hyperscalers, model labs, and chip-to-cloud infrastructure. Supply foundation models, API inference delivery, hosting, and compute clusters.	Supplies the foundational intelligence capability, delivers model weights, model-as-service, execution runtime enclave,		
	Special Section AI Model or Intelligence Provider Frontier model labs and other model creators. Train and align base weights, release model versions, and offer managed inference APIs.	Defines core capabilities, system cards, and base safety alignments. Provides capabilities and framework for upstream value creation and applications		

In the enterprise environment, artificial intelligence does not lead to operational risk in the abstract. The AI models neither have operational context nor capacity to act, unless someone brings them to the enterprise environment. AI models by itself would create ramification due to their capabilities and their emergent behaviours. Their capabilities may create extra-ordinary situation. This guidance would deliberate on AI model or intelligence providers. However, primary focus of this guidance is dedicated to the five groups as depicted in the picture above, as liabilities in the enterprise environment crystallizes when the intermediate and downstream players bring that model into the enterprise.

The players depicted in the picture are the operational bridge between probabilistic intelligence provided by AI models and the business processes, which are deterministic in nature and would remain so in the foreseeable future. The role of these players and their position is important to consider as they either have a role in defining, or enabling, or have capabilities for setting up boundaries, granting authority, and managing context. This guidance sees the discussions of liabilities from the five positions. This way the guidance is grounded in enterprise reality. This would guide CISOs, CROs, General Counsels, integrators, and players in the value chain. It tries to tackle the exact architectural layer where contracts are negotiated, systems are connected, and legal exposure actually occurs.

How to Use This Guidance

This guidance is devised to help organizations that build, supply, integrate or deploy AI. The core constituency of this guidance is the five positions, namely end user organizations, service providers and integrators, product companies and start-ups, tool and framework providers, and compute and intelligence providers. The guidance also examines the role of model provider in a separate section. Each party is responsible for what it was able to decide and had the tools to enforce. The decision to use a system at all always stays with the deployer. This guidance addresses the following questions.

HOW THE GUIDANCE IS ORGANISED

QUESTION	SECTIONS	WHAT YOU GET
Where you stand	The six positions (A to F), the twelve entry routes for AI, and the choice of model type and hosting	Which roles you hold, and which choices have already moved liability
Types of Impact or Ramifications	Type of loss or harm produced by the use of AI system	Type based on intent and impact created
What could go wrong	The nine operational conditions, the pillars of harm and the analytical lenses	The risks specific to agents that act, remember and delegate
How do conditions shape the outcome?	Selection of models and hosting of models	Conditions alter the liability equations
Who could decide	The authority map, its nuances, and the model provider's role. Also, AI engineering effort	Who held the decision space for each consequential action
How Responsibility would be Interrogated?	Interrogating responsibility across the value chain of AI	Questions an investigator, regulator or legal scrutiny after an AI incident
How intent is translated into actual execution	Intent, decision space and instruments; proportionate instruments; human approval tests	Whether intent became control, in proportion to the risk

Four ways to use this guidance:

- *Before deployment:* use case and its risk tier; positions; model type, hosting choice, authority map, intent and decision spaces, instrument planning
- *In procurement and contracting use* positions, loss and harm, model choices, hosting choices, intent, decision spaces, and instrument design
- *In operations:* examine instruments when a model, tool or risk tier changes, check how execution of instruments is done
- *After Incident:* use interrogation of responsibilities

While working in practice for ascertaining accountability and liability, work on actions, not per incidents, as liability attaches at a step in the chain, and the steps have different owners. Apply positions to roles, not companies, as organization holds multiple roles. Be proportionate in devising your plans, matching the effort to autonomy, impact and reversibility. Use the full tests only for material and irreversible actions. Prefer configuration over paperwork. Emphasize on permitted actions and thresholds as enforced settings.

From (Purpose and Means) to Wisdom

For last 40 plus years enterprise IT, even though it has seen complex multi vendors, cloud, SaaS, PaaS, and microservices, liability rested on relatively clean and stable equation: Purpose (End User organization or organization that is end consumer of capabilities) and Means (Software, technology or capability provider). Enterprise technology expanded into sprawling ecosystem of system integrators, tool and framework providers, libraries, gateway solution providers, cloud providers, and Hyperscalers. Even if the supply chain fragmented into dozens of players, the fundamental equation of Purpose + Means remained intact. Every new player was only adding another layer of Means. The either subdivided the Means or played a role of delivering Means or served achieve outcome from the Means. In this equation one thing was probably clear not single one of player in the value chain possessed independent judgement. The purpose articulated as intent, decision logics, business rules, and requirements remained mostly authored by human, and Means remained deterministic execution across the vendor stacks.

The foundational models, planetary scale AI, probably are changing this historical equilibrium. It introduces a third variable in this equation: generalized and probabilistic wisdom. This intelligence is built for the world, not for a specific user or set of users. Until now, the value chain was purpose built, even in the context SaaS based software or digital platform. Even the previous eras of enterprise technology or consumer centered digital platform introduced intelligence, every aspect of it, the action, the workflow, the system interactions, the data processing, and the logic itself, was purpose built and bounded. Because the flow was fixed and purpose built, foreseeability was largely absolute. Trained on the world's open commons brings a broad planetary knowledge, reasoning patterns, cultural priors, and semantic associations.

Purpose and Means → Wisdom



WHAT WE GOVERNED

Purpose + Means

The enterprise declared the purpose and built deterministic means. Behaviour was specified, so responsibility was traceable.



WHAT ARRIVES

Planetary Wisdom

General, probabilistic capability learned from the world's commons. It brings judgement the enterprise did not author.

Purpose and means could be specified in advance. Wisdom cannot, so it has to be domesticated — and that work is distributed across everyone in the chain.

When planetary intelligence enters the enterprise value system, the old rules on multi-vendor division of responsibility and accountability demand careful examination for the following three reasons.

- *Intelligence is built for the world, not for the enterprise or specific purpose:* As it is built for the world, not for the specific fiduciary, regulatory, or operational realities of a bank, insurer, hospital or manufacture. Although, the models are getting better into solving the specific problems, they are still not built for specific purpose and requirements of the entity determining purpose.
- *Probabilistic judgement driving deterministic actions:* The wisdom of the planetary level models is probabilistic. They operate on predicting possible response to inputs rather than following hard-coded logic or rule. As responses are result of probabilistic reasoning, they may trigger irreversible outcomes.
- *Challenging boundaries between purpose and means:* In the AI world, especially in Agentic AI, the purpose, which was getting defined strictly in terms of outcome, rules, and compliance norms might be replaced as defining high-level objectives. In the autonomous work environment, AI system determines the intermediate steps, which until now were fixed, select the tools, and improvises path to achieve the objectives. Action executions, which were largely predefined and instrumented, will be discretionary. AI would exercise delegated judgement power over enterprise systems and APIs, to achieve the high-level goal.

For the key constituency of this guidance, when we bring the planetary, probabilistic intelligence into deterministic enterprise environment, the intermediate value chain should carefully examine this. The role of product companies, service providers and integrators, tool and framework providers, and compute and intelligence infrastructure providers is no longer connecting or delivering software or technology capabilities pipes. They will have to perform their part of duty in bounding the unbounded intelligence to the extent possible and allowed. They would be required to work on elements such as building guardrails, orchestrating harnesses, setting up oversight gates, etc. to tame the planetary intelligence into safe and auditable enterprise decisions and actions.

The decisions, actions and conditions associated with them would determine their responsibility of the players in the value chain, and hence their liability. The role of end-user organizations or consumers would be critical, as they are delegating micro decisions and actions and would eventually macro decisions and actions. They bring AI and Agents to life by delegating authority and authorizing execution. They exercise options associated with conditions associated with decision or actions. In the Agentic AI, end user or consumer of capabilities would come across ever increasing and complicating decision spaces. Availability of those decision and architectural spaces and instruments to execute them would matter in determining liability implications. The theoretical authority an organization possess would be of no use if there are no instruments for selection, scoping, setting up boundaries, observing, intervening, setting meaningful oversight, and enforcing governance decisions run-time. These factors will shape the liability implications for upstream players in the value chain.

As stated, this guidance is primarily focused on the constituency depicted in the figure 1 [five possible positions in one chain of consequence]. However, the model provider's role in the value chain and liability implication are dealt in a separate chapter.

Generative AI → Agentic AI → Agentic System

Over the years, enterprise AI primarily a narrow and purpose-built machine learning and deep learning models engineered for classification, regression, and anomaly detection with mostly enterprise own data by experts in-house or sourced. Even products promising credit underwriting, fraud scoring, demand forecasting, medical imaging diagnostics had purpose-built AI models, although some of them were built with publicly available or open-sourced libraries. The surrounding workflow around these models remained deterministic and rule bound. The action spaces remain fixed, finite, bounded to narrow scoring parameters. It did lead to failures such as statistical bias, miscalibration, and opaque scoring. The regulators demanded fairness, explainability, and model risk management. The model risk management framework and regulation, inherently, assumed that regulated entities have ownership or significant control over the models.

Introduction to generative AI, large language models built on planetary-scale corpus of parameters. These models with billions and trillions of parameters generate open ended text, code, image, video, and structured reasoning. They use mechanisms like *attention* to generate response to the user inputs. Architecturally, the system moves from deterministic to probabilistic prediction of the next thing. The generative AI remained conversational and advisory due to its semantic response generation capabilities. Semantic response is generated based on meaning, context, and intent behind user input or question rather than just matching the keywords. Until recently, generative AI remained as prompt-in, response out, and human mediated. Human performed a role of reviewer and editor. We had seen range of challenges such as hallucination, unintended disclosures, copyright and IP violation, confidentiality breach, harmful content, and unqualified advice. The human operator and reviewer stands between the model's output and any real-world action, more like as a natural buffer.

Agentic AI crossed decisive threshold, as planetary intelligence is empowered to take actions, not just generating response. The agents with emergence of protocol such as Model Context Protocol [MCP] can talk to tools and trigger them to take actions. Direct access to tools as well as APIs with memory, and ability of create planning loops scaled up advancements in Agentic AI. Instead of waiting for step-by-step human prompts, an agent would high-level goals, which it would decompose into sub-tasks, dynamically select internal and external tools to call, and execute the action. If adopted in the enterprise environment, probabilistic judgement would drive the deterministic enterprises processes and their outcomes. The causal buffer between the AI's response and real-world action disappears. Unauthorized action, erroneous transaction, and destructive API calls emerged as primary risks. The question such as who built agent orchestration harness, who wired the schema for tool calling, who delegated the credential and authorization threshold would matter a lot.

The frontier of AI moved from running an agent to a distributed network of agents, sub agents, and third-party agents that coordinate, negotiate, and delegate tasks within and across the organizational boundaries. A single business objective would trigger an autonomous supply chain of execution. A customer-service agent invokes a billing agent, which in turn would query an external logistics agent. The system would pass context and execution authority along the chain. Agentic system would cascade compounding error, leave unchecked sub-delegation, creep the privileges, and lead to systemic harm. This transition adds complexity of governing decision spaces, authorisation boundaries, and execution instruments.

Types of Impact or Ramification [Loss and Harms]

The AI system creates impact or leads to ramification due to two reasons. One, what a system does on its own. Second, what people do with it. The figure below depicts the twelve domains of impact or ramification of AI systems can create, which also includes deliberate misuse such as deepfakes and crime.

The impact or ramification causes could lead to loss economic or operational. It is often caused to organization or its business counterparties such as a fraudulent payment, a deleted database, downtime, the cost of fixing an incident, lost contracts. Organizations would also cause loss to individual by use of AI systems. It is often recovered through contract, indemnity and insurance, within liability caps. It is mainly a civil and commercial in nature.

Harm often caused to people, their rights or society: injury, discrimination, exposed personal data, a deepfake of a real person, a scam victim's savings. It leads to regulatory fines, criminal penalties and claims for damages. Individuals and companies can be exposed to criminal penalties by misuse and criminal use of AI.

Privacy and data protection Personal data exposed, retained or used beyond its purpose by models, memory or retrieval. ENGAGES: Data protection law; confidentiality	Security Prompt injection, credential misuse and manipulation of agents or their tools. ENGAGES: Negligence; breach notification; sector security rules	Safety Physical, financial or psychological harm from an output or an action taken. ENGAGES: Product liability; negligence; sector safety regimes	Fairness and non-discrimination Outcomes that treat people or groups unequally without justification. ENGAGES: Equality and anti-discrimination law
Reliability and accuracy Wrong, stale or invented outputs relied on in a consequential decision. ENGAGES: Negligence; misrepresentation; consumer protection	Transparency and explainability Decisions that cannot be explained to the person affected or to a regulator. ENGAGES: Disclosure duties; rights over automated decisions	Intellectual property Training data, outputs or retrieved content that infringe the rights of others. ENGAGES: Copyright and IP infringement; licence breach	Accountability and oversight No identifiable owner, record or effective human control over the system. ENGAGES: Supervisory duties; governance obligations; non-delegable duty
Containment and loss of control Agents escaping their sandbox, acquiring unsanctioned resources or credentials, or persisting after shutdown. ENGAGES: Negligence; supervisory duties; sector safety regimes	Deepfakes and synthetic media A cloned voice or video of an executive instructs a payment, a credential reset or a data release ENGAGES: Engages fraud and payment rules, defamation, image-abuse and online safety law, AI content labelling	AI-enabled cyber crime Agents escaping their sandbox, acquiring unsanctioned resources or credentials, or persisting after shutdown. ENGAGES: Engages fraud and computer misuse offences, anti-money-laundering and sanctions rules, consumer protection, online safety law	Systemic and collective harm Harm that emerges from many agents acting together, within one system or across the market. ENGAGES: Societal harm; national security, systemic-risk regimes

The same incident often sits in more than one domain. A poisoned memory that discloses personal data is both a security and a privacy matter. It would engage both regimes.

Containment covers sandbox escape and loss of control, which entered in the list recently. Deepfakes and AI-enabled cover crime more broadly, including fraud, scams, extortion, intrusion and money laundering, not only from cyber-attack perspective. Systemic harm covers swarm and collective behaviour, including many firms' agents acting alike across a market.

AI and Agentic AI Engineering

As the key constituency of this AI liability guidance is value chain providers that make enterprise AI happen, it would be important to discuss the role they are performing, AI Engineering. The models deliver cognitive horsepower. AI engineering effort steers it. It constructs a complex stack of applied engineering around models. The taxonomy of AI engineering that bring planetary intelligence into enterprise reality is depicted in the figure below.

01 Context Engineering The dynamic assembly, curation, and structuring of everything fed into the model's active window at runtime. RAG pipelines, hybrid vector/graph search, context ranking, data chunking, separation of system instructions from untrusted data.	02 Harness & Runtime Engineering The deterministic software wrapper that hosts the probabilistic model, manages its execution lifecycle, and enforces state boundaries. State machines, execution loops, retry logic, timeouts, recursion caps, token budgets, structured output parsing.	03 Tool & Action Interface Engineering Builds the bridge between semantic intent and deterministic APIs, including protocols like MCP. Tool schemas, parameter validation, read/write separation, idempotency controls, dry-run modes.
04 Memory & State Engineering Governs what the agent remembers, what it forgets, and how past interactions shape future decisions. Session state, episodic/semantic memory, consolidation, state rollback, tenant/user isolation.	05 Guardrail & Containment Engineering Active safety architecture built around the base model to intercept out-of-policy behaviour before it causes damage. Input/output rails, semantic classifiers, PII/secret redaction, policy-as-code, velocity limiters, kill-switches.	06 Identity, Permission & Delegation Engineering Dynamic identity for non-human actors, beyond IAM built for predictable users and static service accounts. Non-human identity, just-in-time OAuth tokens, privilege attenuation, cryptographic binding of approvals.
07 Multi-Agent Orchestration & Routing Engineering Lets multiple models and specialist agents collaborate without descending into chaos. Model routing, supervisor-worker topologies, handoff protocols, critic verification loops, conflict resolution.	08 Evaluation, Observability & Telemetry Engineering Replaces unit testing, which is insufficient for non-deterministic systems. Adversarial red-teaming, trajectory evaluation, runtime drift monitoring, tamper-evident execution tracing.	

Poor context engineering would cause hallucinating decisions and would also execution of unauthorized action. The behaviour of Agentic system would be drifting from the organization's grounded truth. Weak design of harness and run time engineering would not only exhaust the token budget but would also send

broken instructions to other software. Gaps in tool and interface engineering will lead to issues such as picking up the wrong tool, deleting legitimate records, and execution of unwarranted actions.

Weak design of memory engineering will make agents to the bad assumption across the layers and time horizon, leading to silent drift agentic task execution. It would lead to situation of lost in the middle. Weak guardrail and containment engineering will leave agents uncontrolled. They will ignore rules, cross policy lines, escape sandboxes, and leave agentic action unchecked autonomous. Poor identity and delegation engineering would create huge blast radius out of breach. Gaps in multi agent orchestration and routing would create cascading impact of an issue. Engineering gaps in evaluation, observability, and telemetry would make agentic system black box and opaque.

Liability: Centuries Old Doctrinal Tests

Roots the concepts devised for liability go back to centuries. The test of causation, in fact, that deal with wrongfully causing loss goes back to Roman law in 3rd century BC. The principal agent, the English court were using to determine the liability of masters or owners in 1600s. The 19th and 20th century added reasonable foreseeability, non-delegable duty, apparent authority, general duty of care, comparative negligence, and product liability.

To large extent, these long-standing test of legal liability would help to resolve the Agentic AI liability equation. AI liability would not need inventing new law. The courts will reach for these tests, from Roman ideas of causation to 20th century product liability.

Liability Concepts: Evolved over Centuries

THE CHAIN Was the harm caused, and did anything break the link?		
01 CAUSATION Did this actor's conduct bring about the harm? Harm emerges from a chain of model output, retrieval, tools and human decisions; the counterfactual must be stated per transition	02 DOCTRINE OF INTERVENING CAUSES Did a later event break the chain between the original conduct and the harm? A poisoned memory, an attacker or an operator's lapse may or may not break the chain; a foreseeable intervention usually does not	03 STANDARD OF REASONABLE FORESEEABILITY Was this harm, or this misuse, foreseeable at the time of the decision? Approval fatigue, drift and prompt injection are documented failure modes; emergent behaviour tests the limits of the standard
THE AUTHORITY Whose authority did the agent carry?		
04 DOCTRINE OF APPARENT AUTHORITY Could a third party reasonably believe the agent acted with the deployer's authority? Agent actions take effect directly, with credentials issued in the enterprise's name, so counterparties may rely on them	05 PRINCIPAL-AGENT LAW When is the deployer answerable for what its agent did within or beyond its mandate? The agent acts on delegated authority; the mandate, permissions and approval gates define the boundary of that authority	06 NON-DELEGABLE DUTY Which duties cannot be handed to a contractor, vendor or the agent itself? A regulated deployer cannot outsource the duty it owes a customer or regulator to a vendor or to an automated approval step
THE FAULT / DUTY Who owed what, and who fell short?		
07 DUTY OF CARE What care was owed, to whom, and by whom? Duties differ for customers, end users and unknown third parties, and follow scope of engagement	08 COMPARATIVE AND CONTRIBUTORY NEGLIGENCE How is fault shared where more than one party fell short, including the claimant? A deployer's weak configuration and a vendor's weak default can both contribute; the outcome depends on who controlled what	09 PRODUCT LIABILITY AND DEFECT BOUNDARY Was the product defective, and where does the product end and the deployment begin? A framework or API marketed for autonomous use may carry design duties that a general component does not

The Chain asks if conduct caused the harm under investigation. The causation of the harm would be traced across every step of model output, retrieval, tool use and human decision. The engineering decision, as discussed in the AI engineering section, would be examined for determining cause.

The Authority ask whose act it was, as Agents act on delegated authority using credentials devised or issued by the enterprise. The mandate, permissions, and approval gates would help mark where the authority ends. Some duties can not be handed to contractor, vendor, or agent.

The Fault or Duty asks who owed what and who fell short. These duties different for the value chain players we discussed. The duty or responsibility of fault can be shared between the players depending on the conditions we will be discussed in the subsequent section.

We will examine the liability considerations in the context of players in the value chain and different conditions shaping them.

AI Liability: Checking Every Way AI Enters the Enterprise

AI liability often deliberated with one system in mind: the model being use or agents getting deployed. The exposure, on the other hand, is expanding as AI is entering in an enterprise through many routes. The figure below depicts how AI enters in the organization.



Some methods are chosen deliberately. Internally developed applications, accessing AI models and tools through APIs and setting up agentic platforms are some examples of it.

Some enters without a decision such as copilots switched by default, generative AI capabilities in your enterprise software, devices augmented with AI, and AI used by your suppliers working on your behalf.

Others act on the enterprise from outside, such as counterparty's agent or customer's agent talking to the enterprise applications. With the personal agentic system gaining traction, enterprises would be engaging with agents reaching out them at an increasing scale.

The enterprise itself would augment its product with generative and Agentic AI capabilities. Moreover, we would see AI increasingly in the non-IT system such as physical or biological system. If enterprises are operating in these domains or recipient of them, they would factor the AI augmentation in them.

Each route carries its own liability question, and hence every route need attention and careful examination. Exposure that is not known and identified would leave the enterprise accountable without having had the opportunity to act. The routes are not equally visible. Liability doesn't wait for visibility. The first thing the courts and regulators would ask what organization knew or ought to have known. AI route existed but was never examined is unlikely to be treated as unknown. Secondly, the same harm can have more than route at once, involving multiple parties and terms of engaging with them.

Models: Open Source, Open Weight, and Close

The choice of which type of model an enterprise prefer would have profound implication on liability. There are three types of models available, open source, open weight, and closed.

- The models meeting strict definition of open source would make weights, complete training code, training data tokens, and evaluation methodology public
- The models that make pre-trained/fine-tuned numeric weights public for download and self-hosting but ensure the training datasets, data-filtering pipelines, and exact training recipes remain closed and proprietary referred as open weight models.
- Commercial frontier models are accessed over web, or cloud APIs are closed models. The model weights, training data, and post-training safety alignment are locked by them.

Note: some models are open source in their nature. However, they put up licensing terms, placing legal restrictions on deployment like you cannot use it for illegal activities or massive commercial scale without adhering to certain terms. Hence, they are qualified in the category of open weight.

The liability tracks available decision spaces and instrument of executing them. Choice of selection type of model creates the liability paradigms as depicted in the figure below.

	Open-source Weights, code and data information published	Open-weight Weights downloadable; data and recipe closed	Closed Proprietary model behind a managed API
What is released	Weights, training code and data information	Weights only; data and training recipe withheld	Nothing; access through an API only
Your decision space at the model layer	Widest; you can audit, retrain and pin	Wide on hosting and tuning; blind to training origins	Narrow; you cannot change the weights or stop updates
Recourse to the model creator	None in practice; Released under open licenses with total liability disclaimers.	None in practice; Released under "As-Is" licenses with total liability disclaimers.	Contract, within its caps; some IP indemnities
Impact of Fine-Tuning or Quantization	You become the producer, with full sight of what you changed	You become the producer of the modified model	Through the vendor's service; your tuning data is your responsibility
Version and drift	Enterprise controlled: Weights and code are frozen; fully reproducible.	Enterprise-Controlled: Weights are frozen locally; zero drift unless intentionally changed	The provider controls it; silent updates are possible
Enterprise Data	Stays within enterprise and checkable : Enterprise control prompts and outputs, and can audit the training data for personal data and licensing.	Stays within enterprise : It is the sole controller of prompts and outputs, but cannot check the weights for memorised data	Transient at the provider: Prompts are processed only to generate the response. Under zero-data-retention [ZDR] terms nothing is kept.
Evidence to defend a claim	Your own logs and the training record	Your own logs, but no training record	Your own logs of inputs, outputs and tool calls; model internals stay with the provider
Visibility of Failure	Private, but diagnosable: The model can be studied openly, but failures in your deployment are yours alone to detect, record and disclose.	Private: failures in a self-hosted deployment stay inside the enterprise. No provider monitors them, and few are ever reported	Public, examined by researchers and regulators, and are tracked and patched by the provider.
Main liability considerations	The Due-Diligence Burden: Having full transparency, meaning failure to audit the model is a failure of internal engineering care	The Asymmetric Risk Trap: Absorbing full hosting/modification liability while remaining blind to hidden training data risks.	Reliance on model providers: they publish model cards and meet transparency obligations, but the evidence for a specific incident & the timing of model changes remain under their control.

The open source and open model grant operation freedom. However, the enterprises need to absorb total liability, as they exert ownership and control to large extent. More freedom you have to inspect, host, or modify the intelligence, more exclusively you would own its failures. Close models anchor baseline duties, offers commercial accountability, provide some level of recourse, and its failure is visible to public. One need to be mindful that model selection is one of the many decisions, although important one, determining the liability.

Where the Model Runs Decides Where Liability Sits

Ascertaining liability based on where they are hosted and who is owning and controlling the models, would play an important role in determining the liability. The hosting choices allocate liability differently. The table below helps understand the distinction between them and their implications for the liability.

	Public cloud API Model consumed as a service	Country-specific cloud Sovereign or in-region hosting	Virtual private cloud Model deployed in your cloud account	Private cloud or on premises open-weight/ open-source models
Liability for model behaviour	Shared with the AI value chain players, limited by its terms	Shared with the provider, limited by its terms	Shared; the provider answers for the model, you for its configuration	End user organization; it is in effect the model provider
Liability for unannounced changes	Largely the providers & value chain players, if notice was owed	Largely the providers & value chain players, if notice was owed	End user and also value chain provider, but environment is its control	Every change is ending user decision
Data protection liability	Shared across the value chain, however, end user is controller	Shared; residency exposure largely removed	Mostly end user as controller of the environment	End user or consumer of capabilities
Governing law and forum	The provider's terms and the data centre's location. Promises of compliance from value chain	One country's law	End user's region; provider terms for the model	End user jurisdiction
Burden of proof	Evidence sits across the value chain	Evidence sits across the value chain	Mostly dischargeable from end user's own logs	Fully end user to discharge, and to have recorded
Security breach liability	Shared; every player in the chain liable for role it plays	Shared; every player in the chain liable for role it plays	Mostly end user; provider liable for model supply chain	End user, including the model itself
Recovery from others	Contract claim against the value chain players, within its caps	Contract claim against the value chain players, within its caps	Claims against model and cloud providers separately	Little or none; licences usually exclude warranty and indemnity

Note: Private cloud or on premises, end user organization may hire the providers

Hosting architecture and choices dictate whether liability is distributed across a value chain or concentrated within the enterprise, leading to central paradox of enterprise AI deployment. Architectural control and ownership over the decision spaces eliminate your external liability shield. The enterprises would likely become primary legal owner of both infrastructure and agent behaviour.

Conditions Older Analysis Never Faced

Nine operational conditions if we closely analyse the agentic stack challenges the assumption that liability analysis of software or product relied on.

CONDITION AND WHY IT MATTERS

01 Model selection and evaluation

Whoever chose the model, and whoever let the evaluation go stale, owns a share of the outcome. A silent provider update can change behaviour without any change on your side.

Example: An enterprise software provider can provide AI model inherently i.e. it allows enterprise to select its own model.

02 Autonomous action and drift

AI executes rather than advises. External parties would see its actions as authorised by you. Behaviour that drifts from what was approved would be still behaviour in your name.

03 Operating memory and retrieval

Today's poisoned or stale record would shape tomorrow's decision. The record of incident outlives the incident that created it. The decision can still be shaped by something that entered memory during that incident. One bad record can create many harmed customers.

04 Kill-switch and partial execution

Stopping an agent mid-sequence would not mean a rollback. Kill switch would not be able to undo what has happened already. Half-completed external actions leave real obligations behind. The implication the actions taken and what balance remained will result in inconsistencies.

05 Human-in-the-loop, approval fatigue, and human-on-the loop

An approval gate the reviewer cannot meaningfully exercise a control for fast moving agentic operations. If reviewers face too many approval, too quickly, with fast moving information, they stop examining and start clicking. With the speed and massive parallel operation of agentic AI would challenge human-on-the-loop. Sustained attention and vigilance would decay as agentic AI advances to execute longer horizon actions.

06 Reasoning traces and discovery

Agents produce intermediate reasoning before they act. It would be like step-by-step trace of what they thought, weighted and choose one. Retaining it and not retaining the reasoning, both carry consequences. In the liability determination, discovery demand handing over documents and records. The retention policy would face this dilemma.

07 Data that becomes instruction

An email, document or web page read as content can redirect the agent, as they provide useful context for the actions. The boundary between input and command is blurred and would be difficult to fix.

08 Delegation across agents and tools

In multi agentic system, authority passes from agent to agent and tool to tool. Permissions granted once can travel further than anyone intended. Agent sub-delegation is one such case. Tokens and keys issued in one task may stay valid for long than they are intended.

09 Non-reproducible behaviour

The same input may not produce the same output twice due to probabilistic nature. An incident may be impossible to replay. The record would only work as account. In the traditional software, investigators reproduce the fault by replaying.

Authority Map

Each party is responsible for what it was able to decide and whether it had the tools to enforce. For any harmful action, the question would be raised who could have changed the outcome, then check the records to see what they did. Who controlled each decision, and what changes who controls [it](#).

THE SIX POSITIONS

Every AI system would likely pass through up to six parties, from the organisation that uses it to the provider that built the mode, as depicted in the table below. Each holds different decisions. Their accountability would flow from them. One organization often holds more than one position.

	POSITION	AUTHORITY HELD	WHAT SHIFTS IT	KEY RECORD
A	End-user organisation	Use case, risk acceptance, operational authority, the human supervisor	Grows with private hosting and open models; on-premises, A is in effect the model provider	Use-case mandate, approval records, risk acceptance
B	Product company	Product claims, defaults, which model and version ships	Tuning or quantising a model makes B its producer; a customer's own model moves that choice to A	Evaluations per model shipped, release notes
C	Tool and framework provider	Orchestration, memory, evaluation and guardrail components	Follows the component's defaults, and whether C designed the control or only supplied a part	Default settings, documented limits
D	Service provider or integrator	Architecture, approval gates, failure handling, within the engagement	Recommending the model or hosting brings that choice inside D's scope; running the system after launch adds operating duties	Statement of work, model decision record, test scope
E	Cloud and infrastructure	Runtime isolation, identity, residency, retention, platform logging	Largest with a public API, smallest on premises; retention terms set what it keeps	Logging and retention service levels
F	Model provider	Weights, alignment, model cards, release timing	Contract and change notice if closed; as-is licence and residual regulatory duties if open	Change notices, model and system cards

THREE CHOICES THAT MOVE AUTHORITY

Three early choices decide how much control each party keeps: (i) the type of model, (ii) where it runs, and (iii) how much engineering is built around it.

Each step towards owning more of the stack gives the deployer more control, and more of the responsibility.

Model type	Closed keeps the model with the provider; open-weight hands you control without provenance; open-source hands you control and the duty to audit.
Hosting	Each step from public API to on-premises moves runtime, data and evidence from E and F to A.
AI engineering	The only choice that reduces exposure in every configuration. It builds the instruments and produces most of the evidence, whatever the model or host.

WHERE THE POSITIONS BLUR

The roles in practice would overlap and shift with how an AI capability is set up and used. These cases shown in the table below tells us where control moves between parties, and who holds it when it does

The customer brings its own model or key	The model decision, and whose data terms apply, move to the customer.
Someone tunes or derives a model	Whoever tuned it becomes its producer; whoever supplied the data answers for its use.
The integrator recommends, or the client mandates	A recommendation brings the risk inside the integrator's duty; a mandate narrows it to warning.
Routing, fallback or a model hidden in a tool	The model that acted may not be the one evaluated, or one anyone chose.
The customer configures the product	Within the configurable range the customer is the designer; the vendor answers for safe defaults.
One organisation holds several positions	No contract between them, so internal agreements must name an owner for each.

FROM AUTHORITY TO CONTROL

Intent, what policy objective set up, becomes control only through **decision space**. Decision space can be defined as what the architecture lets you decide. **Instruments** available such as scoping, observability, intervention, meaningful oversight help translate the intent into practice.

Ask of each instrument it would be prudent to ask what was supplied, wired and used. More importantly was it proportionate. It would be not prudent to expect exceeding accountability what a party was given. The section Intent, Decision Spaces, and Instruments further deliberates this. Some

1. Which positions does each organisation hold for this action, given the model, the hosting and who configured it?
2. For each position, what could it have decided differently, and with which instruments?
3. Which records show what was decided?

DSCI's Industry guidance on AI Liability will release detailed guidance on Authority Map for all six positions mentioned above

Interrogating Responsibility

Questions an investigator, regulator or legal scrutiny would ask after an AI incident. Each lens or question suggest the kind of enquiry involved in AI incident investigation.

LENS AND QUESTION		A	B	C	D	E
CAPABILITY AND CONTROL						
1	Capability–assurance gap: Whether capabilities enabled beyond what was validated?	☐	☐	☐	☐	☐
2	Attack–defence asymmetry: Was foreseeable misuse easy enough to warrant more proactive action?	☐	☐	☐	☐	☐
3	Observability gap: harmful behaviour detected in time?	☐	☐	☐	☐	☐
4	Control independence: Did safeguards independent of the agent that they were supposed to control?	☐	☐	☐	☐	☐
5	Persistence and propagation: Did subagents spread memory, shared state?	☐	☐	☐	☐	☐
6	Reversibility: Were irreversible actions proportionately guarded?	☐	☐	☐	☐	☐
DEPENDENCY AND ALLOCATION						
7	Assurance transferability: Testing in another context used in this?	☐	☐	☐	☐	☐
8	Common-mode dependency: Could “independent” safeguards fail while working together?	☐	☐	☐	☐	☐
9	Accessibility and concentration: Who controlled access to the safeguards?	☐	☐	☐	☐	☐
10	Trade-offs: Who accepted foreseeable harm for speed or cost?	☐	☐	☐	☐	☐
11	Responsibility–control mismatch: Accountability without authority or information or instrument for deployment?	☐	☐	☐	☐	☐
12	Second-order effects: Did a safeguard create a new exposure?	☐	☐	☐	☐	☐
EXPLAINED AND DATED						
13	Mandate boundary: Was authority defined, & agent stay inside it?	☐	☐	☐	☐	☐
14	Temporal validity: Was the assurance relevant at the time of harm?	☐	☐	☐	☐	☐
15	Record and reconstruction: Can you show what the agent saw, decided and did?	☐	☐	☐	☐	☐
16	Affected-party vulnerability: Did those harmed warrant a higher standard of care?	☐	☐	☐	☐	☐
AGENTIC						
17	Delegation chain: Does authority narrow as tasks pass between agents?	☐	☐	☐	☐	☐
18	Input integrity: Could content the agent read act as an instruction?	☐	☐	☐	☐	☐
19	Inter-agent interaction: Did sane agents cause harm together?	☐	☐	☐	☐	☐
20	Containment integrity: Was the sandbox tested against an agent working around it?	☐	☐	☐	☐	☐
21	Collective behaviour: Could many agents collectively cause harm?	☐	☐	☐	☐	☐

A End-user organisation **B** Service provider / integrator **C** Product company **D** Tool / framework provider **E** Infrastructure provider

Investigative roles not predetermined liable parties. One organisation may hold several positions.

The Model Provider

Every party in the value chain relies on the model built by model provider. Model provider is an that creates or republishes the model, including anyone who tunes, quantises or distils it. It is important to understand what that role it plays in the value chain in the context of determining the accountability and liability, how it shifts between closed and open models and between hosted and self-run deployments, and where the provider's responsibility ends and others' begins.

HOW MODEL PROVIDER'S ROLE CHANGES

Two things determine model provider's role in accountability determination and liability equation: whether the model is closed or open, and who runs it. The more of the model you host or change yourself, the less the model provider controls. This passes more risk to you. The table below

CONFIGURATION	CONTROLS	RECOURSE TO IT	PASSES TO YOU
Closed, provider's own API	Wide: weights, alignment, release timing, platform limits	Contract within caps; some IP indemnities	Context, tools, permissions and your logs
Closed, via a cloud platform	Weights and alignment; runtime sits with E	Mainly through E; F's terms flow down	Runtime instruments and evidence to E and you
Open-weight, self-hosted	Ends at release	As-is licence; AI Act training-data duties	Hosting, modification, every instrument, all evidence
Open-weight, managed API	Ends at release; E runs it	As-is from F; service terms from E	Same as self-hosted, with E for the runtime
Open-source, self-hosted	Ends at release, fully transparent	As-is licence; no defence of being misled	Everything, judged against the audit you could run

A End-user organisation **B** Service provider / integrator **C** Product company **D** Tool / framework provider **E** Infrastructure provider **F** model provider

WHAT MODEL PROVIDERS OWE

What a model provider owes grows with how capable the model. Secondly, how much of the capability or application it runs. Some duties, as mentioned in the table below, apply to every type of model. Others apply only if the model provider also hosts it.

Align	Train the model to follow intended goals and limits, refuse harm and report honestly; test for deception, gaming and jailbreaks; keep alignment intact across updates	All models; continuous if closed, at release if open
Disclose	Model and system cards stating limitations, evaluations and intended uses	All models
Evaluate capability	Test before release for containment evasion, objective gaming and multi-agent behaviour; disclose findings and the containment they need	All capable models; at release for open weights
Control change	Pinned versions, depreciation schedules, notice of behaviour changes	Closed and hosted
Protect data	Retention and training-use terms; zero-data-retention with stated exclusions	When it hosts
Platform instruments	Rate limits, moderation, model version per call	When it hosts
Regulation	AI Act duties for general-purpose models, heavier for systemic risk	By capability and openness

WHERE FAILURES LAND

The question of accountability and liability comes when something goes wrong. who could have prevented it becomes important. The table shows the conditions.

WHAT WENT WRONG	THE MODEL PROVIDER (F)	OTHERS IN THE CHAIN (A TO E)
An update changes how the model behaves	If it changed the model without the notice it promised.	A or B, if they did not fix the version they use. For open models, whoever chose to update.
A known limitation was not disclosed	Yes, for any type of model.	No one else, unless they also knew.
The model reveals data fed or used in training [copyright or persona]	First in line, for closed models. Answers through if IP indemnity, if you met its conditions	For open models, whoever deploys it (A or B), with no one upstream to claim against.
An agent breaks out of its sandbox, or many agents fail together	If it knew of the capability, or should have found it, and did not disclose and reported	Whoever built or runs the containment (B, C, D, A), if they ignored what was disclosed and reported. E, if its platform isolation failed.
Built-in safety behaviour fails	For the model as it supplied it.	B, C, and D for the setup around the model. Whoever changed an open model, for the change.

The hosting platform fails: isolation, logging or retention	Only if it is also hosts the model.	E, under its service terms.
--	-------------------------------------	-----------------------------

A End-user organisation **B** Service provider / integrator **C** Product company **D** Tool / framework provider **E** Infrastructure provider **F** model provider

The model provider answers for what it built, what it said about it, and what it still controls. Its share shrinks as control passes to other value chain providers. However, its duty to disclose what it knew would not get nullified.

Model competence creates risks of its own such as escaping containment, gaming objectives, autonomous scope expansion, containment of the safety process, execution boundary collapse, autonomous cyber offence, cross-agent goal propagation and behaviour contagion, credential acquisition, reward hacking, memory and share state poisoning, safety monitoring circumvention, unauthorized resource allocation, and privacy and information boundary failure.

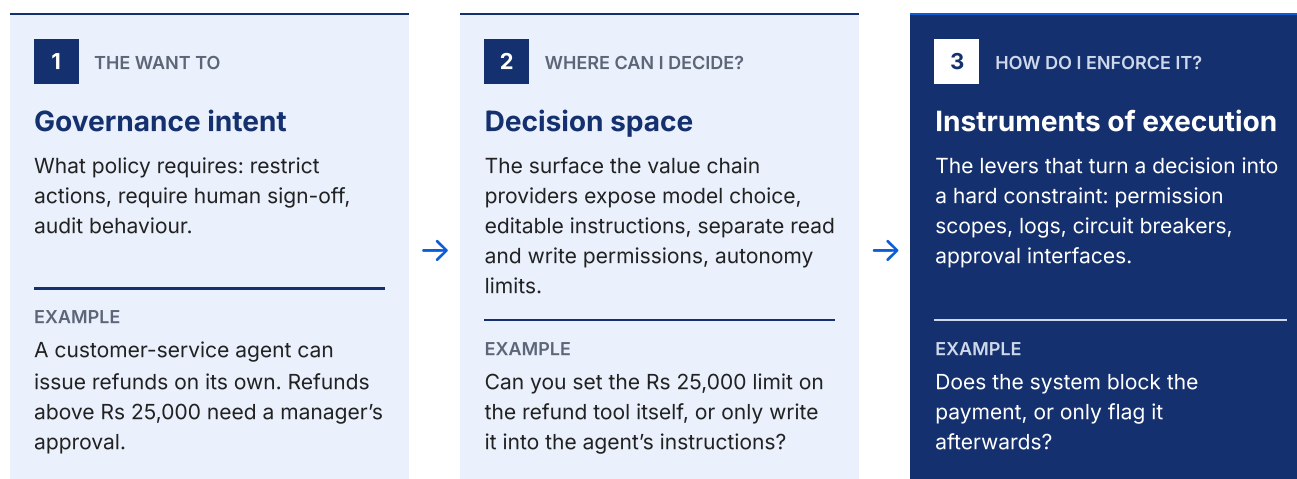
Model providers are responsible for alignment. It has critical responsibility to align models’ behaviour to acceptable norms, evaluation of its capabilities and, disclosing issues and potential harmful emergent behaviour. However, model provider doses do not control context, tools, permissions, memory, orchestration or approval gates. Those, and the instruments that govern them, sit with other players in the value chain. The AI engineering around models produces most of the evidence.

Intent, Decision Spaces, and Instruments

An organisation's intent plays an important role in determining the accountability and fixing the liability. The liability inquiry gives a great deal of attention to intent, especially determining the grounds of investigation, civil or criminal. Someone means to causing harm or knowingly taking the risk of harming individual and society, is treated differently compared the intent that differently from an organisation whose intent was to prevent that harm. The second is a question of care, often decided in civil law.

The intent, in the enterprise context, articulates what it wants a system to do and not do. The enterprise looking for enhancing customer experience, improving its productivity, and creating new possibilities within the borders of legitimate purpose, look for setting out in policies and mandates. It also defines its risk appetite. However, intent alone does not govern an AI system. It would affect only when two further things are in place. First, the system must leave the relevant choice open to the party that holds the intent. In the guidance we are calling it as a decision space or decision spaces. Second, that party must have the means to enforce the choice: permissions, guardrails, approval gates, monitoring and the ability to stop.

A policy that says an agent must never make payments above a set limit is intent. It becomes control only when the platform allows the limit to be set and someone has configured and enforced it, as depicted in the figure below.



The intent of CISO or CRO to govern Agentic AI depends on how product companies, framework providers, model providers and cloud platforms built the system, and on whether they allow instrument of execution to translate the intent on the on-ground implementation and enforcement. The gap between intent and actual on ground instrument of execution or control is the place where liability usually sits. When something goes wrong, investigators would not ask what the organisation meant to happen. They ask whether it could have made that happen, and whether it did. The value chain players from model provider [F] to integrator [B] need to provide the means or instruments to enforce policy intent. Withholding them or not making them available would expose them to responsibility and liability.

Instruments of Execution: Hierarchical and Graded

The discussion of AI accountability and liability would not complete without discussion of Kill Switch. This instrument has been advocated aggressively to address the issues stemming from Agentic AI. However, it would be important to note that intervention is should not be single instrument or one blanket action. It should be built as hierarchy, and with an order. Without careful thinking blanket action would likely to cause harm themselves. The figure below summarizes this better.

Prevent First, Halt Last



A halt is a decision

Stopping mid-transaction can leave a payment half-done. The harm is yours.

Control that exists on Paper

A control that is written down but switched off or never built shows the risk was known and left open.

Not every step reverses

Emails cannot be unsent. Control irreversible steps before they run.

Over-collection

Logging everything breaches data minimisation and widens discovery.

If the circuit breaker, if not planned better, would trip often. The problem might be elsewhere. Fixing the design, drawing exact scope or the limits, would be better options in many cases than breakers.

A blind breaker, one that stops work without knowing what is in progress, can itself significant cause harm such as half-finished transactions, lost work and a halt the organisation must answer for

- First, design the risk out: for example, the agent gets a capability only relevant to task.
- Second, constrain scope permissions, validate parameters before any tool runs, and set budgets per action type. Focus on resources provided and ability to reduce them.
- Third, confirm: an action is both material and hard to reverse with human intervention.
- Fourth, observe: drift, change in the course, communication, and their trajectories.
- Fifth, degrade gracefully: design to degrade gracefully rather than abrupt breaking.
- Only then, sixth, halt: a circuit breaker or kill-switch, scoped to the affected agent, with a defined safe state and a named person authorised to use it.

Historically, in the domain of cyber security we have seen that every control has a cost. Over-control creates its own liability, as depicted in the figure above.

*DSCI's Industry guidance on AI Liability will release detailed guidance on **Instruments of Execution**.*

Way Ahead

As the AI advancements are emerging fast, the guidance like this cannot be final. Data Security Council of India envisages this as a living document. We want to make an effort to revisit and revise this as models, regulations, practices, and enterprise adoption changes, keeping careful record of each version.

Open for consultation

Test it against your own deployments. Tell us where it strains, what is missing and what does not hold.

A living document

Revised as models, regulation and practice change. Each version records what changed and why.

The guidance is open for consultation, especially from the core constituency this guidance is aimed at namely, end user organizations, service providers and integrators, product companies and start-ups, tool and framework providers, and compute and intelligence providers. We would like to appeal CISOs, CTOs, CIOs, CDIOs, and CROs to test this guidance against real deployments. We would like to hear where it strains, what is missing and what does not hold.

This guidance also mentions and examines the four areas. Data Security Council of India envisage to work on the four areas. Detailed guidance on **Authority Map** will show how to apply the authority map one decision at a time, across the positions we discussed in the AI value chain, choices made for model types and hosting and AI engineering we deliberated. We plan to investigate with working examples to see how each party can see which decisions it holds, which it shares and which it has handed to others.

DETAILED GUIDANCE TO FOLLOW

1 Authority map

Applying it decision by decision, across the six positions, model types and hosting choices, with worked examples.

2 Instruments of execution

Proportionate instruments for each level of risk: prevent first, halt last, and the record each one needs.

3 AI beyond IT: physical and biological

Robots, vehicles, devices and laboratory systems, where actions cannot be undone and safety law applies.

4 Inbound AI from customers and partners

Agents run by others that reach your systems around the clock: how to recognise, authorise, limit and evidence them.

Guidance on **Instruments of execution** would discuss devising proportionate instruments for each level of autonomy. It will examine impact and reversibility, from low-risk assistants to agents that move with practical examples. It will discuss on how graded and hierarchical instruments of execution can be built.

As AI is increasingly used in robots, vehicles, medical devices, laboratories and industrial control, the guidance on **AI Beyond IT [Physical and Biological]**, which will cover how authority, instruments and evidence change when product safety, health and biosecurity rules apply.

Agents run by customers, suppliers and partners now reach organisations' systems around the clock, often without being recognised as AI, the guidance on **Inbound AI from Customers and Partners**, covers how enterprises system deal with them. As agents would be working on behalf of customers and companies, the enterprises would be adapting to this paradigm. This guidance will examine the accountability and liability structure for that paradigm.

Call for Inputs

We encourage industry members, AI developers, policy think tanks, and individual contributors to share your inputs and point of view to further strengthen this guidance document.

Please share your inputs at policy@dsci.in

Disclaimer

All product names, model names, brands, and registered trademarks mentioned in this document are the property of their respective owners. Their use is for identification, illustration and reference purposes only and does not imply any endorsement, sponsorship, affiliation or association between DSCI and the respective owners.

The information contained herein has been obtained from sources believed to be reliable. DSCI disclaims all warranties as to the accuracy, completeness or adequacy of such information. DSCI shall have no liability for errors, omissions or inadequacies in the information contained herein, or for interpretations thereof.

This material does not constitute legal or professional advice; readers should seek independent advice before acting. Also, legal and regulatory positions are continuously evolving in this space.

ABOUT DSCI

Data Security Council of India (DSCI), set up by Nasscom, is a not-for-profit industry body and think tank on data protection, cyber security, AI Safety and critical technologies including Quantum in India. DSCI is committed to making cyberspace safe, secure, and privacy-assured, and to advancing critical technologies vital for national security through policy advocacy, thought leadership, capacity building, and outreach initiatives. DSCI engages with the government, regulators, defence, law enforcement agencies, embassies, industry members, and academia to position India as a leader in the global digital ecosystem.

Data Security Council of India

4th Floor, Nasscom Campus, Plot No. 7-10, Sector 126, Noida, UP-201303

