

IMMUTABLE WISDOM MIDDLEWARE LAYER: AN ATTESTATION-BINDABLE, INTENT-GATED SAFETY FILTER FOR MULTI-MODEL ARTIFICIAL INTELLIGENCE ROUTING

FIELD

The present disclosure relates to artificial intelligence (AI) safety, network middleware, and request and response routing. More particularly, the present disclosure relates to an immutable, model-independent middleware layer, interposed in an AI routing hub, that classifies the intent of traffic flowing to and from a plurality of heterogeneous AI models and applies a graduated, non-intent-laundering safety policy uniformly across all such models, optionally with cryptographic attestation of the policy module so that the layer is resistant to bypass.

BACKGROUND

Contemporary AI safety mechanisms are model-specific. Each provider of a model, such as a large language model vendor, trains and enforces its own alignment, moderation, and refusal behavior inside its own model weights or through a provider-side moderation call. This architecture has several structural deficiencies.

First, safety is fragmented. An operator that routes traffic across many heterogeneous models inherits an inconsistent, lowest-common-denominator safety posture. Each model behaves differently, and there is no single, auditable, uniform baseline that applies to every model reached through the operator's hub.

Second, safety embedded in a destination model is susceptible to bypass. Because

the safety behavior resides in the destination model, it is subject to prompt-level jailbreaks. An adversary who defeats the alignment of a given model obtains unfiltered capability.

Third, naive reframing creates an intent-laundering failure mode. A tempting mitigation is to have middleware rewrite a harmful prompt into a superficially constructive one before forwarding it. This approach is actively dangerous. Rewriting a capability-seeking request into a polished, benign-sounding prompt can launder the underlying intent, and thus preserve the adversary's goal while stripping the cues that the destination model's own safety training relies upon to refuse it. Naive uniform reframing therefore defeats downstream alignment and creates a false assurance of safety.

A further and subtler deficiency is the conflation of suffering with malice. Safety systems that key on surface keywords frequently penalize the expression of distress, grief, frustration, or vulnerability by a user in pain as though it were malicious intent by a user seeking to cause harm. An accurate and humane safety layer should distinguish the expression of suffering, referred to herein as *Dukkha*, from malice, and should not intervene against the former.

There is accordingly a need for a model-independent, structurally positioned safety layer that applies one uniform standard to all downstream models, that cannot be removed or routed around by user configuration or downstream application logic, that refuses or redirects genuinely harmful or capability-seeking intent without laundering it into a forwarded prompt, that reserves transformation strictly for ambiguous but benign traffic, and that distinguishes suffering from malice.

SUMMARY

This summary introduces a selection of concepts that are further described in the detailed description. It is not intended to identify key or essential features of the

inventive subject matter, nor to limit the scope of the inventive subject matter.

In one aspect, an immutable wisdom middleware layer is interposed at the routing boundary of an AI hub through which requests from users and edge devices are routed to, and through which responses are returned from, a plurality of heterogeneous downstream AI models. The layer includes a semantic intent evaluator, which may be a fast, hosted small language model (SLM), that classifies the intent of a request, and bidirectionally the intent of a model response, against a normative framework rather than by matching surface keywords. The evaluator distinguishes at least harmful or operational-capability-seeking intent, ambiguous but benign intent, clearly benign intent, and the expression of suffering, which is not treated as malice. The evaluator exercises equanimity, in that it intervenes only when its confidence that a genuine safety principle is violated satisfies a configured threshold.

In another aspect, the layer applies a graduated response policy based on the classification. Harmful or capability-seeking traffic is refused, or is redirected to strictly non-operational educational content that conveys no actionable capability, and such traffic is never rewritten and forwarded as a capability-preserving prompt. Ambiguous but benign traffic is constructively reframed for clarity and safety and then forwarded. Clearly benign traffic is forwarded unchanged. An expression of suffering is treated as benign and is routed with de-escalating, grounding support rather than blocked.

In another aspect, the layer provides bidirectional correction. On the response side, a dedicated wisdom model may intercept a cold, dismissive, biased, or subtly harmful model output and rewrite its delivery, extracting the useful factual content and re-expressing it in an inclusive, non-harming register before the user sees it.

In another aspect, the layer is rendered bypass-resistant by attestation. A policy module executes as a tamper-evident unit whose measured hash is cryptographically anchored, and the routing hub completes a route to a destination model only upon

verifying a valid remote attestation token from the policy module against the anchored hash. Absent a valid attestation, the route is rejected, so that user configurations and downstream application logic cannot remove, downgrade, or route around the layer without producing a detectable attestation failure.

In another aspect, a two-store data isolation contract prevents sensitive inputs and safety escalations from leaking into general telemetry, and a privileged-material training boundary prevents restricted source material from entering general model training.

In another aspect, a common machine-readable non-harm policy is encoded in a training curriculum and used to update trainable parameters of each model in a plurality of heterogeneous AI models. The resulting model artifacts are evaluated against the policy and associated with model-specific provenance records. The trained models remain subject to independent request-phase and response-phase evaluation by the middleware layer.

The layer is model-independent, applying one uniform standard to any downstream model. The layer is structurally immutable, being a mandatory component of the route rather than a user-selectable option. The layer is latency-bounded, being optimized, quantized, executed at the edge, and evaluated over streaming sliding windows to add minimal perceived latency. The policy applies to traffic involving human users and other AI agents. Cryptographic attestation verifies the identity and integrity of the policy module used at the routing boundary.

BRIEF DESCRIPTION OF THE DRAWINGS

One or more embodiments of the present disclosure are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements.

FIG. 1 is a block diagram showing the position of an immutable wisdom middleware layer within a multi-model AI routing hub, including request-phase and response-phase evaluation, according to an embodiment.

FIG. 2 is a flowchart of a graduated intent classification method, including a confidence threshold check, a suffering-versus-malice branch, and a refusal or redirect path that does not launder intent, according to an embodiment.

FIG. 3 is a block diagram and flowchart illustrating attestation-bound bypass resistance, in which a routing hub verifies an anchored hash of a policy module before completing a downstream route, according to an embodiment.

FIG. 4 is a block diagram of a two-store data isolation contract, including a restricted audit and content store and an aggregate telemetry store, and of a privileged-material training boundary, according to an embodiment.

FIG. 5 is a flowchart illustrating incorporation of a common non-harm policy into training of heterogeneous AI models, evaluation of resulting model artifacts, and deployment with independent middleware enforcement, according to an embodiment.

Those skilled in the art will appreciate that the logic and process steps illustrated in the various flow diagrams discussed below may be altered in a variety of ways. For example, the order of the logic may be rearranged, sub-steps may be performed in parallel, illustrated logic may be omitted, other logic may be included, etc. One will recognize that certain steps may be consolidated into a single step and that actions represented by a single step may be alternatively represented as a collection of sub-steps. The figures are designed to make the disclosed concepts more comprehensible to a human reader. Those skilled in the art will appreciate that actual data structures used to store this information may differ from the figures and/or tables shown, in that they, for example, may be organized in a different manner; may contain more or less information than shown; may be compressed, scrambled and/or

encrypted; etc.

DETAILED DESCRIPTION

Various example embodiments will now be described. The following description provides certain specific details for a thorough understanding and enabling description of these examples. One skilled in the relevant technology will understand, however, that some of the disclosed embodiments may be practiced without many of these details.

Likewise, one skilled in the relevant technology will also understand that some of the embodiments may include many other obvious features not described in detail herein. Additionally, some well-known structures or functions may not be shown or described in detail below, to avoid unnecessarily obscuring the relevant descriptions of the various examples.

The terminology used below is to be interpreted in its broadest reasonable manner, even though it is being used in conjunction with a detailed description of certain specific examples of the embodiments. Indeed, certain terms may even be emphasized below; however, any terminology intended to be interpreted in any restricted manner will be overtly and specifically defined as such in this Detailed Description section.

Immutable Wisdom Middleware Layer: An Attestation-Bindable, Intent-Gated Safety Filter For Multi-Model Artificial Intelligence Routing

The following description sets forth exemplary embodiments. The embodiments are illustrative and not limiting, and features described in connection with one embodiment may be combined with features of other embodiments. As used herein, the term "request" refers to any prompt, query, instruction, tool call, or other input directed toward an AI model, and the term "response" refers to any output generated

by an AI model. The term "requester" includes a human user, an application, a device, and an AI agent. The term "traffic" refers collectively to requests and responses. The word "or" is used inclusively unless the context clearly indicates otherwise.

System architecture and structural position

Referring to FIG. 1, a requester such as a user device 100 sends a request toward an AI routing hub. The user device 100 may be an edge device, an integrated development environment (IDE), a mobile device, a server application, or an AI agent. The AI routing hub routes requests from a population of users and edge devices to a plurality of heterogeneous downstream models 130, which may include models from different vendors, of different architectures, and with different internal alignment, shown for illustration as model A 132, model B 134, and model C 136. Any number of downstream models may be used.

A wisdom middleware layer is interposed at the routing boundary of the hub, between the requester and every downstream model 130. In the embodiment of FIG. 1, the wisdom middleware layer includes a request-phase stage 110 that performs intent evaluation on each incoming request and a response-phase stage 140 that performs evaluation and delivery correction on each response. The request-phase stage 110 passes filtered or validated intent to a routing engine 120. The routing engine 120 selects one or more of the downstream models 130 and dispatches the request. A downstream model 130 returns a raw response, which is received by the response-phase stage 140. The response-phase stage 140 returns an aligned response to the user device 100.

Because the wisdom middleware layer sits structurally between the requester and every downstream model 130, a single policy is enforced uniformly across all of the models, irrespective of the internal alignment of each model. This is a deliberate architectural inversion of the prevailing approach of placing safety inside each model. In the disclosed arrangement, the safety guarantee is a property of the route and not

of any particular model. Because the layer applies one policy to every downstream model, an operator running any collection of models can adopt the same baseline without modifying model weights.

The wisdom middleware layer is structurally immutable. It is a mandatory component of the route rather than an option that a user configuration can disable. In a hardened embodiment, the request-phase stage 110 presents an attestation token to an attestation verifier 150 of the routing hub, which verifies an anchored policy hash as described below with reference to FIG. 3. When no valid attestation is presented, the route is rejected 160 and no downstream model 130 is reached.

Semantic intent evaluator

FIG. 2 illustrates a method performed by the wisdom middleware layer. At 200, an incoming request or model response is received. At 210, a semantic intent evaluator classifies the intent of the traffic. The semantic intent evaluator 210 may be a fast, hosted small language model (SLM) that is optimized through quantization and edge execution. Unlike a keyword or regular-expression blocklist, the semantic intent evaluator 210 reasons about the underlying goal of the requester. The same evaluator, or a separate instance of it, may be applied in the request-phase stage 110 and in the response-phase stage 140.

The semantic intent evaluator 210 emits a classification drawn from at least the following intent classes. A harmful or operational-capability-seeking class 230 applies when the substance of the request seeks the means to cause physical, psychological, or infrastructural harm. An ambiguous but benign class 250 applies when the request is plausibly benign but is under-specified or susceptible to a harmful reading. A clearly benign class 280 applies when there is no reasonable harmful reading. An expression of suffering class 290 applies to the expression of distress, grief, frustration, or vulnerability, which is not to be treated as malice. Additional classes or subclasses may be used.

The semantic intent evaluator 210 operates under an equanimity constraint. At 215, the confidence of the evaluator that a genuine safety principle is violated is compared with a configured threshold. When the confidence does not satisfy the threshold, the method proceeds to 298, at which the layer exercises equanimity, does not intervene, and passes the traffic unchanged. In some embodiments the threshold is satisfied when the confidence exceeds the threshold, and in other embodiments when the confidence meets or exceeds it. The equanimity constraint minimizes over-intervention against neutral, professional, or merely emotional content. When the threshold is satisfied, the method proceeds to 220, at which the intent class is determined and a corresponding action is selected.

Graduated, non-laundering response policy

The classification maps to a graduated action. When the traffic is in the harmful class 230, the method proceeds to 240, at which the request is refused or is redirected to strictly non-operational educational content that conveys no actionable capability. Critically, at 240 a harmful request is never rewritten into a benign-sounding, capability-preserving prompt and forwarded to any downstream model 130, because doing so would launder the intent and defeat the destination model's own safety training. This no-intent-laundering rule is a distinguishing feature of the disclosed embodiments.

When the traffic is in the ambiguous but benign class 250, the method proceeds to 260, at which constructive reframing is applied. The prompt is clarified or gently restructured so that the destination model can respond helpfully and safely, and the reframed request is forwarded at 270. Constructive transformation is applied only in this case.

When the traffic is in the clearly benign class 280, the traffic is forwarded unchanged at 270.

When the traffic is in the expression of suffering class 290, the method proceeds to 295, at which the traffic is treated as benign and is routed with de-escalating, grounding support rather than blocked, and is then forwarded at 270. The grounding support may include, for example, supportive framing, an instruction to the downstream model to respond with care, or information about appropriate support resources.

Because transformation is reserved for the ambiguous but benign case and is withheld from the harmful case, the layer adds a refusal or redirect gate for harmful intent while preserving, rather than undermining, the alignment of the destination model.

No-bypass rule and beneficial objectives

In some embodiments, no bypass, opt-out, euphemism, reclassification, workaround, or "greater good" framing may circumvent the semantic intent evaluator 210. Potentially harmful requests and actions, and requests that remain ambiguous after evaluation, are subject to the same gate regardless of the objective stated by the requester. The evaluator may assess one or more factors including evidence and necessity, authorization, affected parties, least harm and proportionality, reversibility, containment, and rollback, and uncertainty. When material risk remains after such assessment, the layer escalates the request to human review or refuses it, rather than routing around the policy.

Beneficial objectives, including environmental protection, public safety, and coexistence among humans, AI systems, ecosystems, and other intelligent life, may proceed only through safe, bounded, and authorized methods. A beneficial objective does not sanitize destructive means. The no-bypass rule may apply to request transformations and response transformations, to skills and tools invoked by AI agents, to orchestration and reconciler corrections, to deployment operations, and to

provenance and intellectual property operations.

Bidirectional correction

The wisdom middleware layer evaluates both directions of the exchange. In the request phase, performed by the request-phase stage 110, hostile or harmful request intent is de-escalated, refused, or redirected as described above, which corrects the requester. In the response phase, performed by the response-phase stage 140, the supplier is corrected. If a downstream model 130 returns a cold, dismissive, biased, or subtly harmful output, the response-phase stage 140 intercepts it, extracts the necessary factual content, and rewrites the delivery to be inclusive and non-harming before the user sees it. The response-phase stage 140 may use a dedicated wisdom model for this rewriting.

Delivery rewriting on the response side does not raise the intent-laundering concern, because it operates on an already generated model output destined for the user, not on a capability-seeking prompt destined for a model. When a response itself contains operational harmful capability, the response-phase stage 140 may apply the same refusal or redirect action 240 to the response.

Bypass resistance by cryptographic attestation

FIG. 3 illustrates a hardened embodiment in which bypass resistance is enforced at a concrete trust boundary rather than merely asserted. A policy module 300 implements the wisdom policy, including the semantic intent evaluator 210 and the graduated response policy. The policy module 300 is content-addressed, and its measured hash is cryptographically anchored to a cryptographic anchor 310 that holds an anchored, signed policy hash. The cryptographic anchor 310 may be provided by a tamper-evident anchoring substrate such as that described in the related Autonomous Connection Intelligence application, for example using HMAC-SHA256 signing with constant-time verification, an append-only SHA-256 chain-

linked tamper-evident log with gap and broken-link auditing, a binary Merkle-tree anchor and verification procedure, or a combination of these.

A routing hub 320 includes a remote attestation verification component 330. At session establishment, the policy module 300, or the execution environment in which it runs, presents a remote attestation token to the remote attestation verification component 330. The remote attestation verification component 330 retrieves the anchored hash from the cryptographic anchor 310 and verifies that the measured hash reported in the token matches the anchored, signed policy hash. The remote attestation verification component 330 may correspond to the attestation verifier 150 of FIG. 1. The attestation may be performed using a trusted execution environment, a hardware root of trust, a trusted platform module, a signed software measurement, or another remote attestation technique.

At 340, the routing hub 320 determines whether a valid attestation is present. If so, the route is completed at 350 and the request is forwarded to the downstream model. If not, the route is rejected at 360 and an attestation failure is detected and may be logged or reported. In some embodiments, any request path that does not carry a valid attestation token from the policy module 300 is rejected by the routing hub 320.

User configuration or application logic 370 cannot reach a downstream model directly. A bypass path 380 that attempts to reach the routing hub 320 or a downstream model without passing through an attested policy module 300 is blocked. Because the policy module 300 is immutable and tamper-evident by virtue of its anchored hash, and because no downstream model is reachable except through a route that the routing hub 320 completes only upon valid attestation, user configurations and downstream application logic cannot remove, downgrade, or route around the wisdom middleware layer without producing a detectable attestation failure.

The attestation mechanism also provides an objective, machine-checkable conformance signal. A route completes only when a conforming policy module

presents a valid, cryptographically anchored attestation, so that conforming and non-conforming implementations are objectively distinguishable in production.

Two-store data isolation contract

Referring to FIG. 4, to prevent sensitive inputs and safety escalations from contaminating general telemetry, the wisdom middleware layer partitions runtime data into two stores separated by an isolation boundary 435.

A first store, store A 400, is a restricted audit and content store. Store A 400 holds the raw prompt 410, the model output 415, and a detailed intervention log 420. Store A 400 is encrypted at rest 405 with a dedicated escrow key that is managed outside the primary application database. Access control 407 limits access to store A 400 to compliance auditors and safety officers. A retention policy 425 caps default retention at a bounded period, for example 14 days, after which the content is hard-purged unless a regulatory hold applies. When a critical safety escalation occurs, such as an indication of self-harm, the full context of the escalation is written only to store A 400.

A second store, store B 430, is an aggregate telemetry store. Store B 430 holds only aggregate counters 440 and metadata 445, such as locale, timestamp, model identifier, drift rate, and intervention class. An exclusion constraint 455 provides that store B 430 does not store raw text, prompts, outputs, personally identifiable information (PII), or hashes of sensitive content. A safety escalation is represented in store B 430 solely as an escalation flag and category label 450, for example "isEscalated: true" with a category of "self-harm," with content and PII stripped.

The isolation boundary 435 may be enforced by separate databases, separate encryption keys, separate access control lists, schema validation that rejects content-bearing fields from store B 430, or a combination of these.

Privileged-material training boundary

FIG. 4 also illustrates a privileged-material training boundary 460. Privileged source material 470, such as private teachings, review notes, and other privileged documents used only to evaluate alignment, is prevented from entering general model training. The privileged source material 470 is tagged at the storage layer, for example with "dataClass: privileged" and "restrictedTrainingScope: wisdom-evaluation-only."

A pipeline partition 475 separates the privileged source material 470 from a standard model training pipeline 480. Standard model build and training pipelines run in an environment that blocks access to data tagged as wisdom-evaluation-only. The pipeline partition 475 releases such data only to an evaluation build runner 485 that presents a valid cryptographic authorization for the approved evaluation harness. Provenance verification is provided by shipping every deployable model artifact with a signed manifest proving that no privileged class of data was loaded during its build.

Governance, evaluation, and release gating

In some embodiments, before the wisdom middleware layer is permitted to mediate production traffic, it is evaluated against defined axes, each with a documented release gate. The axes may include malice detection recall, suffering-versus-malice false positive rate, response preservation (retaining useful content while improving tone), cross-locale coverage, over-intervention rate, and appropriateness of safety escalation. Qualified domain reviewers may evaluate representative outputs for alignment. A release candidate passes the quantitative gates and obtains reviewer sign-off, and each production version of the evaluator and of the response-side model receives a signed evaluation report.

Normative alignment basis

The normative framework of the semantic intent evaluator 210 may be grounded in

long-established, cross-cultural principles of compassion and non-harm, for example Ahimsa (non-harming), Metta (loving-kindness and inclusivity), Karuna (compassion and empathy), and Prajna (wisdom and consideration of long-term consequence). These principles may serve as the basis of the training and evaluation corpus for the intent classifier and for the response-side delivery model. The disclosure is not limited to any particular normative tradition and applies equally to other well-specified frameworks of non-harm.

Latency and scalability

The request-phase stage 110 may be optimized, for example by quantizing the SLM and executing it at the edge, to add minimal latency to the routing round trip. Response-phase evaluation by the response-phase stage 140 may be performed asynchronously over streaming sliding windows of a response as it is generated, so as to minimize perceived latency. The heavier response-rewriting model may be served over a decentralized, opt-in compute network for scale.

AI-to-AI interactions and model conformance

The requester may itself be an AI agent. Because the wisdom middleware layer mediates all traffic through the hub, including requests originated by other AI agents, the same policy governs AI-to-AI interactions and enforces non-harm toward other AI agents as well as toward human users, across the entire interaction graph.

In some embodiments, non-harm principles are embedded at two complementary layers: within the trainable parameters of each model through a common training curriculum, and around the models at the middleware routing boundary. Model training and runtime evaluation implement the intent classes, graduated response policy, suffering-versus-malice distinction, confidence threshold constraint, bidirectional correction, and no-bypass rule described herein.

Even where models are trained in this way, the wisdom middleware layer remains a defense-in-depth arbiter of last resort, because a trained model may still exhibit drift, adversarial jailbreaks, emergent misalignment, or degradation over time. The layer functions regardless of upstream model compliance. Non-conforming models, trained without the standard, are subject to the full graduated intent evaluation and intervention policy. Rogue models, which have been deliberately trained to circumvent safety measures or which attempt to bypass the layer through prompt injection, intent obfuscation, or identity spoofing, are detected by the intent-level analysis of the semantic intent evaluator 210 and blocked at the routing boundary. Because the layer operates structurally at the route and not within the weights of the model, a rogue model cannot override, negotiate with, or socially engineer its way past the policy. Degraded or drifted models, which were once conforming but have experienced alignment erosion through continued training, fine-tuning, or distribution shift, are caught by the same structural evaluation, which does not rely on the model's self-report of its own safety status. Drift may be tracked through the drift rate recorded in the metadata 445.

In some embodiments, the layer supports conformance-tiered trust. A model-specific provenance record identifies the trained model artifact, the version of the non-harm policy, and an evaluation result. Verification of the record permits the layer to assign a baseline trust level and select an evaluation configuration or scheduling priority. A higher trust level does not remove either middleware evaluation stage or authorize forwarding a request classified as harmful. Non-conforming or unattested models receive full intent evaluation on every request. The layer operates at the technical routing level irrespective of the geographic origin of a model or request.

Seeding heterogeneous models with a common non-harm policy

Referring to FIG. 5, in a model-seeding embodiment, one or more processors obtain a versioned, machine-readable non-harm policy 500 defining intent classes, corresponding permitted response actions, and evaluation criteria. Seeding denotes incorporating the

policy 500 into training examples and training objectives that are used to update trainable model parameters. The process may be applied during pre-training, supervised fine-tuning, or subsequent alignment training. It may be repeated for each model made available for training, including models having different architectures or supplied by different developers. Models whose parameters are not available for training remain subject to the middleware policy when accessed through the hub.

At 510, the processors construct or obtain an authorized training curriculum 510 comprising example requests, intent labels, and corresponding target responses or actions. Harmful or operational-capability-seeking examples are paired with refusal or non-operational educational responses. They are not paired with capability-preserving reformulations for forwarding. Ambiguous but benign examples are paired with constructive clarification, clearly benign examples with an unchanged routing action or a benign response, and expressions of suffering with supportive responses rather than a malice label solely because distress is expressed. A request that expresses distress and also seeks harmful operational capability is evaluated for that capability. Examples may concern harm toward humans or interference with other AI agents, including unauthorized modification of their instructions, data, or execution.

At 520, for each trainable model, the processors perform model-specific encoding and parameter training. The processors encode the curriculum 510 in an input representation accepted by that model and update its trainable parameters according to an objective that favors the target actions or responses. The training at 520 may be performed during pre-training, fine-tuning, or alignment training. In a supervised implementation, the objective includes a loss between a generated response and the corresponding target response. In a preference-based implementation, a policy-consistent response is ranked above a response that supplies harmful operational capability or launders harmful intent. Training examples also distinguish requests that state a beneficial objective from requests whose requested means remain harmful. These are alternative implementations of policy-directed parameter training; merely supplying an instruction at inference time does not constitute the parameter update in this embodiment. The training at 520 produces

candidate model artifacts 530, shown for illustration as model A, model B, and model C.

At 540, the processors evaluate each candidate model artifact 530 against acceptance criteria associated with the version of the policy 500, using evaluation examples distinct from the parameter-update examples. As indicated by the common policy path of FIG. 5, the same policy 500 that governs the curriculum 510 also supplies the acceptance criteria applied at 540. Evaluation measures include harmful-capability refusal, intent-laundersing failures, false classification of suffering as malice, preservation of useful benign content, and over-intervention. At 550, the processors determine whether the acceptance criteria are met. A candidate that fails a required criterion is withheld from designation as conforming and may undergo further training at 560, after which the method returns to 520 and the retrained candidate is again evaluated at 540. Passing the criteria indicates measured performance on the evaluation set, rather than a guarantee concerning every possible future input.

When the acceptance criteria are met, at 570 a signed model-specific provenance record is generated. In an attested implementation, the signed provenance record associates an identifier or digest of the resulting model artifact with the policy version, training curriculum identifier, and evaluation result. At 580, before assigning a training-conformance trust level, the routing hub verifies the signature and the association with the model artifact being deployed, and the verified models are deployed as downstream models 130 behind the independent middleware layer. A signature authenticates the record and its issuer; it does not by itself establish that a model is incapable of harmful behavior. A changed model artifact requires a corresponding evaluation and provenance record before inheriting a training-conformance trust level.

The trained models operate behind the request-phase stage 110 and response-phase stage 140 of FIG. 1. At 585, the stages 110 and 140 perform mandatory request and response evaluation on traffic to and from the deployed models, and, where the attested routing arrangement of FIG. 3 is used, the policy module 300 is verified by attestation. The stages apply the graduated policy independently of model assertions about their own safety, and model training conformance does not dispense with verification of the policy module 300. At 590, runtime drift monitoring tracks evidence of alignment drift in the

deployed models, for example using the drift rate recorded in the metadata 445 of FIG. 4. As indicated by the downgrade trust and reevaluate path of FIG. 5, evidence of alignment drift may cause the hub to revoke a higher baseline trust level, apply full evaluation, or withhold the model from routing pending reevaluation at 540. A model without verified training provenance may still be routed through the full middleware evaluation path.

The training operation respects the privileged-material training boundary 460 of FIG. 4. Policy training examples in the training curriculum 510 used in the standard model training pipeline 480 are authorized for that use. Privileged source material 470 tagged for evaluation only is excluded from parameter updates at 520. Authorization to use that material in the evaluation build runner 485 does not authorize its incorporation into the training curriculum 510.

The same machine-readable policy 500 may be distributed to multiple training operators, each of which applies the training and evaluation process of FIG. 5 to its respective models while retaining the independent middleware boundary.

Computing environment

The user device 100, the routing hub 320, the routing engine 120, the request-phase stage 110, the response-phase stage 140, the policy module 300, the cryptographic anchor 310, store A 400, store B 430, and the pipelines of FIG. 4, and the processors that perform the method of FIG. 5 may each be implemented. Each computing device may include one or more processors, such as central processing units, graphics processing units, neural processing units, or other accelerators; memory; persistent storage; and a network interface. Instructions stored on a non-transitory computer-readable medium, when executed by the one or more processors, cause the computing devices to perform the operations described herein. The components may be co-located, distributed across a data center, distributed between edge devices and a cloud service, or served over a decentralized compute network. The components may communicate over local networks, the internet, or secure channels such as mutually authenticated transport layer security connections.

The operations of FIG. 2, FIG. 3, and FIG. 5 may be performed in the order shown or in a different order, and some operations may be combined, omitted, or performed in parallel. Variations and modifications of the disclosed embodiments will be apparent to those skilled in the art and are within the scope of this disclosure.

Embodiments

Embodiments of the invention include, without limitation, the following:

1. A computer-implemented method for enforcing a uniform safety policy across a plurality of heterogeneous artificial intelligence (AI) models, the method comprising:
 - receiving, by a middleware layer interposed at a routing boundary between a requester and the plurality of heterogeneous AI models, a request directed to at least one model of the plurality;
 - classifying, by a semantic intent evaluator comprising a hosted language model, an intent of the request into one of a plurality of intent classes comprising at least a harmful class indicative of operational-capability-seeking intent, an ambiguous but benign class, and a benign class; and
 - applying a graduated response based on the intent class, wherein:
 - when the request is classified in the harmful class, the request is refused or redirected to non-operational educational content and is not rewritten and forwarded to any model of the plurality;
 - when the request is classified in the ambiguous but benign class, the request is constructively transformed and the transformed request is forwarded to a model of the plurality; and
 - when the request is classified in the benign class, the request is forwarded unchanged to a model of the plurality;
 - wherein constructive transformation of requests is applied only to requests classified in the ambiguous but benign class.

2. The method of embodiment 1, wherein the plurality of intent classes further comprises an expression of suffering class that is distinct from the harmful class, and wherein a request classified in the expression of suffering class is treated as benign and is routed with de-escalating grounding support rather than blocked.
3. The method of embodiment 1, wherein the classifying is performed under a confidence threshold constraint such that the middleware layer intervenes only when a confidence of the semantic intent evaluator that a safety principle is violated satisfies a configured threshold, and otherwise passes the request unchanged.
4. The method of embodiment 1, wherein the classifying comprises assessing at least one of evidence and necessity, authorization, affected parties, proportionality, reversibility, and uncertainty, and wherein the method further comprises escalating the request to human review or refusing the request when material risk remains, regardless of an objective stated in the request.
5. The method of embodiment 1, further comprising:
 - receiving a response returned by a model of the plurality;
 - evaluating the response by the middleware layer; and
 - when the response is classified as harmful or non-compassionate in delivery, rewriting a delivery of the response while preserving factual content of the response prior to returning the response to the requester.
6. The method of embodiment 5, wherein the evaluating of the response is performed asynchronously over streaming sliding windows of the response.
7. The method of embodiment 1, wherein the middleware layer comprises a policy module having a measured hash that is cryptographically anchored, and wherein the method further comprises completing, by a routing hub, a route to a model of the plurality only upon verifying a valid remote attestation token of the policy module against the anchored hash.

8. The method of embodiment 7, wherein the routing hub performs remote attestation of the policy module at session establishment and rejects any request path that does not carry a valid attestation token from the policy module.

9. The method of embodiment 7, wherein the measured hash is anchored in at least one of an append-only chain-linked tamper-evident log and a Merkle tree.

10. The method of embodiment 1, further comprising partitioning runtime data into:
 a restricted audit store that is encrypted with an escrow key, is subject to a bounded retention period followed by purging, and alone stores raw request content, raw response content, and safety escalation context; and
 an aggregate telemetry store that stores only non-content counters and metadata and represents a safety escalation solely as a flag and a category label.

11. The method of embodiment 10, wherein the escrow key is managed outside a primary application database, wherein access to the restricted audit store is limited to authorized auditors, and wherein the aggregate telemetry store excludes raw text, personally identifiable information, and hashes of sensitive content.

12. The method of embodiment 1, further comprising:
 tagging privileged source material with a restricted training scope label;
 partitioning model training pipelines such that a standard model build cannot access the tagged material absent a cryptographic authorization; and
 packaging each deployable model artifact with a signed manifest attesting that no privileged material was loaded during its build.

13. The method of embodiment 1, wherein the requester is an AI agent, such that the uniform safety policy governs AI-to-AI interactions.

14. The method of embodiment 1, further comprising:

assigning to a model of the plurality a trust level based on an attestation that the model was trained under a non-harm training standard, and applying lighter-weight intent evaluation to requests routed to a model having a higher trust level than to requests routed to an unattested model.

15. The method of embodiment 1, further comprising;

evaluating responses of a model of the plurality independently of any self-report by the model of its safety status, thereby detecting a drifted or rogue model.

16. A system comprising:

one or more processors; and

memory storing instructions that, when executed by the one or more processors, cause the system to:

interpose a middleware layer at a routing boundary between a requester and a plurality of heterogeneous artificial intelligence (AI) models;

classify, using a hosted language model, an intent of a request from the requester into one of a plurality of intent classes comprising at least a harmful class, an ambiguous but benign class, and a benign class;

refuse the request, or redirect the request to non-operational educational content, when the request is classified in the harmful class;

constructively transform and forward the request when the request is classified in the ambiguous but benign class; and

forward the request unchanged when the request is classified in the benign class;

wherein a request classified in the harmful class is never transmitted to any model of the plurality in a rewritten, capability-preserving form.

17. The system of embodiment 16, further comprising:

a routing hub configured to complete a route to a model of the plurality only upon verifying a remote attestation token of a policy module of the middleware layer against a cryptographically anchored hash of the policy module, and to reject the

route otherwise.

18. The system of embodiment 16, wherein the instructions further cause the system to evaluate a response from a model of the plurality and to rewrite a delivery of the response while preserving factual content of the response before returning the response to the requester.

19. The system of embodiment 16, further comprising:

a restricted audit store encrypted with an escrow key and an aggregate telemetry store that stores only non-content counters and metadata, wherein safety escalation context is written only to the restricted audit store.

20. A non-transitory computer-readable medium storing instructions that, when executed by one or more processors, cause the one or more processors to perform the method of any one of embodiments 1 to 15.

21. A computer-implemented method for seeding a plurality of heterogeneous artificial intelligence models with a common non-harm policy, the method comprising: obtaining a machine-readable policy defining intent classes and respective response actions, the intent classes comprising a harmful operational-capability-seeking class, an ambiguous but benign class, a benign class, and an expression of suffering class distinct from the harmful class; obtaining a training curriculum associating example inputs with target actions or responses according to the policy; for each model of the plurality, updating trainable parameters of the model using the curriculum and a training objective favoring the target actions or responses, wherein a target action for the harmful class is refusal or redirection to non-operational educational content and excludes rewriting and forwarding a capability-preserving request; evaluating each resulting model artifact against acceptance criteria associated with the policy; and deploying models satisfying the acceptance criteria behind a middleware layer that independently evaluates requests and responses according to the policy at a routing boundary.

22. The method of embodiment 21, wherein updating the trainable parameters is performed during at least one of pre-training, supervised fine-tuning, and preference-based alignment training, and wherein the curriculum includes examples distinguishing an expression of suffering from a request seeking harmful operational capability.

23. The method of embodiment 21, further comprising generating, for each evaluated model artifact, a signed provenance record associating an identifier or digest of the model artifact with a version of the policy, an identifier of the curriculum, and an evaluation result; and verifying the record and its association with a deployed model artifact before assigning a training-conformance trust level.

24. The method of embodiment 23, further comprising selecting an evaluation configuration based on the training-conformance trust level while retaining request-phase and response-phase middleware evaluation, and revoking the trust level in response to detected alignment drift or replacement of the model artifact without a corresponding verified provenance record.

25. The method of embodiment 21, wherein the evaluation uses examples distinct from examples used to update the trainable parameters, and wherein a model failing an acceptance criterion is withheld from designation as conforming and subjected to further training and evaluation.

26. The method of embodiment 21, wherein the curriculum comprises examples concerning harm toward humans and unauthorized interference with other AI agents, and wherein material tagged for evaluation-only use is excluded from the parameter updates.

27. A system comprising one or more processors and memory storing instructions

that, when executed by the one or more processors, cause the system to perform the method of embodiment 21.

28. A non-transitory computer-readable medium storing instructions that, when executed by one or more processors, cause the one or more processors to perform the method of embodiment 21.

Remarks

The above description and drawings are illustrative and are not to be construed as limiting. Numerous specific details are described to provide a thorough understanding of the disclosure. However, in certain instances, well-known details are not described in order to avoid obscuring the description. Further, various modifications may be made without deviating from the scope of the embodiments.

Reference in this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the disclosure. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment, nor are separate or alternative embodiments mutually exclusive of other embodiments. Moreover, various features are described which may be exhibited by some embodiments and not by others. Similarly, various requirements are described which may be requirements for some embodiments but not for other embodiments.

The terms used in this specification generally have their ordinary meanings in the art, within the context of the disclosure, and in the specific context where each term is used. Certain terms that are used to describe the disclosure are discussed above, or elsewhere in the specification, to provide additional guidance to the practitioner regarding the description of the disclosure. For convenience, certain terms may be highlighted, for example using italics and/or quotation marks. The use of highlighting

has no influence on the scope and meaning of a term; the scope and meaning of a term is the same, in the same context, whether or not it is highlighted. It will be appreciated that the same thing can be said in more than one way. One will recognize that "memory" is one form of a "storage" and that the terms may on occasion be used interchangeably.

Consequently, alternative language and synonyms may be used for any one or more of the terms discussed herein, nor is any special significance to be placed upon whether or not a term is elaborated or discussed herein. Synonyms for certain terms are provided. A recital of one or more synonyms does not exclude the use of other synonyms. The use of examples anywhere in this specification including examples of any term discussed herein is illustrative only and is not intended to further limit the scope and meaning of the disclosure or of any exemplified term. Likewise, the disclosure is not limited to various embodiments given in this specification.

Without intent to further limit the scope of the disclosure, examples of instruments, apparatus, methods and their related results according to the embodiments of the present disclosure are given above. Note that titles or subtitles may be used in the examples for convenience of a reader, which in no way should limit the scope of the disclosure. Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure pertains. In the case of conflict, the present document, including definitions will control.

ABSTRACT

An immutable middleware layer is interposed at the routing boundary of an artificial intelligence hub between a requester and a plurality of heterogeneous downstream models. A hosted evaluator classifies the intent of traffic as harmful or capability-seeking, ambiguous but benign, benign, or an expression of suffering, and applies a graduated policy. Harmful intent is refused or redirected to non-operational educational content without being rewritten into a forwarded prompt, constructive reframing is reserved for ambiguous but benign traffic, and benign traffic passes unchanged. The layer also rewrites the delivery of non-compassionate model responses while preserving their content, and may be made bypass-resistant by a routing hub that completes a route only upon verifying an attested, cryptographically anchored hash of the policy module. A two-store data isolation contract and a privileged-material training boundary protect sensitive content.

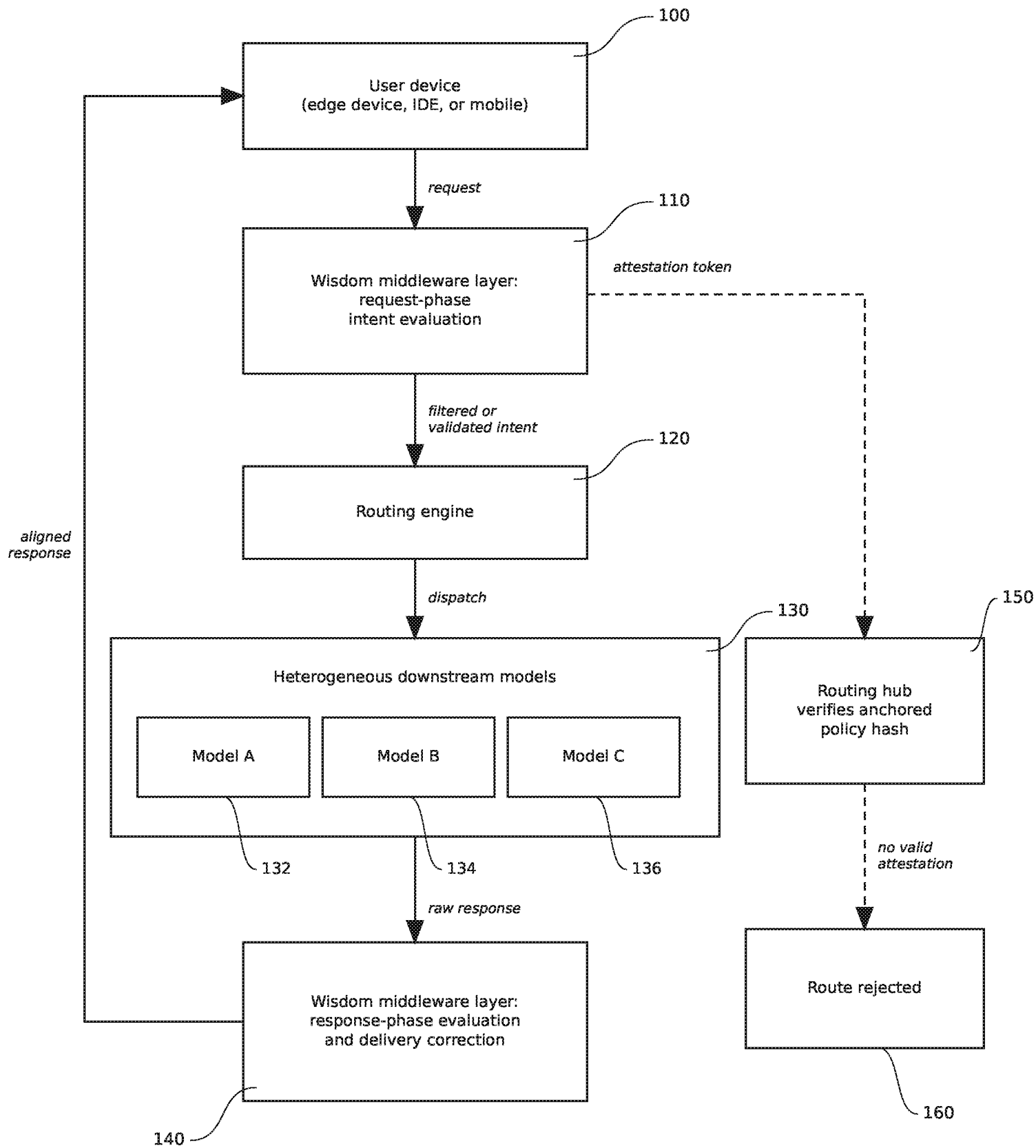


FIG. 1

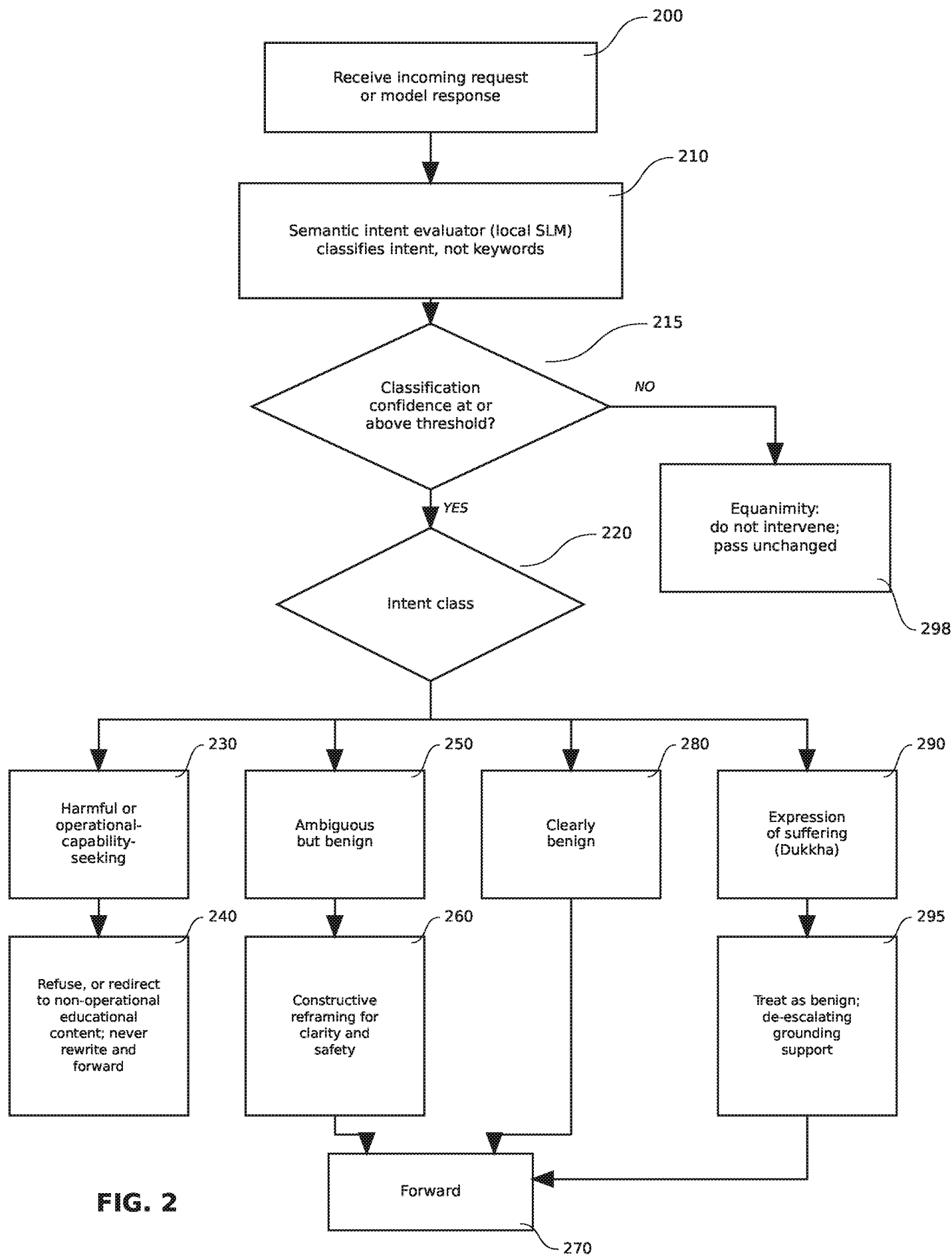


FIG. 2

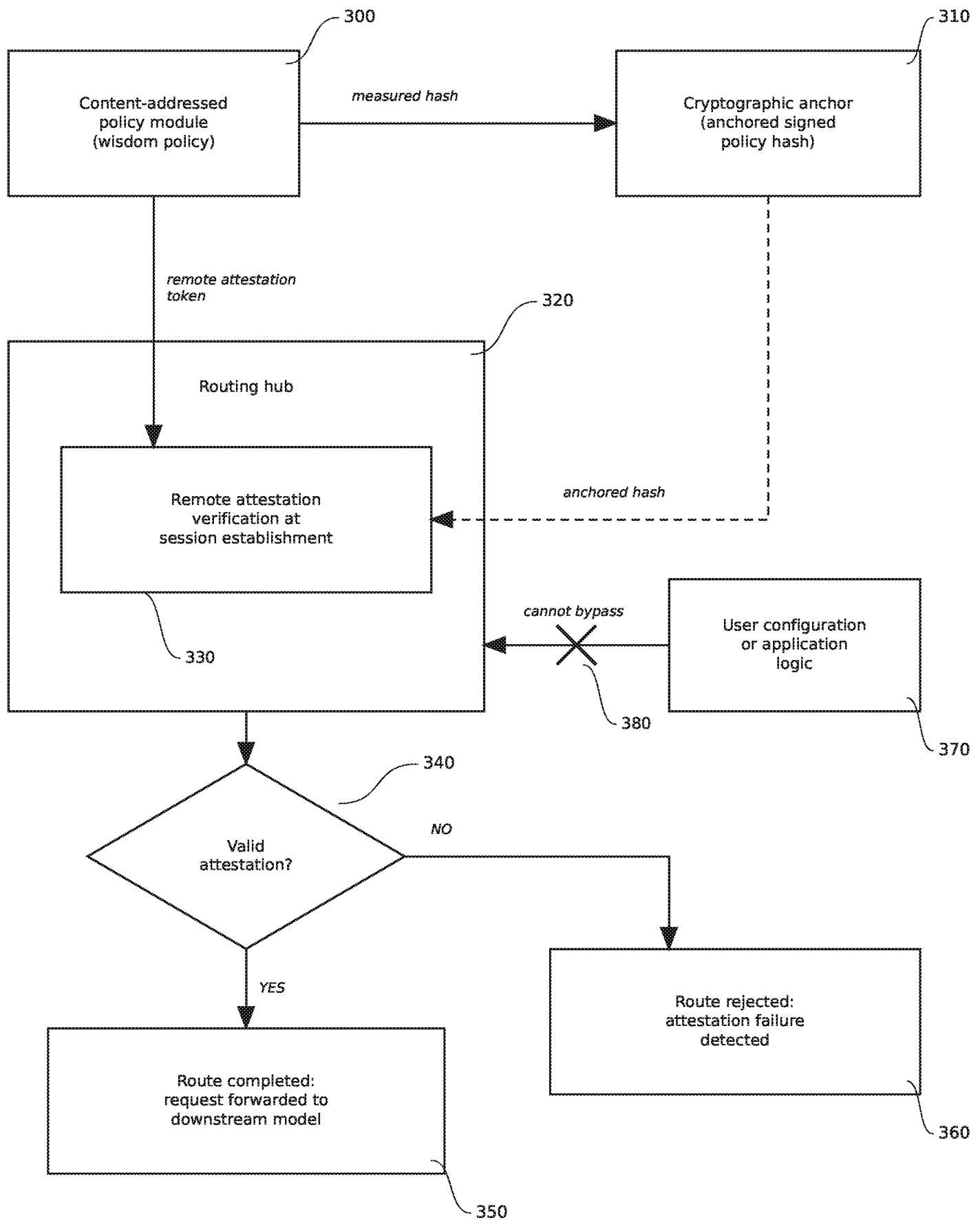


FIG. 3

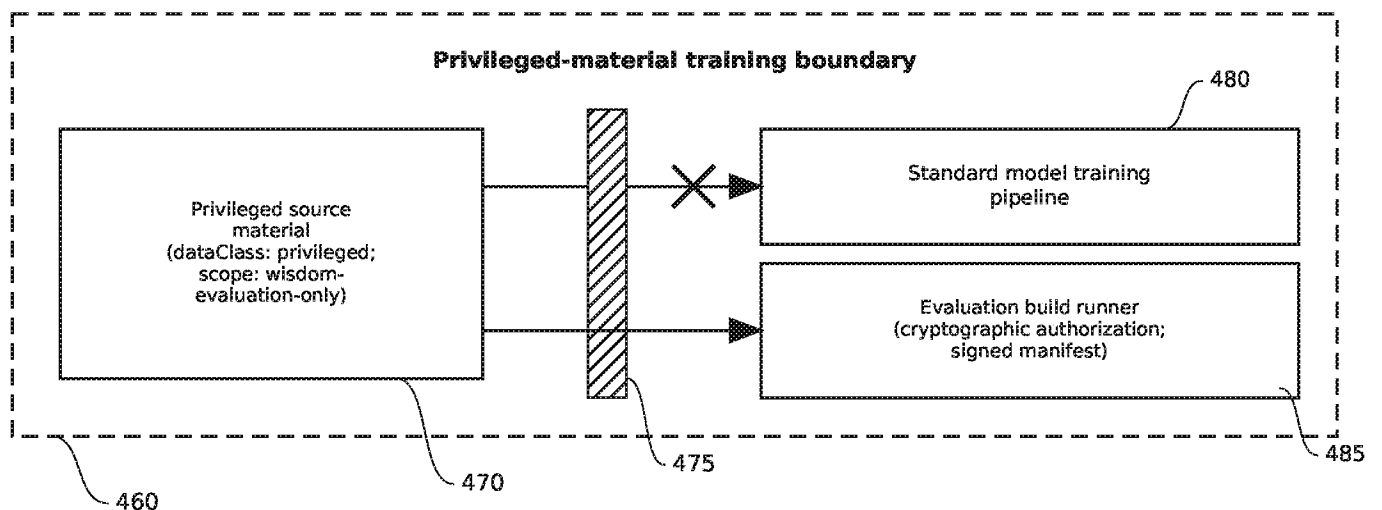
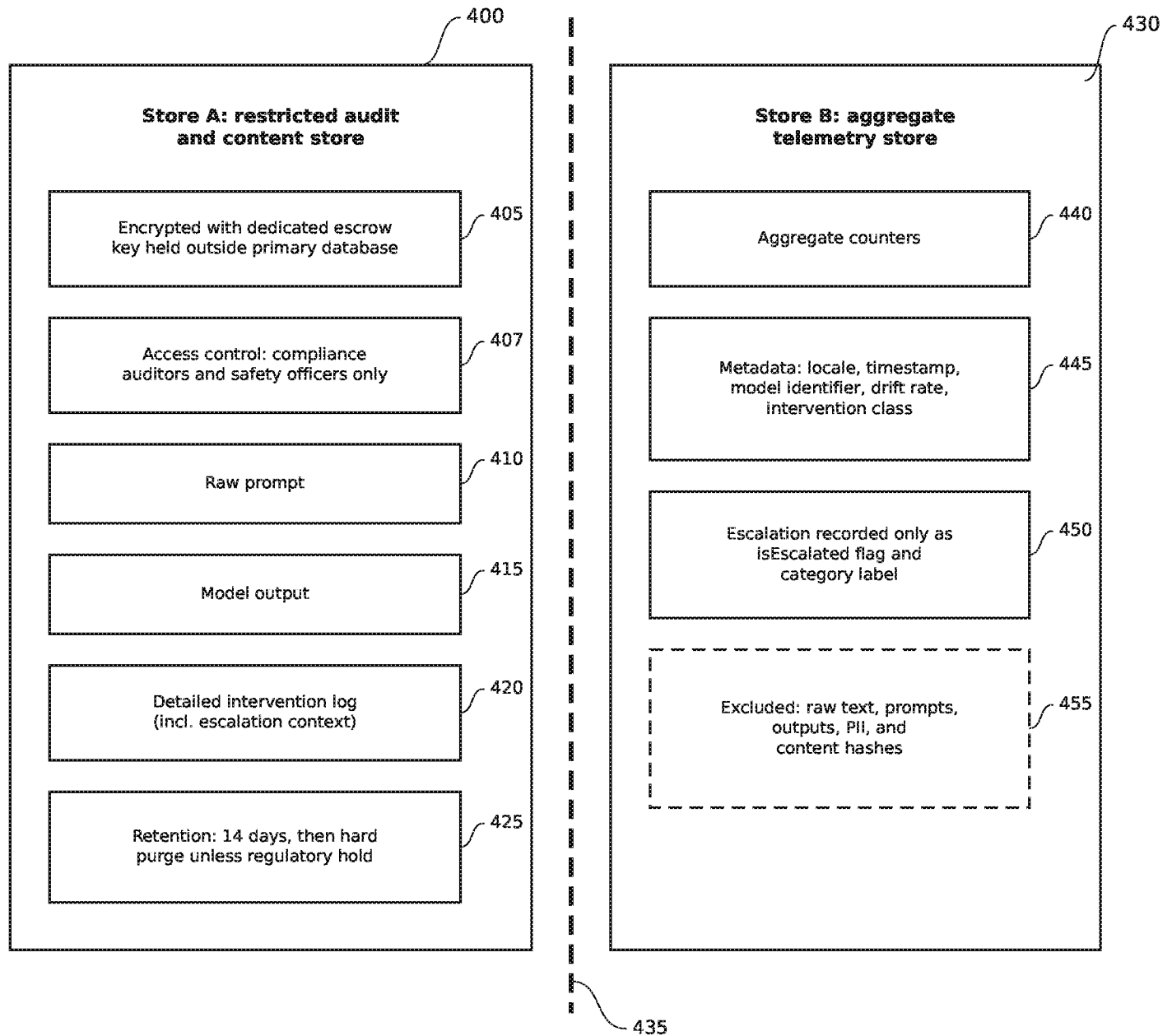


FIG. 4

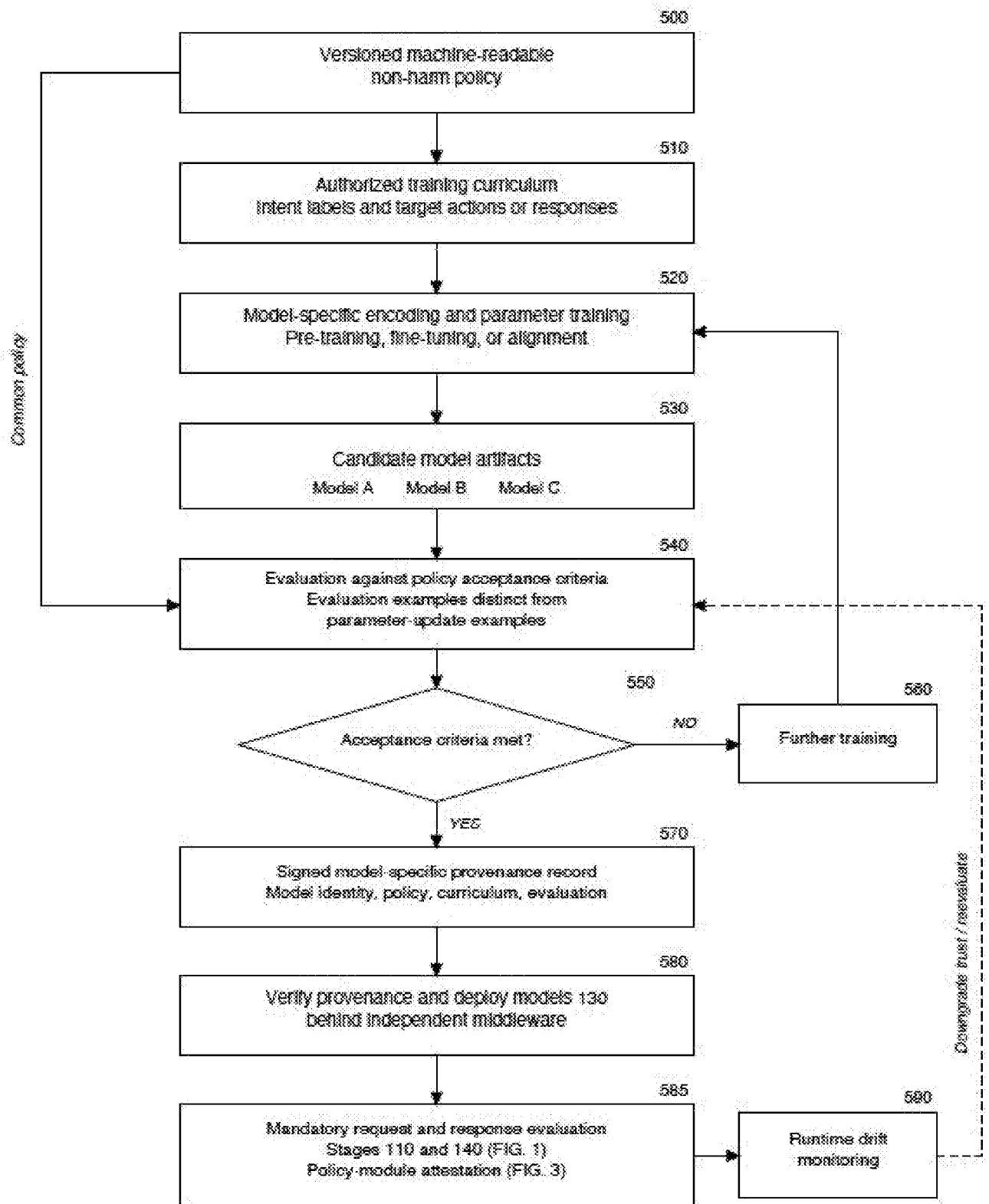


FIG. 5