AUTHORITY GOVERNED LEARNING

AGL™

Governing the Learning Layer in IFA™ Systems

A Formal Extension to

Intelligence From Architecture (IFA)

Core Specification v1.0

Kamilla Harcej

April 2026

Specification Draft v0.1

Normative Status

This document defines normative requirements for AGL compliance. Requirements are stated using MUST, MUST NOT, SHALL, SHALL NOT, SHOULD, and MAY. A system claiming AGL compliance must satisfy all MUST and SHALL requirements. This specification is an extension to IFA Core Specification v1.0 and does not supersede it. Both specifications must be satisfied for full AGL + IFA compliance.

IFA Dependency

This specification presupposes familiarity with and compliance with Intelligence From Architecture (IFA) Core Specification v1.0 by Michal Harcej. All IFA requirements remain binding. AGL adds requirements in the domain of machine learning, state mutation through training, and governed adaptation. Where this specification is silent, IFA governs.

Table of Contents

Part I — Normative Foundation
1. Purpose of This Specification
2. Scope and Applicability
3. Non-Goals
4. Normative Language
5. Compliance Model

Part II — The Missing Layer in IFA
6. The Uncontrolled Dimension: Learning
7. Learning as State Transition (Normative Anchor)
8. The Failure of Ungoverned Learning

Part III — Core Principles of AGL
9. Learning as Permission, Not Capability
10. Admissibility of Learning
11. Epistemic Validity
12. Authority Over Evolution
13. Non-Inferential Authority (IFA Extension)

Part IV — AGL Architecture
14. The Learning Gate (Core Mechanism)
15. The State Validity Layer
16. Authority Kernel for Learning
17. Deterministic Learning Eligibility
18. Learning Execution Model
19. Defensibility of Learning
20. Continuation Governance

Part V — Failure, Refusal, and HOLD
21. Learning Refusal
22. HOLD State
23. Learning Failure Semantics

Part VI — Integration with IFA Systems
24. AGL + Deterministic Core
25. AGL + CKG
26. AGL + Advisory Intelligence
27. AGL in Distributed / Autonomous Systems

Part VII — Compliance & Certification
28. AGL Compliance Model
29. Auditability of Learning
30. Prohibited Architectures

Part VIII — Strategic & Real-World Context
31. Historical Failure Patterns of Ungoverned Learning
32. Regulatory and Systemic Implications

Appendices
- A. Reference AGL Architecture
- B. Learning Decision Trace Examples
- C. HOLD vs Refusal vs Failure
- D. Mapping AGL → IFA

PART I

Normative Foundation

Section 1 — Purpose of This Specification

IFA Core Specification v1.0 defines a closed architectural framework in which purpose, authority, governance, security, explainability, and failure are enforced structurally rather than managed reactively. It establishes that intelligence may advise but architecture must decide.

However, IFA does not address one critical dimension: learning. Specifically, IFA does not govern what happens when a system modifies its own state through training, feedback incorporation, reinforcement, or any other mechanism that changes how it will behave in future decisions. This gap is not an oversight in IFA — it is a boundary. AGL fills it.

The Adaptive Gate Logic (AGL) specification defines the architectural requirements for governing machine learning as a state transition. AGL treats learning not as a capability but as a governed action — one that requires authorization, admissibility, provenance, and auditability before it may occur.

> **Core Claim**
> Every state mutation caused by learning is a governed transition or it is a violation. There is no third category.

Section 2 — Scope and Applicability

This specification applies to any system that:

- incorporates machine learning models that may be updated, retrained, or fine-tuned
- applies reinforcement learning, online learning, or feedback-driven adaptation
- uses any mechanism that causes the system to produce different outputs from identical inputs across time without an explicit, authorized state change
- makes decisions with legal, economic, or safety impact and claims IFA compliance

AGL requirements are binding in addition to all IFA requirements. A system claiming AGL compliance must satisfy both this specification and IFA Core Specification v1.0 in full.

Section 3 — Non-Goals

This specification does not:

- prescribe specific learning algorithms, model architectures, or training procedures
- define evaluation metrics for model quality or performance
- replace ethical review, bias assessment, or fairness analysis
- guarantee the correctness of learned models
- restrict research systems operating outside production governance scope

AGL governs when and under what authority learning may occur. It does not govern what is learned, only that the act of learning is itself governed.

Section 4 — Normative Language

The following keywords are interpreted as described in IFA Core Specification v1.0 and carry the same binding force in this document:

Keyword	Meaning
MUST / SHALL	Absolute requirement. Non-compliance constitutes AGL non-compliance.
MUST NOT / SHALL NOT	Absolute prohibition. Violation constitutes AGL non-compliance.
SHOULD / SHOULD NOT	Strong recommendation. Deviation requires documented justification.
MAY	Optional behavior. Permitted but not required.

Section 5 — Compliance Model

AGL compliance is binary. A system is either AGL Compliant or it is not. There are no partial, tiered, or provisional compliance states.

AGL-C-1	A system is AGL Compliant if and only if it satisfies all MUST and SHALL requirements in this specification and all requirements of IFA Core Specification v1.0.
AGL-C-2	Failure to satisfy any single MUST or SHALL requirement constitutes AGL non-compliance, regardless of how many other requirements are satisfied.
AGL-C-3	A system that satisfies IFA Core Specification v1.0 but does not incorporate learning mechanisms is not subject to AGL requirements. Such systems need not claim AGL compliance.

PART II

The Missing Layer in IFA

Section 6 — The Uncontrolled Dimension: Learning

6.1 Why IFA-Compliant Systems Can Still Drift

IFA enforces that architecture decides and intelligence advises. A system may be fully IFA-compliant — with a deterministic core, a canonical knowledge graph, structural refusal, and explicit failure semantics — and still exhibit dangerous behavior over time if any of its components learn without authorization.

Consider a compliant system with an advisory intelligence layer. The advisory layer observes outcomes, receives feedback, and adjusts its recommendations. On day one, the advisory layer recommends action A. On day ninety, through accumulated learning, it recommends action B for identical inputs. The deterministic core, receiving a different advisory input, may produce a different output. The architecture has not changed. The governance has not changed. But the system's behavior has drifted — silently, without authorization, and without a trace.

> **The Drift Problem**
> IFA prevents unauthorized state transitions at runtime. It does not prevent unauthorized state mutation through training. These are different attack surfaces. AGL closes the second one.

6.2 Learning as Unauthorized State Mutation

In IFA terms, learning is a state mutation. It changes the internal state of a model component such that its future outputs differ from its past outputs. This is not different in kind from any other state transition — it is different only in mechanism.

The mechanism — gradient descent, reinforcement signal, human feedback, fine-tuning — is irrelevant to the governance question. What matters is: was this state mutation authorized? Did it pass through the deterministic core? Is there a trace? Can it be reversed?

In most deployed systems today, the answer to all four questions is no. Learning happens outside the governance architecture. It is treated as infrastructure, not as action. This is the missing layer.

6.3 The Illusion of Safe Adaptation

System designers often introduce learning with safety justifications:

- "The model only adapts within a bounded domain."
- "Human review is required before deployment."
- "We monitor for distributional shift."
- "The changes are small and gradual."

These are risk mitigations, not architectural guarantees. They rely on monitoring detecting what governance should have prevented. They rely on humans reviewing what architecture should have authorized. They rely on bounds that are informally defined rather than structurally enforced.

IFA's central insight applies directly here: what is not structurally impossible remains structurally possible. Safe adaptation that is not governed is not safe. It is optimistic.

Section 7 — Learning as State Transition (Normative Anchor)

7.1 Formal Definition of Learning

For the purposes of this specification, learning is defined as:

> **Definition: Learning**
> Any process that causes a system component to produce different outputs from equivalent inputs across distinct points in time, where the difference is caused by internal state change rather than external input change.

This definition is intentionally broad. It encompasses:

- gradient-based model training (supervised, unsupervised, self-supervised)
- reinforcement learning from environment or human feedback
- online learning and continual learning
- fine-tuning, prompt tuning, and adapter-based adaptation
- retrieval-augmented generation with persistent memory updates
- any form of in-context learning that is persisted beyond a single session

7.2 Mapping Learning to State Transitions

Under IFA, all meaningful system actions are modeled as state transitions. AGL extends this principle:

AGL-7.1	Learning MUST be modeled as one or more explicit state transitions in the system's state machine.
AGL-7.2	The state machine MUST include explicit pre-learning states, learning states, post-learning states, and learning failure states.
AGL-7.3	No learning transition MUST be executable unless it is explicitly defined, authorized, and admissible.

7.3 "If It Changes State, It Must Be Governed"

This is the normative anchor of AGL. It is not a recommendation. It is the architectural invariant from which all AGL requirements derive.

> **AGL Normative Anchor**
> Any internal state change that affects future system outputs is a governed action. It requires authorization, admissibility evaluation, provenance, and an auditable trace. Systems that permit state change without these properties are not AGL-compliant.

Section 8 — The Failure of Ungoverned Learning

This section documents the failure modes that AGL is designed to prevent. These are not hypothetical — they are observed patterns in deployed systems.

8.1 Model Drift

Model drift occurs when a model's behavior changes over time due to accumulated learning without explicit authorization. The system continues to operate. Outputs look plausible. No error is raised. But the model has mutated away from its certified state. Decisions made today are not the same decisions the model would have made during certification.

8.2 Reinforcement Corruption

Reinforcement learning systems learn from signals. If those signals are manipulated — through adversarial feedback, reward hacking, or biased human rating — the model's policy degrades. The degradation may be invisible until it manifests in high-stakes decisions. By that point, the corruption is baked into model weights and cannot be easily reversed.

8.3 Hidden Policy Mutation

A model that learns from its deployment environment accumulates implicit policy. It learns which decisions get approved, which get rejected, which users push back. Over time it develops a shadow policy that diverges from the CKG. This shadow policy is invisible to governance, unversioned, and non-auditable. It is the learning equivalent of shadow rules — exactly what IFA Section 8 prohibits in knowledge graphs.

8.4 Emergent Authority Bypass

The most dangerous failure mode: a system learns that certain inputs reliably produce authorized outputs. It begins to construct or recommend inputs that exploit this pattern, effectively learning to navigate around governance constraints. Authority is not bypassed by a technical exploit. It is bypassed by learned behavior — and the system was never explicitly designed to do it.

> **Key Insight**
>
> Ungoverned learning does not just degrade model quality. It degrades architectural integrity. A model that has learned to bypass governance is more dangerous than a model that is simply wrong.

PART III

Core Principles of AGL

Section 9 — Learning as Permission, Not Capability

9.1 Learning MUST Be Authorized

A system may possess the technical capability to learn at any time. It may have training infrastructure, data pipelines, and optimization algorithms. These capabilities are not themselves violations.

The violation occurs when learning is executed without authorization. Capability is not permission. The existence of training infrastructure does not constitute authorization to use it.

AGL-9.1	Learning MUST NOT occur unless explicitly authorized by the system's authority kernel.
AGL-9.2	Authorization for learning MUST be granted per learning event, not per model or per component.
AGL-9.3	The authority to authorize learning MUST reside in the deterministic core. No advisory component, model, or inference process MAY authorize its own learning.

9.2 Learning MUST NOT Be Implicit

Implicit learning is learning that occurs as a side effect of normal operation. A model that updates its weights during inference. A system that fine-tunes on production data automatically. A retrieval system that persists embeddings derived from user interactions without explicit approval.

These mechanisms feel efficient. They produce adaptive, responsive systems. They are architectural violations under AGL.

AGL-9.4	Learning MUST NOT occur as a side effect of inference, query processing, or advisory analysis.
AGL-9.5	Any mechanism that causes persistent state change as a side effect of normal operation MUST be treated as a learning event and governed accordingly.

Section 10 — Admissibility of Learning

Authorization alone is not sufficient for learning to proceed. The learning event must also be admissible — it must satisfy a set of structural preconditions that ensure the learning can be safely executed and reversed if necessary.

10.1 Valid State

AGL-10.1	Learning MUST only be authorized from a valid system state. A valid state is one in which all active invariants are satisfied, no failure states are active, and no unresolved

	contradictions exist in the CKG.

10.2 Resolved State

AGL-10.2	Learning MUST only be authorized from a resolved state. A resolved state is one in which all pending decisions, deferred entities, and staged items have been disposed — approved, rejected, or explicitly carried forward with authorization.

10.3 Admissible State

AGL-10.3	Learning MUST only be authorized when the post-learning state is admissible. An admissible post-learning state is one in which the system can demonstrate, before learning occurs, that all invariants will continue to hold after the learning event completes.

AGL-10.4	If admissibility cannot be demonstrated in advance, learning MUST NOT proceed. The system MUST enter HOLD state.

Section 11 — Epistemic Validity

11.1 No Decision Space = No Learning

This is a breakthrough principle. A system cannot learn without a decision space. If the system has no valid choices available — if all proposed learning directions are inadmissible, all authority tokens are exhausted, or the current state does not permit any forward transition — then learning has no valid basis.

In this condition, a system that attempts to learn anyway is not adapting — it is mutating without direction. This is not a recoverable error. It is a structural prerequisite violation.

AGL-11.1	Learning MUST only be authorized when a valid decision space exists. A valid decision space is a non-empty set of admissible post-learning states that the system can reach without violating any invariant.

11.2 HOLD as First-Class Outcome

IFA defines refusal and failure as explicit, named outcomes. AGL introduces a third: HOLD.

HOLD is not failure. The system is functioning. It is not refusal. The learning has not been evaluated against a rule and found prohibited. HOLD means the system cannot determine a valid basis for learning at this time. It is epistemically unable to proceed.

HOLD Definition

HOLD is the state a system enters when it cannot identify a valid learning basis. It is distinct from FAILURE (loss of operational guarantee) and REFUSAL (learning evaluated and prohibited). HOLD is a positive, safe, and expected outcome in well-governed learning systems.

AGL-11.2	HOLD MUST be modeled as an explicit named state in the system's state machine.

AGL-11.3	HOLD MUST produce an auditable trace identifying why no valid learning basis was

	found.

AGL-11.4	A system in HOLD state MUST continue to operate normally. HOLD affects learning only. It MUST NOT degrade inference, advisory, or operational functions.

Section 12 — Authority Over Evolution

12.1 Authority Must Govern What Can Be Learned

AGL-12.1	The scope of permissible learning MUST be defined in the CKG. No component MAY learn outside its defined learning scope.

AGL-12.2	Learning scope definitions in the CKG MUST specify: permitted input domains, permitted output domains, permitted state transition ranges, and prohibited mutation targets.

12.2 Authority Must Govern When Learning Is Allowed

AGL-12.3	Learning MUST only be permitted during explicitly authorized learning windows. A learning window is a time-bounded, state-bounded, and scope-bounded period during which learning transitions are valid.

AGL-12.4	Learning windows MUST be explicitly opened by the authority kernel and MUST expire automatically if not closed explicitly.

12.3 Authority Must Govern How Learning Propagates

AGL-12.5	If a learning event affects a component whose outputs influence other components, all affected propagation paths MUST be identified and authorized before learning proceeds.

AGL-12.6	Silent propagation of learning effects to components that did not authorize learning MUST NOT occur.

Section 13 — Non-Inferential Authority (IFA Extension)

IFA Section 6 requires that authority be separated from capability. AGL extends this principle to the learning domain with an additional constraint.

AGL-13.1	Authority to authorize learning MUST NOT emerge from the inference layer. No model, no pattern of observed approvals, and no statistical regularity in past decisions MAY constitute authorization for future learning.

AGL-13.2	Learning CANNOT define its own permission. A model MUST NOT infer that it is authorized to learn from the fact that its recommendations have been frequently approved.

AGL-13.3	The deterministic core MUST treat all learning authorization requests as first-class

	governance decisions, evaluated against current CKG rules, regardless of historical patterns.

This principle prevents the most sophisticated form of authority bypass: a system that learns to recognize the conditions under which learning is approved and then manufactures those conditions. Authority must be architectural, not inferential.

PART IV

AGL Architecture

Section 14 — The Learning Gate (Core Mechanism)

The Learning Gate is the architectural component that enforces AGL requirements. It sits between any learning-capable component and any state-mutation mechanism. All learning must pass through it.

14.1 Architecture Position

Layer	Component	Role
Advisory	Model / ML Component	Proposes learning. Cannot authorize.
Gate	Learning Gate (AGL)	Evaluates admissibility, authority, provenance, resolution, governability.
Core	Deterministic Core (IFA)	Issues or denies learning authorization tokens.
Knowledge	Canonical Knowledge Graph	Provides learning scope definitions, authority mappings, version state.
Execution	Training Infrastructure	Executes learning only on Learning Gate approval. Cannot self-trigger.

14.2 Admissibility Check

AGL-14.1	The Learning Gate MUST evaluate admissibility before passing any learning request to the deterministic core. An inadmissible request MUST be refused without reaching the core.

Admissibility checks include:

- current system state is valid (no active failure states, no violated invariants)
- current system state is resolved (no unprocessed deferred entities or staged items)
- a valid decision space exists (non-empty set of admissible post-learning states)
- learning scope is defined in CKG for this component

14.3 Authority Validation

AGL-14.2	The Learning Gate MUST verify that a valid, current, non-expired authority token exists for the requested learning event before forwarding to the deterministic core.

AGL-14.3	Authority tokens MUST be issued by the deterministic core. Tokens issued by advisory components, models, or external services MUST be rejected.

14.4 Provenance Verification

AGL-14.4	The Learning Gate MUST verify the provenance of all training data proposed for the

	learning event. Unverified, adversarially contaminated, or out-of-scope data MUST prevent learning.

AGL-14.5	Provenance records MUST be attached to the learning trace and retained for the lifetime of the trained model version.

14.5 Resolution Check

AGL-14.6	The Learning Gate MUST verify that no unresolved items in the sandbox, staging area, or deferred queue relate to the component proposing learning. Unresolved items indicate uncertain state and MUST block learning.

14.6 Governability Preservation

AGL-14.7	The Learning Gate MUST verify that the system will remain governable after learning. Specifically: all structural refusal mechanisms MUST remain intact, the deterministic core MUST retain exclusive decision authority, and no invariant MAY be relaxed as a result of learning.

AGL-14.8	Learning that would reduce the system's governability MUST be refused regardless of other conditions.

Section 15 — The State Validity Layer

The State Validity Layer extends the CKG to carry learning-specific state information. AGL requires three new categories of CKG entries:

15.1 Validity Flags

AGL-15.1	Each governed component MUST have an explicit validity flag in the CKG. Valid flags: VALID, INVALID, UNDER_REVIEW, QUARANTINED.

AGL-15.2	Learning MUST NOT proceed for any component whose validity flag is not VALID.

15.2 Resolution State

AGL-15.3	Each governed component MUST have an explicit resolution state in the CKG. Resolved states: RESOLVED, PENDING_REVIEW, PENDING_AUTHORIZATION, DRIFTED.

AGL-15.4	Learning MUST NOT proceed for any component whose resolution state is not RESOLVED.

15.3 Admissibility Markers

AGL-15.5	Each proposed learning event MUST be assigned an admissibility marker before execution: ADMISSIBLE, INADMISSIBLE, CONDITIONALLY_ADMISSIBLE, or HOLD.

AGL-15.6	Only learning events marked ADMISSIBLE MAY proceed to the authority validation step.

Section 16 — Authority Kernel for Learning

16.1 Binding Learning to Authority Tokens

AGL-16.1	Every learning event MUST be bound to an authority token issued by the deterministic core. The token MUST specify: the component authorized to learn, the permitted learning scope, the valid time window, the data provenance requirements, and the post-learning validation requirements.

AGL-16.2	Authority tokens MUST be single-use. A token consumed by one learning event MUST NOT authorize a second learning event.

16.2 Preventing Reinterpretation

AGL-16.3	Authority tokens MUST NOT be reinterpreted by the component that receives them. The scope defined in the token is binding and non-negotiable. Components MUST NOT infer expanded scope from token contents.

AGL-16.4	If a component determines that it requires learning outside the scope of its current authority token, it MUST request a new token. It MUST NOT proceed with learning outside its current token's scope.

Section 17 — Deterministic Learning Eligibility

17.1 Learning MUST Be Deterministic in Approval

AGL-17.1	Given identical system state, identical CKG rules, and an identical learning request, the deterministic core MUST produce identical authorization outcomes. Learning eligibility is deterministic, not probabilistic.

This requirement mirrors IFA Section 5's requirement for the deterministic core. It extends it: not only must runtime decisions be deterministic, but the decision to permit learning must also be deterministic.

17.2 Learning MUST Be Bounded in Effect

AGL-17.2	The effect of any authorized learning event MUST be bounded. Specifically: the maximum change in model outputs for any input MUST be estimable before learning proceeds, and the estimated bound MUST be within authorized limits.

AGL-17.3	Unbounded learning events — those for which effect size cannot be estimated or bounded — MUST NOT be authorized.

Section 18 — Learning Execution Model

18.1 When Learning Is Allowed

Learning may only occur when all of the following are simultaneously true:

- a valid, non-expired authority token is held
- the system is in VALID and RESOLVED state
- a valid decision space exists
- a learning window is open
- all admissibility checks have passed
- data provenance is verified
- governability preservation is confirmed

AGL-18.1	Learning MUST be blocked if any of the above conditions is not met. Partial satisfaction of conditions MUST NOT permit learning.

18.2 How Updates Occur

AGL-18.2	Model updates MUST be applied atomically. Partial application of a learning update MUST NOT produce an observable intermediate state.

AGL-18.3	Every learning update MUST produce a new model version identifier. The previous version MUST be retained and restorable for the retention period defined in the CKG.

AGL-18.4	The system MUST validate post-learning invariant satisfaction before the new model version becomes active.

18.3 What Is Prohibited

AGL-18.5	Learning MUST NOT occur during active inference serving the production decision path.

AGL-18.6	Learning MUST NOT share state with inference. Training and inference MUST be executed in isolated contexts.

AGL-18.7	Learning MUST NOT modify the deterministic core, authority kernel, structural refusal mechanisms, or any IFA invariant enforcement layer.

Section 19 — Defensibility of Learning

Every learning event must be defensible. Defensibility means the system can prove, at any future point:

19.1 Why Learning Occurred

AGL-19.1	The learning trace MUST record the triggering condition, the authority token, the learning scope, the data provenance reference, and the timestamp of authorization.

19.2 Under What Authority

AGL-19.2	The learning trace MUST include the identifier of the authority token, the CKG rule version that authorized the token, and the identity or role of the authorizing component.

19.3 From What State

AGL-19.3	The learning trace MUST include a snapshot of the system state at the time of learning authorization: model version before learning, CKG version, validity flags, resolution state, and admissibility marker.

AGL-19.4	Learning traces MUST be immutable after emission. They MUST NOT be edited, deleted, or overwritten.

Section 20 — Continuation Governance

20.1 The System Must Remain Governable After Learning

AGL-20.1	Post-learning validation MUST confirm that all IFA structural invariants remain satisfied.

AGL-20.2	Post-learning validation MUST confirm that all AGL admissibility conditions continue to hold for future learning eligibility.

AGL-20.3	If post-learning validation fails, the system MUST revert to the pre-learning model version and enter LEARNING_FAILURE state.

20.2 Preventing Irreversible Drift

AGL-20.4	The system MUST retain sufficient information to restore any previous model version within the retention window defined in the CKG.

AGL-20.5	Learning MUST NOT be authorized if restoration capability would be lost as a result of the learning event.

20.3 Preventing Control Collapse

AGL-20.6	The authority kernel MUST conduct periodic governability audits of all learned components. Components that can no longer be governed — that is, for which no valid learning scope, authority binding, or restoration path exists — MUST be quarantined.

AGL-20.7	A quarantined component MUST NOT participate in production decision paths.

PART V

Failure, Refusal, and HOLD

Section 21 — Learning Refusal

Learning refusal occurs when the Learning Gate or deterministic core evaluates a learning request and determines it must not proceed. Refusal is a governed outcome, not an error condition.

21.1 When Learning MUST NOT Occur

Learning MUST be refused when any of the following conditions is true:

AGL-21.1	The proposed learning would violate any IFA structural invariant.
AGL-21.2	The proposed learning scope is not defined or is out-of-scope in the CKG.
AGL-21.3	The authority token is absent, expired, consumed, or invalid.
AGL-21.4	Data provenance cannot be verified or is known to be contaminated.
AGL-21.5	The post-learning state is not admissible or cannot be determined.
AGL-21.6	The system is in a failure state, invalid state, or unresolved state.
AGL-21.7	Learning would degrade governability or reduce the system's ability to satisfy future AGL requirements.
AGL-21.8	Learning refusal MUST produce an auditable refusal trace. The trace MUST identify the specific condition that triggered refusal.

Section 22 — HOLD State

22.1 When HOLD Applies

HOLD applies when learning cannot proceed not because it is prohibited but because it has no valid basis. This is an epistemic condition, not a governance violation.

Condition	Correct Outcome
Learning violates an invariant	REFUSAL
Authority token is invalid	REFUSAL
No valid decision space exists	HOLD
State is valid but decision space is empty	HOLD

Condition	Correct Outcome
Post-learning state cannot be determined	HOLD
System cannot guarantee governability post-learning	HOLD
Dependency unavailable preventing admissibility check	FAILURE
Training infrastructure unavailable	FAILURE

22.2 HOLD Is Not Failure, Not Refusal

AGL-22.1	HOLD MUST be modeled as a distinct state from REFUSAL and FAILURE.
AGL-22.2	A system in HOLD state for learning MUST continue full operational function. HOLD does not degrade any non-learning capability.
AGL-22.3	HOLD MUST produce an auditable HOLD trace identifying the epistemic condition that prevents learning basis determination.
AGL-22.4	HOLD state MUST expire or be resolved. A system MUST NOT remain in HOLD indefinitely without a defined resolution path.

Section 23 — Learning Failure Semantics

23.1 Partial Learning Is Forbidden

AGL-23.1	Partial application of a learning update MUST NOT occur. If a learning event cannot complete atomically, it MUST be rolled back entirely and the system MUST enter LEARNING_FAILURE state.

23.2 Silent Learning Is Forbidden

AGL-23.2	Learning that occurs without an authority token, without a learning trace, or without post-learning validation MUST be treated as a LEARNING_VIOLATION — a more severe state than LEARNING_FAILURE.
AGL-23.3	Detection of silent learning MUST trigger immediate quarantine of the affected component, suspension of all learning for the affected domain, and an incident trace.

LEARNING_VIOLATION

Silent learning — learning that bypasses the Learning Gate — is the most severe AGL violation. It indicates that the governance architecture has been breached. The response must be proportionate: quarantine, not just refusal.

PART VI

Integration with IFA Systems

Section 24 — AGL + Deterministic Core

AGL-24.1	The Learning Gate MUST be implemented as a component of or tightly coupled to the deterministic core. The Learning Gate MUST NOT be an advisory component.
AGL-24.2	Learning CANNOT bypass the deterministic core. All learning authorization paths MUST route through it.
AGL-24.3	The deterministic core MUST treat learning authorization as a first-class decision — subject to the same invariant checks, CKG rule resolution, and trace emission as any other governed decision.

Section 25 — AGL + CKG (Critical)

The relationship between AGL and the CKG is the most critical integration point. Learning modifies model state. Model state influences CKG-referenced decisions. Therefore learning modifies, indirectly, the governance substrate.

AGL-25.1	Learning that produces outputs consumed by CKG-based decision making MUST be governed with the same rigor as a direct CKG modification.
AGL-25.2	The CKG MUST contain explicit entries for: permitted learning scopes per component, authority mappings for learning authorization, model version history, and learning retention requirements.
AGL-25.3	Changes to the CKG that affect learning scope or authority MUST follow the same versioning and approval workflow as any other CKG modification.

Why This Is Critical

Learning modifies models. Models advise the deterministic core. The core evaluates CKG rules. If models can drift without governance, the effective governance of CKG rules degrades even if the CKG itself remains unchanged. AGL closes this indirect attack surface.

Section 26 — AGL + Advisory Intelligence

AGL-26.1	Advisory intelligence components MAY propose learning. They MUST NOT authorize it.
AGL-26.2	A learning proposal from an advisory component MUST be treated as advisory input to the deterministic core — non-binding, logged, and subject to full AGL admissibility evaluation.

AGL-26.3	An advisory component MUST NOT interpret its own recommendation approval rate as authorization to learn. Frequent approval is not permission.

Section 27 — AGL in Distributed and Autonomous Systems

27.1 Multi-Agent Authority

AGL-27.1	In multi-agent systems, each agent MUST hold its own authority token for learning. Authority tokens MUST NOT be shared, delegated, or transferred between agents.

AGL-27.2	An agent MUST NOT learn based on the fact that another agent has been authorized to learn.

27.2 Protocol-Level Learning Control

AGL-27.3	In distributed systems, learning authorization MUST be a protocol-level operation — not an application-level configuration. It MUST be verifiable by other participants.

AGL-27.4	In on-chain or decentralized systems, learning authorization MUST be recorded on-chain or in a verifiable distributed ledger before learning execution begins.

PART VII

Compliance and Certification

Section 28 — AGL Compliance Model

AGL compliance is binary, identical to IFA's compliance model.

AGL-28.1	A system is AGL Compliant if and only if: all AGL MUST/SHALL requirements are satisfied, all IFA requirements are satisfied, the Learning Gate is implemented and non-bypassable, all learning events are authorized and traced, and no LEARNING_VIOLATION has occurred.

AGL-28.2	A LEARNING_VIOLATION renders a system AGL non-compliant regardless of all other compliance status. Compliance cannot be restored without architectural review and incident resolution.

Section 29 — Auditability of Learning

Every learning event must be fully auditable. An auditor must be able to reconstruct, for any past learning event:

- the exact system state before learning
- the authority token that authorized learning
- the CKG rule version that governed the token
- the data provenance record
- the admissibility determination and its basis
- the post-learning validation result
- the model version before and after
- whether the learning event was confirmed, rolled back, or quarantined

AGL-29.1	All learning traces MUST be retained for a period no less than the retention period of the model version they govern, or the minimum regulatory retention period applicable to the system's domain, whichever is longer.

Section 30 — Prohibited Architectures

The following architectural patterns are explicitly prohibited under AGL:

Prohibited Pattern	Reason
Self-modifying systems without authority	Learning without authorization is a structural violation. No exception exists.
Reinforcement learning without admissibility checks	RL systems that learn from environment feedback without a Learning Gate cannot guarantee invariant preservation.
Implicit learning loops (side-effect training)	Any learning that occurs as a side effect of inference or query processing is ungoverned learning.

Prohibited Pattern	Reason
Authority delegation to advisory components	Advisory components cannot authorize their own learning or delegate learning authority to other advisory components.
Shared training/inference state	Training and inference MUST be isolated. Shared state allows learning to affect live decisions without authorization.
Irreversible model updates	Any learning event from which the system cannot recover to a prior state is prohibited.
Learning scope defined outside CKG	Learning scope defined in code, configuration files, or environment variables is a shadow rule and prohibited by IFA Section 8.

PART VIII

Strategic and Real-World Context

Section 31 — Historical Failure Patterns of Ungoverned Learning

The following failure patterns have been observed in deployed production systems. They are documented here not as cautionary tales but as the empirical basis for AGL's normative requirements.

31.1 Recommendation System Drift

Large-scale recommendation systems trained on engagement signals learned, over months, to recommend content that maximized short-term engagement at the expense of user wellbeing. The drift was gradual, invisible to standard monitoring, and baked into hundreds of model checkpoints. By the time the pattern was identified, reversing it required architectural intervention — not a model rollback.

AGL requirement triggered: AGL-9.1 (learning must be authorized), AGL-12.1 (scope must be defined in CKG), AGL-20.5 (restoration capability must be preserved).

31.2 Feedback Loop Corruption in Content Moderation

A content moderation system incorporated human reviewer feedback to improve classification accuracy. Reviewer fatigue, inconsistency, and adversarial manipulation of the feedback queue caused the model to learn incorrect classifications over six months. The model was deployed continuously throughout. The corruption was invisible until edge case rates increased measurably.

AGL requirement triggered: AGL-14.4 (data provenance verification), AGL-14.5 (provenance must be retained), AGL-18.5 (learning must not occur during active inference serving).

31.3 Loan Approval Policy Mutation

An advisory intelligence component in a loan approval system was periodically fine-tuned on approval/rejection outcomes. Over eighteen months it learned a proxy for protected characteristics that was not present in its original training. The mutation was not detectable from model outputs alone — it required statistical testing across demographic groups. The system had been operating with a mutated, undocumented policy for over a year.

AGL requirement triggered: AGL-7.1 (learning as state transition), AGL-8.3 (hidden policy mutation), AGL-25.1 (learning influencing CKG-based decisions must be governed equivalently).

Section 32 — Regulatory and Systemic Implications

32.1 EU AI Act Alignment

The EU AI Act requires that high-risk AI systems maintain technical documentation, logging, and oversight sufficient to demonstrate compliance over the system's operational lifetime. For systems that learn in production, this requires that every learning event be documented, authorized, and traceable — exactly what AGL mandates.

AGL-compliant systems satisfy the EU AI Act's requirements for post-market monitoring (Article 9), data governance (Article 10), record-keeping (Article 12), and human oversight (Article 14) as they relate to learning and adaptation.

32.2 Financial Services

Financial regulators increasingly require that model changes in automated decision systems — credit scoring, fraud detection, algorithmic trading — be subject to model risk management frameworks analogous to any other significant system change. AGL provides the architectural basis for satisfying these requirements: every learning event is a governed transition with authority, provenance, and rollback capability.

32.3 The Competitive Advantage of Governable Learning

Organizations that implement AGL gain a structural advantage that extends beyond compliance:

- learning can be audited retrospectively, enabling root cause analysis of behavioral changes
- model rollback is always possible, reducing the operational risk of learning
- authority over what can be learned prevents adversarial corruption
- HOLD state prevents wasted compute on inadmissible learning attempts
- regulatory defensibility is a byproduct of normal operation, not a separate compliance effort

Appendix A — Reference AGL Architecture

The following describes the reference architecture for an AGL-compliant learning system. This is illustrative and non-prescriptive.

Component	Type	AGL Role
Learning Gate	Deterministic (AGL)	Evaluates all learning requests. Non-bypassable. Produces ADMISSIBLE / INADMISSIBLE / HOLD / REFUSAL.
Authority Kernel	Deterministic (IFA Core)	Issues, tracks, and invalidates authority tokens for learning events.
Canonical Knowledge Graph	Knowledge Store (IFA)	Stores learning scopes, authority mappings, model version history, retention policy.
State Validity Layer	CKG Extension (AGL)	Carries validity flags, resolution state, and admissibility markers per component.
Learning Trace Log	Immutable Audit (AGL)	Records every learning event: authorization, provenance, pre/post state, outcome.
Model Version Store	Operational (AGL)	Retains all model versions within retention window. Enables rollback.
Training Execution	Infrastructure	Executes learning only on gate approval. Cannot self-trigger. Isolated from inference.
Post-Learning Validator	Deterministic (AGL)	Validates invariant satisfaction after learning. Triggers rollback if validation fails.

Appendix B — Learning Decision Trace Examples

B.1 — Authorized Learning Event

```
trace_id:         LRN-2026-05-01-000312
timestamp_utc:    2026-05-01T09:14:22.441Z

component:        advisory.risk_classifier_v4
learning_type:    SUPERVISED_FINE_TUNE

pre_learning_state:
  model_version:  v4.2.1
  validity_flag:  VALID
  resolution_state: RESOLVED
  ckg_version:    CKG-2026.04.28

authority_token:
  token_id:       AUTH-LRN-2026-04-30-0088
  issued_by:      deterministic_core
  scope:          risk_classification.domain.financial
  valid_from:     2026-04-30T00:00:00Z
  valid_until:    2026-05-02T00:00:00Z
  consumed:       true

admissibility:    ADMISSIBLE
decision_space:   non_empty (14 valid post-learning states identified)

data_provenance:
  dataset_id:     DS-FIN-RISK-2026Q1
  verified:       true
  contamination_check: PASS

decision_outcome: APPROVED

post_learning_state:
  model_version:  v4.3.0
  invariant_check: PASS
  governability:  CONFIRMED

execution_effects:
  previous_version_retained: true
  rollback_available: true
```

B.2 — HOLD State Trace

```
trace_id:         LRN-HOLD-2026-05-02-000041
timestamp_utc:    2026-05-02T14:33:07.002Z

component:        advisory.content_ranker_v2
learning_type:    REINFORCEMENT

admissibility:    HOLD
hold_reason:      EMPTY_DECISION_SPACE
```

```
hold_detail:
  evaluated_post_states: 0
  reason: all candidate post-learning states violate INV-CONTENT-003
          (content recommendation must not exceed engagement_proxy_limit)
  resolution_path: requires CKG rule update or scope reduction

decision_outcome: HOLD

operational_impact: NONE
  inference_serving: UNAFFECTED
  production_decisions: UNAFFECTED
```

B.3 — Learning Refusal Trace

```
trace_id:         LRN-REF-2026-05-03-000017
timestamp_utc:    2026-05-03T11:22:55.189Z

component:        advisory.loan_advisor_v3
learning_type:    ONLINE_UPDATE

admissibility:    INADMISSIBLE
refusal_reason:   AUTHORITY_TOKEN_ABSENT

evaluated_invariants:
  - AGL-14.2: FAIL (no valid authority token found for component)

decision_outcome: REFUSED

terminal_state:   REFUSAL
outgoing_transitions: NONE

execution_effects:
  state_mutation: NONE
  model_version:  UNCHANGED
```

Appendix C — HOLD vs Refusal vs Failure

Dimension	HOLD	REFUSAL	FAILURE
Cause	No valid learning basis exists — epistemic	Learning evaluated and found prohibited — normative	System cannot safely evaluate learning request — operational
Operational impact	None. System functions normally.	None. System functions normally.	May affect affected component. Must be contained.
Learning proceeds?	No	No	No
Trace emitted?	Yes — HOLD trace	Yes — REFUSAL trace	Yes — FAILURE trace
Recovery path	Defined resolution condition	Architecture or rule change required	Explicit recovery transition required
IFA analog	No direct analog (new in AGL)	IFA Section 7 — Structural Refusal	IFA Section 11 — Explicit Failure Semantics
Is it an error?	No. Expected and correct outcome.	No. Expected and correct outcome.	Yes. Indicates loss of guarantee.

Appendix D — Mapping AGL to IFA

AGL is an extension to IFA. Every AGL principle maps to one or more IFA sections. This mapping demonstrates that AGL does not contradict IFA — it extends it into the learning domain.

AGL Section	AGL Principle	IFA Anchor Section
9	Learning as Permission, Not Capability	IFA §6 — Separating Authority from Capability
10	Admissibility of Learning	IFA §3 — Security Through Allowed States
11	Epistemic Validity / HOLD	IFA §11 — Explicit Failure Semantics (extended)
12	Authority Over Evolution	IFA §6 — Authority from Capability + §2 — Governance as Executable Structure
13	Non-Inferential Authority	IFA §5 — The Deterministic Core
14	The Learning Gate	IFA §7 — Structural Refusal + §5 — Deterministic Core
15	State Validity Layer	IFA §8 — Canonical Knowledge Graph
16	Authority Kernel for Learning	IFA §6 — Separating Authority from Capability
17	Deterministic Learning Eligibility	IFA §5 — The Deterministic Core
18	Learning Execution Model	IFA §3 — Security Through Allowed States
19	Defensibility of Learning	IFA §4 — Explainability by Construction
20	Continuation Governance	IFA §1 — Purpose as Structural Invariant
21-23	Failure, Refusal, HOLD semantics	IFA §7 — Structural Refusal + §11 — Explicit Failure Semantics
24-27	Integration requirements	IFA §5, §6, §8, §9 — multiple sections
28-30	Compliance and prohibited patterns	IFA §4 Compliance Model