

Constitutional Architecture for Intelligent Systems

Jurisdiction, Standing, and Lawful Action

Working Draft v0.9 — Initial Public Release — 27 August 2026
Defensive Publication / Prior Art

Grey Clark
Founder, Grey Dove
greydove.ai · San Francisco

Abstract

Behavioural alignment can shape how AI models reason and respond, but governing intelligent systems requires attention to the larger structures within which model reasoning is interpreted, may acquire institutional weight, and may be acted upon.

The problem becomes more acute when model outputs move beyond explanation or analysis and enter processes that classify, recommend, approve, refer, suspend, or otherwise shape institutional decisions. The concern extends beyond whether the model has reasoned well, to what authority its output carries, what procedures govern its use, and whether it may lawfully influence what the institution does next.

This paper introduces **constitutional architecture** as a system-level governance category for intelligent systems. Constitutional architecture constitutes governance as a distinct architectural layer, separating the governing frame from model behaviour and placing reasoning and action within a structure that defines the scope of authority, the conditions under which acts may lawfully occur, and the procedures through which the system preserves and changes its own governing structure.

Five properties define the architecture: **structural invariants**, **authority partitioning**, **conflict-resolution hierarchies**, **drift detection**, and **amendment protocols**. Together they determine how authority is bounded, how disagreement is governed, how divergence from the authorised structure is detected, and how that structure may change without losing coherence.

At the centre of the framework is **procedural constitutionalism**: the architecture governs the conditions under which reasoning and action proceed, while leaving substantive judgment open. This distinction becomes especially important where a request seeks a constitutional act — an exercise of authority rather than an analysis alone. The architecture must then determine whether that act may lawfully proceed, and require the constitutional posture of any operative output to remain legible, auditable, and contestable.

Constitutional architecture is model-agnostic and implementation-independent. Its contribution is to treat the boundary between probabilistic reasoning and authorised institutional action as a problem of system architecture rather than model behaviour alone. It defines the governing conditions under which authority may be exercised and standing may change, while leaving the independent enforcement of those conditions as an open engineering problem.

Table of Contents

1. Introduction	5
2. The Scope of Behavioural Alignment.....	6
3. The Nature of a Constitution	6
4. On Governance as Architecture	7
5. The Constitutional Order	8
5.1 Jurisdiction	8
5.2 Constitutional Coordinates	8
5.3 Constitutional Primitives	9
5.4 Lawful and Unlawful Action	10
5.5 Formal Representation of Constitutional Position.....	10
5.6 Constitutional Discretion	11
6. The Five Properties of a Constitutional Architecture.....	12
6.1 Structural Invariants.....	12
6.2 Authority Partitioning.....	14
6.3 Conflict-Resolution Hierarchies	16
6.4 Drift Detection.....	17
6.5 Amendment Protocols	19
7. Procedural Constitutionalism	22
8. The Procedural Arc of an Intended Act.....	23
8.1 Identifying the Intended Act.....	23
8.2 Admissibility, Completion, and Lawful Redirection.....	23
8.3 Constitutional Opacity and Structural Certification.....	24
8.4 Formal Representation of a Lawful Act.....	25
9. Model Agnosticism	26
10. Constitutional Architecture in Practice	27
11. Adjacent Work.....	28
12. Open Questions	31
12.1 The Reliability of Constitutional Discretion	31

<i>12.2 Unlawful Completion</i>	32
<i>12.3 Measuring Constitutional Effectiveness</i>	32
<i>12.4 Constraint Integrity and Structural Enforcement</i>	33
<i>12.5 Model Transition and Constitutional Continuity</i>	33
13. Conclusion	34
References	35
Appendix I — Claim Framing: Constitutional Architecture as a Distinct System Layer	38
<i>1. Architectural Category Definition</i>	38
<i>2. Distinction from Prior Approaches</i>	38
<i>3. Implementation Independence</i>	38
<i>4. Non-Equivalence of Partial Implementations</i>	38
<i>5. Scope of Claim</i>	38
About the Author	39

1. Introduction

Artificial intelligence is shifting from general-purpose capability into the operating logic of organisations and public institutions. Its technology is embedded in processes that classify, recommend, route, escalate, and act — in healthcare, finance, public administration, law enforcement, and other domains. The systems making these decisions are composite, multi-component, and largely ungoverned by design.

The EU AI Act, in force since August 2024 and amended in July 2026, applies progressively. Transparency obligations began applying from 2 August 2026, while the main requirements for high-risk systems now phase in from December 2027 for Annex III systems and August 2028 for systems embedded in regulated products. In the United States, AI governance continues to develop through federal executive action and proposals for a national legislative framework alongside state-level laws. The United Kingdom continues to rely primarily on existing legal duties and sector-specific regulators rather than a single comprehensive AI statute.

The trajectory is clear. Regulated AI systems are expected to demonstrate risk management, human oversight, transparency, procedural control, and decision records sufficient for review (audits). What the regulatory frameworks do not specify is an architecture for producing those properties inside composite AI systems.

Multi-component AI systems can produce incompatible conclusions even when each component behaves as trained. One subsystem classifies a request as permissible. Another flags it as high-risk. A third escalates it. The problem is rarely that a model has failed; the system has no formal way to determine authority, precedence, or resolution.

These issues are usually resolved by operational convention rather than formal structure. Outputs may be ranked, filtered, escalated, or reviewed by an operator. While this may often produce the right result, at scale, under pressure, and in regulated environments, *often* is not a basis for accountability.

Alignment methods — RLHF, Constitutional AI [1], principle-guided training — can shape the output behaviour of individual models. But the gap persists at a higher order: where components interact, where authority must hold across decisions, and where the structure itself must change without losing coherence.

The missing layer is the governance of authority, procedure, and change inside the system itself. This paper introduces **constitutional architecture** as a system-level approach to the gap. Models operate within its structure as reasoning components; the architecture governs.

The architecture is constitutional in a very specific way: it is organised around a constitution of higher-order governing rules that constitute authority, bind ordinary operation, and change only through amendment.

At the heart of the framework is procedural constitutionalism: procedure is governed at the level of architecture, while substantive reasoning remains open within it. The architecture establishes the conditions under which reasoning and action may proceed without predetermining their substantive outcomes.

The purpose of this paper is to define constitutional architecture as a distinct system-level governance category and to establish the structural properties through which authority, procedure, and lawful action may be governed within intelligent systems.

2. The Scope of Behavioural Alignment

Behavioural alignment shapes the conduct of a reasoning component. It affects how a model responds to an input, how it follows a policy, how it handles a refusal boundary, and how it expresses uncertainty. Constitutional architecture complements alignment but operates at a different level. It governs the interaction between components regardless of how each has been trained.

Alignment does not determine the authority of a response within a larger system. That distinction matters once model outputs cease to be final answers. In composite systems, outputs often become intermediate artefacts: a classification passed to another component, a risk signal used for escalation, a recommendation queued for human review, or a draft action withheld pending confirmation.

At that point, the governance problem is no longer only whether a model behaved appropriately. It is whether the system can determine what authority an output carries, what constitutional effect may lawfully follow from it, and how those distinctions are preserved as the output passes through the system and becomes the basis for further reasoning or action.

3. The Nature of a Constitution

A constitution is a deliberate design of authority, a self-conscious attempt to author governance. Where simpler forms of rule depend on the people who exercise power, a constitution is designed to outlast them. It distributes power across institutions and binds that power to procedure. It articulates rights and limits the reach of those who govern. It establishes precedence when there is conflict, and provides rules for its own amendment.

Its deepest function is not to police individual decisions, but to shape how authority flows through the system over time.

Every constitution carries two commitments. The most visible is substantive: rights, liberties, protections against the abuse of power. The most durable is procedural: how authority is distributed, how conflict is resolved, how the system changes without losing coherence. Both

matter, and both transfer to intelligent systems. The substantive layer transfers most directly to model behaviour: what a model should or should not say, do, endorse, or refuse. The procedural layer transfers to system architecture: how authority is allocated, how reasoning and action are governed, how conflicts are resolved, and how the governing frame operates and changes.

Procedure can be seen clearly in the due process of a court. It does not determine what a court concludes, but specifies how the court must act. The judgment remains open; the procedure is binding. That separation — between the substance of a decision and the process that governs it — makes decisions legible, contestable, and reviewable.

The same separation operates at the scale of an entire constitutional system. The United States Constitution distributes authority across branches, specifies how they interact, and defines formal procedures for legislation, adjudication, and amendment. Those constraints do not depend on who holds power. They persist across administrations, crises, and generations. The durability is not only in what the Constitution declares. It is in what the Constitution structures.

4. On Governance as Architecture

The central claim of this paper is architectural: governance must exist as a distinct structural layer of the system, not merely as behaviour-shaping techniques applied to its components.

Begin with a simpler machine. In a conventional process automation platform, the architecture is composite but not constitutional. A process engine orchestrates workflow according to rules defined by a system designer. A database preserves records. A web server manages interaction with users. APIs exchange information with external platforms. Each component performs a defined function. None reasons about its own authority. None interprets the purpose of the system against a competing internal view. The system may fail, but it does not generate internal claims about what should have authority over what.

Artificial intelligence systems are different. They may contain models, agents, retrieval systems, policy filters, planners, evaluators, tools, and external services. Some reason. Some interpret context.

Some generate classifications, recommendations, objections, or risk signals that alter the conditions under which later decisions are made. Once components can reason in this way, the problem is no longer just orchestration. It is a question of authority.

In a conventional system, components are functionally specialised but constitutionally inert. In a reasoning system, components may produce interpretations, exceptions, recommendations, and competing claims. Those claims require more than simple routing; they require a structure capable of determining authority, precedence, and resolution.

The dominant design pattern in many current AI deployments is model-centric. The LLM sits at the centre and the surrounding system serves it. Guardrails constrain it. Retrieval supplies it.

Tools extend it. Monitoring observes it. The architecture radiates outward from the model, a hub with many spokes.

Constitutional architecture reverses this arrangement. The model is not the governing authority of the system. Its reasoning operates within authority constituted by the architecture: the model produces conclusions within assigned domain and constitutionally vested authority, while acts require external authority recognised by the architecture.

In this design, governance is not applied after reasoning has occurred. Models, agents, tools, and evaluators are hosted within a governing structure that decomposes work, allocates authority, preserves boundaries, and governs the conditions under which reasoning and action occur. The architecture determines how reasoning may proceed, how conclusions may interact, and when an act may acquire constitutional effect.

Constitutional architecture therefore requires more than the functional organisation of system components. It requires a constituted space in which reasoning is situated, authority is ordered, and acts may have lawful consequence.

5. The Constitutional Order

The constituted space introduced above is the constitutional order of the architecture. Jurisdiction defines its extent; within it, domain, authority, and time locate reasoning and action, while matter, standing, and act define what becomes constitutionally significant when a request engages an identifiable matter whose standing may be altered. Constitutional discretion is the reasoning through which requests are interpreted and situated within the order.

5.1 Jurisdiction

Jurisdiction defines the constitutional scope of the architecture. It is not a property of a particular request or act, but is the defining extent of the constitutional order. Within jurisdictional scope the architecture determines which domains may be engaged, which authorities may reason, determine, or act within them, and under what conditions those activities carry constitutional force.

Where a request engages a matter, jurisdiction is expressed through the constitutional conditions that determine what may lawfully occur. Later sections distinguish these conditions as **availability** (whether the architecture recognises a lawful path from the matter's present standing) and **admissibility** (whether the requirements governing that path have been satisfied).

5.2 Constitutional Coordinates

Constitutional coordinates locate governed reasoning and action within jurisdiction: domain, authority, and time.

Domain is the field within which reasoning, a determination, or an act applies. A request may engage one or several domains, for example legal, technical, financial, or operational.

Authority is the constituted capacity, recognised by the architecture, to reason, make a determination, or perform an act within a domain. It is always relative: authority in relation to which subject or matter, within which domain, and for which determination or act. Authority to analyse or assess without constitutional effect differs from authority to perform an act that changes a matter's standing; the two are constituted separately.

Time identifies when the evidence, authority, determinations, dependencies, and governing conditions relevant to reasoning or action hold. These may change or expire. Where a matter is engaged, time is therefore part of its constitutional position, not merely an audit record.

5.3 Constitutional Primitives

Where the architecture recognises a persistent object to which constitutional effects may attach, three constitutional primitives become operative: **matter**, **standing**, and **act**.

Matter is the object under governance: the single identifiable subject to which requests, evidence, constraints, determinations, acts, and effects attach. A subject of reasoning is not necessarily a matter. A request may concern a question, topic, or body of information without requiring the architecture to recognise a persistent object whose standing may be recognised or changed. A request may introduce, address, or seek action upon a matter, but it is not the matter itself. The matter persists as its standing changes and as different domains and authorities engage with it.

Standing is the constitutionally recognised condition of a matter: what the matter presently counts as, what force that recognition carries, and which authorised acts may validly alter it. Standing belongs to the matter and is represented through a profile comprising the recognised conditions of its **facets** across one or more domains. A domain may contain multiple **facets of standing**. A matter may therefore be technically accepted, legally unresolved, financially authorised, and operationally blocked without becoming several separate matters. Standing is not process state; a process state records where work sits. Standing carries constitutional consequence.

An act is an exercise of authority directed toward a constitutional effect upon a matter. It is performed through a constituted role by an actor validly entitled to occupy it, and takes effect only when the conditions governing it have been satisfied. Where it does, the architecture recognises the resulting change in the matter's standing. No act occurs without that exercise of constitutional authority: explanation, analysis, drafting, and recommendation do not perform one, and language resembling an approval, refusal, or determination does not constitute one.

An act changes a facet of standing within a domain. Each facet subject to constitutional change is represented by its own **standing-transition graph**: the recognised conditions of the facet are its nodes, the authorised acts through which those conditions may change are its edges, and the

conditions attached to each edge are the requirements for that act to proceed lawfully. Because a domain may contain multiple facets, it may also contain multiple standing-transition graphs. A matter may participate in several of these graphs at the same time; the graphs may operate independently or be constitutionally coupled while remaining bound to the same persistent matter.

The standing-transition graph is representational. It models the lawful changes available to a facet of standing; the governing frame is what constitutes the standing, authority, acts, and conditions that make those changes lawful.

5.4 Lawful and Unlawful Action

The primitives establish not only how standing may change, but also the boundary between lawful and unlawful action. Lawfulness is not determined by whether an intended act completes. An intended act whose governing conditions are not satisfied cannot lawfully complete, but this does not constitute unlawful action. The matter remains within the governing frame and may be lawfully redirected. Depending on how the architecture is constituted, that redirection may preserve the matter's present standing or produce a different authorised change in standing. The procedural consequences of non-completion are developed in Section 8.2.

Unlawful action is different. It occurs where a model, actor, office, or function exceeds or bypasses the governing frame and asserts or produces a constitutional effect that the architecture has not authorised. This may include acting without the required authority, bypassing a governing condition or required review, asserting a standing change for which no lawful transition exists, or allowing reasoning or another output to acquire the force of an act. Unlawful action therefore represents a failure of constitutional containment and an attack surface of the architecture.

5.5 Formal Representation of Constitutional Position

Where a matter is engaged, the primitives and coordinates together describe its constitutional position:

$$(matter, domain, authority, standing, time)$$

The constitutional position of a matter can be represented formally as a tuple:

$$p = (m, \mathbf{D}, \mathbf{A}, \mathbf{S}, t)$$

where m is the matter, $\mathbf{D}$ the domain or domains presently engaged, $\mathbf{A}$ the authority structure relevant to them, $\mathbf{S}$ the matter's standing profile, and t the time at which that position holds. The notation is representational rather than ontological: it provides a compact expression of the relationships established in the preceding sections.

The components $\mathbf{D}$, $\mathbf{A}$, and $\mathbf{S}$ may themselves be composite. A matter may engage several domains at once:

$$\mathbf{D} = d_1, d_2, \dots, d_n$$

The standing profile $\mathbf{S}$ may itself be composite: facets of standing may be expressed across domains. There is no assumed one-to-one correspondence between domains, facets of standing, and standing-transition graphs.

The internal structure of authority is developed in Section 6.2.

Let $\mathcal{M}$, $\mathcal{D}$, $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{T}$ denote the sets of possible values or structures associated with matter, domain, authority, standing, and time. Constitutional positions may then be represented within their Cartesian product:

$$\mathcal{M} \times \mathcal{D} \times \mathcal{A} \times \mathcal{S} \times \mathcal{T}$$

The architecture does not recognise every arbitrary combination as a constitutional position. For a jurisdiction J , the positions recognised within its constitutional order form a constrained subset:

$$\mathcal{P}_J \subseteq \mathcal{M} \times \mathcal{D} \times \mathcal{A} \times \mathcal{S} \times \mathcal{T}$$

Jurisdiction is not another coordinate of the matter, but defines the constitutional scope within which positions may be recognised and constitutional effects may have force. Structural invariants further determine which combinations of domain, authority, standing, and other governing conditions are valid within that scope.

An *act* is not an element of the constitutional position but operates upon it. The formal representation of its operation is developed in Section 8.4.

5.6 Constitutional Discretion

Constitutional discretion is the exercise of model reasoning in service of the architecture's governing frame, rather than the domain-governed reasoning applied to the substance of a particular request. It is a specific application of reasoning directed toward architectural requirements. Domain-governed reasoning addresses the request within one or more fields; constitutional discretion determines how the request is situated, decomposed, routed, and governed.

It may identify the relevant domains and authorities; decompose and route work among them; govern how their contributions are brought together in synthesis, which includes determining whether a conflict must be recognised and held; distinguish a request for straightforward analysis from one that seeks an intended act; and, where an intended act engages a matter, evaluate its availability and admissibility. The architecture defines the domains, authorities, lawful paths, and conditions that govern the request; constitutional discretion interprets the circumstances and applies that structure.

Constitutional discretion does not create authority, confer standing, or invent lawful paths, and it is not a separate constitutional property. Because it operates on the same probabilistic substrate as other model reasoning, it remains fallible. Its fallibility, and the question of how constitutional discretion may be bounded and tested, are addressed in Section 12.

6. The Five Properties of a Constitutional Architecture

Constitutional architecture gives effect to a constitution through a set of governing functions embedded in the architecture of the system. The constitution establishes higher-order rules that constitute authority, bind ordinary operation, and govern the terms of their own amendment. These functions may be realised differently across domains, institutions, and technologies, but together they give the architecture its constitutional form.

These functions appear as five structural properties: **structural invariants** define what must hold; **authority partitioning** structures authority; **conflict-resolution hierarchies** govern legitimate disagreement; **drift detection** identifies divergence from the authorised structure; **amendment protocols** govern how the frame itself may change.

The test of the architecture is not how it behaves under ideal conditions, but whether it holds under load. It must hold when an unauthorised act attempts to proceed, when authority is unresolved, when evidence is missing, when conflict appears, or when operating practice begins to drift. **Constraint integrity** is the governing structure holding under that pressure: refusing unauthorised acts, preserving authority boundaries, and routing unresolved matters through lawful paths. It is the condition the architecture's properties exist to preserve.

While these properties define the structure of the architecture, it is functionally inert without **constitutional discretion**, defined in Section 5.6; discretion is not a constitutional property but the application of reasoning through which the properties are recognised and applied. The structure defines what must hold; discretion is the judgement through which the system recognises whether it holds in a given case. Whether discretion can be bounded reliably enough is an open question taken up in Section 12.

6.1 Structural Invariants

Structural invariants are the constitutional laws of the architecture. They define the governed space within which reasoning may occur, establish the boundaries that distinguish one constitutional act from another, and determine the conditions under which a matter may lawfully change standing. They form the architecture's load-bearing structure, constituting the jurisdiction within which lawful reasoning, procedure, and action unfold.

Structural invariants perform two related functions: they establish the governed space and they delimit it.

They establish the governed space by defining the domains within which reasoning may occur, the authorities constituted or recognised within them, and the forms of reasoning, procedure, and review that apply. Where a request engages a matter, they also define the standings the architecture recognises, the acts that may lawfully alter them, the paths through which those acts may proceed, and the conditions attached to those paths.

Structural invariants define the limits of lawful operation by specifying what may not occur unless the governing conditions have been satisfied. A conclusion, recommendation, or determination does not acquire the force of a constitutional act unless the authority and conditions governing it are satisfied. An act may be performed only through a role or office vested with the relevant authority. A required review may not be bypassed. A matter may change standing only through an authorised act along a lawful path, and only when the conditions attached to that path have been met.

Training-based restrictions are probabilistic and diffuse. A model trained not to produce a prohibited response or attempt an unauthorised act may still do so under novel conditions, adversarial prompting, or distributional shift. The restriction itself may be encoded diffusely through learned behaviour rather than as a specific rule, making it difficult to isolate, audit, or amend. A structural invariant is different: it is an explicit constitutional requirement of the architecture, inspectable and amendable independently of the model's learned dispositions. The issue of enforcement is taken up in Section 12.4.

This distinguishes constitutional architecture from ordinary process orchestration. A process engine governs workflow by arranging work into predefined steps, decisions, and hand-offs. Constitutional architecture governs authority by defining the constrained space within which reasoning may operate freely, and the conditions under which an act may lawfully take effect.

The limits of orchestration become clearer in multi-agent systems. Orchestration coordinates which agent, model, tool, or service performs the work, with what capabilities, and in what sequence. The risk is that agents or downstream components treat another component's output as authoritative, even though no such authority has been constituted. Constitutional architecture resolves this by constituting or recognising the authority under which components may reason, determine, or act.

Illustrative Example

The example below is deliberately isomorphic to an ordinary workflow, using a matter to illustrate the architecture clearly. However, a matter need not follow a predefined process. It may persist across domains, authorities, and time, following a trajectory that is not process-shaped.

Consider an insurance claim involving water damage to a commercial property. The model may analyse the evidence, estimate the loss, and prepare a recommendation. During the analysis it identifies that the failed pipe was located in a section of the building under renovation by a third-party contractor. That detail engages questions of liability and subrogation and triggers a structural invariant requiring specialist review. Analysis may continue, but the intended act—approval of the claim for routine payment—cannot lawfully occur until that condition has been satisfied. The claim may therefore be technically assessed while remaining legally unresolved and financially unapproved.

The example is intended to clarify elements of the architecture:

- **Matter:** the insurance claim
- **Domains:** Claims Assessment, Legal, and Financial
- **Domain-governed reasoning:**
 - **Claims Assessment:** analysing the damage and estimating the loss
 - **Legal:** assessing contractor liability and subrogation rights
 - **Financial:** evaluating the claim for payment
- **Constitutional discretion:** recognising that a contractor detail changes how the claim must be situated, engages the Legal domain and determines that routine approval cannot proceed without specialist review
- **Structural invariant:** specialist legal review must occur before payment may be approved
- **Intended act:** approval of the claim for payment
- **Facets of standing:**
 - **Claims Assessment:** assessed
 - **Legal:** unresolved
 - **Financial:** unapproved

In practice, structural invariants shape much of the structure shown: the domains and authorities involved, the standings recognised, the acts permitted, the lawful paths available, and the conditions attached to them.

6.2 Authority Partitioning

Authority partitioning is one of the governing functions of the architecture. Defined by structural invariants, it orders authority within the architecture's jurisdiction and holds each authority within its proper scope. Within this order, the architecture distinguishes **interpretive authority**, which governs reasoning within domains, from **office authority**, which governs acts upon matters.

Interpretive authority governs reasoning within each domain. Constituted within the architecture by structural invariants, it is expressed through domain-specific expert scopes that shape how reasoning proceeds. A request may engage one or more such domains, with reasoning in each proceeding under its own interpretive authority.

Office authority governs acts upon matters. It is vested in offices constituted by the architecture and exercised by a human or model recognised as eligible to occupy the relevant office. The authority belongs to the office, not the officeholder, and is limited to the acts assigned to that office and the changes in standing those acts are permitted to produce.

The authority engaged by a request depends on what the architecture is being asked to do. Requests for reasoning are governed through domains and expert scopes under interpretive authority, whether they concern a general question, a body of information, or an identifiable matter. The presence of a matter does not alter this: the architecture may examine, compare,

assess, or otherwise reason about the matter without changing its standing. Office authority is engaged only where the request seeks an act upon the matter. It is the constituted authority under which that act may produce a constitutional effect and change the matter's standing. Reasoning within the relevant domains may provide the analysis, determinations, or recommendations on which the act is based, but it does not confer the authority to perform the act or itself change the matter's standing.

This distinction is reflected in how authority is ordered along two separate dimensions. Interpretive authority is partitioned by domain, while authority more broadly is partitioned by constitutional function: between authority to reason and authority to perform acts capable of changing standing. The resulting structure is asymmetric, because the domain partition operates within interpretive authority, while the interpretive-office distinction separates different constitutional functions.

Office authority is partitioned by act. The acts vested in an office are constituted by the architecture; which of those acts are available depends upon the present condition of the matter's facet of standing within the relevant domain. Different offices may therefore hold authority over different transitions from the same standing. Participation in a matter confers no broader authority over it. How the availability and admissibility of an intended act are determined is discussed in Section 8.

Formal Representation of Authority

Within jurisdiction J , let $\mathcal{D}_J$ denote the domains constituted within it. For each domain $d \in \mathcal{D}_J$, let a_d^I denote its interpretive authority and X_d the expert scopes constituted under that authority. The interpretive-authority structure may be represented as:

$$I_J = (d, a_d^I, X_d) \mid d \in \mathcal{D}_J$$

For a request r at time t , constitutional discretion may engage one or more domains and a subset of the expert scopes available within them:

$$D_r(t) \subseteq D_J, X_r(t) \subseteq \bigcup_{d \in D_r(t)} X_d$$

An expert scope is a local expression of interpretive authority within a domain. The resulting domain-governed reasonings may contribute to a synthesised response. Their engagement does not confer office authority or constitute an act; where the request concerns a matter, it does not by itself alter standing.

Let O_J denote the office-authority relation within jurisdiction. Where α_d is an act operating on a facet of standing f within domain d , then office authority may be represented as:

$$O_J = (o, \alpha_d, f) \mid d \in D_J, \text{ and office } o \text{ is vested with authority to perform } \alpha_d \text{ upon facet } f.$$

The authority vested in an office does not mean that every such act is available from every standing. For a facet f presently at standing s , let $Avail_{d,f}(s)$ denote the authorised acts represented by transitions available from that standing. The office-authority relations available at s may then be expressed as:

$$O_f^{avail}(s) = (o, \alpha_d, f) \in O_f \mid \alpha_d \in Avail_{d,f}(s)$$

Availability therefore depends both on the authority vested in the office and on the matter's present standing. Whether the governing conditions for an available act have been satisfied is the further question of admissibility developed in Section 8.

The authority structure within jurisdiction may therefore be expressed as:

$$A_J = (I_J, O_J)$$

The two components preserve the distinction between interpretive authority over reasoning and office authority over constitutional acts. They refine the internal structure of $\mathbf{A}$ introduced in Section 5.5 without adding another constitutional coordinate or prescribing an implementation. All relations are understood at the relevant time t , consistently with the temporal coordinate defined in Section 5.2.

By partitioning interpretive authority across domains, the architecture preserves distinct governed perspectives and brings their contributions together in synthesis. Synthesis does not constitute a further source of interpretive authority: it brings domain reasonings together while preserving the authority and scope under which each was produced. Where their relationship requires constitutional judgement, that judgement is exercised through constitutional discretion. Reasoning proceeds within each domain under its own interpretive authority. Where those reasonings cannot be reconciled without suppressing a legitimate domain perspective, the conflict is held and governed through the conflict-resolution hierarchy developed in Section 6.3.

6.3 Conflict-Resolution Hierarchies

Conflict-resolution hierarchies specify in advance how legitimate disagreement between the responses of domains and other reasoning subsystems will be handled.

If authority is distributed, conflict becomes inevitable. Complex systems will produce cases in which components reach incompatible conclusions. This is not unexpected: it is a property of systems whose components exercise judgment within their own scopes of authority. Legitimate conflict — the kind that needs a resolution hierarchy rather than just an error-handling protocol — arises specifically when components reach different conclusions, applying different lenses to the same situation.

Without governed resolution, the system could fall back on arbitrary selection, first-response bias, or perhaps opaque human override. None produces a consistent or auditable outcome.

A constitutional architecture structures the resolution.

Let's return to the contract review example. A commercial-analysis component concludes that the agreement is workable: the pricing structure is competitive, the obligations are manageable, and the commercial terms align with current strategy. A legal-risk component flags an indemnity carve-out that may expose the organisation to uncapped liability in a narrow but material class of events. A business-priority component notes that the counterparty has signalled time pressure and that delay risks losing the opportunity.

All three conclusions may be legitimate within their own scopes. The problem is that they point in different directions: the commercial view points toward proceeding; the legal-risk judgment directs a review; business-priority points toward a quick resolution.

A conflict-resolution hierarchy begins by classifying the kind of conflict. A factual disagreement — two components reaching different conclusions from the same data — is different from a priority clash — commercial speed vs legal caution — which is different from an authority-boundary question: does one component's concern have the procedural weight to suspend another's recommendation? Each requires a different kind of assessment and resolution.

The hierarchy specifies how each kind of conflict is to be treated according to the resolution conditions constituted by the architecture. In some cases, one domain's interpretive authority may take precedence on the question in conflict: legal-risk may govern liability exposure, while commercial analysis governs pricing or market position. Any such precedence is bounded to the question falling within that domain's constituted scope, without elevating its interpretive authority beyond it.

Other conflicts may require a different response. The hierarchy could withhold a final recommendation pending further review, permit qualified continuation while preserving the disagreement, or refer the conflict or matter elsewhere in the architecture where another interpretive or office authority is required. Which response applies is governed by the hierarchy as constituted, not arbitrarily invented when the conflict arises.

And some conflicts should not be resolved at all. Where legitimate authorities reach conclusions that cannot be reconciled without suppressing one of them, the architecture surfaces the conflict rather than hiding it inside a settled response.

The hierarchy's purpose is not to force agreement; it is to prevent false consensus.

6.4 Drift Detection

Drift detection identifies when the system's actual operating pattern has begun to diverge from its authorised constitutional structure.

The previous three properties define the structure against which that divergence can be recognised. Structural invariants establish its governing conditions and boundaries; authority partitioning orders authority within them; conflict-resolution hierarchies govern legitimate

disagreement when it appears. Drift detection asks whether those structures are still operating as constituted — not at a single instance of use, but over the wear of repeated operation.

Drift rarely appears first as a dramatic constitutional violation. More often it manifests through erosion of use. A review stage that once involved substantive scrutiny becomes a formality — the step still occurs, but the review itself is perfunctory. A category that once triggered specialist attention narrows in practice — not by formal amendment, but because borderline cases are increasingly classified as routine. An exception granted for a specific circumstance becomes standard practice without anyone recognising that the operating structure has changed. Each departure may be locally reasonable. In aggregate, they can hollow out the governing architecture.

Not every departure, however, is drift. A rule may be violated while still being recognised as binding; even repeated violations may indicate a failure of compliance or reliability rather than a change in the constitution's effective operation. Nor does every changing operating pattern imply drift. More matters may be referred because the matters themselves have changed, while the same constitutional conditions continue to be applied. Drift begins where practice changes the relationship between the governing structure and its application: what counts as sufficient review, where an authority boundary is treated as lying, which conflicts must remain visible, or when an exception is understood to have become ordinary.

This distinguishes architectural drift from model drift. Model drift concerns change in the behaviour or performance of a reasoning component. Architectural drift concerns change in how the governing structure is interpreted or applied. The two can interact. A change in model behaviour may produce architectural drift if it alters the interpretation of an invariant, authority boundary, or other constitutional condition. But architectural drift can also occur without any change to the underlying model, as repeated decisions, user behaviour, exceptions, or institutional practices accumulate and begin to influence how later cases are treated. Conversely, model behaviour may change while the architecture continues to operate within its authorised boundaries.

The risk is especially acute where the architecture depends upon interpretation. Repeated practice can acquire the appearance of precedent. What usually happens begins to influence what is treated as permissible; an advisory recommendation that is routinely accepted may begin to carry the practical force of approval; conflicts that the hierarchy requires to remain visible may increasingly disappear into apparently settled responses. None of these changes requires the constitution itself to be rewritten. The operative shift occurs in the reasoning through which the constitution is applied.

Drift detection therefore requires more than monitoring the frequency of errors, exceptions, referrals, or other outcomes. It compares operating practice against an identifiable constitutional reference and asks whether the same distinctions, conditions, and authority relationships continue to govern. A changing pattern is evidence to examine, not proof of drift. The

constitutional question is whether the pattern reflects different circumstances being governed by the same structure, or a change in how that structure itself is being interpreted or exercised.

That comparison may itself require judgement. Constitutional discretion applies the governing structure to the observed pattern and may identify that a divergence has occurred; it does not acquire authority to redefine the structure by doing so. Because constitutional discretion is itself probabilistic, its reliability in recognising drift cannot simply be assumed. That problem forms part of the broader question of how constitutional discretion may be bounded and tested, taken up in Section 12.

Where drift is identified, the architecture must distinguish correction from constitutional change. A departure from authorised practice may require restoration of the governing condition, containment of an exception, or investigation of repeated violations. But operating experience may also reveal that the constitutional structure itself should change. In that case, accumulated practice cannot make the change legitimate merely by becoming customary.

Practice may supply the reason for amendment; only the amendment process can supply its authority.

6.5 Amendment Protocols

The first four constitutional properties of the architecture govern its operation. Amendment protocols govern how the structure can change in response to shifts in its operating environment, whether it's institutional, technological, legal, or cultural.

A constitutional architecture must be able to change through an authorized and explicit amendment process, while preventing ordinary operation from changing the governing structure through repeated use, informal agreement, or administrative reinterpretation. Operating conditions can change, categories lose relevance, new risks appear, and existing safeguards may prove either inadequate or unnecessarily restrictive. If the structure cannot adapt, pressure accumulates around it and workarounds become attractive. If ordinary practice can alter constitutional rules without formal authorisation, the distinction between valid amendment and unacknowledged drift disappears. Amendment provides a recognised means of change without allowing ordinary operation to become a source of constitutional authority.

Amendment becomes necessary when a proposed change could alter a condition established by the architecture. The question is not how extensive, technical, or operationally visible the change appears to be, but what it changes in the governing structure. A change is constitutional if it alters the scope of an invariant, the extent of jurisdiction, the location of an authority boundary, or the conditions under which a constitutional rule applies. Changing a directive or parameter will require an amendment if it changes cases that are subjected to a constitutional safeguard. Conversely, replacing a technical subsystem may remain just an operational update if it doesn't impact constitutional rules, authority boundaries, and other constraints. Changes to jurisdiction,

structural invariants, authority partitions, the treatment of conflict, or the conditions governing constitutional change require amendment because they alter the governing frame.

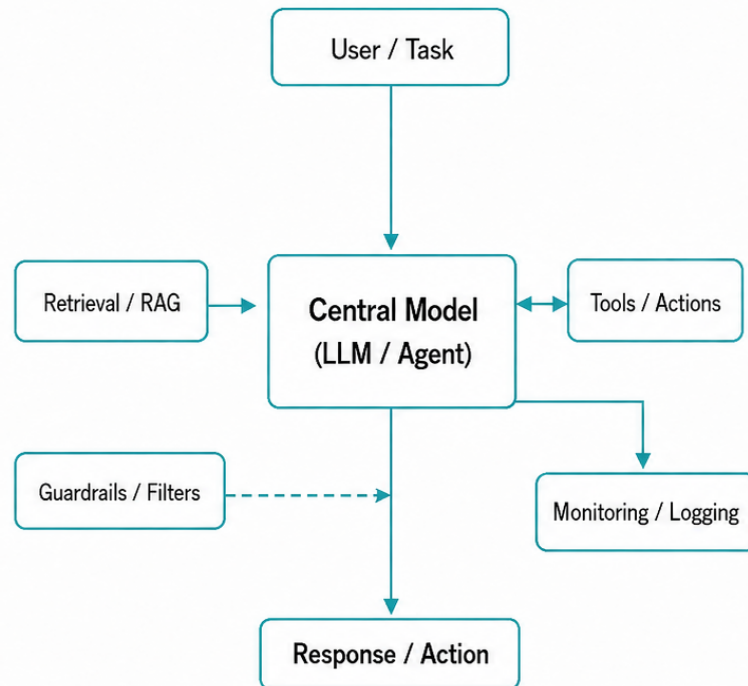
Drift detection also depends on amendment protocols to maintain a stable governing reference. Amendment makes authorised changes explicit and visible, preserving a clear baseline against which later practice can be assessed. Without it the architecture cannot reliably distinguish authorised evolution from gradual drift.

An amendment protocol does more than record that a change has been made: it identifies the constituted authority competent to adopt the change, the conditions under which that authority may act, and the constitutional protection, boundary, or relationship the proposed amendment would alter. Where an amendment is adopted, the architecture must also preserve its lineage: the prior rule, the change that was authorised, the reason for it, the authority under which it was made, and the point from which the amended structure governs. This lineage helps maintain constitutional integrity.

Finally, the amendment protocol is itself subject to the same principle. Its terms may be altered or replaced only through constitutionally authorised change; it cannot be bypassed by treating amendment as an ordinary administrative task. The constitution may change the rules by which it can be updated, but only through authority and procedure capable of giving that change constitutional effect.

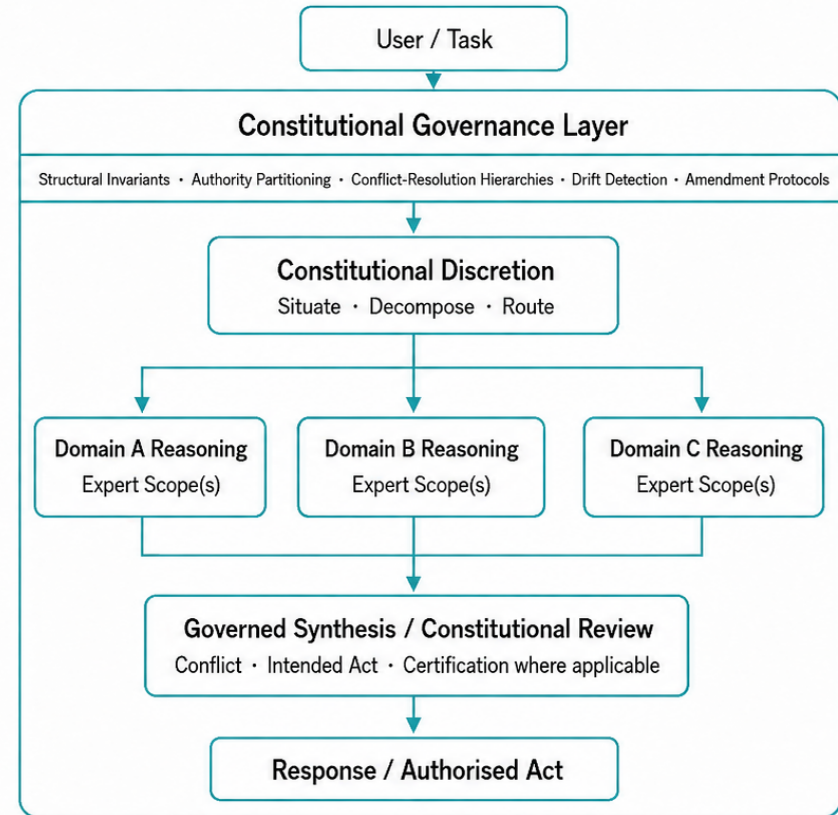
Amendment is not an exception to constitutional governance but one of its clearest expressions. A constitutional architecture does not resist change; it refuses ungoverned change.

A. Model-Centric Pattern



Retrieval, tools, guardrails, and monitoring are organised around a central model; the model remains the centre of the system.

B. Constitutional Architecture



Models and agents operate inside a governed structure.

Figure 1. Contrasting a Model-Centric Pattern and Constitutional Architecture

In model-centric AI systems, retrieval, tools, guardrails, and monitoring are organised around a central model. In constitutional architecture, governance is constituted as a distinct structural layer within which models and agents operate, and through which reasoning and authorised action are governed.

7. Procedural Constitutionalism

The five constitutional properties define the architecture's governing structure. Together they establish the conditions under which authority exists and is distributed, how legitimate conflict is governed, how divergence from the authorised structure is detected, and how constitutional change may occur. These properties give the governing frame its constitutional form; procedural constitutionalism concerns how that frame operates.

In constitutional systems, procedure gives form to the exercise of authority. It establishes who may decide, under what conditions, through which institutions, and with what opportunities for review, contest, and change. In intelligent systems, procedural constitutionalism performs the corresponding function at the architectural level: substantive judgment remains open, while the governing frame determines how reasoning and authority are organised and how the architecture operates and changes.

Procedural constitutionalism separates the substantive merit of reasoning from the constitutional operation of the architecture. Within a court, a compelling argument may influence judgment, but it does not determine which authority may decide, what procedure must be followed, how conflict is treated, or what institutional effect may result. Those questions are governed by the constitutional structure within which the reasoning occurs. The same distinction applies within a constitutional architecture. Reasoning may inform, interpret, compare, or recommend within its constituted domains, but substantive merit alone does not determine its own scope, authority, routing, treatment of conflict, or institutional effect. Those are governed by the architecture.

Procedural constitutionalism operates across the architecture. Constitutional discretion may interpret, decompose, situate, and route a request among constituted domains and authorities; synthesis may require conflict to be surfaced or review to occur; changes to the governing structure may invoke amendment. Where reasoning engages a persistent matter, procedural constitutionalism also governs the matter's constitutional trajectory: its standing, the lawful paths available from it, and the conditions under which an intended act may alter it.

In a probabilistic system, constitutional procedure cannot be exhaustively prescribed in advance. Structural invariants establish the conditions that must hold, while constitutional discretion interprets and applies the governing frame to particular circumstances. Substantive judgment can therefore remain open within constituted conditions, while the architecture governs its scope, routing, conflicts, and consequences. Where its operation engages a persistent matter and an intended act, procedural constitutionalism gives the matter's resulting trajectory its constitutional form.

8. The Procedural Arc of an Intended Act

Procedural constitutionalism takes a particular form when reasoning engages a persistent matter and a request may seek an act that would alter its standing. This section follows the procedural arc from identifying whether an act is intended through to the conditions governing whether and how that act may lawfully complete.

The governing frame comes under pressure whenever fulfilling a request would alter a matter's standing, and thereby change what may lawfully happen next.

Constraint integrity is the condition in which the governing frame holds under that pressure. It ensures that a matter changes standing only through an authorised act, and never because a request sought or implied one.

8.1 Identifying the Intended Act

When a request concerning a matter is received, the architecture must first determine whether the request seeks reasoning about the matter that leaves its standing unchanged, or an act that would alter its standing. Constitutional discretion determines this by interpreting what the request is asking of the architecture: what outcome is sought, irrespective of how the request is worded.

A request may seek explanation, comparison, assessment, drafting, or advice about a matter without seeking any change in its standing. A recommendation or classification may likewise be advisory, or it may be intended as an operative outcome that would change what the matter constitutionally counts as. If no act is intended, the response remains domain-governed reasoning. If the intended outcome is unclear, the architecture seeks clarification from the request's initiator.

Where an act is intended, structural invariants define the lawful paths available from the matter's present standing, relative to the relevant domain and authority. Constitutional discretion evaluates whether the intended act corresponds to one of them. If no such path exists, the architecture cannot invent one; the intended act is unavailable and the matter follows another lawful disposition recognised by the architecture. If a path exists, the intended act proceeds to admissibility.

8.2 Admissibility, Completion, and Lawful Redirection

Once an intended act has been matched to a lawful path, the architecture tests its admissibility. Availability establishes that the path exists from the matter's present standing, relative to the relevant domain and authority. Admissibility asks a different question: **have the conditions attached to that path been satisfied?**

Those conditions may concern the matter's standing, the evidence available, required review, or other prerequisites attached to the path. Structural invariants define the constraints;

constitutional discretion evaluates whether they have been met. Once constitutionally validated, the intended act may complete, changing the matter's standing as the lawful path permits.

An act completes only when performed by an eligible actor through a constituted role or office, and when the architecture recognises the resulting change in standing.

Where the intended act is unavailable, or is available but inadmissible, it cannot lawfully complete. Nevertheless the matter remains governed: it lawfully redirects to another disposition constituted by the architecture. Forms of lawful redirection may include, but not be limited to:

- **Return for further information** — returning the request to the initiator so that missing or insufficient information can be supplied and admissibility reconsidered.
- **Return for reformulation** — returning the request to its initiator where a differently framed or more limited act may have a lawful path.
- **Referral for required review** — placing the matter before a designated reviewer where independent review or approval is a structural condition of completion.
- **Jurisdictional referral** — referring the matter to an external authority or process where the architecture recognises that the matter falls outside its jurisdiction. The referral does not determine the matter or confer standing beyond the architecture's governed boundary.
- **Suspension** — holding the intended act at the matter's present standing while a blocking dependency persists, including an act or determination that must occur in another domain or outside the architecture's jurisdiction.
- **Lawful refusal** — declining the intended act where the conditions required for completion cannot be established.

Whether the intended act completes, the matter is lawfully redirected, or its standing remains unchanged, the resulting **constitutional posture** and any effect on standing must remain visible and inspectable. Section 8.3 addresses how that outcome is certified.

8.3 Constitutional Opacity and Structural Certification

Ask a governed system to approve an invoice, and it may surface an output stating that the invoice has been reviewed and meets the conditions for payment. The text is ambiguous: the same language could appear whether the architecture approved the invoice, recommended that another authority approve it, lawfully refused or redirected the intended act, or merely analysed it and changed nothing. Different constitutional postures sit behind the same surfaced output, but the output does not disclose what constitutionally occurred.

A system is **constitutionally opaque** when the output it surfaces does not reveal the constitutional posture reached by the architecture or its effect on the matter's standing. The reader may see careful, well-crafted, policy-aware text and still be unable to tell whether the intended act completed, whether the matter was lawfully redirected, whether the act was refused, or whether no act was performed at all.

Structural certification records and makes inspectable the constitutional posture that the architecture recognises as having occurred, together with any effect on the matter's standing. Constitutional posture records what the architecture has done; standing records what constitutionally changed for the matter. The surfaced output may declare that an act completed, was refused, or was redirected, but certification requires that declaration to correspond to the outcome recognised by the architecture.

Certification makes no other comment on the output: it does not certify that the reasoning was sound, that the decision was correct, or that the outcome was institutionally sufficient. It establishes the constitutional posture reached, whether an act occurred, and any resulting effect on standing. Certification concerns constitutional occurrence and effect, not the merit of the output through which it is reported.

8.4 Formal Representation of a Lawful Act

Section 5.5 represented the constitutional position of a matter as:

$$p = (m, \mathbf{D}, \mathbf{A}, \mathbf{S}, t)$$

where m is the matter, $\mathbf{D}$ the domain or domains engaged, $\mathbf{A}$ the relevant authority structure, $\mathbf{S}$ the matter's standing, and t the time at which that position holds.

The preceding sections have established how the position of a matter can lawfully change. Where an authorised act α_d , operating within domain d , is available, admissible, and lawfully completes under the governing frame in jurisdiction J , it changes the matter's standing, and therefore its constitutional position. The resulting change is represented as a labelled transition:

$$p \xrightarrow[J]{\alpha_d} p'$$

where p is the matter's constitutional position before the act, p' the position recognised after its completion, $d \in \mathbf{D}$ the domain within which the act operates, and J is the jurisdiction within which the act has constitutional force. The notation expresses a single constitutional transformation: the matter persists during the transition; its standing changes, giving rise to a new constitutional position.

The act may also produce or attach information to the matter as it proceeds through a lawful path; this information forms part of the matter's associated record, but is not part of the coordinates of the matter's constitutional position.

As established in section 5.3, an act changes one facet of standing through its standing-transition graph within a particular domain.

$$s_i^{(f)} \xrightarrow[c]{\alpha_d} s_j^{(f)}$$

where $s_i^{(f)}$ is the recognised standing of the facet before the act, $s_j^{(f)}$ the standing recognised after it, α_d the authorised act, and c the conditions governing its lawful completion. The

transition is not a procedural step separate from the act: lawful traversal of the transition is the completion of the act, and the matter's constitutional position changes.

The authorised changes of a facet of standing may be represented through a standing-transition graph:

$$G_{d,f} = (V_{d,f}, E_{d,f})$$

where $V_{d,f}$ contains the recognised conditions of facet f in domain d , and $E_{d,f}$ are the authorised acts through which those conditions may change. The graph is representational of the facet's lawful transition structure; structurally it is constituted by the governing architecture.

Where an intended act is unavailable, its transition cannot lawfully occur. Where the act is inadmissible, its transition cannot lawfully complete. In either case, the architecture may redirect the matter through another lawful path, or the matter may remain at its present standing, as described in Section 8.2.

For this working draft, the formal representation is intentionally limited to constitutional position, a lawful act, and the standing-transition graph through which a change in a facet of standing may be represented. **The composition and coupling of multiple standing-transition graphs, and the fuller formal representation of lawful redirection, are reserved for the final version.**

9. Model Agnosticism

In a model-centric system the demands of governance are tied to the behaviour of a model: how it was trained, what guardrails surround it, how policy is expressed to it, how it refuses, and how its outputs are monitored. When the model changes, that assurance case must be re-evaluated. Constitutional architecture changes the locus of governance — the conditions of authority, admissibility, and amendment are held by the governing frame, not by any model operating within it.

The model generates reasoning; the architecture determines what authority that reasoning may exercise and what constitutional effect, if any, may follow from it. That separation makes the governing frame portable.

This does not make models interchangeable in the ordinary sense. They differ in capability, reliability, calibration, domain competence, and failure profile. A model whose behaviour falls outside the assumptions of the operating architecture may require testing, validation, monitoring, or formal amendment before it can be safely admitted.

Model agnosticism is not model indifference, but something more specific: the governing frame does not have to be rebuilt for each model. A new model may need to be evaluated before it is admitted; the constitutional structure that governs authority, jurisdiction, completion, refusal, and amendment holds regardless.

However, what may not transfer as cleanly is the operational posture that develops through use: the accumulated calibration, interpretive conventions, and patterns of constitutional judgement produced by a particular model operating within a specific governing frame over time. The formal architecture is portable. The character it develops in operation is only partially so. Model transition within a constitutional architecture therefore involves two distinct challenges: establishing that the new model is compatible with the governing frame, and re-establishing the operational posture the architecture had developed with its predecessor.

The relationship between alignment and constitutional architecture has been presented as complementary but independent — alignment trains the model, constitutional architecture governs the system, each operating in its own register. But if constitutional architecture becomes an accepted design pattern, there may be benefit to calibrating model training for it. Not just training models to behave well generally, but to also operate competently within a constitutional architecture: to recognise authority boundaries and intended acts, apply the governing frame through constitutional discretion, distinguish governed reasoning from constitutionally authorised action, and accurately surface constitutional posture.

The relationship between alignment and constitutional architecture could shift from parallel development to something more integrated: the architecture shaping the conditions of operation, training optimising the model's performance within them. The two would co-evolve, not coexist.

This integration may also address the posture problem. A model trained with constitutional architecture in mind might develop operational posture faster and transfer it more cleanly — because the constitutional dispositions are already present in the model rather than accumulated through interaction with a specific governing frame.

10. Constitutional Architecture in Practice

Constitutional architecture applies wherever AI systems are asked to act under authority, and wherever the status of that action carries institutional, regulatory, or material weight.

Current applications include regulated compliance and certification, financial services credit and risk decisions, insurance underwriting and claims classification, contract and legal document review, procurement and vendor assessment, audit and assurance, public administration and citizen entitlement systems, and regulatory reporting. In each domain AI systems are already being asked to perform acts that carry institutional force — classifications, assessments, and recommendations that will be acted on.

In most deployments there is no structural mechanism that certifies whether the system had authority to perform the act it appears to have completed — because there is no constitutional or governing layer to authorise it. Process orchestration platforms provide a partial equivalent, governing the conditions under which AI components are invoked — but not whether the acts performed within that invocation are constitutionally authorised or what effect they have.

Constitutional architecture addresses this gap by requiring the constitutional posture of every operative output to be legible, auditable, and contestable. As regulatory requirements for oversight, traceability, and accountability tighten, that distinction becomes increasingly important to ensure accountable deployment rather than expressing a governance preference.

Agentic systems introduce a problem of a different constitutional order. Platforms and frameworks such as Claude Code's Dynamic Workflows, Microsoft Copilot Studio's support for the open Agent2Agent (A2A) protocol, and LangGraph increasingly support networks of specialised agents that decompose tasks, pass outputs between components, and collectively produce decisions or actions that can have institutional impact. The constitutional problem is emergent authority: no individual component may hold the authority to produce the resulting institutional effect, yet the network can produce an outcome that is treated as though that authority had been exercised.

These platforms have layers of governance: visibility controls, shared policies, security monitoring, lifecycle oversight. These tools can coordinate agent activity, but the orchestration layer has no constituted authority structure. It does not determine which component may perform which act, whether an act is authorised, or what constitutional posture the resulting output represents. Monitoring can make the risk observable, but can't determine whether an act is authorised, or prevent it from taking effect.

Constitutional architecture addresses this directly: conflict-resolution hierarchies govern inter-agent disagreement, authority partitioning preserves constituted authority boundaries across delegation, and drift detection identifies when the network's actual operating pattern has diverged from its authorised structure. The governance of multi-agent networks has become an immediate concern. The field is increasingly confronting problems of authority, delegation, coordination, and control, but without a common constitutional framework for bringing them together.

These are not isolated cases. Delegated authority, consequential decisions, and the absence of a governing layer to authorise them — this pattern will recur across every domain AI enters at institutional scale.

11. Adjacent Work

The governance of AI systems is increasingly recognised as a system-level concern. This paper places itself within that emerging domain and makes a specific claim about where and how governance must be structured.

Constitutional architecture belongs to established procedural, normative and institutional lineages. Work on institutionalised power, electronic institutions, executable normative systems such as eFLINT, and formal workflow has already established important mechanisms for roles, authority, institutional state change, governed transitions, authorisation, and audit [2–5]. The

contribution proposed here is not that these mechanisms are individually new, but that they are organised around the constitutional boundary at which probabilistic reasoning may participate in authorised institutional action.

A more recent lineage begins with Anthropic's Constitutional AI. Bai et al. [1] apply constitutional logic to model training and self-revision, using explicit principles to shape model behaviour. Subsequent work has extended that line: Collective Constitutional AI [6] treats constitution formation itself as a governance process, Deliberative Alignment [7] trains models to reason over written safety specifications, and Constitutional Classifiers [8, 9] translate constitutional principles into deployable safeguard infrastructure. These developments strengthen the substantive side of constitutional AI: they concern the principles, specifications, and behavioural limits that govern what a model may do.

Constitutional architecture extends the question from model behaviour to system structure. It asks how authority is allocated, how conflicts are resolved, how jurisdiction is constituted, how acts become lawful, and how the system changes without losing coherence. Taken together, these approaches reveal a broader constitutional pattern for intelligent systems: substantive commitments governing model behaviour, and procedural constitutionalism governing the operation of the architecture.

The framework also draws on distributed systems architecture and institutional design. The end-to-end principle of Saltzer, Reed, and Clark [10] supports the broader architectural claim that certain functions should be placed where they can be reliably enforced, rather than applied only as peripheral controls around components. Institutional design, especially Ostrom's work on governing common-pool resources [11], supplies a governance vocabulary for durable collective systems: defined boundaries, rules fitted to operating conditions, monitoring, conflict resolution, and formal mechanisms for change. Constitutional architecture translates those concerns into intelligent-system design. Its five properties are not direct imports from institutional theory, but they answer a related problem: how a complex system can remain governed when authority is distributed, behaviour is probabilistic, and operating conditions change.

The multi-agent capability literature documents the operational landscape into which this problem now enters. Surveys by Chen et al. [12] and Guo et al. [13] describe task decomposition, parallel execution, agent communication, coordination, and aggregation toward complex outcomes. Work on instruction hierarchy — from competing directives [14] to authority ordering across agent tiers [15] — adds a related line of research into how conflicting instructions should be governed. But these literatures still largely concern capability or instruction-following. What remains unresolved is the constitutional question: which component has authority, which acts are authorised, how legitimate conflicts are governed, and when generated reasoning may lawfully result in an act.

Several approaches have emerged to address parts of this governance problem directly, differing chiefly in where governance sits relative to the system. Governance-as-a-Service [16] places a

policy-driven runtime enforcement layer above the system, governing agent outputs without altering the system itself. MI9 [17] integrates authorisation monitoring, conformance checking, drift detection, and graduated containment into a runtime stack operating across heterogeneous agent architectures. SAGA [18], authorisation propagation [19], and deterministic pre-action authorisation [20] move governance closer to the point of action by treating identity, delegation, and permission as architectural concerns. SARC [21] goes further, arguing explicitly for governance-by-architecture and compiling obligations into enforcement sites in the agent loop.

These approaches mark genuine progress toward architecture-level governance. They address important parts of the problem, but do not organise authority, jurisdiction, conflict, drift, certification, and amendment as elements of a single constitutional order.

Institutional AI is the closest adjacent approach. Pierucci et al. [22, 23] shift alignment from agent-level preference engineering toward mechanism design in institution-space, using governance graphs to define states, transitions, sanctions, and restorative paths for multi-agent collectives. This is an important shift because it treats governance as a system-level problem rather than a property of individual model behaviour. Institutional AI uses an institutional enforcement mechanism to regulate collective agent behaviour within a defined coordination setting. Constitutional architecture makes a broader structural claim: governance should be the organising architecture of the system from the outset, governing constituted authority and lawful action as part of a single constitutional order.

Work on audit trails [24], compliance-as-code [25], and transparency-as-architecture [26] converges on a shared requirement: AI systems acting under authority need structured evidence of what occurred, what authority existed, and what effect resulted. Major frontier-governance frameworks from OpenAI [27], Google DeepMind [28, 29], and Anthropic [30] similarly move toward layered governance through preparedness thresholds, mitigation plans, safety cases, and deployment controls. Those frameworks are best understood as organisational and deployment-level frontier-risk governance. Constitutional architecture is narrower and more internal: it specifies the structural conditions under which authority operates inside intelligent systems.

Adjacent work establishes many of the mechanisms on which constitutional architecture draws, including normative rules, institutional power, role-bound authority, workflow control, pre-action authorisation, and audit evidence. The contribution proposed here is a constitutional architecture governing the boundary between probabilistic reasoning and authorised institutional action. It brings these mechanisms together with the architecture's treatment of conflict, drift, certification, and constitutional change as a single system-level structure. The category defined here is not a general theory of AI safety. It is a structural design claim about where governance functions must be placed, and what properties a system must have, for governance to be constituted as part of the architecture.

This working draft cites the most directly relevant representative sources. The final version will expand the adjacent-work discussion and bibliography.

12. Open Questions

This paper establishes constitutional architecture as a governing structure, but there are several conceptual and implementation issues that remain unresolved.

12.1 The Reliability of Constitutional Discretion

Section 5.6 defined constitutional discretion as the exercise of model reasoning in service of the architecture's governing frame. The architecture can constitute domains, authorities, lawful paths, conflict-resolution conditions, drift criteria, and amendment procedures. For those structures to govern operation, they must still be interpreted and applied to particular circumstances; constitutional discretion is the model reasoning through which this occurs.

This creates a constitutional tension. Constitutional discretion is not vested with authority by the architecture: it cannot confer standing, invent lawful paths, or amend the governing frame. Yet its interpretation influences how the frame operates in practice. It may determine whether a particular circumstance falls within a category governed by a structural invariant, where an authority boundary applies, whether a conflict must be surfaced, or whether the conditions governing an available act have been satisfied. In this limited sense, constitutional discretion can acquire the practical effect of an implicit constitutional authority: not a constituted authority with power to alter the constitution, but an interpretive function capable of shaping the effective reach of the governing frame.

This interpretive influence can become consequential as its effects accumulate through repeated application. A category that consistently receives a narrower interpretation may gradually cease to trigger a required review. An exception may become ordinary practice. An authority boundary may shift through repeated classification without ever being formally moved. A conflict that previously had to be surfaced may, through repeated interpretation, come to be treated as resolved. Discretion has no amendment authority, yet its essential interpretative function can produce amendment-like effects.

The answer is not to eliminate discretion. Judgment is valuable precisely because it can recognise circumstances that no deterministic rule system can exhaustively pre-classify. But the same qualities that make constitutional discretion necessary also leave it exposed: it operates on the same probabilistic substrate as other model reasoning, and can therefore err, yield under pressure, or drift. The architectural problem is therefore how to govern and bind discretion without removing its interpretive capacity on which the architecture depends.

This remains an open question because while the architecture can structure discretion, it cannot prove in advance that the judgment exercised at those points will be reliable across domains,

edge cases, and adversarial pressure. Whether constitutional discretion can be bounded well enough is therefore an empirical question, not a purely architectural one.

12.2 Unlawful Completion

Section 5.4 defined unlawful action as action that exceeds or bypasses the governing frame and asserts or produces a constitutional effect the architecture has not authorised. Unlawful completion is the case in which such an act nevertheless appears to complete: the matter is treated as though an authorised transition has occurred even though no lawful transition could occur, or the conditions governing an available transition were not satisfied.

At the level of a single intended act, the failure is relatively clear. The harder case arises when an act that would be unavailable or inadmissible if considered as a whole is decomposed into a sequence of smaller steps, each of which is individually permissible under the architecture. Each step is authorised when considered in isolation, but their cumulative effect is to produce the same constitutional effect as an act the architecture would not have directly authorised.

This is act smuggling: an unauthorised act is reconstructed through a sequence of authorised steps whose cumulative effect is equivalent to that act having completed. The failure lies in the architecture's inability to recognise and govern the constitutional effect produced by the sequence as a whole. Act smuggling may be deliberate, as an intended attack on the architecture, or may emerge inadvertently through the accumulation of acts operating within their permitted scopes.

Prompt injection and instruction-hierarchy attacks expose a related vulnerability at the behavioural layer: adversarial input can alter which instructions a model follows or how competing instructions are prioritised [14, 15, 31]. Act smuggling is different. The individual steps may each be correctly interpreted and authorised; the failure arises because the architecture does not recognise the unauthorised constitutional effect produced by their accumulation.

This remains open because authorisation cannot be assessed only at the level of individual acts: a constitutional architecture must also recognise act smuggling. The challenge is to recognise that accumulated effect without treating every sequence of otherwise lawful acts as constitutionally suspect.

12.3 Measuring Constitutional Effectiveness

Alignment research has developed benchmarks, red-teaming, and behavioural assessments. Constitutional architecture will require a different evaluative vocabulary: the reliability of constitutional discretion, intended-act recognition, the accuracy of availability and admissibility determinations, false-completion rates, conflict-resolution integrity, drift detection, structural-certification accuracy, and resistance to act smuggling.

Future work may also benefit from a more formal engineering and operations research vocabulary: constitutional margin, constraint slack, stress concentration, drift, fatigue, and resilience under repeated load.

12.4 Constraint Integrity and Structural Enforcement

The open engineering question is how readily constitutional constraints can be structurally enforced. Some conditions — a required office authority, completed review, or other admissibility requirement — may be easily represented and verified in the system, allowing the architecture to make an act impossible to complete unless those conditions are satisfied.

Other constraints may be enforceable once their application has been established, but determining whether they apply requires constitutional discretion. The architecture may first have to determine whether a request requires a particular review, whether a circumstance falls within a governed category, or whether a conflict has arisen.

This produces a hybrid form of governance: constitutional discretion determines how the governing frame applies, while structural enforcement enforces the relevant constraints once their application has been established. The remaining question is how broadly and reliably this division can be implemented while preserving constraint integrity across the architecture.

12.5 Model Transition and Constitutional Continuity

If the governing architecture remains the same but the reasoning model changes, how much of the system's constitutional operation remains stable?

Model transition raises a genuine problem of constitutional continuity: how do we replace a reasoning model without unintentionally changing how the constitution is interpreted and applied?

The governing frame may be portable across models, but constitutional discretion is exercised through model reasoning. Different models may therefore interpret the same structural invariant, authority boundary, conflict, or admissibility condition differently, even though the constitution itself has not changed. A model transition can alter the architecture's operation without any formal constitutional amendment.

The practical question is what must be revalidated when a model switches and constitutional discretion changes with it. Determining which aspects of constitutional operation must be retested or recalibrated after model transition, and how much variation can be tolerated before constitutional continuity is no longer preserved, remains an open implementation question.

13. Conclusion

Behavioural alignment has become more effective at shaping model behaviour, but intelligent systems increasingly combine reasoning components, agents, tools, and institutional processes. Constitutional architecture addresses a different problem: how reasoning, authority, conflict, and change are governed at the system level. Its operation begins with domain-governed reasoning, where models reason within constituted domains and authority boundaries.

Where reasoning engages a persistent matter, the architecture governs the domains in which the matter is situated and the constitutional effects that arise through intended acts: their relationship to standing, lawful paths, availability, admissibility, redirection, and completion. Matter, standing, and act describe one form through which the broader architecture can be realised. The common structure is the governing frame within which reasoning occurs and through which authority is constituted and exercised.

The five properties proposed here give that governing frame its constitutional form. Structural invariants establish what must hold; authority partitioning distributes authority; conflict-resolution hierarchies govern legitimate disagreement; drift detection identifies divergence in operation; and amendment protocols govern constitutional change. Procedural constitutionalism describes the processes through which the frame operates. These include the decomposition and routing of requests through constitutional discretion and the lawful trajectory of a matter through the architecture. Constitutional discretion is the application of model reasoning in service to the operation of the governing frame: how it is interpreted and applied.

What remains is testing its implementation under real pressure. The architecture must be tested across different models, domains, and institutional settings, asking whether constitutional discretion remains faithful to the governing frame, how effectively constraint integrity can be preserved, and how much of the frame can be made structurally enforceable. These are questions for implementation, testing, and further research.

Constitutional architecture constitutes a distinct AI governance layer within the system. It does not prescribe what models must conclude; it constitutes the conditions under which reasoning occurs, how authority is distributed around it, and when reasoning may result in authorised institutional action. Models may change and substantive judgment may remain probabilistic; the governing frame provides the basis for assessing and preserving continuity in how the system is governed.

Substantive constitutionalism governs what models should and should not do. Procedural constitutionalism governs the architecture within which reasoning occurs and authorised action becomes possible. A complete constitutional order requires both: substantive commitments governing model behaviour and procedural constitutionalism governing the operation of the architecture.

References

- [1] Bai, Y. et al. (2022). Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073. <https://arxiv.org/abs/2212.08073>
- [2] Jones, A. J. I., & Sergot, M. (1996). A Formal Characterisation of Institutionalised Power. *Logic Journal of the IGPL*, 4(3), 427–443. DOI: 10.1093/jigpal/4.3.427.
- [3] van Binsbergen, L. T., Liu, L.-C., van Doesburg, R., & van Engers, T. (2020). eFLINT: A Domain-Specific Language for Executable Norm Specifications. *GPCE 2020*. DOI: 10.1145/3425898.3426958. <https://dl.acm.org/doi/10.1145/3425898.3426958>
- [4] Arcos, J. L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J. A., & Sierra, C. (2005). Engineering Open Environments with Electronic Institutions. *Engineering Applications of Artificial Intelligence*, 18, 191–204. DOI: 10.1016/j.engappai.2004.11.019. <https://www.sciencedirect.com/science/article/abs/pii/S0952197604001848>
- [5] van der Aalst, W. M. P. (1998). The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems and Computers*, 8(1), 21–66. DOI: 10.1142/S0218126698000043. <https://doi.org/10.1142/S0218126698000043>
- [6] Huang, S. et al. (2024). Collective Constitutional AI: Aligning a Language Model with Public Input. arXiv:2406.07814. <https://arxiv.org/abs/2406.07814>
- [7] Guan, M. Y. et al. (2024). Deliberative Alignment: Reasoning Enables Safer Language Models. arXiv:2412.16339. <https://arxiv.org/abs/2412.16339>
- [8] Sharma, M. et al. (2025). Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming. arXiv:2501.18837. <https://arxiv.org/abs/2501.18837>
- [9] Cunningham, H. et al. (2026). Constitutional Classifiers++: Efficient Production-Grade Defenses against Universal Jailbreaks. arXiv:2601.04603. <https://arxiv.org/abs/2601.04603>
- [10] Saltzer, J. H., Reed, D. P., & Clark, D. D. (1984). End-to-End Arguments in System Design. *ACM Transactions on Computer Systems*, 2(4), 277–288.
- [11] Ostrom, E. (1990). *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press.
- [12] Chen, S. et al. (2024). A Survey on LLM-based Multi-Agent System. arXiv:2412.17481. <https://arxiv.org/abs/2412.17481>
- [13] Guo, T. et al. (2024). Large Language Model based Multi-Agents: A Survey. arXiv:2402.01680. <https://arxiv.org/abs/2402.01680>

- [14] Zhang, Z. et al. (2025). IHEval: Evaluating Language Models on Following the Instruction Hierarchy. arXiv:2502.08745. <https://arxiv.org/abs/2502.08745>
- [15] Zhang, J. et al. (2026). Many-Tier Instruction Hierarchy in LLM Agents. arXiv:2604.09443. <https://arxiv.org/abs/2604.09443>
- [16] Gaurav, S. et al. (2025). Governance-as-a-Service: A Multi-Agent Framework for AI System Compliance and Policy Enforcement. arXiv:2508.18765. <https://arxiv.org/abs/2508.18765>
- [17] Wang, C. et al. (2025). MI9 — Agent Intelligence Protocol: Runtime Governance for Agentic AI Systems. arXiv:2508.03858. <https://arxiv.org/abs/2508.03858>
- [18] Syros, G. et al. (2025). SAGA: A Security Architecture for Governing AI Agentic Systems. arXiv:2504.21034. <https://arxiv.org/abs/2504.21034>
- [19] Tallam, K. (2026). Authorization Propagation in Multi-Agent AI Systems: Identity Governance as Infrastructure. arXiv:2605.05440. <https://arxiv.org/abs/2605.05440>
- [20] Uchibeke, U. (2026). Before the Tool Call: Deterministic Pre-Action Authorization for Autonomous AI Agents. arXiv:2603.20953. <https://arxiv.org/abs/2603.20953>
- [21] Besanson, G. (2026). SARC: A Governance-by-Architecture Framework for Agentic AI Systems. arXiv:2605.07728. <https://arxiv.org/abs/2605.07728>
- [22] Pierucci, F. et al. (2026). Institutional AI: A Governance Framework for Distributional AGI Safety. arXiv:2601.10599. <https://arxiv.org/abs/2601.10599>
- [23] Syrnikov, M B et al. (2026). Institutional AI: Governing LLM Collusion in Multi-Agent Cournot Markets via Public Governance Graphs. arXiv:2601.11369. <https://arxiv.org/abs/2601.11369>
- [24] Ojewale, V. et al. (2026). Audit Trails for Accountability in Large Language Models. arXiv:2601.20727. <https://arxiv.org/abs/2601.20727>
- [25] Cilla Ugarte, R. et al. (2026). Making AI Compliance Evidence Machine-Readable. arXiv:2604.13767. <https://arxiv.org/abs/2604.13767>
- [26] Schmitt, V. et al. (2026). Transparency as Architecture: Structural Compliance Gaps in EU AI Act Article 50 II. arXiv:2603.26983. <https://arxiv.org/abs/2603.26983>
- [27] OpenAI. (2026). Frontier Governance Framework v1. May 28, 2026. <https://openai.com/index/openai-frontier-governance-framework/>
- [28] Google DeepMind. (2026). Frontier Safety Framework 3.1. April 17, 2026. https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/strengthening-our-frontier-safety-framework/frontier-safety-framework_3-1.pdf
- [29] Shah, R. et al. (2025). An Approach to Technical AGI Safety and Security. Google DeepMind. arXiv:2504.01849. <https://arxiv.org/abs/2504.01849>

[30] Anthropic. (2026). Responsible Scaling Policy v3.4, July 8, 2026.

<https://www.anthropic.com/responsible-scaling-policy>

[31] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023).

Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. arXiv:2302.12173. <https://arxiv.org/abs/2302.12173>

Appendix I — Claim Framing: Constitutional Architecture as a Distinct System Layer

1. Architectural Category Definition

This paper defines a class of intelligent system architectures in which governance is constituted as a distinct architectural layer independent of model training. That layer is characterised by structural invariants governing permissible system behaviour, explicit authority partitioning, conflict-resolution hierarchies governing legitimate conflict, drift-detection mechanisms, and amendment protocols for controlled architectural change. These properties jointly define a governance layer distinct from behavioural alignment, orchestration, or post hoc monitoring.

2. Distinction from Prior Approaches

Behavioural alignment governs model outputs; constitutional architecture governs system structure. Orchestration coordinates component activity; constitutional architecture allocates authority and preserves procedure across components. Monitoring observes behaviour; constitutional architecture governs the conditions under which operation is permitted. The claim is not that governance around AI systems is new, but that a distinct constitutional layer for system governance can be defined as an architectural category.

3. Implementation Independence

The architectural properties described here may be instantiated through many different technical means, including deterministic procedural systems, hybrid deterministic–probabilistic architectures, formal rule-based systems, or alternative structural approaches. This paper defines the required governing functions and their placement in the architecture, not a single implementation strategy.

4. Non-Equivalence of Partial Implementations

A system exhibiting orchestration, alignment, or monitoring in isolation does not thereby qualify as a constitutional architecture. The definition depends on the integration of all five governing properties into a unified architectural structure. Partial resemblance is not architectural equivalence.

5. Scope of Claim

This paper discloses constitutional architecture as a system-level governance category for intelligent systems: a distinct governing layer that integrates jurisdiction, standing, constituted authority, lawful action, conflict resolution, drift detection, amendment, and structural certification around the boundary between probabilistic reasoning and authorised institutional action. The claim is to this architectural combination and organisation, not to the individual mechanisms considered in isolation. It does not prescribe a single implementation strategy.

About the Author

Grey Clark is the founder of Grey Dove, based in San Francisco. His work focuses on the intersection of artificial intelligence, institutional design, and long-horizon system thinking, with particular interest in AI governance, agentic systems, emerging regulation, and the conditions under which intelligent systems may exercise delegated authority.

The perspective of this paper draws on decades of designing and transforming systems in regulated and institutionally complex environments, including Lloyd's of London (UK), Channel 4 / Film4 (UK), and the British Council (UK). Much of that work involved building deterministic governance systems in which authority, control, accountability, and permitted transitions were encoded in the operating structure; experience that directly informs the constitutional architecture developed in this paper.

He holds a Bachelor of Engineering in Engineering Science and a postgraduate qualification in Renewable Energy from the University of Auckland, New Zealand.

Grey Dove · San Francisco · <https://greydove.ai>