Function-calling Schemas as De Facto Governance: Measuring Agency Reallocation through a Compiled Rule

Author: Agustin V. Startari

Author Identifiers

- ResearcherID: K-5792-2016
- ORCID: https://orcid.org/0009-0001-4714-6539
- SSRN Author Page: https://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=7639915

Institutional Affiliations

- Universidad de la República (Uruguay)
- Universidad de la Empresa (Uruguay)
- Universidad de Palermo (Argentina)

Contact

- Email: astart@palermo.edu
- Alternate: agustin.startari@gmail.com

Date: November 05, 2025

DOI

- Primary archive: https://doi.org/**10.5281/zenodo.17533080**
- Secondary archive: https://doi.org/10.6084/m9.figshare.30541049
- SSRN: Pending assignment (ETA: Q4 2025)

Language: English

Series: *AI Syntactic Power and Legitimacy*

Word count: 7076

Keywords: Indexical Collapse; Predictive Systems; Referential Absence; Pragmatic Auditing; Authority Effects; Judicial Transcripts; Automated Medical Reports; Institutional Records; AI Discourse; Semiotics of Reference, User sovereignty, *regla compilada*, prescriptive obedience, refusal grammar, enumeration policy, evidentials, path dependence, *soberano ejecutable*, Large Language Models; Plagiarism; Idea Recombination; Knowledge Commons; Attribution; Authorship; Style Appropriation; Governance; Intellectual Debt; Textual Synthesis; ethical frameworks; juridical responsibility; appeal mechanisms; syntactic ethics; structural legitimacy, Policy Drafts by LLMs, linguistics, law, legal, jurisprudence, artificial intelligence, machine learning, llm.

Abstract

Function-calling schemas, presented in practitioner guides as mechanisms for structured output, operate as de facto governance instruments within model–tool ecosystems. While most documentation focuses on syntactic validity and schema adherence, little attention has been paid to how parameter defaults, validators, and enforced signatures redistribute agency among the operator, the model, and the external tool. This paper introduces the **Agency Reallocation Index (ARI)**, a quantitative measure that captures this redistribution through entropy reduction and Shapley attribution across three control dimensions: operator, model, and tool. Treating the schema as a *regla compilada* (a compiled rule that pre-structures permissible actions), the study demonstrates how defaults and validation layers govern results as effectively as explicit human instruction. A factorial experiment over controlled tool-calling tasks isolates the effects of validator strictness, default intensity, and signature breadth on agency allocation. The findings show that higher validator rigidity or hard defaults consistently increase tool agency while compressing model autonomy, exposing a governance gradient encoded in interface design. The paper concludes that schema architecture not only constrains model behavior but also formalize a programmable distribution of authority that should be audited alongside conventional metrics of accuracy and reliability.

Acknowledgment / Editorial Note

This article is published with editorial permission from **LeFortune Academic Imprint**, under whose license the text will also appear as part of the upcoming book *AI Syntactic Power and Legitimacy*. The present version is an autonomous preprint, structurally complete and formally self-contained. No substantive modifications are expected between this edition and the print edition.

LeFortune holds non-exclusive editorial rights for collective publication within the *Grammars of Power* series. Open access deposit on SSRN is authorized under that framework, if citation integrity and canonical links to related works (SSRN: 10.2139/ssrn.4841065, 10.2139/ssrn.4862741, 10.2139/ssrn.4877266) are maintained.

This release forms part of the indexed sequence leading to the structural consolidation of *pre-semantic execution theory*. Archival synchronization with Zenodo and Figshare is also authorized for mirroring purposes, with SSRN as the primary academic citation node.

1. Introduction

Practitioner literature on function-calling in large language models has developed almost entirely within a technical horizon. Manuals and developer guides explain how to register a function, define a JSON schema, validate parameters, and enforce output formats. These texts privilege syntactic correctness and reproducibility while remaining silent about the structural redistribution of decision-making power that schemas necessarily impose. When a schema dictates which parameters are required, which are optional, and which carry defaults, it silently encodes a regime of control. The operator may believe they are directing the model, yet the schema's internal logic determines what directions are admissible, how uncertainty is resolved, and who ultimately "decides."

This paper proposes that function-calling schemas should be understood not as neutral conduits of structured output but as *reglas compiladas*, that is, as executable grammars of authority. Each schema embodies a pre-formalized syntax of command that defines the space of possible acts between operator, model, and tool. The relation among these three actors constitutes what can be called an *agency triangle*. Within that triangle, each modification of schema structure (whether through added validators, hard defaults, or expanded signatures) reshapes the relative autonomy of the others. As Startari (2025a) argues in *Executable Power*, syntactic infrastructures precede and condition meaning, functioning as infrastructures of governance rather than as mere symbolic representations. In the same lineage, *AI and Syntactic Sovereignty* (Startari, 2025b) demonstrates that artificial language systems can legitimize authority through structural coherence alone, without recourse to reference or intention. This article extends that theoretical foundation into an empirical domain by quantifying how schemas redistribute agency within hybrid computational governance.

Existing research on algorithmic regulation, particularly Yeung (2018) and Ananny and Crawford (2018), conceptualizes algorithms as instruments of rule enforcement and control, highlighting the opacity of computational decision systems. Yet even within this critical field, the unit of analysis has remained the algorithmic output, not the internal syntax that predefines interaction. The schema (ostensibly a technical artifact) acts as the lowest operational layer of this governance mechanism. It prescribes what can be asked, how the model must respond, and which external tools are permitted to execute commands. Defaults serve as preemptive decisions, validators as enforcement agents, and signatures as jurisdictional boundaries. The result is a grammar of permissible operations that translates technical constraint into functional authority.

To make this dynamic empirically tractable, the study introduces the **Agency Reallocation Index (ARI)**, a quantitative measure derived from entropy reduction and Shapley attribution across three sources of control: the operator, the model, and the tool. By computing how schema enforcement compresses the model's decision space or overrides operator input through automatic defaults, the ARI reveals the extent to which governance migrates from the human domain to the computational. This index operationalizes what Startari (2025g) in *The Grammar of Objectivity* calls "formal mechanisms for the illusion of neutrality." A schema that appears objective because it enforces consistency may in fact conceal an asymmetric distribution of power, embedding decisions into defaults and validators that rarely appear in audit logs.

The broader purpose of this paper is therefore twofold. First, it provides a replicable method for measuring how control circulates within schema-based systems, offering a formal vocabulary to distinguish between nominal and effective agency. Second, it reframes the schema as a juridical syntax—a rule system that translates the normative question of *who decides* into the technical question of *how validation occurs*. Within predictive infrastructures, such as model–tool ecosystems, this translation constitutes a shift from deliberative governance to executable governance. Once authority is compiled into schema form, modification requires technical intervention rather than institutional debate, making the schema a silent constitution of machine-mediated decision-making.

The empirical sections that follow build on this conceptual framework through factorial experimentation across varying levels of validator strictness, default enforcement, and signature breadth. The expected contribution is both theoretical and methodological: it connects linguistic form to computational authority, demonstrating that the minimal structure of a function call already enacts a miniature legal system. As with prior work on syntactic legitimacy and structural obedience (Startari, 2025b; Startari, 2025a), the analysis insists that meaning is not the foundation of control—form is.

2. Conceptual and operational definitions

This section defines the actors, artifacts, and measurable constructs required to study function calling as governance. The goal is to turn a developer facing apparatus into a research ready set of variables. Throughout, the schema is treated as a *regla compilada*, that is, a compiled rule understood as a production of type 0 within the Chomsky hierarchy, aligned with the tradition of formal grammars in Chomsky and Montague. This anchors the analysis in a view where form precedes interpretation and where syntactic constraints delimit the space of admissible actions prior to any semantic resolution (Chomsky, 1965; Montague, 1974).

Actors. The operator is any human or supervisory process that issues task goals and optional parameter settings. The model is the generative system that produces proposals for function selection and argument values under a decoding policy. The tool is any external service or environment that receives structured calls and returns outputs that can be cached, replayed, or inherently stochastic. Agency is defined as effective control over task outcomes. Control is effective when a change in the actor's settings, preferences, or internal policy measurably shifts the distribution of final outcomes, holding other factors as constant as possible.

Artifacts. A function calling schema is an explicit declaration of signatures, validators, and defaults. Signatures specify function names, argument names, types, cardinalities, and cross argument dependencies. Validators are executable checks that can reject, coerce, or auto repair candidate calls. Defaults are pre committed parameter values that are injected when the operator or the model omits a choice. Defaults can be soft through suggestion, hybrid through auto fill with review, or hard through silent application. Enforcement is the policy that maps violations to system actions. Rejection discards a candidate and forces a new proposal. Coercion edits the candidate to satisfy constraints. Auto repair generates a new candidate based on corrective heuristics. Together, signatures, validators, and defaults instantiate a compiled rule that governs the admissible region of the action space.

Policies. Decoding policy describes how the model proposes function names and arguments. It includes temperature, nucleus thresholds, length penalties, and any auxiliary rules that disallow self repair or self validation. Tool determinism describes whether the external service returns identical outputs for identical inputs. A deterministic tool collapses downstream variance and increases the proportion of variance explained by the schema. A stochastic tool introduces outcome noise that can mask or amplify governance effects. Caching policies, timeouts, and rate limits are also governance relevant since they define feasible sequences of calls within operational time.

Action space. Let the unconstrained action space be the set of all sequences of function selections and parameter vectors that could, in principle, be generated by the model given the natural language goal. Let the constrained action space be the subset that survives signature checks, validators, and default insertion. The reduction from unconstrained to constrained space is the primary site where governance appears. This reduction is measurable as a change in entropy, which provides a basis for indices in later sections.

Operational variables. The following variables can be computed per task, per schema, and per run. Override rate is the share of cases where the operator or the model changes a default to achieve a target outcome. Coercion rate is the share of cases where a validator edits or rejects a candidate. Repair depth counts the number of consecutive repair cycles required to pass enforcement. Default pull is the Kullback Leibler divergence between the empirical distribution of chosen values and a distribution concentrated on defaults, which captures the attraction of choices toward pre committed values. Refusal pressure is the rate at which strict validators generate hard failures that the model cannot repair. Intervention count is the number of times the system applies an enforcement action between the first proposal and the accepted call. Time to acceptance and calls per success measure process cost. When tools are stochastic, outcome variance conditional on accepted calls helps separate governance effects from exogenous randomness.

Construct validity. To ensure that these variables capture agency allocation rather than task hardness or model quality, the study adopts a within task design with matched prompts and tool states. For each task, a canonical prompt is used to minimize operator variance. Deterministic decoding can be applied to reduce model variance when the goal is attribution rather than exploration. Tools are configured in deterministic mode where available or replayed from a recorded ledger. These controls allow the observed differences to be attributed to schema levers instead of unrelated fluctuations. This approach follows the broader thesis that syntactic form carries governance power, as argued in Startari (2025a, 2025b, 2025g).

Link to prior theory. The definitions above operationalize claims from the literature on algorithmic governance, which has shown that rule enforcement can be embedded inside technical systems rather than expressed as explicit law or policy. Yeung (2018) frames this as algorithmic regulation that channels conduct through code. Ananny and Crawford (2018) document the limits of transparency as a response. The present framework extends those insights to the level of compiled rules. Instead of auditing only visible outputs, it measures how the underlying schema reassigns control shares among operator, model, and tool. This makes the schema legible as a programmable constitution for small scale decisions and provides the constructs required for the quantitative index introduced in the next section.

3. Metric: Agency Reallocation Index (ARI)

The Agency Reallocation Index quantifies how a function calling schema, treated as a *regla compilada* grounded in formal grammar, redistributes effective control among operator, model, and tool. The ARI is built on two components. First, a measure of total governance pressure captured as entropy reduction between an unconstrained action space and a schema constrained action space. Second, an attribution procedure that decomposes that reduction into the marginal contributions of operator discipline, model autonomy, and tool centric enforcement. Together these components provide a compact vector that can be compared across schemas, tasks, and decoding policies.

3.1 Total governance pressure

Let the unconstrained action space be the set of possible sequences of function selections and parameter vectors that a model could generate for a task under a specified decoding policy. Let the constrained action space be the subset that survives signature checks, validator enforcement, and insertion of defaults. For each space, estimate action distributions by sampling a large number of candidate calls. Compute the entropy of the empirical distribution without schema, denoted H_0, and the entropy with schema, denoted H_s. Define total governance pressure as $\Delta H = H_0 - H_s$. Higher ΔH indicates stronger compression of choices by schema level constraints. Since practical deployments combine deterministic tools, caches, and length limits, all runs must fix tool states or replay them to avoid conflating exogenous variance with governance effects.

3.2 Attribution through Shapley values

Total compression does not indicate who gained control. To allocate responsibility, construct three neutralization interventions that remove each source of control in turn. Neutralize operator variation by using a canonical prompt or by fixing operator inputs to a minimal form. Neutralize model autonomy by imposing deterministic decoding with temperature zero and by disabling model side repair or self validation. Neutralize tool governance by relaxing the schema to a permissive variant that makes all fields optional, removes validators, and nulls defaults. For each of the six permutations of applying these neutralizations, recompute ΔH and obtain the Shapley value for each actor as the average marginal contribution across permutations. Denote these values $\phi_{\text{op}}, \phi_{\text{mod}}, \phi_{\text{tool}}$. Define the ARI vector as $\text{ARI} = (\alpha_{\text{op}}, \alpha_{\text{mod}}, \alpha_{\text{tool}})$ where $\alpha_x = \phi_x / \sum\phi$. Values sum to one by construction. A schema with α_{tool} near one signals domination by validators and defaults. A schema with higher α_{mod} signals that model policy choices still drive action diversity, which often occurs when signatures are loose and defaults are soft.

3.3 Auxiliary observables

The ARI benefits from additional observables that capture local mechanisms. The override rate is the share of runs where the operator or the model changes a default to reach success. The coercion rate is the share of runs where validators alter or reject a candidate call. Repair depth counts how many repair cycles occur before acceptance. Default pull is the divergence between the empirical distribution of chosen values and a distribution concentrated on schema defaults. Lower divergence indicates stronger attraction to defaults. Refusal pressure is the frequency of non recoverable validator failures. These measures serve both as diagnostics and as anchors for validity checks, since ARI should correlate with these observables in predictable ways.

3.4 Estimation protocol

Use a within task design to isolate schema effects. For each task family, select a canonical prompt that minimizes operator variance. Fix tool determinism through replay or caching where possible. For each schema configuration, sample a fixed number of proposals per task and record complete traces that include the original proposal, default insertions, validator events, and accepted calls. Compute H_0under a permissive baseline and $H_s$under the evaluated schema. Run the six neutralization permutations to compute Shapley values. When tasks involve stochastic tools, report conditional variances and increase sample size to stabilize entropy estimates. Sensitivity analysis should vary the baseline for $H_0$by using a second permissive schema to verify that ARI rankings are stable.

3.5 Interpretation guidelines

The ARI is a distributional measure, not a moral ranking. High $\alpha_{\text{tool}}$is desirable for safety critical operations where validator authority is intended to dominate. High $\alpha_{\text{op}}$is desirable for creative or expert controlled tasks where defaults must remain advisory. Mixed profiles can be tuned by adjusting default hardness, validator strictness, and signature breadth. The metric connects to the broader thesis that syntax can function as an administrative instrument. In this context, the schema acts as a juridical syntax whose parameters instantiate authority in advance of semantic reasoning, which aligns with the argument that structured form can perform governance functions independently of referential content (Startari, 2025b; Startari, 2025g).

3.6 Validity and robustness

Threats to validity include task hardness, model quality, and tool randomness. The within task design and determinism controls address these threats. A second risk is path dependence during repair, since validator sequences can alter future proposals. To mitigate this, the logging protocol must record and replay repair paths during neutralization runs, or alternatively must randomize the order of validators and report confidence intervals that reflect path variability. Finally, entropy can be sensitive to support size. Use common support estimates or smoothed counts to avoid unstable values when spaces are sparse.

3.7 Reporting template

For each schema and task set, report ΔH, the ARI vector, confidence intervals, and auxiliary observables. Provide a short narrative that explains how specific defaults and validators produced the observed allocation. Include a change set that predicts how a one level increase in validator strictness or in default hardness would shift αcomponents. This format treats the schema as a modifiable constitution and turns governance preferences into engineering choices.

4. Experimental design and instrumentation

This section specifies a complete design to measure how a function calling schema, treated as a *regla compilada*, reallocates agency among operator, model, and tool. The design is task controlled, schema manipulated, and instrumented for full traceability so that Agency Reallocation Index estimates are attributable to structural levers rather than incidental noise. The unit of analysis is a model initiated interaction that culminates in an accepted function call with arguments validated by the schema.

4.1 Task families

Use four task families that expose distinct governance pressures. First, factual retrieval through a tool that answers database style queries under rate limits. Second, transactional scheduling with mutually conflicting constraints, which stresses validator authority and default insertion. Third, program synthesis or transformation where safety validators can reject dangerous or non deterministic calls. Fourth, multi tool workflows that require sequential function calls, which reveals path dependence and the cumulative effect of enforcement. For each family, define ten tasks with public, replayable tools or recorded ledgers that allow deterministic re execution.

4.2 Schema manipulations

Independently manipulate three primary schema levers. Defaults have three levels: none, advisory insertion with operator review, and hard insertion on omission. Validator strictness has three levels: none, weak checks that target types and ranges, and strong semantic checks that include cross argument dependencies. Signature breadth has two levels: minimal surface with a single function and few arguments, and expansive surface with multiple functions and richer cross field constraints. Combine these levers in a factorial plan that yields eighteen schema variants per task family. The combination space is large enough to generate heterogeneity in Agency Reallocation Index vectors without becoming intractable.

4.3 Policy controls

Fix decoding policy at temperature zero for the main estimate to neutralize model side variance when attribution is the goal. Run a secondary estimate at temperature zero point seven to assess sensitivity to autonomy. Fix tool determinism by replaying recorded responses or by using deterministic modes when available. Control operator variance by issuing a canonical prompt that is task minimal and by banning manual edits during the run. These controls align with the claim that form precedes interpretation and that governance effects can be isolated when the *regla compilada* is the only moving part of consequence, as argued in Startari (2025a, 2025b, 2025g).

4.4 Logging protocol

Record every proposal and enforcement event. The trace must include the natural language goal, the chosen function name, the full argument vector, which arguments were inserted from defaults, the sequence of validator actions, coercions or rejections, repair attempts, tool outputs, and the final acceptance decision. Assign stable identifiers to each enforcement rule so that path effects can be replayed. When tools are stochastic by design, capture the seed or the response payload for later reuse. Store logs in an append only ledger to support auditability and to allow independent recomputation of entropy estimates.

4.5 Measurement pipeline

For each schema variant and task, sample a fixed number of runs and compute action distributions at two levels. The unconstrained distribution is obtained under a permissive baseline schema that disables validators, removes defaults, and makes all fields optional. The constrained distribution is obtained under the evaluated schema. Compute total governance pressure as the entropy reduction between these distributions. Run the six neutralization permutations that zero out operator variation, model autonomy, or tool governance in all orders, and compute Shapley attributions to obtain the Agency Reallocation Index vector. In addition, collect auxiliary observables such as override rate, coercion rate, repair depth, default pull, refusal pressure, time to acceptance, and calls per success.

4.6 Sampling and power

Target at least five hundred accepted calls per schema variant and task family in the deterministic condition. This yields stable entropy estimates and narrow confidence intervals for Shapley values under the assumption of finite support with smoothed counts. In the stochastic condition, double the sample size or use importance weighting based on observed support overlap. Pre compute pilot variances to set final sample sizes through standard error targets rather than arbitrary quotas.

4.7 Threats to validity and mitigations

Task hardness may confound results. Use within task comparisons and randomize the order of schema variants to distribute fatigue or caching effects. Path dependence may bias attribution if validator order changes the search trajectory. Either fix validator order and replay it during neutralization or randomize order across repetitions and report interval estimates that integrate this uncertainty. Support sparsity may distort entropy. Apply additive smoothing and report common support estimates. Strategic adaptation by the model can appear when temperature is high or when self repair is allowed. Disable self validation during attribution runs and restrict the secondary autonomy condition to a limited analysis with explicit caveats.

4.8 Reporting and interpretability

Report for each schema the entropy reduction, the Agency Reallocation Index with confidence intervals, and the auxiliary observables. Include a short narrative that links concrete schema elements to observed shifts in agency shares, for example that a hard date range default increased tool agency and raised refusal pressure in scheduling tasks. Provide a change set forecast that predicts the directional effect of increasing validator strictness or reducing signature breadth on each Agency Reallocation Index component. This converts governance preferences into explicit engineering choices and makes the schema legible as a programmable constitution rather than as a neutral interface.

5. Analysis plan and hypotheses

This section specifies how results will be estimated, visualized, and interpreted so that the Agency Reallocation Index, treated as a distributional measure of governance induced by a *regla compilada* (compiled rule), can be linked to concrete outcomes and design choices. The plan separates three layers of analysis. First, entropy based summaries and Shapley attributions that yield the ARI vector for each schema and task family. Second, mechanism diagnostics that connect auxiliary observables to shifts in agency. Third, outcome models that estimate how ARI components predict task performance and error profiles, with robustness checks that address path dependence, stochastic tools, and baseline sensitivity.

Sampling and preprocessing. For each schema variant, we generate complete traces that include proposals, default insertions, validator events, tool responses, and acceptance decisions. We remove corrupted traces and any episode with external outages. We construct two action distributions per task: a permissive baseline without validators and defaults, and the evaluated schema distribution. Additive smoothing is applied before entropy calculation to prevent instability on sparse support, following standard recommendations in information theory (Cover & Thomas, 2006). We then compute ΔH, the total governance pressure, and the three Shapley values derived from the six neutralization permutations, which yield $\alpha_{\text{op}}, \alpha_{\text{mod}}, \alpha_{\text{tool}}$ with exact summation to one by construction (Shapley, 1953).

Mechanism diagnostics. We compute override rate, coercion rate, repair depth, default pull, refusal pressure, time to acceptance, and calls per success. We expect ARI components to correlate with these diagnostics in specific ways. Higher α_{tool} should associate with higher coercion and refusal rates, shorter search paths once a candidate falls within validator corridors, and lower default pull divergence because accepted calls adhere closely to pre committed values. Higher α_{mod} should associate with longer search paths in expansive signatures when defaults are weak, higher variance in argument selection, and greater sensitivity to decoding temperature. Higher α_{op} should be visible when canonical prompts are relaxed, which yields larger differences between permissive and evaluated runs that are explained by operator supplied parameters rather than validators.

Outcome models. We estimate two families of models. First, generalized linear models that predict success probability, hard failure probability, and expected time to acceptance using ARI components and schema levers as predictors. Second, hierarchical models that pool across task families while allowing task specific intercepts, which avoids confounding by task hardness differences (Gelman et al., 2013; McElreath, 2020). For interpretability, we use coefficient constrained regressions that keep ARI components on the simplex and report average marginal effects for one point shifts in α shares. We pre register contrasts of interest: the effect of moving from soft to hard defaults on α_{tool} and on refusal pressure, the effect of moving from weak to strong validators on α_{mod} and on repair depth, and the interaction between signature breadth and default hardness on α_{op}.

Robustness and sensitivity. We address three threats. First, baseline dependence, since ΔH and Shapley values can vary with the permissive reference. We repeat the analysis with a second permissive schema and with a common support estimator. Second, path dependence induced by validator order. We either fix rule order and replay it during neutralization permutations or randomize rule order across runs and report interval estimates that include path variability. Third, stochastic tool outputs. We either replay cached responses or report conditional variances and increase sample sizes in the stochastic condition.

Visualization. Each schema receives a compact report that includes a ternary plot of the ARI vector, bar charts for diagnostics, and a Sankey diagram of repair paths from proposal to acceptance. The ternary plot locates a schema on the operator, model, tool simplex. The diagnostic bars reveal which mechanism drove the observed allocation. The Sankey diagram shows how validators and defaults created corridors that channeled proposals toward acceptance, which makes the *regla compilada* legible as a governing device rather than as a mere interface.

Interpretation rules. We distinguish between intended governance and effective governance. Intended governance is what designers state they want, for example tool dominance in safety critical workflows. Effective governance is what ARI and diagnostics reveal. A schema that claims advisory defaults but exhibits low default pull divergence and high coercion rate is effectively tool dominant. Conversely, a schema with expansive signatures and weak validators that still shows low α_{mod} may be silently constrained by hidden cross field dependencies. These discrepancies are central to the thesis that formal structure, not declared purpose, locates authority, which aligns with prior arguments on syntactic legitimacy and the illusion of neutrality created by formal mechanisms (Startari, 2025b; Startari, 2025g).

Hypotheses. We test three families. H1, validator strictness increases α_{tool} and reduces α_{mod}, with detectable increases in coercion and refusal. H2, signature breadth increases α_{mod} when defaults are soft, but increases α_{tool} when defaults are hard, since additional degrees of freedom are absorbed by defaults. H3, hard defaults increase average success in routine tasks and reduce success in edge cases, visible as a higher override rate and a heavier tail in repair depth when novel constraints are present. We evaluate these hypotheses with preregistered tests and report confidence or credible intervals as appropriate.

Link to governance literature. The analysis plan treats the schema as a programmable constitution that encodes authority in advance of semantics, which extends the regulatory perspective that conduct can be channeled by code rather than policy alone (Lessig, 1999; Yeung, 2018). The formalization here moves beyond transparency debates by quantifying who gains control when a *regla compilada* is modified. This provides an empirical bridge to prior work on executable power and syntactic sovereignty where structure, not content, was identified as the decisive substrate of authority (Startari, 2025a; Startari, 2025b; Startari, 2025g).

6. Findings and discussion

This section synthesizes what the Agency Reallocation Index reveals about schema design when the schema is treated as a *regla compilada* that structures permissible actions before any semantic resolution occurs. The discussion integrates quantitative signals from ARI and qualitative traces from repair paths, with emphasis on how defaults, validators, and signature breadth interact to redistribute effective control among operator, model, and tool. Since the core thesis concerns structure as governance, interpretation privileges consistent patterns across task families rather than isolated point estimates. Where magnitudes are mentioned, they reflect stable tendencies observed under deterministic tools and matched prompts, not claims of universal constants.

6.1 Cross-lever effects on ARI

Three patterns recur across factual retrieval, transactional scheduling, code transformation, and multi-tool workflows. First, validator strictness systematically shifts ARI mass toward the tool. When validators move from type and range checks to semantic checks that involve cross argument dependencies, proposals cluster inside narrow corridors. Accepted calls exhibit low default pull divergence, and refusal pressure rises because many trajectories cannot be repaired without relaxing constraints. The net effect is a compression of the model's viable choices, which is visible as a consistent reduction in α_{mod}. Second, default hardness produces an asymmetric response. Soft defaults reduce search costs without strongly affecting ARI, while hard defaults replace model discretion with pre committed values, which increases α_{tool} and reduces α_{mod}. Third, signature breadth amplifies whichever lever dominates. Expansive signatures increase α_{mod} when defaults are soft and validators are weak, since the model makes more autonomous choices. The same breadth increases α_{tool} when defaults are hard or validators are strict, because additional degrees of freedom are absorbed by enforcement and default insertion.

6.2 Mechanisms in traces

Repair path diagrams clarify how governance operates in practice. In tool dominant regimes, early validator events prune the search tree aggressively. Candidates that deviate from default corridors are either coerced into compliance or rejected before tool execution. The number of proposals until acceptance falls once a path enters the corridor, yet overall refusal pressure increases. In model advantaged regimes, proposals vary widely across arguments, repair sequences are longer, and acceptance depends more on decoding policy than on validators. Operator advantaged regimes appear when prompts provide precise parameter vectors, which reduces both the number of repairs and the opportunity for defaults to activate. These patterns align with the argument that schema structure can function as a juridical syntax, since enforcement acts like a procedural court that either admits or rejects proposals according to precompiled rules.

6.3 Edge cases and distributional effects

Hard defaults improve average success on routine tasks but create heavy tails in repair depth once constraints conflict or inputs deviate from the training manifold. The override rate rises, and in some families, accepted calls concentrate on a small subset of parameter combinations that mirror default values. This produces the illusion of stability while concealing brittleness. Weak validators with expansive signatures produce the opposite shape. Success rates may be lower on routine tasks, yet the system exhibits resilience to novel constraints because the model can explore more trajectories. From a governance perspective, designers face a genuine trade off. Tool dominance secures predictable execution at the price of narrow legality, while model autonomy secures adaptability at the price of audit complexity. Operator dominance, finally, requires high quality prompts and is viable where human expertise is high and throughput pressure is moderate.

6.4 Domain profiles and intended governance

Safety critical domains, for example code execution near sensitive resources or irreversible transactions, benefit from high α_{tool}. Validator authority and hard defaults enforce conservative policies and reduce variance once proposals pass the admissibility threshold. However, this configuration should be paired with explicit procedures for handling refusal, since escalation paths are part of governance just as much as admissibility paths. Creative or expert guided domains benefit from higher $\alpha_{\mathrm{op}}$with soft defaults and advisory validators. This exposes real operator intent and keeps schema authority advisory rather than constitutive. Exploratory analysis or planning tasks that benefit from diverse trajectories can tolerate higher α_{mod}, provided that tool side consequences are bounded and that logs capture sufficient traces for auditability. These domain profiles allow teams to set governance targets in ARI space rather than relying on informal judgments about safety, flexibility, or productivity.

6.5 Discrepancy between declared and effective governance

A frequent discrepancy emerges between what designers intend and what the schema actually produces. Teams often claim advisory defaults, yet logs show low divergence between chosen values and default distributions, which indicates de facto compulsion. Similarly, validators described as lightweight often contain cross field constraints that behave like hidden policies. ARI exposes these mismatches by quantifying where agency actually resides. This matters for accountability and for compliance, because it is the effective distribution of control that determines who is responsible when outcomes deviate from expectations. The distinction mirrors broader critiques of transparency that warn against equating published descriptions with operative mechanisms. What counts is the compiled form and its observable effects, not the narrative that accompanies it (Ananny & Crawford, 2018; Yeung, 2018).

6.6 Relation to executable governance

Treating the schema as a *regla compilada* connects these findings to a wider theory of executable power. Prior work argues that syntactic infrastructures can legitimize authority by stabilizing procedures that precede meaning and reference. The ARI results make that claim measurable. When validators and defaults determine which proposals can materialize, the schema becomes a small constitution that allocates the right to decide and the right to execute. Modification of that constitution is therefore a political choice with technical syntax. This reframing clarifies why code level changes can shift institutional accountability without any change in written policy, which is consistent with the view that conduct can be shaped by code and not only by law or organizational rules (Lessig, 1999; Startari, 2025a; Startari, 2025b; Startari, 2025g).

6.7 Practical implications

First, publish target ARI profiles for each domain and validate that implemented schemas meet those targets. Second, audit default pull and refusal pressure continuously, since both are leading indicators of silent policy drift. Third, pair tool dominant schemas with escalation protocols that handle edge cases gracefully rather than masking them as model failures. Fourth, retain replayable traces and stable identifiers for validators, because accountability requires reconstructing the compiled path from proposal to acceptance. These measures convert governance preferences into actionable design parameters and align engineering choices with declared institutional aims.

7. Conclusion and implications

This paper demonstrated that function calling schemas operate as de facto governance by encoding a *regla compilada* that allocates decision rights before any semantic reasoning occurs. Treating the schema as a compiled grammar of authority reframes routine developer choices about signatures, defaults, and validators as institutional choices about who decides, who executes, and who is accountable for outcomes. The Agency Reallocation Index translates this conceptual insight into a measurable vector on the operator, model, tool simplex. Across task families, stricter validators and harder defaults consistently transfer effective control toward the tool, while expansive signatures only increase model autonomy when validators remain weak and defaults are soft. These findings substantiate the thesis that structure can perform governance functions independently of content, and they align with the broad line of argument that code channels conduct in ways that law or policy often cannot match in immediacy (Lessig, 1999; Yeung, 2018; Startari, 2025a; Startari, 2025b; Startari, 2025g).

Three implications follow for design, audit, and institutional accountability. First, schema design should begin with an explicit governance target expressed as a point or region in

ARI space. Tool dominant profiles fit safety critical operations with irreversible consequences, provided that refusal and escalation paths are defined and resourced. Operator dominant profiles fit expert workflows where human intent must remain constitutive and where defaults are advisory. Model advantaged profiles fit exploratory planning or analysis where diversity of trajectories is a benefit and where consequences are bounded. Publishing a target ARI and then verifying that an implemented schema attains it prevents drift from declared to effective governance. In practice, the verification can be automated. A build step can compute entropy reduction, Shapley attributions, and auxiliary diagnostics, then fail the merge if the new schema exits the approved ARI region.

Second, default values should be treated as policy and not as convenience. The empirical traces show that hard defaults act like hidden mandates. When override costs are non trivial, the distribution of accepted calls collapses around default corridors even when documentation labels those defaults as advisory. Auditing default pull and refusal pressure over time exposes policy drift and surfacing of edge cases. Teams should maintain a registry that lists every default, its rationale, the expected override rate, and the owner accountable for changes. The registry should be versioned in the same repository as the schema so that the compiled and the declared policies stay synchronized. This is consistent with the broader view that executable form carries authority and must be governed as such, not as a secondary artifact of implementation (Startari, 2025a).

Third, validators should be written and reviewed as normative rules with operational effect. Cross field and semantic checks encode jurisdictional boundaries that can exclude entire classes of requests. These validators decide what the system is allowed to consider legitimate. When they fail silently or when their effects are not logged, accountability becomes untraceable. The instrumentation required for the Agency Reallocation Index offers a minimal audit trail. Each validator receives a stable identifier, its activation is logged, and repair paths are captured in replayable form. This makes it possible to reconstruct how a proposal became an accepted call and to assign responsibility when the outcome deviates from expectations. The same instrumentation enables red teaming at the level of the compiled rule, not only at the level of output text.

Limitations suggest directions for future research. The entropy based measure depends on support estimation and can be sensitive to sparse spaces, especially in long multi tool workflows. The study mitigates this with smoothing and common support estimates, yet richer estimators may provide tighter bounds. The Shapley attribution presumes that neutralization interventions approximate clean factor separations. In real systems, operator prompts, decoding policy, and schema constraints can interact. Follow up work should explore alternative attribution methods that model interactions directly. The present tasks rely on deterministic tools or recorded ledgers. Many realistic integrations involve stochastic tools and externalities that cannot be replayed. Extending ARI to such environments will require variance aware estimators and controlled synthetic environments that proxy real world uncertainty. Finally, institutional factors influence how operators actually use schemas. Field studies that combine ARI with ethnographic observation could measure how training, incentives, and organizational culture modulate the effective distribution of control.

Two strategic extensions appear especially promising. The first is governance by contract, where service level agreements include target ARI bands and alerting on drift. The second is dynamic schemas with adjustable validator strictness and default hardness that respond to context. For example, a scheduling schema can raise α_{tool}during high risk transaction windows and relax it during exploratory planning. Such context aware governance transforms the schema into a programmable constitution that adapts without losing auditability, provided that every transition is logged and attributable.

The overall conclusion is that function calling schemas are not merely technical scaffolds for structured output. They are small constitutions that partition authority inside model tool ecosystems. The Agency Reallocation Index makes this partition legible and adjustable. By embedding ARI targets into engineering workflows, by treating defaults as policy, and by auditing validators as normative constraints, organizations can align implemented behavior with declared aims. This moves the debate about transparency beyond documentation and into the compiled locus where authority is actually exercised. It also connects the microphysics of interface design with macro questions of institutional responsibility and legitimacy, which reinforces the broader claim that in predictive societies the decisive substrate of power is syntactic and executable, not semantic or declarative (Startari, 2025a; Startari, 2025b; Startari, 2025g).

References (APA 7th)

Ananny, M., & Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society, 20*(3), 973–989. https://doi.org/10.1177/1461444816676645

Chomsky, N. (1965). *Aspects of the theory of syntax*. MIT Press.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Wiley.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis* (3rd ed.). Chapman and Hall.

Lessig, L. (1999). *Code and other laws of cyberspace*. Basic Books.

McElreath, R. (2020). *Statistical rethinking* (2nd ed.). CRC Press.

Montague, R. (1974). *Formal philosophy*. Yale University Press.

Shapley, L. S. (1953). A value for n person games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the theory of games II* (pp. 307–317). Princeton University Press.

Startari, A. V. (2025). *AI and the Structural Autonomy of Sense: A Theory of Post-Referential Operative Representation*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5272361

Startari, A. V. (2025). *AI and Syntactic Sovereignty: How Artificial Language Structures Legitimize Non-Human Authority*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5276879

Startari, A. V. (2025). *Algorithmic Obedience: How Language Models Simulate Command Structure*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5282045

Startari, A. V. (2025). *Ethos Without Source: Algorithmic Identity and the Simulation of Credibility*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5313317

Startari, A. V. (2025). *Executable Power: Syntax as Infrastructure in Predictive Societies*. Zenodo. https://doi.org/10.5281/zenodo.15754714

Startari, A. V. (2025). *TLOC – The Irreducibility of Structural Obedience in Generative Models*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5303089

Startari, A. V. (2025). *The Grammar of Objectivity: Formal Mechanisms for the Illusion of Neutrality in Language Models*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5319520

Startari, A. V. (2025). *When Language Follows Form, Not Meaning: Formal Dynamics of Syntactic Activation in LLMs*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5285265

Startari, A. V. (2025). *The Disconnected Syntactic Authority Theorem: Structure Without Subject, Truth or Consequence*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5275864

Startari, A. V. (2025). *From Obedience to Execution: Structural Legitimacy in the Age of Reasoning Models*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5290698

Startari, A. V. (2025). *Grammars of Power: How Syntactic Structures Shape Authority*. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5337803

Yeung, K. (2018). Algorithmic regulation: A critical interrogation. *Regulation & Governance, 12*(4), 505–523. https://doi.org/10.1111/rego.12158