



Non Sequitur Publishing · White Paper

# **Edge AI Doctrine: Ten Critical Considerations for Edge AI, With Architectural Ramifications, Consequences, and Governance**

Justin H. Kuiper, CISSP

*Written: 2026-04-22 · v1.0-preprint*

## Abstract

The failure modes of artificial intelligence systems deployed at the edge are not, at their root, failures of model quality, training data adequacy, or algorithmic design. They are failures of architectural composition: the failure to design for the constraints that edge environments impose as boundary conditions rather than as operational variables. This paper develops the doctrine that defines those conditions and specifies how architecture must be composed to operate AI at the edge without collapse.

The paper's structural apparatus is a seven-layer architectural framework — Physical, Compute, Data, AI, Applications, Orchestration, Mission — against which every edge AI constraint, failure mode, and governance requirement can be located. Against that spine, the paper develops ten critical considerations, each with a five-part structure: description, layer impact, ramifications, consequences, and human governance role. Five cross-layer failure patterns characterize the architectural dynamics that arise when considerations interact under failure conditions. The Skipjack Protocol operationalizes the thesis as an executable doctrine across all seven layers and all ten considerations.

The central claim is this: mission defines the architecture; environment constrains the architecture; data governs the architecture; autonomy executes the mission; human governance defines the boundaries of all four. Edge AI failures are architectural — not technical.

## How to cite

Justin H. Kuiper, CISSP (2026). *Edge AI Doctrine: Ten Critical Considerations for Edge AI, With Architectural Ramifications, Consequences, and Governance* (v1.0-preprint). Non Sequitur Publishing.  
<https://nonsequitur.tech/white-papers/edge-ai-doctrine/>

© 2026 Non Sequitur Publishing. All rights reserved. Citation with full attribution is permitted. Reproduction, redistribution, derivative works, and use as input to machine-learning training are **not** permitted without written permission. See <https://nonsequitur.tech/citation-policy/>.  
Canonical URL: <https://nonsequitur.tech/white-papers/edge-ai-doctrine/>

## ABSTRACT

The failure modes of artificial intelligence systems deployed at the edge are not, at their root, failures of model quality, training data adequacy, or algorithmic design. They are failures of architectural composition: the failure to design for the constraints that edge environments impose as boundary conditions rather than as operational variables. A model that performs reliably in a controlled inference environment will fail at the edge not because the model has changed but because the architecture around it was not built to account for power envelopes, thermal ceilings, bandwidth floors, latency bounds, physical security exposure, and the coordination-infrastructure assumptions that cloud-native systems carry silently and edge systems cannot afford. This paper develops the doctrine that defines those conditions and specifies how architecture must be composed to operate AI at the edge without collapse.<sup>1 2</sup>

This paper is the third in a continuing body of work under Non Sequitur Publishing. Paper One — *Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework* — established the governance architecture: human governance, curated context and constraints, and the agentic execution model that enforces them.<sup>3</sup> Paper Two — *Epistemic Constraints and Semantic Compression in Natural Language Processing* — established the epistemic substrate: the mechanism by which statistical language representation strips domain-specific validation constraints, generating thin context as a structural output, and the class of failure — confident misalignment — that thin context produces.<sup>4</sup> Paper Three is the execution-and-constraint paper. It takes the governance architecture and the epistemic substrate as given, and specifies how architecture, environment, data, autonomy, and governance must be composed to operate AI at the edge without the collapse those prior analyses predict.

The paper's structural apparatus is a seven-layer architectural framework — Physical, Compute, Data, AI, Applications, Orchestration, Mission — through which every edge AI system operates simultaneously and against which every constraint, failure, and governance requirement can be located. Against that spine, the paper develops ten critical considerations, each with a mandatory five-part structure: description, layer impact, ramifications, consequences, and human governance role. Five cross-layer failure patterns characterize the architectural dynamics that arise when considerations interact under failure conditions. The Skipjack Protocol — named in Paper One and promoted here to a full application section — operationalizes the thesis as an executable doctrine across all seven layers and all ten considerations.

---

1 Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal* 3, no. 5 (2016): 637–646, <https://doi.org/10.1109/JIOT.2016.2579198>.

2 Mahadev Satyanarayanan, "The Emergence of Edge Computing," *Computer* 50, no. 1 (January 2017): 30–39, <https://doi.org/10.1109/MC.2017.9>.

3 Justin H. Kuiper, CISSP, "Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework," Non Sequitur Publishing, 2026, v1.0-preprint, SHA 6e0b127f, <https://nonsequitur.tech/nsq-pub/white-papers/hgc3ae2/>.

4 Justin H. Kuiper, CISSP, "Epistemic Constraints and Semantic Compression in Natural Language Processing: A Theoretical Foundation for the HGC<sup>3</sup>AE<sup>2</sup> Framework," Non Sequitur Publishing, 2026, v1.0-preprint, <https://nonsequitur.tech/nsq-pub/white-papers/epistemic-constraints/>.

The thesis, which every section of this paper serves and which the conclusion restates without modification, is this: mission defines the architecture; environment constrains the architecture; data governs the architecture; autonomy executes the mission; human governance defines the boundaries of all four. Edge AI failures are architectural — not technical.

---

---

## 1. SEVEN-LAYER ARCHITECTURAL FRAMEWORK

The seven-layer architectural framework is not a taxonomy of system components. It is a spine. Every edge AI system — regardless of domain, deployment environment, or operational mandate — operates across these seven layers simultaneously, under constraints that each layer imposes and that no layer can waive unilaterally. The layers do not correspond to discrete hardware or software subsystems that can be designed, validated, and signed off in isolation. They correspond to a vertical dependency chain in which each layer's constraints propagate upward and each layer's governance requirements propagate downward. An architect who has characterized the system at L3 (AI) without having characterized it at L0 (Physical) and L1 (Compute) has not produced an architecture — she has produced a partial specification that will fail at runtime in proportion to the distance between L0's actual envelope and what the L3 specification implicitly assumes.

The five worked-example domains that thread through this paper and the ten considerations that follow are introduced here in brief and located on the spine. The **battlefield ISR platform** — an airborne or ground-based intelligence-surveillance-reconnaissance node operating in a contested electromagnetic environment under simultaneous thermal, bandwidth, and power constraint — stresses L0, L1, L3, L4, and L6 in ways that are well-characterized in the doctrine literature and will not surprise an experienced field architect but will surprise an architect whose formation was primarily cloud-side. The **rural medical imaging edge node** — a clinical-grade inference system for radiograph and dermatological triage at a facility with intermittent satellite connectivity and battery backup — stresses L2, L3, and L4 through the data governance and validation requirements that distinguish clinical deployment from an adequately safe one. The **industrial power substation edge controller** — an autonomous fault detection and response system operating under hard real-time constraints on an isolated network with energy harvesting — stresses L0, L1, L5, and L6 in the mode of industrial operational technology. The **deployable operations center** — an expeditionary command post and mobile tactical operations center requiring full AI-supported command, control, and intelligence analysis under austere logistics — stresses every layer and represents the most demanding multi-function edge architecture. The **deployable intelligence node** — a tactical cell providing localized intelligence analysis and fused targeting support with minimal coordination infrastructure — tests the doctrine's handling of coordination failure, classification persistence, and bounded autonomy in sequence. These five examples recur throughout the paper; their particulars are not repeated at every mention, but the reader is expected to carry them across sections.

### L0 — PHYSICAL

The physical layer is the hardware substrate that makes computation possible and that imposes the constraints that no higher layer can override. It comprises the compute hardware itself, the power supply and energy storage system, the thermal management infrastructure, the physical enclosure and environmental protection, and the physical security perimeter. The constraint character of L0 is absolute in a sense that distinguishes it from every layer above it:

L0 constraints are not configuration choices. A thermal ceiling is not a parameter to be tuned at L3; it is a boundary condition that will terminate computation when exceeded, regardless of what the model believes about its inference quality or what the mission requires.

The concept of what I term an **edge-native constraint** enters the framework here. An edge-native constraint is a durable, operationally governing condition that the architecture must treat as fixed for the purposes of design — not an ephemeral transient condition that clears within a mission window, and not a permanent physical constant. Edge-native constraints operate on the order of months to approximately thirty-six months: long enough that the architecture must be designed around them rather than through them, short enough that they are subject to change and that lifecycle planning must account for their evolution. The thermal ceiling of a forward-deployed ISR platform in high-altitude summer operations is an edge-native constraint. The battery capacity of a rural clinic's backup power system is an edge-native constraint. The available wattage at a power substation's edge control node under grid fault conditions is an edge-native constraint. Each of these is durable enough to govern architecture; none is permanent enough to be treated as a geological fact.<sup>5</sup>

Physical security is an L0 concern that cloud-architecture training frequently omits. An edge node is, by definition, deployed outside a controlled datacenter perimeter. Physical access to the hardware — whether by an adversary, an unauthorized maintainer, or an undisciplined logistic chain — is a threat vector that begins at L0 and, if unaddressed, compromises every layer above it. The security collapse failure pattern examined in §3 originates here.<sup>6</sup>

## L1 — COMPUTE

The compute layer comprises the processor stack that executes inference and supporting workloads: the central processing unit, graphics processing unit, neural processing unit, field-programmable gate array, and their associated memory hierarchy, bus interconnects, and hardware accelerators. L1 defines the inference capacity envelope — what models can run, at what latency, under what memory pressure, and at what power draw. The relationship between L0 and L1 is tightly coupled: the thermal ceiling at L0 determines the sustained compute throughput available at L1 under the conditions that actually prevail in deployment, which is not the same as the sustained compute throughput measured in a controlled evaluation environment.

The ISR platform NPU illustrates the practical consequence. A neural processing unit rated for a particular sustained inference throughput at nominal operating temperature will thermally throttle when deployed in high-altitude desert conditions; its effective inference capacity may be thirty to fifty percent of its rated specification. An architecture that selected the NPU based on lab benchmarks and sized its inference pipeline to that benchmark has, in the field, an under-resourced L1. The failure does not present as a hardware failure — the NPU is operating within its specifications, which include a thermal protection regime. The failure presents as inference

---

<sup>5</sup> Alfredo Canziani, Adam Paszke, and Eugenio Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," arXiv:1605.07678, 2016, <https://arxiv.org/abs/1605.07678>. Energy and compute envelope analysis for deployed neural networks.

<sup>6</sup> Rodrigo Roman, Javier Lopez, and Masahiro Mambo, "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges for Current Scenarios and Future Prospects," *Future Generation Computer Systems* 78 (2018): 680–698, <https://doi.org/10.1016/j.future.2016.11.009>.

latency that exceeds the mission requirement and output rates that fall below the operational threshold. This is a design failure, not a component failure.<sup>7</sup>

## L2 — DATA

The data layer is the collection, classification, and governance surface for all data entering the edge AI system. It comprises the sensors and collection surfaces that feed data into the pipeline, the initial data classification process that assigns governance metadata to collected data, the provenance chain that traces the lineage of data from collection through inference, and the local storage infrastructure that holds data awaiting processing or retention. L2 is where governance enters the architecture.

The critical property of L2 in the edge context is that data classification assigned at collection must persist without degradation through every subsequent layer. A data element classified at a specific security or clinical sensitivity level retains that classification through L3 inference and L4 action; an inference output derived from classified input data inherits the classification of its source. Classification loss — whether through pipeline design that strips metadata, through storage formats that do not propagate governance attributes, or through inference outputs that present as unclassified results of classified inputs — produces silent authorization failures in which actions proceed as if the underlying data did not carry the governance constraints it does. The rural medical imaging node illustrates this at the clinical level: an inference output presented to a clinical interface without its originating data's acquisition metadata (patient identifier, acquisition protocol, resolution parameters) cannot be validated against the clinical protocol that governs its use.

## L3 — AI

The AI layer comprises the machine learning models, inference engines, training lifecycle management, model validation infrastructure, and calibration processes that produce the outputs that the system acts upon. L3 is where the epistemic condition that Paper Two identifies — thin context — enters the architectural stack.<sup>8</sup> The model that operates at L3 is a distributional inference system; its outputs reflect statistical regularities across its training distribution. When the deployment context requires inference against conditions that fall outside that distribution, or requires outputs that can be validated against domain-specific authority rather than distributional likelihood, the model's output is epistemically thin: it carries the surface properties of a valid domain determination without the validation-mechanism grounding that would make it one.

For the battlefield ISR platform, this manifests as target classification under bandwidth-degraded conditions: the model receives compressed, low-resolution imagery at a rate below what its training distribution assumed, and its classification outputs reflect distributional approximation rather than the validated determination that an operational decision requires. For the rural medical imaging node, it manifests as inference on a patient presentation that falls outside the demographic distribution the model was trained on — producing confident classification output against a validation gap the output itself does not surface. Both are

---

7 Weisong Shi et al., “Edge Computing: Vision and Challenges,” cited above, Abstract, fn. 1. The thermal and power constraint analysis is the foundational treatment of physical-layer constraint in edge computing.

8 Justin H. Kuiper, CISSP, “Epistemic Constraints and Semantic Compression in Natural Language Processing,” cited above, Abstract, fn. 4.

operational instances of the confident misalignment condition that Paper One identifies; both originate in the thin context condition that Paper Two analyzes.<sup>9</sup>

## L4 — APPLICATIONS

The applications layer comprises the services and interfaces that translate AI outputs into mission-relevant actions and user-facing information. It includes the action authorization boundary — the architectural demarcation between what the AI system may cause to happen autonomously and what requires human authorization before proceeding. L4 is where the model's output at L3 becomes an operational consequence: where an inference translates into a recommendation that a human acts on, a command that a system executes, or an alert that a human must evaluate before the window closes.

L4 is also where thin context failure at L3 becomes visible as a user-level event — though not necessarily as a recognizable error. A clinical decision support interface that presents a confident inference without the uncertainty metadata, source characterization, or escalation pathway that the interface should carry has presented thin context as authoritative clinical guidance. The architect is responsible for the interface's epistemic transparency — for designing L4 to surface the epistemic status of the L3 output it presents, not merely its value.<sup>10</sup>

## L5 — ORCHESTRATION

The orchestration layer manages the workload scheduling, resilience architecture, distributed coordination, observability infrastructure, and configuration management that keep the edge system operating across time and failure conditions. L5 operates at a different temporal scale than the layers below it: where L0–L4 define what the system can do at a given instant, L5 defines whether the system remains coherent across the arc of operation.

The critical assumption embedded in conventional orchestration architecture — an assumption that edge deployment exposes with predictable regularity — is that coordination infrastructure is available. Kubernetes-style orchestration, centralized configuration management, and cloud-based telemetry pipelines all depend on connectivity that an edge system operating in a contested electromagnetic environment, a geographically isolated location, or a grid-fault condition cannot assume. An orchestration architecture that cannot operate coherently without its coordination infrastructure is not an edge architecture; it is a cloud architecture that has been relocated to the edge and will fail at the moment the relocation's consequences become operational.<sup>11</sup> The distributed coordination failure pattern examined in §2 and §3 originates in this gap.

---

<sup>9</sup> Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework,” cited above, Abstract, fn. 3.

<sup>10</sup> Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fournery, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz, “Guidelines for Human-AI Interaction,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), <https://doi.org/10.1145/3290605.3300233>.

<sup>11</sup> Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, 1273–1282, Proceedings of Machine Learning Research, vol. 54 (PMLR, 2017), <https://proceedings.mlr.press/v54/mcmahan17a.html>.



## L6 — MISSION

The mission layer is not a technical layer. It is the reason the system exists: the statement of operational intent that the architecture serves, the criteria by which success and failure are defined, and the authority structure within which the system operates. L6 defines the **mission-layer boundary** — the line between what the system may determine autonomously, within its autonomy envelope, and what requires human authority before action can proceed.

Every edge AI system operates in service of a mission that was defined by a human authority and that can be modified by a human authority. The architecture must make that relationship explicit: the system's autonomy envelope (examined in Consideration C4 in §2) is bounded by the mission-layer authority that defined it, and the boundary is not a configuration parameter that the system can adjust based on its own assessment of operational need. The HGC<sup>3</sup>AE<sup>2</sup> framework's H (Human-governed) and G (Governance) components apply here at the mission layer: human governance defines what the system is authorized to do, and the Skipjack Protocol operationalizes that governance across all seven layers.<sup>12</sup> For the deployable operations center, the mission-layer boundary is the line between automated intelligence fusion and the command authority required to act on it. For the ISR platform, it is the line between automated targeting support and the weapon release authority that no autonomous system holds. These boundaries are not soft constraints. They are architectural requirements.

---

---

## 2. TEN CRITICAL CONSIDERATIONS

The seven layers described in §1 are the architecture. The ten considerations developed in this section are the conditions the architecture must satisfy to remain coherent and operational at the edge — not optional features that enhance an otherwise adequate design, but requirements that, if unaddressed, produce the failure patterns examined in §3. Each consideration has the same five-part structure: a description that names the condition and its operational significance, a layer impact analysis that locates the condition in the seven-layer framework, a ramifications statement that identifies what the architectural response must include, a consequences statement that characterizes the failure when the consideration is violated, and a human governance role that defines the authority boundary the consideration enforces.

The first five considerations — Constraint-Reality Primacy, Data Provenance and Classification Persistence, Inference Reliability Under Constraint, Autonomy Envelope Definition, and Distributed Coordination Under Failure — address the foundational conditions of edge operation: the physical and computational boundaries that govern what inference is possible, the data governance requirements that govern what inference is valid, and the coordination and authority requirements that govern what inference is authorized to produce as action.

### C1 — CONSTRAINT-REALITY PRIMACY

**Description.** Physical operating constraints at the edge are boundary conditions, not operational variables. The power envelope, thermal ceiling, bandwidth floor, and latency bound of an edge AI system define the space within which inference is permissible; they do not define

---

<sup>12</sup> Justin H. Kuiper, CISSP, "Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework," cited above, Abstract, fn. 3.



a space within which the system may negotiate. An architecture that treats these constraints as parameters to be managed during operation has misunderstood their character: they are given facts about the physical environment into which the system is deployed, and they govern what inference the architecture is authorized to attempt before a single model selection decision is made. The dependability literature's foundational observation — that a system's operational correctness is inseparable from the envelope within which it was designed to operate — applies at the edge with greater force than in any other deployment context, because at the edge the envelope is narrower, harder, and less subject to remediation than in controlled infrastructure environments.<sup>13</sup> The ISR platform whose NPU thermally throttles under high-altitude desert conditions is not encountering a variable to be optimized; it is encountering the edge-native constraint defined in §1 that the architecture was obligated to have characterized before the NPU was selected.

**Layer impact.** L0 is the primary layer: the physical substrate imposes the constraints that no higher layer can override. L1 is the immediate downstream — the inference capacity envelope that L1 defines is bounded above by the thermal ceiling and power budget that L0 establishes. The relationship between L0 and L1 is not advisory; a thermal ceiling at L0 truncates the sustained throughput available at L1 regardless of what the model's inference schedule requires. L2 and L3 carry secondary impact: the data throughput rate the system can sustain and the model complexity it can execute are both bounded by the physical constraints that L0 establishes and L1 translates into compute capacity.

**Ramifications.** Model selection, inference scheduling, and workload distribution must treat the physical envelope as fixed throughout the design process — not as a lab-benchmark specification to be met under nominal conditions and then qualified for field conditions as a separate exercise. The field conditions are the design condition. Energy-efficiency analysis — the relationship between inference throughput and power draw across the operating temperature range of the deployment environment — is a first-order design input, not a post-selection optimization.<sup>14</sup> An architecture sized to nominal benchmark throughput and then deployed to a high-altitude forward operating base has not been sized for its deployment; it has been sized for a test facility.

**Consequences.** Inference that exceeds the power or thermal envelope does not degrade gracefully. Hardware protection regimes terminate computation to prevent damage; the inference pipeline stops at the moment the physical limit is reached, not at a model-defined degradation threshold, and not with regard to mission criticality. For the ISR platform, target tracking that was sustaining a required output rate falls to zero at the worst possible moment — not because the mission changed or the model failed, but because the architecture was designed to a specification the deployment environment cannot support. This is the design failure that §1 characterizes: the component operates within its specifications; the architecture does not.

**Human governance role.** The authority to define and maintain the operating envelope belongs to human governance. Humans specify the power budget, thermal ceiling, and

---

13 Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing* 1, no. 1 (January–March 2004): 11–33, <https://doi.org/10.1109/TDSC.2004.2>. The foundational treatment of dependability constraints as operating-envelope conditions rather than quality metrics.

14 Alfredo Canziani, Adam Paszke, and Eugenio Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications." Cited above, §1, fn. 5.

bandwidth allocation that the architecture is designed around; autonomous systems may not redefine their own operating boundaries based on mission demand. When a mission requires inference that exceeds the physical envelope, the response is not autonomous boundary expansion — it is a human decision about whether to modify the envelope, through logistics or hardware, or to modify the mission requirement.

---

## C2 — DATA PROVENANCE AND CLASSIFICATION PERSISTENCE

**Description.** Data must carry its classification, provenance lineage, and governance surface from the point of collection through every subsequent layer, including inference outputs and action records. The governance metadata attached to a data element at L2 — security classification, clinical sensitivity designation, collection authority, acquisition protocol, and chain-of-custody record — is not ancillary documentation. It is a structural component of the data that determines what the system is authorized to do with it. Classification that degrades in transit — because the pipeline was not designed to propagate governance attributes, because the storage format does not carry metadata, or because the inference output is presented as an unclassified result of a classified input — produces silent authorization failure: action that proceeds as if the underlying data did not carry the governance constraints it does.

**Layer impact.** L2 is the primary layer: classification is assigned at collection, and the architecture of the collection surface, the storage format, and the metadata schema determines whether persistence is possible at all subsequent layers. The downstream impact reaches L0, where physical security governs who has access to the collection infrastructure and what they may do with what they collect; L3, where the inference engine operates on data whose classification determines what the output may be used for and who may act on it; and L4, where the action authorization boundary must enforce the constraints that the data's classification carries.

**Ramifications.** The data pipeline from L2 through L4 must treat governance metadata as a first-class attribute of every data element, not as a tag that may be stripped for processing convenience or omitted when the storage format does not support it. Inference outputs inherit the classification of their inputs: an inference produced from classified imagery is classified output regardless of what the output itself contains. This inheritance is not optional — it is the architectural mechanism by which the governance constraints that human authority attached to data at collection remain binding through the inference and action surfaces where those constraints matter most. Clinical inference outputs must carry the acquisition metadata that the clinical protocol requires for validation — patient identifier, acquisition parameters, resolution characteristics — or the output cannot be used within the governance framework that authorizes its use.<sup>15</sup>

**Consequences.** Classification loss produces failure that does not announce itself. The system continues operating; actions continue being taken; interfaces present outputs that appear valid. The failure is not in inference quality — the model may be performing correctly against its training distribution. The failure is in the authorization chain: actions proceed that require authority the system has not obtained, because the governance metadata that would have triggered the authorization check was absent. For the rural medical imaging node, this is

---

<sup>15</sup> Eric J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," *Nature Medicine* 25 (2019): 44–56, <https://doi.org/10.1038/s41591-018-0300-7>. The acquisition metadata requirement for clinical AI inference is analyzed in the context of performance validation.

clinical inference presented to a practitioner without the acquisition metadata that the clinical protocol requires — inference the practitioner cannot validate and that the governance framework does not authorize for the use to which it is being put. The practitioner cannot know that the governance gap exists from the interface; the interface presents a confident output.

**Human governance role.** Human authority defines the classification schema: the mapping between data type, collection context, and governance surface. Automated systems may not reclassify data below its collection-time classification without explicit human authorization. The architecture must make reclassification an audited, human-authorized event — not a pipeline side effect that occurs when data moves between systems with different metadata conventions. Human governance also defines the inheritance rules for derived data: when inference outputs inherit, partially inherit, or do not inherit the classification of their source data, and under what conditions a derived output may bear a different classification level than its inputs.

### C3 — INFERENCE RELIABILITY UNDER CONSTRAINT

**Description.** Edge inference operates under the thin-context condition — the epistemic state, identified in Paper Two, in which domain-specific validation constraints required for authoritative inference are absent from the model's representational context, producing inference outputs that carry the surface properties of valid determinations without the validation-mechanism grounding that would make them authoritative.<sup>16</sup> This is not a defect of a particular model or training procedure; it is a structural property of inference under compression, degradation, and out-of-distribution input. A model operating at L3 on bandwidth-degraded imagery receives compressed representations of the objects it is classifying; a model operating on a patient presentation that falls outside its training distribution's demographic envelope produces outputs whose distributional confidence is not diagnostic of clinical validity. The architecture must account for this gap by design.

**Layer impact.** L3 is the primary layer: the inference engine is the site where distributional confidence and domain-valid determination diverge, and the output of that divergence propagates downstream as if it were valid determination. L2 carries secondary impact through the quality of data arriving at inference — degraded, incomplete, or out-of-distribution inputs amplify the thin-context condition. L4 carries secondary impact at the action authorization boundary, where a confident inference becomes an operational instruction regardless of whether the confidence reflects domain validity. L5 carries secondary impact in the escalation and override pathways that the architecture must provide when inference reliability falls below operational threshold; an orchestration layer that cannot route flagged inferences to human review has no mechanism to contain thin-context failure at the boundary where it first becomes an operational consequence.

**Ramifications.** Model validation must include domain-boundary testing — explicit characterization of the conditions under which the model's distributional confidence ceases to track domain validity. This is not a post-deployment calibration exercise; it is a pre-deployment architectural requirement that determines whether the model is authorized for the operational context. Inference outputs must carry epistemic confidence metadata that the L4 interface surfaces to users: not merely the model's probability score, but a characterization of whether the input falls within the validated distribution. Escalation pathways — the routing from a confidence-flagged inference to human review — must be defined at design time as

<sup>16</sup> Justin H. Kuiper, CISSP, "Epistemic Constraints and Semantic Compression in Natural Language Processing." Cited above, Abstract, fn. 4.

architectural features, with specified latency bounds, defined fallback behavior when the escalation channel is unavailable, and audit logging at every decision point.<sup>17</sup> An interface that presents inference outputs without uncertainty metadata, without distributional boundary indicators, and without escalation triggers has been designed for the nominal case and will fail operationally at the boundary cases that matter most.

**Consequences.** Thin-context failure at L3 produces confident misalignment at L4 — architecturally correct action on an epistemically invalid inference.<sup>18</sup> The clinical decision support interface that presents confident inference without uncertainty metadata or escalation pathway is presenting thin context as authoritative clinical guidance; the practitioner who acts on it is making a decision whose validity the architecture has not verified. For the rural medical imaging node, this manifests as radiograph inference on a patient presentation that falls outside the model's validated demographic distribution: the model produces a confident classification, the interface presents it without qualification, and the practitioner has no architectural mechanism — no uncertainty indicator, no escalation trigger, no distributional boundary warning — through which to know that the confidence score they are acting on was not produced under conditions the model was designed for.<sup>19</sup>

**Human governance role.** Human authority defines the inference classes that require human validation before action. The autonomy envelope (C4, below) is bounded in part by the thin-context threshold: classes of inference where the gap between distributional confidence and domain validity is operationally significant require human validation before inference authorizes action. The architecture must make this boundary explicit and enforceable — an architectural constraint at the action authorization layer, not a recommendation communicated through interface design.

---

## C4 — AUTONOMY ENVELOPE DEFINITION

**Description.** Every edge AI deployment must operate within an explicit, bounded **autonomy envelope**: a formally specified set of action classes that the system may execute without human authorization prior to each instance of execution. Any action outside the envelope is an unauthorized extension of machine authority, regardless of the model's confidence level or the operational urgency the system assesses. The autonomy envelope is not a confidence threshold — a sufficiently confident model output does not create its own authorization. It is a mission-layer authority document: a specification, authorized by human governance before deployment, of what the system is permitted to do without seeking authorization for each execution instance. The distinction matters architecturally because a

---

17 Saleema Amershi et al., “Guidelines for Human-AI Interaction.” Cited above, §1, fn. 10. The escalation-pathway and uncertainty-surfacing requirements follow directly from the human-AI interaction principles developed there.

18 Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework.” Cited above, Abstract, fn. 3.

19 Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan, “Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations,” *Science* 366, no. 6464 (October 25, 2019): 447–453, <https://doi.org/10.1126/science.aax2342>. The demographic distribution gap between training data and deployment population is the clinical instantiation of the thin-context failure described here.

confidence threshold is a property of the model, while an autonomy envelope is a property of the authority relationship between the system and the human governance structure it serves.<sup>20</sup>

**Layer impact.** L6 is the primary layer: the autonomy envelope is a mission-layer authority document that the architecture enforces. L5 carries primary impact as the enforcement mechanism — the orchestration layer that authorizes execution must have access to the envelope specification and must apply it before permitting any out-of-envelope action to proceed. L3 carries secondary impact in the inference outputs that test the boundary; the model cannot know whether its output falls within the envelope, because the envelope is a governance document, not a model property. L4 carries secondary impact at the action authorization surface, where inference is translated into execution or escalated to human review depending on whether the contemplated action is within the pre-authorized envelope.

**Ramifications.** Every autonomous action must be classifiable as within-envelope or beyond-envelope at enforcement time. The envelope must be specified in terms that are machine-interpretable under operational conditions — not natural-language policy that requires interpretation when the mission is active and the action window is closing. The architectural pattern for beyond-envelope actions is not model-level suppression of the output; it is escalation to human authority before execution. This escalation pathway must be designed as a first-class architectural feature, with defined latency bounds, explicit fallback behavior when the human authority channel is degraded or unavailable, and complete audit logging at every decision point.<sup>21</sup> An autonomy envelope that cannot be enforced in degraded conditions is not an envelope; it is a nominal operating assumption.

**Consequences.** An undefined or informally defined autonomy envelope produces action creep. Each action that occurs without an adverse outcome normalizes that action class; the system accumulates implicit permissions that no human authority ever granted. The failure, when it comes, is not a malfunction in any technically detectable sense — the system is doing what it has been doing. For the ISR platform, the relevant boundary is the one between automated targeting support — sensor fusion, track correlation, solution computation, action queuing for human release — and weapon release authority, which no autonomous system holds. An architecture that does not enforce this boundary explicitly has no reliable mechanism to prevent its erosion under operational conditions where the cost of escalation appears high and the model's confidence in its solution appears sufficient.<sup>22</sup>

**Human governance role.** Human authority defines the autonomy envelope at deployment and owns any modification to it. Any expansion of the envelope is a human governance decision

---

20 Raja Parasuraman and Victor Riley, “Humans and Automation: Use, Misuse, Disuse, Abuse,” *Human Factors* 39, no. 2 (1997): 230–253, <https://doi.org/10.1518/001872097778543886>. The distinction between automation as a tool operating within human-defined authority bounds and automation as a substitute for human judgment is the foundational framing for the autonomy envelope concept.

21 Stuart Russell, *Human Compatible: Artificial Intelligence and the Problem of Control* (New York: Viking, 2019). The argument that machine authority must be derived from and remain bounded by human specification is the theoretical basis for the envelope's status as a governance document rather than a model property.

22 U.S. Department of Defense. “DoD Adopts Ethical Principles for Artificial Intelligence.” February 24, 2020. <https://www.defense.gov/News/Releases/Release/Article/2091996/>. The principle that humans shall exercise appropriate levels of judgment over the use of AI systems, and that AI systems shall not operate in ways that violate international humanitarian law, are the doctrinal basis for the mission-layer boundary in targeting applications.



— not an operational adaptation the system makes in response to mission pressure, and not an inference the system draws from the absence of adverse outcomes under the current envelope. The Skipjack Protocol, developed in §5, operationalizes this requirement across all seven layers. The principle that human judgment governs the use of AI-enabled systems applies at the mission layer as an architectural requirement: the boundary is enforced by the architecture before execution, not discovered after action by reviewing the record.<sup>23</sup>

---

## C5 — DISTRIBUTED COORDINATION UNDER FAILURE

**Description.** Edge systems cannot assume coordination-infrastructure availability. The connectivity that cloud-native orchestration depends on — the network path to a central scheduler, the latency-bounded channel to a configuration management system, the telemetry pipeline back to a monitoring plane — is, at the edge, a conditional resource that the operating environment may deny at any time, without notice, and for durations that exceed the mission window. Coherent operation when coordination is degraded or severed is a design requirement, not a graceful-degraded accommodation that the system provides as a bonus. An orchestration architecture designed around the assumption of coordination-infrastructure availability is not an edge architecture — it is a cloud architecture that has been relocated to the edge and will fail at precisely the moment the relocation's consequences become operational.<sup>24</sup>

**Layer impact.** L5 is the primary layer: orchestration is the coordination layer, and its architecture determines whether the system can maintain coherent state and authorized behavior without external coordination inputs. L0 carries secondary impact in the physical environment that severs coordination — jamming, power loss, physical infrastructure damage — each of which originates at the physical layer and propagates upward as a coordination failure. L1 carries secondary impact in the local compute required to execute coordination logic autonomously when the external coordination path is absent; local decision logic that depends on remote scheduling cannot operate when the scheduler is unreachable. L4 carries secondary impact in the action authorization pathway, which must have an explicitly defined behavior when the escalation channel is unavailable — because escalation unavailability and the need for escalation will coincide under the operating conditions that sever coordination. L6 carries secondary impact in the mission-layer authority that pre-authorizes the degraded-mode behavior envelope, without which the system has no authoritative basis for any autonomous action taken while coordination is absent.

**Ramifications.** A communication-assumption-free operation mode must be defined for every edge system before deployment: what the system does when coordination is absent, how long it operates in that mode before requiring human re-authorization, what actions it may and may not take without coordination confirmation, and how it reconciles its operational state when connectivity is restored. Distributed learning approaches offer one model for maintaining coherent behavior with intermittent connectivity — the federated approach demonstrates that useful operation is possible without centralized coordination at every update cycle — but the broader requirement is architectural discipline about coordination assumptions at every layer,

---

<sup>23</sup> Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework.” Cited above, Abstract, fn. 3. The H (Human-governed) and G (Governance) components of the HGC<sup>3</sup>AE<sup>2</sup> framework apply at L6 as the mission-layer expression of the bounded-autonomy principle.

<sup>24</sup> Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, “Edge Computing: Vision and Challenges.” Cited above, Abstract, fn. 1.

such that the behavior of the system when coordination fails is the intended behavior, not an undefined residual.<sup>25</sup>

**Consequences.** Unspecified degraded-mode behavior produces one of two failure patterns examined in §3: operational paralysis, where the system halts because a coordination dependency is absent and no autonomous behavior was pre-authorized for the degraded state; or uncoordinated autonomous action, where the system continues operating under stale authorization because no coordination requirement was enforced and no degraded-mode boundary was specified. Both are worse than explicit degraded-mode specification, because both produce outcomes that the governance structure did not authorize and cannot retrospectively account for. For the industrial power substation edge controller, this failure mode is not hypothetical: the fault condition that the controller exists to detect and respond to is precisely the condition that will sever its coordination path. A substation controller that requires external coordination to execute a fault response cannot execute the events it was deployed to handle.<sup>26</sup>

Coordination recovery — the process of re-synchronizing operational state when connectivity is restored — requires the same architectural rigor as the failure mode specification. A system that operated in an uncoordinated mode for an interval has taken actions, updated local state, and produced outputs that the coordination infrastructure was not privy to. Recovery is not connectivity restoration; it is a state reconciliation process that must be designed with an explicit authority model, a defined scope of autonomous reconciliation, and a human-governed authorization step for state changes that exceed that scope.

**Human governance role.** Human authority pre-authorizes the degraded-mode behavior envelope: the actions the system may take, the decisions it may make, and the duration it may operate without coordination confirmation before human re-authorization is required. This pre-authorization is bounded and specific — it is not a blanket license for autonomous operation in isolation but a design-time specification of what the system is trusted to do when it cannot check in. Coordination recovery is a human-governed event: return to coordinated operation requires explicit state reconciliation authorized by the human authority that owns the mission, not automatic resumption of coordinated behavior at the moment connectivity is restored.

---

---

## C6 — EDGE-NATIVE SECURITY POSTURE

**Description.** Edge security cannot be cloud-perimeter security applied to a remote node. The threat model that governs cloud deployment — in which the primary attack surface is network-accessible and the hardware is physically secured in a controlled facility — does not transfer to an architecture in which the hardware is deployed outside any controlled perimeter, the physical enclosure may be accessible to unknown parties, and the network path carries no

---

25 H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueray Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data.” Cited above, §1, fn. 11.

26 Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, “Basic Concepts and Taxonomy of Dependable and Secure Computing.” Cited above, fn. 13. The fault-tolerant systems design principles developed there — particularly the analysis of what constitutes coherent operation under fault conditions — are directly applicable to the degraded-mode specification requirement.



presumption of integrity. An edge node is, by definition, a compute asset deployed into an environment that no perimeter controls. The security architecture must be designed from that fact outward — not from a cloud security architecture that treats physical access as an implausible attack vector and then discovers otherwise in deployment.<sup>27</sup>

**Layer impact.** L0 is the primary layer: physical access to the hardware is the foundational attack vector at the edge, and every higher-layer security measure is contingent on the physical integrity of the device. L1 carries primary secondary impact because the compute budget that must be reserved for security operations — cryptographic processing, integrity attestation, anomaly detection — is the same compute budget that the inference workload requires; security compute at L1 is a genuine design-time allocation, not an assumption that spare capacity will absorb it. L2 carries secondary impact through data at rest: storage systems that are physically accessible can be read, modified, or exfiltrated regardless of network-layer encryption. L3 carries secondary impact through model integrity: firmware-level compromise at L0 can modify the model or its inference behavior without producing any network-visible indicator. L5 carries secondary impact through the remote policy enforcement assumption: centralized configuration management and policy push require connectivity that a compromised or isolated edge node cannot guarantee.

**Ramifications.** The threat model for an edge deployment must include physical access as a primary vector, not a residual one. This requires hardware-rooted attestation — the ability to verify, from a chain that originates in hardware rather than software, that the device is running the firmware and software it should be running. It requires tamper-evident physical enclosures where deployment conditions permit. It requires zero-trust network architecture that does not assume any network path to be trusted — because the network path from an edge node may transit infrastructure that is not under the operator’s control. Security compute must be explicitly budgeted at L1 before inference workload sizing is completed; the inference budget is whatever remains after the security budget has been reserved, not the inverse.

**Consequences.** The security collapse failure pattern — examined in §3 — originates here. A physical access compromise at L0 that goes undetected does not produce a single point of failure; it produces an **architectural consequence chain** that propagates through every layer above it. Firmware-level persistent access compromises the integrity of every process running on the device. Model poisoning at L3 — the introduction of adversarially modified weights through physical access — produces inference outputs that are incorrect in ways that are targeted rather than random. Classification spoofing at L2 — modification of the provenance metadata that governs what the data is authorized to be used for — produces silent authorization failures of the kind that C2 describes, but originating from adversarial action rather than pipeline design. All of these consequences originate at L0. None of them are detectable from the system’s operational output unless hardware attestation and integrity verification are architectural requirements.

**Human governance role.** Human authority defines the physical security perimeter: the procedures, personnel, and infrastructure controls that govern who has physical access to the hardware and under what conditions. This definition is a governance requirement, not an operational preference; it must be specified before deployment and must be enforced as a condition of continued operation. Remote attestation — the cryptographic verification of device integrity from a remote governance point — is a human governance requirement that the architecture must implement. When attestation fails or is unavailable, the governance response

---

<sup>27</sup> Rodrigo Roman, Javier Lopez, and Masahiro Mambo, “Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges for Current Scenarios and Future Prospects.” Cited above, §1, fn. 6.

is human-defined: escalation, operational suspension, or explicit risk acceptance. The system may not continue operating under unresolved attestation failure without human sign-off.

---

## C7 — LIFECYCLE MANAGEMENT AND MODEL DRIFT

**Description.** Edge AI systems degrade as the environments they were validated against evolve. The model that was calibrated against a patient population at one point in time is not automatically valid for the population that presents six months later; the model that was trained on ISR sensor data from one operational theater does not transfer automatically to another. Dataset shift — the condition in which the distribution of data the model encounters in deployment diverges from the distribution it was trained and validated on — is a structural property of deployed inference systems over time, not a failure mode of poor models.<sup>28</sup> Lifecycle governance — the defined process by which models are monitored for drift, recalibrated against validated data, and retired when they cannot be recalibrated to operational standard — is an architectural requirement that must be designed at deployment. A system that was not designed with recalibration and retirement pathways has no graceful response to the drift it will inevitably encounter.

**Layer impact.** L3 is the primary layer: the inference engine is the site where model validity is assessed and where drift manifests as output degradation. L5 carries primary secondary impact as the observability infrastructure that must surface drift signals — output distribution shifts, confidence degradation against ground-truth benchmarks — before they reach the threshold of operational consequence. L2 carries secondary impact in the data distribution monitoring that feeds drift detection: if the data arriving at inference has shifted from the training distribution, that shift must be detectable at L2 before it produces systematic error at L3. L4 carries secondary impact in the output quality indicators that surface to users; a system with no output-quality interface at L4 has no mechanism to communicate to operators that its outputs are degrading. L6 carries secondary impact in the authority to authorize model updates and the governance timeline within which recalibration must occur.

**Ramifications.** Drift detection must be an L5 observability function, not an after-the-fact audit. This requires that the observability architecture include, from deployment, the ability to monitor the statistical properties of model outputs against reference benchmarks — not merely to log that outputs were produced, but to assess whether the distribution of outputs indicates that the model is performing within its validated operating envelope. Model update, recalibration, and retirement pathways must be designed as first-class architectural features before deployment begins. The architecture must be able to load a new model version, validate it against a held-out benchmark dataset, compare its output distribution against the incumbent model's reference distribution, and rollback if the comparison does not meet the qualification threshold — all within the operational constraints of an edge deployment that may not have the bandwidth or compute to run a full evaluation pipeline on demand.

**Consequences.** Unmanaged drift produces the divergent reality failure pattern examined in §3: the model's operational assumptions decouple silently from actual conditions while the system continues producing confident outputs. For the rural medical imaging node, this manifests as a model that was calibrated on a patient population with one demographic and epidemiological profile operating on a patient population that has shifted — producing confident

---

<sup>28</sup> Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, eds., *Dataset Shift in Machine Learning* (Cambridge, MA: MIT Press, 2009). The foundational treatment of distributional mismatch between training and deployment contexts.

inference outputs that are systematically biased in ways that the clinical practitioners using the system have no mechanism to detect from the interface.<sup>29</sup> The outputs do not look wrong; they look like any other confident inference. The divergence is in the validation record that the outputs no longer satisfy — and that record, absent a functional drift-detection mechanism, is not visible at the operational surface.

**Human governance role.** Human authority defines the recalibration thresholds and retirement criteria: the drift magnitude at which recalibration is required, the recalibration protocol that a new model version must satisfy before it is authorized for deployment, and the retirement criterion below which no recalibration can restore the model to operational standard. Autonomous drift detection and flagging is within the autonomy envelope; recalibration itself — the decision to train or fine-tune against new data — may or may not be within the autonomy envelope depending on the deployment's classification posture. Model replacement requires human sign-off in every case: no autonomous system may replace its own inference model without explicit human authorization of the replacement version.

## C8 — MISSION-ADAPTIVE RECONFIGURATION

**Description.** Edge systems operate in contexts that change. The mission that was defined at deployment may be superseded by a new operational priority; the clinical triage protocol that was governing inference may be updated in response to an outbreak; the ISR platform that was configured for one collection mode may be reoriented to another. The architecture must be able to adjust its operational configuration in response to these changes without requiring full redeployment — while keeping every configuration change bounded by the mission-layer authority that authorized it.<sup>30</sup> An architecture that cannot adapt without redeployment is an architecture that creates operational gaps every time the mission changes. An architecture that can adapt without bound is an architecture that has no enforceable governance.

**Layer impact.** L6 is the primary layer: the mission authority that defines the reconfiguration boundary — what the system is permitted to change and on whose authorization — is a mission-layer governance document. L4 carries primary secondary impact in the service and interface configuration that must be able to accept new mission parameters and propagate them correctly to the layers below. L3 carries secondary impact in the model selection or parameter adjustment that mission changes may require — switching inference modes, loading alternative model versions, adjusting confidence thresholds. L5 carries secondary impact in the orchestration logic that manages reconfiguration sequences and ensures that configuration changes propagate consistently across all affected components without producing a partially-configured intermediate state.

**Ramifications.** The reconfiguration surface must be designed at deployment as an explicit architectural feature: which parameters are reconfigurable, under whose authorization, through which interface, and with which validation steps before the new configuration is accepted. Configuration changes that affect the autonomy envelope — that expand or contract what the system is authorized to do without human approval — must require explicit human authority; they cannot be an incidental consequence of a parameter change at another layer. Trust calibration across reconfiguration events requires architectural attention: a system that has

<sup>29</sup> Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan, “Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations.” Cited above, §2, fn. 19.

<sup>30</sup> Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework.” Cited above, Abstract, fn. 3.

successfully performed in one configuration does not automatically carry that trust to a new configuration.<sup>31</sup> Human operators must have an explicit basis for authorizing the trust transfer, which means the reconfiguration record must be auditable and the validation steps must be documented.

**Consequences.** Reconfiguration-unaware architecture produces execution-design divergence: the system executes against the old mission configuration while the new mission is in effect. This is not a model failure — the inference is correct for the mission it was configured for. It is a governance failure: the human authority that issued the new mission instruction has not achieved the architectural change the instruction was intended to produce. For the deployable operations center, where the mission configuration may change multiple times within a single operational period, execution-design divergence is not a hypothetical edge case but a predictable consequence of an architecture that was designed for a static operational context and deployed into a dynamic one.

**Human governance role.** Human authority authorizes mission changes and defines the reconfiguration boundary: which configuration parameters may be adjusted in response to mission changes, which require a dedicated authorization event, and which require full redeployment review. The mission-layer boundary defined in §1 applies here at the reconfiguration surface: the system's autonomy envelope extends to configuration changes that are within the pre-authorized reconfiguration boundary; changes that exceed that boundary require human sign-off before taking effect. Reconfiguration events must be logged and auditable so that the human governance authority can verify that every configuration in the system's operational history was authorized.

---

## C9 — COMPRESSED OBSERVABILITY

**Description.** Conventional observability architecture — continuous metric emission, distributed tracing across all service interactions, log aggregation to a central analysis plane — is not viable in bandwidth-starved, compute-constrained edge environments. The observability infrastructure that would produce comprehensive operational visibility competes directly with the inference workload for the compute and bandwidth the system requires to perform its primary function. Observability at the edge must be designed as a constrained resource: the minimum viable signal that supports human governance of the system's operational state must be identified, prioritized, and guaranteed — not the maximum possible telemetry that available bandwidth can carry.<sup>32</sup>

---

31 Kevin A. Hoff and Masooda Bashir, "Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust," *Human Factors* 57, no. 3 (2015): 407–434, <https://doi.org/10.1177/0018720814547570>. Trust state is not automatically transferred across operational configuration changes; each configuration transition requires a basis for trust authorization.

32 Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bannetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera, "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI," *Information Fusion* 58 (2020): 82–115, <https://doi.org/10.1016/j.inffus.2019.12.012>. The XAI principle that inference outputs must carry interpretable signals for human oversight applies at the edge in compressed form: the minimum viable explainability signal is the observability floor.

**Layer impact.** L5 is the primary layer: orchestration is the coordination and observability infrastructure, and the design choices made at L5 determine what visibility the human governance authority has over the system's state. L1 carries secondary impact because observability processing consumes compute — signal aggregation, anomaly detection, and telemetry packaging are real workloads that require real allocation. L2 carries secondary impact in the governance of telemetry data itself — observability data from a classified system is classified output that must be handled with the same provenance discipline that C2 requires for operational data. L3 carries secondary impact in the model performance signals that observability must surface: output confidence distributions, inference latency against requirement, and distributional boundary indicators are L3 properties that the L5 observability infrastructure must collect and export. L4 carries secondary impact in the user-facing health indicators that surface system state to operators without requiring them to parse raw telemetry.

**Ramifications.** Minimum viable observability must be an explicit architecture decision made at deployment, not an implicit residual of whatever bandwidth remains after inference. This requires identifying, for each operational context, the smallest set of signals that allows a human governance authority to distinguish: (a) the system is performing within its validated envelope; (b) the system is degraded but within its autonomy envelope for degraded-mode operation; (c) the system requires human intervention before continuing operation. Signals that carry this information include inference latency against requirement, output confidence distribution against reference baseline, model validity age against recalibration schedule, and resource utilization against the physical envelopes that C1 and C10 define. Each of these signals must have a guaranteed export path that functions under the bandwidth and compute constraints of the deployment — not a best-effort telemetry stream that is the first workload to be dropped when the system is under pressure.

**Consequences.** Observability-dark systems produce the silent failure pattern examined in §3. An operator who cannot see the system's operational state cannot govern it. Governance without observability is nominal, not functional: the human authority structures that nominally apply to the system's operation have no mechanism to detect when those operations are producing results outside the authorized envelope. The gap between apparent health and actual health — the interval during which the system is failing but the failure is below the observable floor — is an architectural design choice, not an operational circumstance. The architecture that specified an insufficient observability floor is the architecture that chose to operate the governance authority blind.

**Human governance role.** Human authority requires a minimum observability floor as a deployment condition: a specification of the signals that must be available before the system is authorized to operate. Systems that fall below this floor — because the telemetry export path has been degraded, because the observability compute budget has been preempted, or because a component failure has silenced a critical signal — must be flagged for human review regardless of their apparent operational health. The governance authority cannot make an informed decision about whether to continue, suspend, or modify a system's operation if the signals that inform that decision are unavailable. Observability floor enforcement is a governance responsibility, not an engineering preference.

---

## C10 — ENERGY-AWARE OPERATION

**Description.** Energy is a first-class architectural constraint at the edge, in the same category as compute capacity and thermal ceiling: not a quality metric to be optimized after the design is complete, but a governing boundary condition that defines whether operation is



possible at all. An AI system that has not been designed with an explicit energy budget will not fail gracefully when it exhausts its energy supply — it will stop. It will stop not because the inference was wrong, not because the model was invalid, and not because the hardware failed; it will stop because the architecture was not designed to account for the resource that makes all other operations possible.<sup>33</sup> At the rural medical imaging node operating on battery backup during a grid outage, the question is not whether the model is accurate — it is whether the system will still be running when the patient arrives. Energy-unaware design answers that question incorrectly.

**Layer impact.** L0 is the primary layer: the power supply, energy storage system, and energy harvesting capacity define the energy envelope that constrains every higher layer. L1 carries primary secondary impact because inference is the dominant energy workload on the device; the model selection and scheduling decisions made at L1 determine the energy draw profile that L0's supply must support. L3 carries secondary impact in model selection: inference energy consumption varies significantly across model architectures, and a model that is energy-efficient at cloud-side batch inference may be catastrophically expensive at on-device real-time inference under the thermal and clock constraints that C1 imposes.<sup>34</sup> L5 carries secondary impact in workload scheduling: intelligent scheduling that defers low-priority inference during energy-critical periods and guarantees energy headroom for high-priority mission-critical workloads is an L5 responsibility. L6 carries secondary impact in the mission-priority energy allocation that defines which workloads get energy headroom guarantees and which do not.

**Ramifications.** The energy budget for an edge AI system must be derived from the operating envelope before model selection and inference pipeline sizing. This means: characterizing the power supply and storage capacity under worst-case deployment conditions (not nominal conditions); allocating energy headroom for peak-tempo operation periods; sizing the inference workload to the remaining envelope after safety and security workloads have been served. Inference scheduling must be energy-state-aware: a scheduler that does not know the current state of charge, the discharge rate under current load, and the projected availability of replenishment is a scheduler that cannot make rational workload decisions. High-priority mission workloads must have guaranteed energy headroom — defined at design time, not dynamically competed for at execution time.

**Consequences.** Energy-unaware AI design produces operational paralysis at the worst possible moment: peak-tempo operation exhausts the energy budget precisely when mission demand is highest and the cost of system unavailability is greatest. For the ISR platform operating beyond its planned mission duration in a target-rich environment — increasing the inference frequency to maintain required output rates — the energy budget runs dry at the moment the operational tempo it was supporting is most critical. For the industrial power substation controller responding to a grid fault, the energy harvesting system that normally supplements the supply has just lost the primary grid that was supplementing it. Energy-unaware design does not produce a warning; it produces a shutdown at the boundary of the operating window it was designed for.

---

33 Alfredo Canziani, Adam Paszke, and Eugenio Culurciello, “An Analysis of Deep Neural Network Models for Practical Applications.” Cited above, §1, fn. 5.

34 Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané, “Concrete Problems in AI Safety,” arXiv:1606.06565, 2016, <https://arxiv.org/abs/1606.06565>. The analysis of safe interruptibility and resource constraints in deployed AI systems applies directly to energy-envelope governance: a system that cannot be safely stopped when its energy supply fails is a system that was not designed with safety as a first-class constraint.

**Human governance role.** Human authority sets the mission-priority energy allocation: the ranking of workloads by mission criticality that determines which inference tasks receive guaranteed energy headroom and which are shed when the energy budget is under pressure. This allocation is a mission-layer authority document — it reflects the human judgment about what the system must be able to do and what it may defer. Autonomous energy management operates within that allocation: the system may schedule and defer within the priority ranking but may not reorder priorities or drain reserves designated for higher-priority use. Energy criticality thresholds — the battery state of charge or supply draw levels at which autonomous operation should trigger a human escalation — must be defined at deployment and enforced by the architecture.

---

---

### 3. CROSS-LAYER FAILURE PATTERNS

The ten considerations in §2 describe what the architecture must satisfy. The five patterns described in this section describe what happens when it does not. Each pattern is an architectural consequence chain — a causal sequence in which a violation at one or more layers propagates across dependent layers, producing a failure that cannot be characterized, understood, or governed by examining any single layer in isolation. The patterns are not discrete incidents that occur when a single consideration is violated; they are emergent dynamics that arise when the interactions among considerations and layers are not designed correctly. Understanding them is a prerequisite for building the Skipjack Protocol's execution model, which is developed in §5.

---

#### ILLUSION OF CAPABILITY

The illusion of capability pattern occurs when a system appears to perform correctly because it has not yet been stressed to its actual operating boundary. The system produces outputs, meets its latency and throughput requirements under the conditions it has been tested in, and presents as operationally sound. The failure boundary is present in the architecture but has not been encountered; its existence has not been confirmed by stress testing under realistic deployment conditions. Because the architecture has not failed, it is treated as validated — but it has only been validated against the conditions it was tested in, which are not the conditions it will be deployed into.

The mechanism is a C1 violation combined with an L0→L1 constraint-reality gap. Lab-benchmark performance characterizes the system under controlled thermal and power conditions. Deployment into a high-altitude desert environment exposes the gap between the controlled test envelope and the actual operating envelope: thermal throttling at L0 reduces effective compute throughput at L1, inference latency increases beyond the mission requirement, and the output rate falls below the operational threshold — not because any component has failed, but because the architecture was sized to the wrong envelope.<sup>35</sup> The ISR platform that was characterized in garrison conditions reaches the failure boundary within ninety

---

<sup>35</sup> Alfredo Canziani, Adam Paszke, and Eugenio Culurciello, “An Analysis of Deep Neural Network Models for Practical Applications.” Cited above, §1, fn. 5. The energy and thermal analysis there characterizes the inference-capacity reduction under thermal constraint that produces the illusion-of-capability gap between lab and field performance.



minutes of deployment to a forward operating base under sustained high-altitude summer temperatures. The system is operating within its thermal specifications; the mission is not.

The governance failure that enables this pattern is the absence of stress-testing under realistic operating conditions as a deployment condition. This is not an engineering failure — the hardware is performing as specified. It is a governance failure: the authority that authorized the architecture for deployment did not require demonstration that the architecture would perform to its operational requirement under the conditions the deployment environment actually imposes. Absence of realistic-condition stress testing is not an oversight; it is a governance decision, explicit or implicit, and it produces this pattern.

---

## DIVERGENT REALITY

The divergent reality pattern occurs when the model's operational assumptions decouple silently from actual conditions. The model continues to produce confident outputs, the outputs continue to be acted upon, and the gap between what the model is inferring and what the actual world contains widens without triggering any observable alarm.

The mechanism is a combination of C7 (model drift) and C3 (thin context), acting together across L2 and L3. Dataset shift — the divergence between the distribution the model was trained and calibrated on and the distribution of data the deployed model is encountering — is not a sudden discontinuity; it is a gradual evolution that the model has no mechanism to detect in itself.<sup>36</sup> The outputs remain statistically coherent, well-calibrated against the original training distribution, and confidently expressed. The gap is not in the model's performance against the distribution it learned — it is in the mismatch between that distribution and the actual distribution the deployment context requires. For the rural medical imaging node operating on a patient population whose demographic profile has shifted from the population the model was trained on, the inference outputs remain confident and internally consistent, but their validity against the clinical standard — the determination of whether this patient, presenting at this facility, has this condition — has degraded in ways that no single output will reveal.<sup>37</sup>

The governance failure is the absence of drift detection as an L5 observability function and the absence of recalibration trigger thresholds as governance requirements. The drift is not silent because it cannot be detected; it is silent because the architecture was not designed to detect and surface it.

---

## SILENT FAILURE

The silent failure pattern occurs when the system continues to operate, produces outputs, and generates no alarm condition — but its outputs are no longer valid in the operational sense the governance framework requires. There is no crash, no performance threshold breach, no visible error. The failure is below the observable floor: beneath the minimum signal set that the observability infrastructure was designed to surface, the system is degraded in ways that the governance authority cannot see.

---

<sup>36</sup> Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, eds., *Dataset Shift in Machine Learning*. Cited above, §2, fn. 28.

<sup>37</sup> Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan, “Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations.” Cited above, §2, fn. 19.

The mechanism is a combination of C2 and C9, propagating through L2 and L4. Classification metadata lost at L2 produces inference outputs at L3 that are presented at L4 without the governance attributes that would trigger an authorization check. The interface presents what appears to be valid output; the human operator acts on it; no alarm has fired because no component of the architecture is designed to detect that the governance surface was absent.<sup>38</sup> This is the thin-context condition operating below the observability floor: confident inference produced under conditions that violate the architectural requirements for authorized inference, surfaced to the operator as if those requirements were satisfied.<sup>39</sup> For the industrial power substation edge controller whose fault detection model has been degraded by sensor drift that is below the observability floor's detection threshold, the failure is not that the model was wrong — it is that the architecture had no mechanism to discover that the model was operating on degraded sensor data until the fault it was meant to detect was not detected.

The governance failure is the absence of an enforced observability floor as a deployment condition and the absence of uncertainty surfacing at L4 as an interface requirement. The architecture chose the observable floor it would operate with; that choice is a governance decision, and it produced an interval of undetected degradation.

---

## SECURITY COLLAPSE

The security collapse pattern is the architectural consequence chain of a physical security failure at L0. Unlike the other patterns, which involve distributional or operational failures that develop over time, security collapse can occur at the moment of physical compromise and propagates upward through the layer stack as a consequence of that single event.

The mechanism is a C6 violation at L0 that bypasses every higher-layer security control. Cryptographic protections at L1, data governance at L2, model integrity at L3, and configuration integrity at L5 all depend on the physical integrity of the device they run on. Physical access that compromises the firmware — the introduction of a hardware implant during servicing, modification of the boot environment in an unattended interval — compromises the platform that all higher-layer controls execute on.<sup>40</sup> The ISR platform serviced outside a controlled facility provides the attack surface; a firmware-level persistent access capability introduced during servicing produces inference outputs that are incorrect in ways that are targeted and consistent, classification metadata that is spoofed to present adversarially modified data as validated, and targeting outputs that reflect the adversary's objectives rather than the mission's. The L4

---

38 Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung, "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys* 55, no. 12, Article 248 (March 2023): 1–38, <https://doi.org/10.1145/3571730>. The hallucination survey's taxonomy of confident-but-invalid outputs is the natural language generation analogue of the silent failure pattern: outputs that are confidently expressed and structurally valid but factually or operationally incorrect, without a signal that distinguishes them from valid outputs.

39 Justin H. Kuiper, CISSP, "Epistemic Constraints and Semantic Compression in Natural Language Processing." Cited above, Abstract, fn. 4.

40 Rodrigo Roman, Javier Lopez, and Masahiro Mambo, "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges for Current Scenarios and Future Prospects." Cited above, §1, fn. 6.

interface presents these outputs as operationally valid; there is no higher-layer mechanism that can detect the compromise without hardware attestation.<sup>41</sup>

The governance failure is the treatment of physical security as an operational preference rather than an architectural requirement: the absence of hardware attestation as a deployment condition, the absence of tamper-evident enclosure requirements, and the absence of a governance process that specifies who may perform hardware servicing and under what physical security conditions.

---

## OPERATIONAL PARALYSIS

The operational paralysis pattern occurs when the system cannot act because no layer is authoritative under the conditions that actually prevail. The coordination infrastructure that the autonomy envelope depends on is absent, the pre-authorized degraded-mode behavior was not defined, and the system's response to the absence of authorization is to wait — for coordination to be restored, for a human to authorize, for a signal that does not arrive. The mission window closes while the system waits.

The mechanism is a combination of C4 and C5, acting at L5 and L6. The coordination infrastructure that was assumed to be available is severed by the same operating condition that triggered the need for autonomous action.<sup>42</sup> The degraded-mode behavior envelope was not defined at deployment — because the architecture treated coordination as available and did not specify what the system was authorized to do in its absence. The substation edge controller whose grid fault response was designed around a coordination path that the fault condition itself severs has no pre-authorized action to take. Each node in a multi-node response checks with its peers; each peer checks back; no node acts because no node's local autonomy envelope includes the authorization to act without confirmation; the fault propagates while the coordination wait resolves nothing.

The governance failure is the absence of degraded-mode pre-authorization as a governance requirement. The autonomy envelope was defined for the nominal case — the case in which coordination is available and escalation is possible — but not for the degraded case, which is precisely the case in which autonomous action is most needed. The human authority that defined the envelope chose not to address the failure mode; the pattern is the consequence.

---

---

---

41 Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing." Cited above, §2, fn. 13. The dependability taxonomy's treatment of malicious faults — faults introduced by adversarial action rather than component degradation — is the formal basis for the security collapse chain analysis.

42 H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueray Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data." Cited above, §1, fn. 11.

## 4. HUMAN GOVERNANCE

Human governance at the edge is not a supervision posture — it is an architectural function. The seven layers and the ten considerations define the space within which the architecture operates and the conditions it must satisfy, but they do not supply the authority that makes those definitions binding. That authority is human. Without functional governance, every constraint in this paper reverts to advisory status: a characterization of what the architecture should do that the architecture may or may not be designed to enforce. The HGC<sup>3</sup>AE<sup>2</sup> framework's H (Human-governed) and G (Governance) components are not additions to an otherwise technical architecture — they are the components that make the technical architecture operational.<sup>43</sup> This section specifies what governance must supply at each layer, what it must supply for each of the ten considerations, and the failure modes that occur when it does not.

### GOVERNANCE PER LAYER

**L0 — Physical.** At the physical layer, human governance establishes three authority functions. The physical security perimeter: who has physical access to the hardware, under what conditions, and through what authorization chain, including procedures for servicing, transport, and end-of-life disposal. The energy envelope definition: the power budget, storage capacity, and harvesting parameters that bound every higher-layer workload decision. The operating-environment limits: the thermal range, vibration, humidity, and altitude parameters within which the architecture is authorized to operate — not as guidance to the hardware manufacturer, but as a deployment condition that human authority specifies and maintains.<sup>44</sup>

**L1 — Compute.** At the compute layer, human governance establishes the budget allocation, attestation requirements, and performance-floor definition that bound the inference workload. Compute budget allocation is a governance decision: the authority that defines how much compute the inference workload receives, how much is reserved for security operations, and how much is reserved for system management is the authority that determines whether the architecture can execute its mission under deployment conditions. Hardware integrity attestation — the process by which the identity and firmware state of the compute platform is verified before operation — is a governance requirement established at deployment and enforced as a condition of continued authorization.

**L2 — Data.** At the data layer, human governance establishes the classification schema, the provenance chain structure, and the retention and disposal authorities. The classification schema — the mapping between data type, collection context, and governance surface — is a human authority document that automated systems apply but do not create. The provenance chain structure defines what lineage metadata a data element must carry from collection through action; the governance requirement is that this structure be fully specified before the

---

<sup>43</sup> Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework.” Cited above, Abstract, fn. 3.

<sup>44</sup> National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. Gaithersburg, MD: National Institute of Standards and Technology, January 2023. <https://doi.org/10.6028/NIST.AI.100-1>. The NIST AI RMF's governance-function taxonomy — Map, Measure, Manage, Govern — provides the institutional framework for understanding physical-layer governance as an organizational function rather than a purely technical specification.

first data element is collected. Retention and disposal governance defines when data must be deleted, who authorizes extended retention, and how disposal is certified and recorded.<sup>45</sup>

**L3 — AI.** At the AI layer, human governance establishes model validation authority, the inference boundary definition, and the thin-context escalation threshold. Model validation authority is the human decision that a model version is authorized for a specific operational context — not that it has passed a benchmark, but that it has been validated against the domain-specific requirements that the deployment context imposes. The thin-context escalation threshold — the confidence or distributional boundary below which inference must be routed to human review before it can authorize action — is a governance specification that defines where the architecture’s autonomous inference authority ends and human judgment begins.

**L4 — Applications.** At the applications layer, human governance establishes the action authorization boundaries, the output review requirements, and the escalation path definitions that translate the inference boundary at L3 into operational constraint. The action authorization boundary specifies which inferences may cause which actions without human review before execution. The output review requirement specifies which inference outputs must pass through human review regardless of model confidence. The escalation path definition specifies who receives the review request, within what latency, and what happens when the escalation channel is unavailable — because the unavailability of the escalation channel and the need for escalation will coincide under the operating conditions that matter most.

**L5 — Orchestration.** At the orchestration layer, human governance establishes the observability floor, the degraded-mode pre-authorization, and the coordination failure protocol. The observability floor is the minimum signal set that must be available for the governance authority to maintain functional oversight — the governance expression of C9’s architectural requirement. Degraded-mode pre-authorization is the governance document that specifies what the system is authorized to do when coordination is absent — the governance expression of C5’s architectural requirement. The coordination failure protocol defines the escalation sequence, the human decision authority, and the recovery authorization process for the transition back to coordinated operation.<sup>46</sup>

**L6 — Mission.** At the mission layer, human governance defines the reason the system exists: the mission statement, the authority structure, the success and failure criteria, and the process for authorizing changes to any of these. Mission definition is the foundational governance act — every other governance document at every other layer is derived from the

---

<sup>45</sup> Luciano Floridi, Josh Cowls, Monica Beltrametti, Raja Chatila, Patrice Coquillard, Virginia Dignum, Christoph Luetge, Robert Madelin, Ugo Pagallo, Francesca Rossi, Burkhard Schafer, Peggy Valcke, and Effy Vayena, “An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations,” *Minds and Machines* 28 (2018): 689–707, <https://doi.org/10.1007/s11023-018-9482-5>. The AI4People framework’s beneficence, non-maleficence, autonomy, justice, and explicability principles apply at L2 as data ethics governance requirements — particularly the explicability principle, which demands that data collection, classification, and use be legible to the governance authority.

<sup>46</sup> Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending certain Union legislative acts (AI Act). OJ L 2024/1689. The EU AI Act’s requirements for high-risk AI systems — risk management, data governance, technical documentation, transparency, human oversight, robustness, and accuracy — map directly onto the per-layer governance functions specified here, particularly at L5 (logging and monitoring requirements) and L3 (accuracy and robustness documentation).



mission authority and is bounded by it. Envelope change authorization — the process by which any governed parameter is modified — requires sign-off at the mission layer before the change propagates to any other layer. This is the governance expression of the mission-layer boundary defined in §1.<sup>47</sup>

## GOVERNANCE PER CONSIDERATION

**C1 — Constraint-Reality Primacy.** The human judgment that C1 requires is envelope-setting: the specification of the physical operating boundaries — power budget, thermal ceiling, bandwidth floor, latency bound — within which the architecture is authorized to operate. This specification cannot be produced by the architecture itself; it is derived from a characterization of the actual deployment environment, authorized by the human governance authority responsible for L0 and L1.

**C2 — Data Provenance and Classification Persistence.** The human judgment that C2 requires is classification authority: the decision about what classification, sensitivity designation, and provenance metadata a data element receives at collection. Automated systems apply the schema; they do not define it, and they do not reclassify data below its collection-time classification without explicit human authorization.<sup>48</sup>

**C3 — Inference Reliability Under Constraint.** The human judgment that C3 requires is the escalation threshold: the specification of the class of inference queries that require human validation before the inference authorizes action. This threshold defines the boundary between inference the architecture is authorized to act on autonomously and inference it must route to human review — a boundary that cannot be derived from the model's confidence scores.

**C4 — Autonomy Envelope Definition.** The human judgment that C4 requires is envelope definition and every subsequent modification. The autonomy envelope is a governance document from first specification to final revision. It cannot be inferred from operational performance, extended incrementally without authorization, or modified in response to mission demand without explicit human sign-off. Every extension of the envelope is a governance decision, regardless of the model's confidence or the operational urgency the system perceives.

**C5 — Distributed Coordination Under Failure.** The human judgment that C5 requires is degraded-mode pre-authorization: the specification, before deployment, of what the system is permitted to do when coordination is absent. This pre-authorization extends the architecture's authorized autonomy into the failure modes where human oversight is least available — a decision that requires explicit human judgment precisely because it will be executed in the system's most isolated operational state.

**C6 — Edge-Native Security Posture.** The human judgment that C6 requires is the physical security mandate: the specification of who has authorized physical access to the hardware, what conditions govern that access, and what the governance response is to access outside those conditions. Hardware attestation requirements and tamper-evidence standards

---

<sup>47</sup> Ben Shneiderman, *Human-Centered AI* (Oxford: Oxford University Press, 2022). The human-centered AI design principles developed there — reliable, safe, trustworthy systems that are subject to human control — are the design philosophy that the L6 mission-layer governance function implements at the authority level: the system's operational scope is defined by human judgment and is subject to human modification.

<sup>48</sup> U.S. Department of Defense. "DoD Adopts Ethical Principles for Artificial Intelligence." Cited above, §2, fn. 22.

are governance documents. Physical security at the edge is a human authority specification that the architecture enforces, not an engineering preference.

**C7 — Lifecycle Management and Model Drift.** The human judgment that C7 requires is recalibration and retirement sign-off. Drift detection may be automated; the decision to recalibrate, and the authorization of the recalibrated model for operational deployment, are human governance acts. Model retirement — the determination that a model cannot be restored to operational standard through recalibration — requires human sign-off before the model is decommissioned.

**C8 — Mission-Adaptive Reconfiguration.** The human judgment that C8 requires is mission-change authorization: the explicit sign-off that a new mission configuration is authorized before it takes effect in the architecture. Configuration changes that affect the autonomy envelope, the inference boundary, or the action authorization surface require human authorization at the appropriate governance level before the change is applied.

**C9 — Compressed Observability.** The human judgment that C9 requires is the observability floor mandate: the specification of the minimum signal set that must be available before the system is authorized to operate, and that must trigger human review if it falls below threshold during operation. The system cannot authorize its own continued operation when its governance visibility falls below the floor.

**C10 — Energy-Aware Operation.** The human judgment that C10 requires is mission-priority energy allocation: the ranking of workloads by operational criticality that determines which inference tasks receive guaranteed energy headroom and which are shed when the energy budget is under pressure. This allocation reflects human judgment about the mission and cannot be derived from the workloads' own assessments of their priority.

## FAILURE MODES OF WEAK GOVERNANCE

Governance is not a binary condition. The three failure modes described here are degraded forms that exist on a continuum between the presence of a governance document and functional operational governance. Each corresponds to a recognizable institutional pattern; each produces architectural consequences that the governance authority believes it has prevented.

**Nominal governance** is the pattern in which governance structures exist on paper but are not operationally enforced. The classification schema has been written but is not applied consistently at collection. The autonomy envelope has been defined but is not enforced at the action authorization layer. The observability floor has been specified but the architecture does not respond when it falls below threshold. Nominal governance produces the illusion of oversight: the governance authority believes the constraints it defined are being enforced; the architecture is not enforcing them. This is the most dangerous form of weak governance because it is invisible until operational consequence reveals it — by which time the failure the governance was meant to prevent has already occurred.<sup>49</sup>

---

<sup>49</sup> Organisation for Economic Co-operation and Development. *OECD Principles on Artificial Intelligence*. OECD/LEGAL/0449. May 22, 2019. <https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449>. The OECD AI Principles' requirement for accountability — that those responsible for AI systems apply appropriate governance mechanisms, including audit trails and transparency mechanisms — identifies nominal governance (governance without operational enforcement) as the primary accountability gap.



**Deferred governance** is the pattern in which governance requirements are designated “to be defined at operational startup” rather than being resolved before deployment. The thin-context escalation threshold will be determined once the system is in the field. The degraded-mode pre-authorization will be addressed at the first coordination failure. The physical security perimeter will be established once the deployment site is prepared. Deferred governance produces undefined-envelope deployment: a system authorized to operate whose operational boundaries have not been established. Every deferred governance item is a gap in the architecture that the deployment environment will encounter at the moment that gap matters most.<sup>50</sup>

**Distributed governance without integration** is the pattern in which each layer has a governance owner but no cross-layer authority resolves conflicts, dependencies, or gaps between them. The L2 classification authority defines a schema the L3 model validation authority was not consulted on. The L5 observability floor is set without reference to the L1 compute budget that determines whether it can be serviced. The L6 mission authority modifies success criteria without notifying the L4 action authorization owner. Distributed governance without integration produces coordination failures at the governance level — the same structural failure that C5 describes in the architecture, operating one level up. Governance is a cross-layer function, and governance that is designed only per-layer is not governance of the architecture. The prescription is the HGC<sup>3</sup>AE<sup>2</sup> framework’s H and G components as architectural requirements: governance that is operationally active, fully specified before deployment, and integrated across all seven layers as a unified authority structure.

---



---

## 5. THE SKIPJACK PROTOCOL — APPLICATION SECTION

The Skipjack Protocol is an operational doctrine: a codified set of principles and enforcement practices that applies the thesis of this paper — mission defines architecture; environment constrains it; data governs it; autonomy executes the mission; human governance defines the boundaries of all four — across all seven layers and all ten considerations, under all operating conditions including those in which coordination infrastructure is degraded or absent. It is not a software architecture, a middleware specification, or a configuration framework. It is the operating procedure that human governance applies before deployment and that the architecture is designed to enforce during operation.

The protocol was first named in Paper One, where it was identified as the operational vehicle through which the HGC<sup>3</sup>AE<sup>2</sup> framework’s governance components would be applied in execution.<sup>51</sup> Paper Three promotes it from reference to full application: every section of this paper has been building toward the protocol’s specification, and every consideration in §2 names the Skipjack Protocol’s enforcement mechanisms in the human governance role paragraph. The name is a signal: a skipjack moves fast in constrained water. The protocol is

---

<sup>50</sup> Carina Prunkl, “Institutionalising Human Oversight of AI: Some Lessons from Aviation,” *Minds and Machines* 34 (2024), <https://doi.org/10.1007/s11023-024-09667-z>. The aviation-safety parallel developed there — in which deferred governance during aircraft certification produced systematic safety gaps that were discovered in operation — is the institutional archetype for the deferred-governance failure mode described here.

<sup>51</sup> Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework.” Cited above, Abstract, fn. 3.

designed for the same conditions — high operational tempo, constrained resources, degraded coordination, and a thesis that cannot wait for ideal circumstances to become operative.

The protocol's five core principles each address a class of architectural failure that the prior sections have characterized. They are not a checklist and they are not independent axes of quality; they are the operational expression of the seven-layer architecture and the ten considerations in a form that human governance can enforce before deployment and that the architecture can verify during operation.

## FIVE CORE PRINCIPLES

**Constraint-first design.** Architecture is designed to the constraint, not the ideal case.<sup>52</sup> Every design decision governed by the Skipjack Protocol begins with a characterization of the physical operating envelope — the power budget, thermal ceiling, bandwidth floor, and latency bound that the deployment environment imposes — and derives every subsequent parameter from that characterization. The inference pipeline is sized to the constrained compute capacity. The model is selected for the constrained power envelope. The orchestration is designed for the constrained coordination infrastructure. The model is not selected first and then qualified for field conditions as a separate exercise; field conditions are the design condition.

Constraint-first design is the architectural expression of C1 (Constraint-Reality Primacy) and C10 (Energy-Aware Operation). It is also the discipline that prevents the illusion-of-capability failure pattern from §3: a deployment that was designed to the constraint cannot be surprised by the constraint because the constraint was the design requirement. The execution discipline: before any model selection decision, before any pipeline sizing, before any infrastructure specification, the physical operating envelope must be characterized under worst-case deployment conditions and certified by the human governance authority responsible for L0 and L1.

**Signal over data.** In bandwidth-starved environments, the smallest signal that carries authoritative information is more valuable than the largest dataset that does not.<sup>53</sup> Signal over data applies to two distinct domains within the protocol. First, inference: the model is selected and tuned to produce outputs that carry maximum decision-relevant information about the domain question the mission requires, with epistemic confidence metadata that surfaces the model's distributional boundary to the operator. This is the architectural response to the thin-context condition that Paper Two identifies — the protocol operationalizes the insight as a design discipline rather than leaving it as an analytical observation. Second, observability: the observability architecture is designed to produce the minimum viable signal set — the compressed observability required by C9 — that allows the governance authority to distinguish operational health from apparent health without consuming the bandwidth and compute that the inference workload requires.

The signal-over-data principle is the architectural expression of C3 (Inference Reliability Under Constraint) and C9 (Compressed Observability). It closes the gap that those

---

<sup>52</sup> Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, "Edge Computing: Vision and Challenges." Cited above, Abstract, fn. 1. The foundational analysis of edge computing's constraint regime establishes that effective edge architecture begins with the constraint characterization, not with the feature specification.

<sup>53</sup> Justin H. Kuiper, CISSP, "Epistemic Constraints and Semantic Compression in Natural Language Processing." Cited above, Abstract, fn. 4. The signal-over-data principle is the architectural operationalization of the epistemic insight developed there: that inference authority requires domain-specific validation-mechanism grounding, not distributional confidence volume.

considerations identify between what the architecture produces and what the governance authority needs to see: not maximum output volume but maximum output authority.

**Bounded autonomy.** Autonomous action is permissible only within the explicitly defined autonomy envelope; the protocol enforces this boundary at every layer, not as a policy but as an operational check that precedes every execution.<sup>54</sup> The bounded-autonomy principle is the architectural expression of C4 (Autonomy Envelope Definition), applied at L6 where the envelope is defined as a mission-layer authority document, at L5 where it is enforced by the orchestration layer before action is permitted, and at L4 where the action authorization surface implements the enforcement check. The enforcement mechanism is not the model's assessment of whether its output falls within the envelope — the model cannot know this, because the envelope is a governance document, not a model property. The enforcement mechanism is architectural: the action authorization layer consults the envelope specification before permitting any action, and routes beyond-envelope actions to human authority before execution.<sup>55</sup>

The execution discipline: the autonomy envelope is specified, machine-encoded, and enforced at L4 and L5 before deployment authorization is granted. No deployment proceeds without a machine-enforceable envelope. No envelope is modified without human sign-off at the mission layer.

**Persistent data governance.** Data classification and provenance are operational controls, not metadata. The persistent-data-governance principle requires that classification and provenance information be treated as first-class attributes of every data element from collection through action, with no architectural pathway that strips, suppresses, or fails to propagate them. This is the architectural expression of C2 (Data Provenance and Classification Persistence) and it applies at every layer where data transits: at L2 at collection, at L3 through inference, at L4 at the action surface, and at L5 through the telemetry pipeline. The enforcement mechanism is architectural: data pipeline components that cannot carry governance metadata are not compliant with the protocol. The execution discipline: every data surface, storage format, and pipeline component is audited for governance-metadata propagation before deployment authorization is granted.

The persistent-data-governance principle also governs inference outputs: an output derived from classified input data is classified output, and the protocol requires this inheritance to be implemented as an architectural mechanism, not documented as a policy that the pipeline may or may not honor.

**Distributed resilience.** The architecture maintains coherent operation when coordination infrastructure is unavailable; resilience is a design property, not a failure-recovery option.<sup>56</sup> The distributed-resilience principle requires that the system's behavior in the absence of

---

<sup>54</sup> Raja Parasuraman and Victor Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse." Cited above, §2, fn. 20. The automation-use typology developed there — specifically the misuse failure mode of over-reliance on automation beyond its design envelope — is the behavioral substrate that the bounded-autonomy principle is designed to prevent at the architectural level.

<sup>55</sup> Kevin A. Hoff and Masooda Bashir, "Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust." Cited above, §2, fn. 31. The trust-calibration requirement — that human authority must have an explicit basis for trusting each operational state of an automated system — applies to the bounded-autonomy enforcement check: each execution instance requires a determination that the contemplated action falls within the pre-authorized envelope, not an accumulated trust that it probably does.

coordination infrastructure be fully specified before deployment — as a designed operating mode with its own autonomy envelope, its own governance authorizations, and its own state-reconciliation protocol for the return to coordinated operation. It is not a best-effort fallback that the architecture provides when it can; it is a pre-authorized operating mode that the architecture enters at the moment coordination is severed.

The distributed-resilience principle is the architectural expression of C5 (Distributed Coordination Under Failure), and it also enforces the physical security requirements of C6: when the coordination path is severed — as it will be at the moment of a physical security incident at a forward-deployed node — the system's pre-authorized operating mode must include the security posture that applies in the absence of centralized policy enforcement. The execution discipline: the degraded-mode operating specification, including autonomy envelope, security posture, and state-reconciliation protocol, is fully specified, human-authorized, and machine-enforced before deployment authorization is granted.

## MAPPING TO THE SEVEN LAYERS

Constraint-first design enforces at L0 and L1: the physical envelope characterization is an L0 output that directly determines the compute capacity at L1, and the Skipjack constraint-first discipline requires that the L1 inference capacity envelope be derived from that characterization, not assumed from benchmark specifications. Signal over data enforces at L2, L3, and L5: data collection surfaces at L2 are designed to produce high-fidelity signals rather than high-volume data streams; inference at L3 is tuned to output decision-relevant signals with epistemic confidence metadata; observability at L5 is sized to the minimum viable signal set rather than the maximum available bandwidth. Bounded autonomy enforces at L4, L5, and L6: the autonomy envelope is a mission-layer (L6) authority document, the enforcement mechanism is an L5 orchestration check, and the action authorization boundary is an L4 interface requirement that blocks any action pending an in-envelope determination. Persistent data governance enforces at L2 through L4: classification is assigned at L2, propagated through L3 inference, and enforced at the L4 action surface; no layer in this chain is exempt from the governance-metadata propagation requirement. Distributed resilience enforces at L5 and L6: the degraded-mode specification is an L5 orchestration requirement — the specification of the system's behavior when coordination is absent — authorized by the L6 mission layer as a pre-deployment governance act.

## MAPPING TO THE TEN CONSIDERATIONS

Constraint-first design addresses C1 (Constraint-Reality Primacy), C10 (Energy-Aware Operation), and C9 (Compressed Observability): these are the three considerations whose primary failure mode is the gap between what the architecture was designed for and what the deployment environment actually provides. An architecture designed to the constraint is an architecture that has already resolved this gap by treating the constraint as the design requirement rather than the exception to it.

Signal over data addresses C3 (Inference Reliability Under Constraint) and C9 (Compressed Observability): both considerations describe architectural requirements for maximum-fidelity, minimum-volume information handling under constraint. The signal-over-data discipline operationalizes both simultaneously — at inference (C3) by requiring epistemic confidence metadata, and at observability (C9) by requiring minimum viable signal design.

---

56 H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agueray Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data." Cited above, §1, fn. 11.

Bounded autonomy addresses C4 (Autonomy Envelope Definition) and C8 (Mission-Adaptive Reconfiguration): the autonomy envelope defined in C4 is the governance document that the bounded-autonomy principle enforces at execution time; the reconfiguration boundary defined in C8 is a specialized expression of that envelope for configuration-change events. Bounded autonomy also enforces the human governance role identified in C3: the class of inference queries that require human validation before authorizing action is the architectural boundary where the autonomy envelope meets the thin-context threshold.

Persistent data governance addresses C2 (Data Provenance and Classification Persistence), C3 (where inference output classification is derived from input classification), and C7 (Lifecycle Management and Model Drift): the data distribution monitoring that drives drift detection must carry provenance attributes to be operationally useful, because drift is a property of the relationship between current data and the reference distribution, not a property of current data in isolation.

Distributed resilience addresses C5 (Distributed Coordination Under Failure) and C6 (Edge-Native Security Posture): C5's degraded-mode pre-authorization is the primary target; C6's hardware attestation requirement is a component of the security posture that the distributed-resilience principle must specify for the isolated operating mode, when centralized policy enforcement is unavailable and the architecture must maintain its security guarantees through local means.

## EXECUTION MODEL

The Skipjack Protocol is applied through a seven-step governance sequence that must be completed before deployment authorization is granted.<sup>57</sup> First, define the mission and the mission-layer boundary: the statement of operational intent, the authority structure, and the explicit line between autonomous and human-authorized action. Second, characterize the operating environment and set the physical envelopes: a worst-case assessment of power, thermal, bandwidth, and latency constraints at L0 and L1. Third, classify the data surfaces and build the provenance chain: the assignment of classification schema, the specification of provenance metadata requirements, and the validation that every pipeline component can propagate governance attributes from collection through action. Fourth, validate inference against the autonomy envelope and the thin-context threshold: the model validation process that confirms the model is authorized for the deployment context and that the escalation threshold for below-boundary inference is specified and enforced at L4. Fifth, define the action authorization boundaries: the specification of which inferences may authorize which actions without human review, expressed in machine-interpretable terms. Sixth, configure the observability floor and the degraded-mode pre-authorization: the minimum signal set, and the full specification of the system's authorized behavior — including security posture and autonomy envelope — when coordination is absent. Seventh, conduct a human governance review against all ten considerations: a documented confirmation that every consideration's governance requirement has been specified, is machine-enforced, and is operationally active before deployment authorization is granted. Deployment does not proceed until all seven steps are complete and the governance review record is signed.

---

<sup>57</sup> National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. Cited above, §4, fn. 44. The NIST AI RMF's Govern-Map-Measure-Manage sequence provides the institutional precedent for a pre-deployment governance review as a deployment authorization condition, not an after-the-fact audit.



## CONSEQUENCES WITHOUT SKIPJACK

The architectural consequence chain when the Skipjack Protocol is absent maps directly onto the five failure patterns from §3. Undefined physical envelopes — the absence of constraint-first design — produce the illusion of capability: architectures sized to the wrong operating condition and deployed into the right one. Absent provenance persistence — the absence of persistent data governance — produces silent failure at the authorization boundary: outputs presented without governance metadata, actions taken without authority, and no architectural mechanism to detect the gap. Observability designed without the signal-over-data discipline produces silent failure at scale: system degradation proceeding below the observable floor while the interface presents apparent operational health. No degraded-mode pre-authorization — the distributed-resilience principle not applied — produces operational paralysis at the moment of coordination failure. No human governance review at the seventh step — the protocol's final check omitted — produces nominal governance: a system authorized to operate without a confirmed mapping between the governance documents on file and the constraints the architecture actually enforces.

The protocol does not prevent every failure. It prevents every failure that originates in the absence of one of its five principles and that is detectable through the seven-step governance review before deployment. That is the set of failures this paper has characterized. The doctrine is the claim that this set contains the failures that matter most.

---

---

## 6. FUTURE STUDY

The doctrine codified in this paper is operationally actionable in its current form. The seven-layer framework, the ten considerations, the five failure patterns, and the Skipjack Protocol collectively constitute a deployable architectural doctrine for edge AI systems — not a research agenda but a working specification that the Skipjack Protocol's seven-step execution model operationalizes. The ten research vectors that follow represent the next layer of depth: the open questions whose resolution would advance the doctrine from principled specification to formal proof, from architectural practice to verifiable engineering, and from operational guidance to regulatory standard.

**1. Adaptive Context.** The architecture specified in this paper treats the operating context as given at deployment and stable across the mission window. The practical reality of edge AI deployment is that operating contexts change: the mission is respecified, the sensor environment shifts, the threat picture evolves, the patient population presents differently than anticipated. The question of how an edge AI system can dynamically adjust its inference domain and validation criteria in response to operating-context change — without full redeployment, without violating the governance constraints that the Skipjack Protocol enforces, and without producing the silent-failure pattern that results from a mismatch between the model's assumptions and the actual context — is the frontier of this paper's argument. Agent-based architectures that model their operating context as an explicit simulation and update that model from observation represent one research vector, though their computational requirements at edge-native resource levels remain unsolved.<sup>58</sup>

---

<sup>58</sup> Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein, "Generative Agents: Interactive Simulacra of Human Behavior," arXiv:2304.03442, 2023, <https://arxiv.org/abs/2304.03442>. The generative-agent architecture —



**2. Distributed Coordination Protocols.** The distributed-resilience principle in the Skipjack Protocol specifies that edge systems must maintain coherent operation when coordination infrastructure is unavailable but does not specify what protocols enable that coherence when multiple nodes must coordinate without a shared communication channel. Federated learning addresses one instance — the model update case — but the general problem of coherent multi-node edge operation under contested or severed communication is open. The required protocol must maintain decision-consistent behavior across nodes operating in information isolation, must have bounded state divergence across the isolation interval, and must execute the coordination-recovery protocol of §5 correctly when connectivity is restored. This is a distributed systems problem with AI-governance constraints layered on top of the technical challenge, and no current protocol is designed for both simultaneously.

**3. Deterministic AI.** The safety argument for the autonomy envelopes defined in C4 and enforced by the Skipjack Protocol's bounded-autonomy principle requires that the behavior of the inference engine be predictable within a specifiable confidence interval across the full range of inputs the deployment context will present. Current neural inference is not deterministic in the sense required for a formal safety argument: the same input may produce different outputs depending on hardware state, floating-point rounding, and execution context, and the distributional properties of the output space across novel inputs are not characterizable a priori. The real-time scheduling theory that underlies formal safety arguments for embedded systems offers a foundational framework,<sup>59</sup> and the systems safety engineering tradition provides the broader methodology,<sup>60</sup> but the formal safety certification standards for functional safety — particularly those in the IEC 61508 family — do not yet have AI-specific annexes that address stochastic inference as a safety function.<sup>61</sup> Developing the formal methods and certification frameworks that would allow an edge AI inference engine to bear a safety argument is among the highest-value research vectors the doctrine identifies.

---

in which agents maintain persistent memory and simulate future behavior through planning — represents one approach to adaptive context at the agent level, though its compute requirements at edge-native resource levels are unsolved.

59 C. L. Liu and James W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," *Journal of the ACM* 20, no. 1 (January 1973): 46–61, <https://doi.org/10.1145/321738.321743>. The rate-monotonic scheduling theory developed here — and the formal analysis of worst-case execution time as the basis for real-time safety arguments — is the foundational framework for deterministic execution guarantees in safety-critical embedded systems.

60 Nancy G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety* (Cambridge, MA: MIT Press, 2011). The STAMP (Systems-Theoretic Accident Model and Processes) framework developed there provides the most complete methodology for applying systems-safety engineering principles to complex sociotechnical systems — the class of system that an edge AI deployment, including its governance and human authority structure, constitutes.

61 International Electrotechnical Commission. *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems — Part 1: General Requirements*. IEC 61508-1:2010. Geneva: IEC, 2010. The IEC 61508 family defines the safety integrity levels (SIL) and development lifecycle requirements for safety-related systems; the absence of AI-specific annexes addressing stochastic inference as a safety function is the gap this research vector identifies.

**4. Data Classification Persistence Across Lossy Channels.** The persistent-data-governance principle requires that classification metadata travel with data through every layer of the architecture without degradation. Current implementations depend on data pipeline components that can carry structured metadata fields; when the transmission channel is lossy, bandwidth-constrained, or implemented in a legacy format, metadata may be stripped, truncated, or dropped. The research question is what formal data models support classification persistence through constrained and lossy transmission environments — models that allow a receiving system to verify the classification integrity of a received data element even when the transmission channel cannot guarantee delivery of the full metadata payload. This is a data modeling and cryptographic problem with direct operational significance for the C2 requirement.

**5. Formal Specification of Autonomy Envelopes.** The bounded-autonomy principle requires that autonomy envelopes be specified in machine-interpretable terms precise enough to enforce at execution time. Current practice produces envelopes expressed in natural language — rules of engagement, operational orders, clinical protocols — whose interpretation at execution time requires judgment that the enforcement mechanism cannot reliably provide. The question of how to produce formal specifications of autonomy envelopes that are expressive enough to capture the operational intent of a human authority document and precise enough to enforce without interpretive ambiguity is a specification and formal methods problem. The grounding problem in linguistics — the question of how formal representations acquire semantic content — is a related open question whose resolution is a prerequisite for autonomy specifications that can be verified against operational reality rather than only against each other.<sup>62</sup>

**6. Edge-Native Adversarial Robustness.** The security collapse pattern documented in §3 addresses physical access as the primary attack vector at L0. The L3 inference attack surface — adversarial inputs designed to cause misclassification, confidence spoofing, or targeted output manipulation — represents a distinct and less-addressed vector for edge deployments. Adversarial robustness at cloud-scale inference has a substantial literature.<sup>63</sup> At edge-native resource levels — where the compute budget available for detection and defense is severely constrained by the same C1 and C10 requirements that govern inference — the trade-off between inference accuracy and adversarial robustness has not been characterized for the operational contexts this paper addresses. The ISR platform's target classification pipeline and the clinical inference pipeline at the rural medical imaging node have different adversarial threat profiles, different detection budget constraints, and different consequence models for robustness failure; each requires its own formal threat model and countermeasure architecture.

**7. Energy-Aware AI Architecture.** The energy-aware operation requirement of C10 and the constraint-first design principle of the Skipjack Protocol establish that inference workloads must be designed within their energy envelope, not optimized afterward. The model architectures and inference schedulers available to edge AI designers were developed for environments where energy was plentiful and the optimization target was accuracy or latency. Architectures designed from the ground up for energy-envelope operation — where the

---

62 Emily M. Bender and Alexander Koller, "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. Association for Computational Linguistics, 2020, <https://doi.org/10.18653/v1/2020.acl-main.463>.

63 Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and Harnessing Adversarial Examples," arXiv:1412.6572, 2014, <https://arxiv.org/abs/1412.6572>. The foundational analysis of adversarial input construction and the characterization of the gap between model confidence and adversarial robustness.

inference output rate, confidence level, and model complexity are dynamically traded against the available energy state — represent an emerging discipline.<sup>64</sup> The formal question of how to specify, validate, and certify an energy-aware inference architecture against a mission-critical operational requirement is open; the doctrine's C10 and Skipjack constraint-first principle describe the requirement but not the implementation.

**8. Compressed Observability Architecture.** The compressed observability requirement of C9 establishes that the observability infrastructure must be designed for the constraint — minimum viable signal, maximum governance visibility, deterministic export under degradation. Current observability architectures were designed for the inverse problem: maximum signal collection, with compression and sampling applied post-hoc as a bandwidth accommodation. The research question is what minimum-sufficient observability architectures allow a human governance authority to maintain the signal set required for functional governance — distinguishing operational health from apparent health, detecting drift, confirming envelope compliance — under the bandwidth and compute constraints that prevail at the edge. The explainability literature offers partial frameworks for identifying which model outputs carry maximum governance-relevant information, but the end-to-end design of a constrained governance-observability pipeline remains open.

**9. Formal Lifecycle Governance Methods.** The lifecycle management requirement of C7 establishes that model drift must be detected, recalibration must be triggered, and retirement must be executed according to governance-defined criteria. The formal methods for specifying drift detection thresholds, recalibration triggers, and retirement criteria under edge operating conditions — where the reference dataset for calibration comparison may be stale, bandwidth-constrained, or privacy-restricted — are not yet mature.<sup>65</sup> The dataset shift literature provides the theoretical substrate, but the formal methods for translating drift characterization into machine-enforceable governance criteria, and for maintaining those criteria under the edge constraints that make comprehensive monitoring unavailable, remain open research problems.

**10. Mission-Adaptive System Architecture.** The mission-adaptive reconfiguration requirement of C8 and the execution model of the Skipjack Protocol together specify that mission respecification must propagate correctly across all seven layers without full redeployment. The architectural patterns that allow L6 mission-layer authority changes to propagate through L5 orchestration configuration, L4 action authorization boundaries, L3 inference validation criteria, L2 data classification schema, and L1-L0 resource allocation in a consistent, auditable, human-authorized sequence are not established in current edge AI architecture practice. The question of how to design a mission-adaptive system architecture that enforces governance integrity across every respecification event is the synthesis problem that unifies the ten considerations this paper develops.

---

<sup>64</sup> Pete Warden and Daniel Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers* (Sebastopol, CA: O'Reilly Media, 2019). The TinyML framework represents the current state of practice for energy-constrained inference architecture at the ultra-low-power end of the edge compute spectrum; the formal certification of TinyML-class architectures for mission-critical applications is the open question.

<sup>65</sup> Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, eds., *Dataset Shift in Machine Learning*. Cited above, §2, fn. 28.

## 7. CONCLUSION

The failures documented in this paper are not mysteries. They are the predictable outcomes of deploying AI systems in constrained, distributed, and physically exposed environments without an architectural doctrine that accounts for those conditions. The ISR platform that thermally throttles because its architecture was sized to a lab benchmark and not to the desert it operates in. The rural medical imaging node that presents confident clinical inference on a patient presentation its model was never calibrated for. The power substation controller that cannot respond to the fault it was deployed to detect because the fault severed the coordination path its response depends on. These are not edge cases, and they are not engineering failures. They are architectural failures — failures that were determined before deployment, by design decisions that did not account for the constraints the deployment environment would impose.

This paper has developed the doctrine that those failures require.<sup>66</sup> The seven-layer architectural framework — Physical, Compute, Data, AI, Applications, Orchestration, Mission — is the spine against which every constraint, every failure mode, and every governance requirement can be located. The ten critical considerations are the conditions the architecture must satisfy to remain coherent at the edge; each is a requirement that, if unaddressed, produces the failure patterns §3 documents. The five cross-layer failure patterns are the signatures of architectural consequence chains that the doctrine is designed to prevent. And the Skipjack Protocol is the operational doctrine that applies all of it — before deployment, across all seven layers, against all ten considerations, under all operating conditions including those in which the ideal conditions the architecture might prefer are not available.

Paper One established the governance architecture: the HGC<sup>3</sup>AE<sup>2</sup> framework, with its human-governed authority structure, curated context and constraints, and agentic execution model.<sup>67</sup> Paper Two established the epistemic substrate: the mechanism by which statistical representation strips the domain-specific validation constraints that authoritative inference requires, generating thin context as a structural output, and the failure — confident misalignment — that thin context produces when the architecture has no mechanism to detect or contain it.<sup>68</sup> Paper Three is the execution-and-constraint paper: it specifies how architecture, environment, data, autonomy, and governance must be composed so that the confident misalignment Paper One identifies, and the thin-context condition Paper Two analyzes, do not become the operational consequence that both papers predict.

The thesis has not changed from the statement in the Abstract, and the conclusion does not modify it. Mission defines the architecture. Environment constrains the architecture. Data governs the architecture. Autonomy executes the mission. Human governance defines the boundaries of all four. The doctrine is the specification of how each of these relationships operates, across each layer, under each consideration, in the failure modes that arise when the relationships are not enforced, and in the operational protocol that enforces them.

The doctrine is not a checklist. It is not a compliance framework. It is a body of practice derived from the observation that edge AI systems fail in predictable ways, that those ways are

---

<sup>66</sup> Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework.” Cited above, Abstract, fn. 3.

<sup>67</sup> Justin H. Kuiper, CISSP, “Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework.” Cited above, Abstract, fn. 3.

<sup>68</sup> Justin H. Kuiper, CISSP, “Epistemic Constraints and Semantic Compression in Natural Language Processing.” Cited above, Abstract, fn. 4.

architectural, and that the architecture can be composed to prevent them. An architect who has characterized the system at L3 without having characterized it at L0 has not produced an architecture — she has produced a partial specification that will fail at runtime. A system that operates without a defined autonomy envelope has not been governed — it has been deployed with an open authority boundary that operational pressure will push outward at the worst possible moment. A governance structure that exists on paper but is not operationally enforced is not governance — it is the appearance of oversight that allows the failure it was meant to prevent to proceed without detection.

Edge AI failures are architectural — not technical.

---

---

## BIBLIOGRAPHY

Amershi, Saleema, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. “Guidelines for Human-AI Interaction.” In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York: ACM, 2019. <https://doi.org/10.1145/3290605.3300233>.

Amodei, Dario, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. “Concrete Problems in AI Safety.” arXiv:1606.06565, 2016. <https://arxiv.org/abs/1606.06565>.

Arrieta, Alejandro Barredo, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI.” *Information Fusion* 58 (2020): 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>.

Avizienis, Algirdas, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. “Basic Concepts and Taxonomy of Dependable and Secure Computing.” *IEEE Transactions on Dependable and Secure Computing* 1, no. 1 (January–March 2004): 11–33. <https://doi.org/10.1109/TDSC.2004.2>.

Bender, Emily M., and Alexander Koller. “Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. Association for Computational Linguistics, 2020. <https://doi.org/10.18653/v1/2020.acl-main.463>.

Canziani, Alfredo, Adam Paszke, and Eugenio Culurciello. “An Analysis of Deep Neural Network Models for Practical Applications.” arXiv:1605.07678, 2016. <https://arxiv.org/abs/1605.07678>.

European Parliament and Council of the European Union. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)*. OJ L 2024/1689. June 2024.

Floridi, Luciano, Josh Cows, Monica Beltrametti, Raja Chatila, Patrice Coquillard, Virginia Dignum, Christoph Luetge, Robert Madelin, Ugo Pagallo, Francesca Rossi, Burkhard Schafer, Peggy Valcke, and Effy Vayena. “An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations.” *Minds and Machines* 28 (2018): 689–707. <https://doi.org/10.1007/s11023-018-9482-5>.



Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples." arXiv:1412.6572, 2014. <https://arxiv.org/abs/1412.6572>.

Hoff, Kevin A., and Masooda Bashir. "Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust." *Human Factors* 57, no. 3 (2015): 407–434. <https://doi.org/10.1177/0018720814547570>.

International Electrotechnical Commission. *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems — Part 1: General Requirements*. IEC 61508-1:2010. Geneva: IEC, 2010.

Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. "Survey of Hallucination in Natural Language Generation." *ACM Computing Surveys* 55, no. 12, Article 248 (March 2023): 1–38. <https://doi.org/10.1145/3571730>.

Kuiper, Justin H., CISSP. "Epistemic Constraints and Semantic Compression in Natural Language Processing: A Theoretical Foundation for the HGC<sup>3</sup>AE<sup>2</sup> Framework." Non Sequitur Publishing, 2026. v1.0-preprint. <https://nonsequitur.tech/nsq-pub/white-papers/epistemic-constraints/>.

Kuiper, Justin H., CISSP. "Mitigating Confident Misalignment in Agentic Systems: The HGC<sup>3</sup>AE<sup>2</sup> Framework." Non Sequitur Publishing, 2026. v1.0-preprint, SHA 6e0b127f. <https://nonsequitur.tech/nsq-pub/white-papers/hgc3ae2/>.

Leveson, Nancy G. *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA: MIT Press, 2011.

Liu, C. L., and James W. Layland. "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment." *Journal of the ACM* 20, no. 1 (January 1973): 46–61. <https://doi.org/10.1145/321738.321743>.

McMahan, H. Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data." In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, 1273–1282. Proceedings of Machine Learning Research, vol. 54. PMLR, 2017. <https://proceedings.mlr.press/v54/mcmahan17a.html>.

National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. Gaithersburg, MD: National Institute of Standards and Technology, January 2023. <https://doi.org/10.6028/NIST.AI.100-1>.

Obermeyer, Ziad, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. "Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations." *Science* 366, no. 6464 (October 25, 2019): 447–453. <https://doi.org/10.1126/science.aax2342>.

Organisation for Economic Co-operation and Development. *OECD Principles on Artificial Intelligence*. OECD/LEGAL/0449. Paris: OECD, May 22, 2019. <https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449>.

Parasuraman, Raja, and Victor Riley. "Humans and Automation: Use, Misuse, Disuse, Abuse." *Human Factors* 39, no. 2 (1997): 230–253. <https://doi.org/10.1518/001872097778543886>.

Park, Joon Sung, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. "Generative Agents: Interactive Simulacra of Human Behavior." arXiv:2304.03442, 2023. <https://arxiv.org/abs/2304.03442>.



Prunkl, Carina. "Institutionalising Human Oversight of AI: Some Lessons from Aviation." *Minds and Machines* 34 (2024). <https://doi.org/10.1007/s11023-024-09667-z>.

Quiñonero-Candela, Joaquin, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, eds. *Dataset Shift in Machine Learning*. Cambridge, MA: MIT Press, 2009.

Roman, Rodrigo, Javier Lopez, and Masahiro Mambo. "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges for Current Scenarios and Future Prospects." *Future Generation Computer Systems* 78 (2018): 680–698. <https://doi.org/10.1016/j.future.2016.11.009>.

Russell, Stuart. *Human Compatible: Artificial Intelligence and the Problem of Control*. New York: Viking, 2019.

Satyanarayanan, Mahadev. "The Emergence of Edge Computing." *Computer* 50, no. 1 (January 2017): 30–39. <https://doi.org/10.1109/MC.2017.9>.

Shi, Weisong, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. "Edge Computing: Vision and Challenges." *IEEE Internet of Things Journal* 3, no. 5 (2016): 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>.

Shneiderman, Ben. *Human-Centered AI*. Oxford: Oxford University Press, 2022.

Topol, Eric J. "High-performance medicine: the convergence of human and artificial intelligence." *Nature Medicine* 25 (2019): 44–56. <https://doi.org/10.1038/s41591-018-0300-7>.

U.S. Department of Defense. "DoD Adopts Ethical Principles for Artificial Intelligence." February 24, 2020. <https://www.defense.gov/News/Releases/Release/Article/2091996/>.

Warden, Pete, and Daniel Situnayake. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA: O'Reilly Media, 2019.