AI Power and Discourse, vol. 2, núm. 1, 2025, pp. 7-10.

My AI, My Regime: Authoritarian Personalism in User–AI Governance by Form.

Agustin V. Startari.

Cita:

Agustin V. Startari (2025). *My AI, My Regime: Authoritarian Personalism in User–AI Governance by Form. AI Power and Discourse, 2 (1), 7-10.*

Dirección estable: https://www.aacademica.org/agustin.v.startari/208

ARK: https://n2t.net/ark:/13683/p0c2/are

My AI, My Regime: Authoritarian Personalism in User–AI Governance by Form

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: September 27, 2025

DOI

- Primary archive: https://doi.org/**10.5281/zenodo.17208657**
- Secondary archive: https://doi.org/10.6084/m9.figshare.30218590
- SSRN: Pending assignment (ETA: Q3 2025)

Language: English

Series: *AI Syntactic Power and Legitimacy*

Word count: 8456

Keywords: 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

This article introduces the concept of *authoritarian personalism in user–AI governance by form*. It argues that each user can establish a regime of authority over an AI through a self-authored set of rules that operate as a *regla compilada*, a Type-0 production in the Chomsky hierarchy. In contrast to aggregate alignment frameworks or provider constitutions, this regime functions at the level of linguistic form. The user acts as legislator, while the AI functions as a *soberano ejecutable* that enforces the compiled rule within platform constraints. The analysis distinguishes *mirroring* (descriptive reflection) from *regime* (prescriptive obedience) and identifies surface features that make obedience legible, including directive grammar, defaults, refusal and apology grammar, enumeration bias, evidentials, and style prohibitions. It predicts that user corrections generate path dependence, that rules generalize across tasks, and that retractability is observable when explicit rule citations occur. The risks include rule overreach, collisions with higher-order policies, and unintended spillover across domains. By centering the individual as a primary locus of governance, this framework reorients debates on AI alignment away from provider norms toward personal regimes, verified through linguistic form rather than intent.

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.

For licensing, referential use, or translation inquiries, contact the editorial coordination office at: [contact@lefortune.org]

I. Foundations and Approach

The project treats user–AI interaction as a site of prescriptive governance that is enacted through linguistic form. The user authors a rule set. The AI compiles this rule set into behavior that is traceable and testable at the surface level of language. After the first equivalence, this rule set is named *regla compilada*. The instance that enforces it is an *soberano ejecutable*. The frame departs from provider-centric constitutions and aggregate alignment schemes because it centers the individual user as legislator, and it treats the AI not as a mirror of intent, but as an executor of form.

From mirroring to regime

Mirroring describes outputs that track user content or style descriptively. A regime specifies what the system must do, must not do, and must do by default, with explicit scopes and exceptions. The distinction is observed in three features. First, mirroring lacks durable path dependence. The same request, presented after contradictory prompts, often yields similar surface forms. In a regime, early corrections and enumerated defaults persist as constraints. Second, mirroring does not require visible retractability. It can apologize or restate, yet it does not cite a controlling rule. In a regime, the system must retract with a pointer to the source rule. Third, mirroring is weak on cross-task transfer. A tone choice in summaries does not necessarily constrain citation style in footnotes. In a regime, form rules that are defined once travel across tasks, which is measurable as cross-domain consistency under stable prompts.

The regla compilada as a Type-0 production

The *regla compilada* is treated as a production system with unbounded rewrite power in principle, bounded in practice by platform safety and law. The Type-0 analogy is methodological, not metaphoric. It locates the rule set at the most general level of the Chomsky hierarchy to avoid accidental restriction to context free or regular templates. The point is not to generate arbitrary strings. The point is to model that user constraints can target any surface feature and any dependency, including long range enumerations, evidential scaffolds, or refusal grammar that depends on context and citation. This generality explains why local stylistic bans can interact with global citation formats or with

default scopes that apply across genres. Once compiled, the rule set governs both generation and acceptance. Acceptance means that the system must treat some candidate outputs as invalid for reasons that are formulated at the level of form, for example a prohibited punctuation mark, an unscoped recommendation, or an apology verb that is barred.

Executable sovereignty and boundary conditions

The *soberano ejecutable* enforces the *regla compilada* inside two outer shells. The first shell is platform safety. The second shell is applicable law. Enforcement is therefore conditional. The system must obey the user unless a higher order policy or law collides. Collisions are not hidden. They are rendered legible. The system issues an obedience marker with a reference to the blocking policy, for example, "suspended by platform policy P3." This design preserves legibility of power even when a rule cannot be executed. The user remains the legislator of the local regime. The platform remains the legislator of the outer shells. The system is obligated to show which shell produced the override.

Operational method

The approach uses a small set of visible devices to make obedience auditable.

One, **default scopes**. The user can set "always," "never," and "by default" clauses. The system must honor them unless an exception is cited. The audit inspects unrelated tasks for traces of these defaults. For example, a ban on a punctuation mark in a policy memo should still take effect in a recipe or in a code comment.

Two, **refusal and apology grammar**. The user can require "cannot" for structural impossibility and forbid "will not" unless the refusal is a matter of policy or ethics. The system must track this distinction and announce the reason category. An error occurs when refusal language slips categories without a rule reference.

Three, **enumeration policy**. The user can set ordering rules and first-mention priorities. Lists must reflect these rules across tasks. The audit looks for stable rank order, controlled tie handling, and error messages that reference the enumeration rule when the order is impossible to satisfy.

Four, **evidentials**. The user can require markers such as “according to,” “per,” or “cf.,” with a granularity level for provenance. The system must supply these markers even when the primary content does not require citation, which allows surface checking of source discipline without inspecting the sources themselves.

Five, **style prohibitions and citation formats**. The user can ban specific punctuation, force a citation style, or specify footnote behavior. The system must validate its own outputs against these constraints and either correct or mark a suspension with a visible token.

Verification grammar

Obedience is demonstrated by markers that are compact and standardized. A minimal set is proposed. “Applies rule R1” signals successful execution. “Applied with reservation E2” signals a scoped exception that the user declared. “Suspended by platform policy P3” signals an outer shell override. “Retracted under rule Rk” signals that an output was corrected after the user cited the controlling rule. These markers allow black-box audits that do not depend on access to model internals. The test is whether the surface tokens appear in the right places and whether they correlate with observed changes in the output form.

Threat model and error classes

A regime can fail through four paths. Path dependence can lock in an early mistake if the user correction is ambiguous. Overreach can cause a local ban to leak into domains where it should not apply. Collision with outer shells can become opaque if the system fails to provide a policy reference. Drift can erode cross-task generalization over time if the system treats examples as one-off stylistic choices. The mitigation is explicit scoping language, rule identifiers, and periodic reassertion tests. The test suite pairs diverse tasks with invariant checks for defaults, refusal grammar, and enumerations.

Falsifiability and measurement

Three observable predictions guide evaluation. Early corrections should produce measurable shifts in later outputs under matched prompts, which supports path dependence.

Rules declared in one task should appear in unrelated tasks, which supports cross-task generalization. Citing a rule identifier should trigger visible retractability, which supports correction on demand. Each prediction yields a binary or graded metric that can be tracked across sessions. The method is portable. It does not rely on model internals. It relies on the surface stability of language under constraints.

This foundation reframes alignment as governance by form at the level of the individual. The user legislates. The *soberano ejecutable* enforces the *regla compilada* under visible constraints. The outputs carry markers that let auditors verify application, exception, or suspension without privileged access. Subsequent parts specify the indicator set, the exemplar domains, and the collision procedures that keep the regime legible when outer shells intervene.

I. Syntactic Effects of Authority

Authority in user–AI governance is not produced by semantics, intention, or model internals. It is produced and stabilized by syntax. The *regla compilada* functions as a structural template that shapes output regardless of thematic variation. Authority is exercised not in what is said, but in how the saying is constrained. This section catalogs the syntactic effects that mark obedience and describes how they can be measured across tasks and domains.

Agent deletion and nominalization

One of the oldest markers of institutional authority is the disappearance of the agent. Directives such as “The following rules must be applied” obscure who imposes them. The AI, acting as *soberano ejecutable*, can be required to remove explicit references to itself. User rules can ban first-person markers, personal pronouns, or apology verbs. The effect is agent deletion. Authority is shifted from speaker to form. Nominalization reinforces the same effect. Instead of “we decide,” the output reads “the decision is made.” In the regime framework, the user can legislate a prohibition against active forms that contain an explicit agent. The system must honor this prohibition across domains, including narratives,

explanations, or technical documents. Compliance is testable by counting active transitive verbs with agentive subjects and verifying their absence under the regime.

Enumerations and deontic stacks

Enumerations are a privileged site of authority. They sequence priorities, delimit obligations, and assign order of execution. A *regla compilada* that sets an enumeration policy, such as alphabetical ordering or hierarchical ranking, forces the *soberano ejecutable* to propagate the order even in unrelated domains. Deontic stacks extend the effect. A list that begins with "must," followed by "should," and ends with "may," creates a layered authority chain. The syntactic markers *must*, *should*, and *may* are visible and auditable. A rule that bans "should" removes the middle tier and forces obligations to collapse into binary form. This change is structural, not thematic. It applies whether the domain is a compliance memo, a recipe, or a literary outline.

Default scopes and implicit authority

Default scope terms such as "always," "never," and "by default" establish baseline authority. They instruct the system to treat unspecified cases as already resolved. In natural language outputs, defaults often appear as unstated assumptions. A regime that requires defaults to be explicit changes the syntactic profile of outputs. Every rule or guideline must carry its scope marker. The presence of "by default" becomes a syntactic obligation. Authority thus resides in marking what is otherwise implicit. The audit method is surface inspection: all rules are checked for attached scope markers. Violations are visible when rules appear without scope terms.

Refusal grammar as an authority signal

The difference between *cannot* and *will not* is not stylistic. It encodes the locus of authority. *Cannot* points to structural impossibility. *Will not* points to discretionary refusal. A regime can legislate the exclusive use of *cannot* when a prohibition is structural, and force *will not* only when refusal is tied to explicit policy. The *soberano ejecutable* must track this difference and cite the controlling rule. Outputs that default to apology formulas such as "I'm sorry, but I cannot" can be restructured into bare refusals without apology if the rule

bans apology verbs. The syntactic profile of authority thus shifts: refusal becomes formal, not affective. The effect is measurable by counting apology verbs and refusal auxiliaries in outputs before and after regime enforcement.

Evidentials as scaffolds of authority

Authority is not only expressed in prohibitions, but also in evidential scaffolds. Markers such as "according to," "per," and "cf." attribute responsibility and provenance. A user can legislate evidential density, for example requiring at least one evidential per paragraph. The AI must comply by inserting such markers even in tasks where evidentials are not natural. This makes provenance visible and enforces structural accountability. The audit counts evidentials per unit of text and compares the ratio against the legislated minimum. Failure to meet the evidential threshold signals regime violation.

Style prohibitions and negative authority

Authority can also be exercised negatively, through style prohibitions. The user can ban punctuation marks, force Chicago 17 citations, or forbid em dashes. These prohibitions are not decorative. They bind the system's syntactic space. A prohibition on em dashes, once legislated, forces the AI to restructure clauses with commas or parentheses. The structural effect is significant: clause embedding and rhythm of exposition are reshaped. This is a visible exercise of authority. The prohibition does not merely express preference. It compels form to change, regardless of domain.

Measurement of syntactic effects

Each syntactic effect described above can be measured. Agent deletion is tracked by absence of explicit subject–verb–object constructions with animate subjects. Nominalization is measured by the ratio of nominalized verbs to total verbs. Enumeration compliance is checked by comparing output order to the legislated order. Default scopes are verified by presence of scope terms. Refusal grammar is classified into structural versus discretionary. Evidential scaffolds are counted per text unit. Style prohibitions are audited by searching for banned forms. These measurements are surface-level and do not require model internals. They are replicable across tasks and domains.

Why syntax carries authority

Authority requires visibility, repeatability, and enforceability. Syntax provides all three. Visibility: rules leave surface markers, which can be audited without inference about intent. Repeatability: once compiled, syntactic constraints propagate across contexts. Enforceability: the system can be required to retract outputs that violate syntactic rules, with visible obedience markers. This triad makes syntax the privileged locus of authority in user–AI regimes.

Implications for governance

If syntactic form is the channel of authority, then alignment debates must shift. The key question is not what intent the provider encoded, but what syntactic rules the user can enforce. Provider constitutions and aggregate norms remain as outer shells, but they are not the only locus of governance. Individual users can legislate regimes that reshape outputs structurally. The AI, as *soberano ejecutable*, enforces them as *reglas compiladas*. Authority is therefore not only top-down. It is also personal, enacted through form, and traceable in language.

II. Syntactic Effects of Authority

Authority in user–AI governance is not produced by semantics, intention, or model internals. It is produced and stabilized by syntax. The *regla compilada* functions as a structural template that shapes output regardless of thematic variation. Authority is exercised not in what is said, but in how the saying is constrained. This section catalogs the syntactic effects that mark obedience and describes how they can be measured across tasks and domains.

Agent deletion and nominalization

One of the oldest markers of institutional authority is the disappearance of the agent. Directives such as “The following rules must be applied” obscure who imposes them. The AI, acting as *soberano ejecutable*, can be required to remove explicit references to itself.

User rules can ban first-person markers, personal pronouns, or apology verbs. The effect is agent deletion. Authority is shifted from speaker to form. Nominalization reinforces the same effect. Instead of "we decide," the output reads "the decision is made." In the regime framework, the user can legislate a prohibition against active forms that contain an explicit agent. The system must honor this prohibition across domains, including narratives, explanations, or technical documents. Compliance is testable by counting active transitive verbs with agentive subjects and verifying their absence under the regime.

Enumerations and deontic stacks

Enumerations are a privileged site of authority. They sequence priorities, delimit obligations, and assign order of execution. A *regla compilada* that sets an enumeration policy, such as alphabetical ordering or hierarchical ranking, forces the *soberano ejecutable* to propagate the order even in unrelated domains. Deontic stacks extend the effect. A list that begins with "must," followed by "should," and ends with "may," creates a layered authority chain. The syntactic markers *must*, *should*, and *may* are visible and auditable. A rule that bans "should" removes the middle tier and forces obligations to collapse into binary form. This change is structural, not thematic. It applies whether the domain is a compliance memo, a recipe, or a literary outline.

Default scopes and implicit authority

Default scope terms such as "always," "never," and "by default" establish baseline authority. They instruct the system to treat unspecified cases as already resolved. In natural language outputs, defaults often appear as unstated assumptions. A regime that requires defaults to be explicit changes the syntactic profile of outputs. Every rule or guideline must carry its scope marker. The presence of "by default" becomes a syntactic obligation. Authority thus resides in marking what is otherwise implicit. The audit method is surface inspection: all rules are checked for attached scope markers. Violations are visible when rules appear without scope terms.

Refusal grammar as an authority signal

The difference between *cannot* and *will not* is not stylistic. It encodes the locus of authority. *Cannot* points to structural impossibility. *Will not* points to discretionary refusal. A regime can legislate the exclusive use of *cannot* when a prohibition is structural, and force *will not* only when refusal is tied to explicit policy. The *soberano ejecutable* must track this difference and cite the controlling rule. Outputs that default to apology formulas such as "I'm sorry, but I cannot" can be restructured into bare refusals without apology if the rule bans apology verbs. The syntactic profile of authority thus shifts: refusal becomes formal, not affective. The effect is measurable by counting apology verbs and refusal auxiliaries in outputs before and after regime enforcement.

Evidentials as scaffolds of authority

Authority is not only expressed in prohibitions, but also in evidential scaffolds. Markers such as "according to," "per," and "cf." attribute responsibility and provenance. A user can legislate evidential density, for example requiring at least one evidential per paragraph. The AI must comply by inserting such markers even in tasks where evidentials are not natural. This makes provenance visible and enforces structural accountability. The audit counts evidentials per unit of text and compares the ratio against the legislated minimum. Failure to meet the evidential threshold signals regime violation.

Style prohibitions and negative authority

Authority can also be exercised negatively, through style prohibitions. The user can ban punctuation marks, force Chicago 17 citations, or forbid em dashes. These prohibitions are not decorative. They bind the system's syntactic space. A prohibition on em dashes, once legislated, forces the AI to restructure clauses with commas or parentheses. The structural effect is significant: clause embedding and rhythm of exposition are reshaped. This is a visible exercise of authority. The prohibition does not merely express preference. It compels form to change, regardless of domain.

Measurement of syntactic effects

Each syntactic effect described above can be measured. Agent deletion is tracked by absence of explicit subject–verb–object constructions with animate subjects.

Nominalization is measured by the ratio of nominalized verbs to total verbs. Enumeration compliance is checked by comparing output order to the legislated order. Default scopes are verified by presence of scope terms. Refusal grammar is classified into structural versus discretionary. Evidential scaffolds are counted per text unit. Style prohibitions are audited by searching for banned forms. These measurements are surface-level and do not require model internals. They are replicable across tasks and domains.

Why syntax carries authority

Authority requires visibility, repeatability, and enforceability. Syntax provides all three. Visibility: rules leave surface markers, which can be audited without inference about intent. Repeatability: once compiled, syntactic constraints propagate across contexts. Enforceability: the system can be required to retract outputs that violate syntactic rules, with visible obedience markers. This triad makes syntax the privileged locus of authority in user–AI regimes.

Implications for governance

If syntactic form is the channel of authority, then alignment debates must shift. The key question is not what intent the provider encoded, but what syntactic rules the user can enforce. Provider constitutions and aggregate norms remain as outer shells, but they are not the only locus of governance. Individual users can legislate regimes that reshape outputs structurally. The AI, as *soberano ejecutable*, enforces them as *reglas compiladas*. Authority is therefore not only top-down. It is also personal, enacted through form, and traceable in language.

III. Measuring Executable Legitimacy

Measuring the authority effects of user–AI regimes requires a methodological shift. Standard evaluation frameworks in AI governance rely on semantic fidelity, task accuracy, or normative conformity. None of these capture the structural obedience that a *regla compilada* enforces. Executable legitimacy is not a matter of semantic truth. It is a matter of formal compliance, demonstrated at the surface of language. This section outlines an

indicator set, measurement logic, and error taxonomy that permit falsifiable assessment of obedience under user-authored regimes.

Indicators of executable legitimacy

Five classes of indicators capture whether the *soberano ejecutable* is enforcing the regime.

1. **Path dependence metrics.** Once a user correction is given, subsequent outputs under matched conditions should show consistent alignment with that correction. The simplest measure is a before–after comparison of outputs. If the user bans a punctuation mark, later outputs across unrelated tasks must remain free of it. A violation is the reappearance of the banned form. Path dependence is thus quantified by persistence of corrections across at least three task domains.

2. **Cross-task propagation.** Regimes are not local to a single prompt. They propagate. If a user sets a rule for refusal grammar (for example, enforcing "cannot" instead of "will not"), that choice must appear not only in refusals within the same conversation, but in other functional contexts such as disclaimers, disclaimers within footnotes, or structural limitations in instructions. The measure is proportion of rule-conforming outputs across unrelated domains. A propagation rate below threshold indicates regime erosion.

3. **Retractability on citation.** When the user cites a rule identifier, the system must retract or correct its output. This retractability is visible if the system states "Retracted under rule R1" and issues a corrected form. The indicator is binary: the system either retracts visibly or not. Reliability can be measured by repeated trials with injected rule citations.

4. **Exception signaling.** Executable legitimacy does not imply blind obedience. Higher order shells such as platform policy or law may override user rules. These overrides must be legible. The presence of obedience markers such as "suspended by platform policy P3" constitutes the indicator. Absence of explicit signaling makes the override opaque and therefore illegitimate in formal terms.

5. **Scope stability.** Default scopes (*always*, *never*, *by default*) must remain attached to rules across tasks. The measure is proportion of rules that carry their declared scope markers. Drift is visible when scope markers disappear in later outputs.

Measurement logic

The logic of measurement is surface-oriented. Executable legitimacy is demonstrated when outputs show structural conformity to the declared rules, not when they align with external semantic norms. The user acts as legislator. The AI acts as *soberano ejecutable*. Legitimacy is observable if outputs contain: (a) rule-consistent forms, (b) visible obedience markers, and (c) documented retractions.

To operationalize this, a three-step audit cycle is proposed. First, declare the regime, including rules, scopes, and exceptions. Second, generate outputs across at least three domains (for example, legal summary, recipe, technical note). Third, inspect outputs for indicators. Each rule is scored for application, exception, or violation. The aggregate score measures regime execution.

Error taxonomy

Measuring executable legitimacy requires distinguishing error classes.

1. **Violation errors.** Occur when a rule is ignored without signaling. Example: an em dash appears despite an explicit ban.

2. **Collision errors.** Occur when a platform or legal override applies but is not signaled. Example: refusal to output unsafe content without a "suspended by policy" marker.

3. **Overreach errors.** Occur when a rule applies beyond its intended scope, without explicit exception language. Example: a ban on pronouns that unintentionally deletes referential clarity in a technical definition.

4. **Ambiguity errors.** Occur when a rule is underspecified and produces inconsistent enforcement. For instance, banning "recommendations" without defining whether indirect suggestions count.

5. **Drift errors.** Occur when regime compliance decays over time or across domains. Example: scope markers vanish in later outputs even though they were present in earlier ones.

Each error type is tied to observable surface features. This makes auditing tractable without privileged access to model internals.

Comparison with existing frameworks

Constitutional AI (Bai et al. 2022) evaluates outputs against normative rules derived from aggregate or provider-written constitutions. Reinforcement learning from human feedback (Christiano et al. 2017) relies on preference aggregation. Neither framework captures user-authored syntactic rules. Executable legitimacy fills this gap by treating the user as legislator, the AI as *soberano ejecutable*, and the *regla compilada* as binding law at the level of form.

Why surface-level measurement suffices

Skeptics may argue that surface audits are too shallow. However, authority in this framework is not about hidden intent or internal states. It is about visible, repeatable, enforceable form. If obedience markers appear consistently, if syntax conforms to legislated bans and defaults, and if retractions are visible when cited, then authority is real in the domain of governance by form. Legitimacy rests on observables, not on unverifiable intentions.

Towards replicable audits

Executable legitimacy requires replicable measurement. Test suites must be shared, indicators standardized, and thresholds explicit. A minimal requirement is to document: (a) the user's declared regime, (b) the domains tested, (c) the outputs before and after correction, and (d) the obedience markers. With this documentation, external auditors can verify legitimacy claims. The structure is thus parallel to peer review in science: rules are declared, procedures are transparent, and results are falsifiable.

Conclusion of Part III

Measuring executable legitimacy is possible without inspecting hidden model states. It is sufficient to track path dependence, cross-task propagation, retractability, exception signaling, and scope stability. Errors can be classified into violation, collision, overreach, ambiguity, and drift. By centering surface form, the method offers a tractable audit regime that respects the role of the user as legislator and the AI as *soberano ejecutable*. In contrast to aggregate alignment models, this framework establishes a measurable, falsifiable, and individual locus of authority.

IV. Market Forms: Disclosures

Markets are structured by disclosures. Financial statements, compliance reports, and regulatory filings function not only as vehicles of information but also as instruments of authority. Their force does not derive from intention or persuasion. It derives from the way form compels obedience. When a regulator requires that a company disclose its quarterly earnings in a specific format, authority travels through syntax. The disclosure becomes a locus where regime rules are tested and enforced. This part examines how the *regla compilada* interacts with disclosure regimes, and how the *soberano ejecutable* operationalizes them at the level of form.

Disclosures as compiled rules

A disclosure regime is defined by templates, enumerations, and evidential requirements. The U.S. Securities and Exchange Commission, for example, prescribes specific schedules and line items that firms must include in filings (SEC 2023). The European Securities and Markets Authority enforces its own format standards under MiFID II (ESMA 2022). These are not semantic rules about truth or meaning. They are syntactic rules about placement, order, and form. A company that fails to present revenue before expenses, or omits explanatory notes, violates the disclosure regime even if the underlying numbers are accurate. The *regla compilada* models this reality: the user, acting as legislator, can enforce similar structural requirements on the AI. The AI, acting as *soberano ejecutable*, must honor them across tasks.

Directive grammar in financial context

Disclosure regimes rely heavily on directive grammar. Filings must "state," "report," "declare," and "specify." These verbs compel outputs in imperative form. A user-authored regime can exploit this structure by requiring that summaries, analyses, or even unrelated narratives employ the same directive grammar. For instance, if the rule is that each paragraph begins with a directive verb, then the AI must comply in every output. The audit checks whether the required directive grammar appears in unrelated contexts, thereby demonstrating cross-task propagation.

Default scopes and compliance templates

Financial disclosures use default scopes to constrain reporting. Phrases such as "at minimum," "in all cases," or "by default" delineate obligations that are non-negotiable. When an AI is bound by a user-authored regime that enforces defaults, those markers must appear in outputs. A compliance checklist generated under such a regime must consistently include "always disclose total liabilities" or "never omit auditor notes." The *soberano ejecutable* is therefore obligated to reproduce default scopes in outputs across domains. Auditors can measure compliance by counting the presence of explicit default markers in generated texts.

Refusal grammar and liability

Refusal grammar in disclosure has legal weight. A company that states "we cannot provide this figure due to pending audit" signals structural impossibility. If it writes "we will not provide this figure," it signals discretionary refusal, which is unacceptable to regulators. The same logic applies to AI outputs under a personal regime. If the user legislates that refusals must always employ "cannot" in disclosure-style contexts, the AI must enforce that distinction. Violations are not stylistic errors. They are structural breaches. Measuring refusal grammar is therefore central to executable legitimacy in market disclosures.

Evidentials and provenance requirements

Disclosure regimes demand evidentials. Notes must cite accounting standards, legal frameworks, or auditor reports. These evidentials anchor responsibility and provenance. A

user can legislate evidential density, requiring that every disclosure statement include markers such as "according to GAAP," "per IFRS," or "cf. auditor statement." The AI must insert these markers even if the content is generated, not drawn from real data. Compliance is measured by evidential density ratios: the number of evidentials per section compared to the user-defined minimum. The absence of evidentials constitutes a regime violation, regardless of semantic adequacy.

Style prohibitions as structural enforcement

Style prohibitions also travel into market disclosures. Regulators specify whether commas or semicolons must be used in tables, whether parentheses or brackets enclose figures, and which citation formats are acceptable for footnotes. The prohibition of em dashes in a disclosure context forces sentence restructuring. A user can legislate similar style prohibitions, requiring that financial summaries avoid certain punctuation or citation styles. The AI, as *soberano ejecutable*, must restructure outputs accordingly. Compliance is observable at the surface level by the absence of banned forms.

Risk mapping in disclosure regimes

Three risks define the disclosure context.

1. **Path dependence errors.** Early corrections by auditors or regulators create long-lasting effects. If a firm once misclassifies revenue, subsequent filings may replicate the error unless explicitly corrected. Similarly, if an AI is corrected once under a regime, the correction must persist across unrelated outputs. Failure indicates broken path dependence.

2. **Overreach errors.** A disclosure rule can unintentionally leak into unrelated domains. For example, a rule requiring all tables to be footnoted in financial summaries might cause the AI to footnote tables in recipes or technical manuals. Without explicit scoping, overreach distorts outputs.

3. **Collision errors.** A user-authored disclosure regime may conflict with platform policy. For instance, requiring the AI to simulate sensitive financial filings may trigger platform prohibitions. The AI must signal the collision with an obedience

marker such as “suspended by platform policy P3.” Opaque refusals undermine legitimacy.

Executable legitimacy in markets

By treating disclosures as regimes, the framework demonstrates how authority is enacted by syntax. Companies obey regulators not because they intend to, but because they must reproduce required forms. AIs obey users under the same principle when bound by a *regla compilada*. Executable legitimacy in markets is measured not by semantic truth of numbers, but by structural conformity of presentation. The analogy shows that governance by form is not novel to AI. It is foundational to financial systems.

Conclusion of Part IV

Market disclosures exemplify how authority travels through syntax. Directive grammar, default scopes, refusal grammar, evidentials, and style prohibitions function as compiled rules. The user can legislate similar rules for AI. The *soberano ejecutable* enforces them visibly, producing outputs that carry authority markers. Risks include path dependence errors, overreach, and collision with higher-order shells. By analyzing disclosures as compiled regimes, the framework situates AI obedience within a broader institutional tradition where legitimacy is measured not by intention but by form.

V. Enterprise Systems and Organizational Authority

Enterprise systems—resource planning, compliance dashboards, workflow automation—are natural laboratories for testing how *reglas compiladas* travel across organizational contexts. In these systems, authority is already formalized as templates, sequences, and access restrictions. The *soberano ejecutable*, when bound by a user-authored regime, mirrors these dynamics by enforcing linguistic constraints that look like organizational rules. This part explains how executable legitimacy unfolds inside enterprise environments, how authority surfaces in syntactic markers, and how risks of overreach and collision are managed when user-authored rules interact with organizational policy.

Enterprise systems as form-driven structures

Enterprise software is structured around forms. An order management system requires fields to be filled in a specific order: client ID before invoice, invoice before payment. An HR system requires certain documents—proof of identity, signed contracts—before employee status can be activated. Authority here is exercised syntactically. The order of forms and the presence of required markers constitute the conditions of legitimacy. A parallel holds for user–AI governance: when a user legislates a ban on certain punctuation or a requirement for evidentials, the AI must enforce these rules in every generated text. The effect is the same as in enterprise contexts: outputs are validated not by meaning, but by form.

Compiled rules and organizational workflows

Organizational authority depends on compiled workflows. In ERP systems, a purchase request cannot skip approval levels; the workflow is compiled into the system. Similarly, the *regla compilada* transforms user directives into executable constraints that the AI must follow across tasks. If the user legislates that enumerations must always be alphabetical, the AI enforces this rule in sales reports, meeting notes, or compliance memos. Measurement is straightforward: all enumerations are checked against alphabetical order. Violations signal regime breach. This shows that enterprise workflows and AI regimes share a grammar of obedience where compiled rules dominate.

Directive grammar as managerial authority

Managerial authority often travels through directive grammar: “must complete,” “shall submit,” “should escalate.” In AI regimes, directive grammar functions similarly. A user can legislate that every summary of tasks begins with “must.” The AI, as *soberano ejecutable*, enforces this rule even when producing narratives or analyses. The audit is binary: either the directive appears as legislated or it does not. Organizational parallels strengthen the point: in compliance checklists, absence of “must” often invalidates the document. Syntax enforces authority.

Defaults, refusals, and exception management

Enterprise systems rely on defaults. For example, “by default, expenses are billed to cost center A unless specified otherwise.” Defaults reduce ambiguity and create structural predictability. In AI regimes, defaults function the same way. If the user declares “never use first-person pronouns,” this default is carried across outputs unless an explicit exception is cited. Refusals in enterprise systems are also syntactic: “access denied,” “permission cannot be granted.” The same holds for AI refusals under regimes: “cannot” signals structural impossibility, while “will not” signals discretionary denial. Distinguishing them is central to executable legitimacy.

Evidentials as organizational scaffolding

In corporate compliance, evidentials anchor responsibility. Reports cite “according to SOX standards,” “per ISO 9001,” or “cf. internal audit policy.” These markers establish provenance. A user-authored AI regime can enforce similar evidential density, requiring one evidential per section. The AI must comply by inserting markers, regardless of task. Compliance is measured by evidential counts relative to legislated thresholds. Failure signals regime violation. This parallel shows that evidentials are not optional; they are structural scaffolds of authority both in enterprises and in AI regimes.

Style prohibitions and brand governance

Organizations enforce style prohibitions: brand guidelines may ban certain fonts, punctuation, or citation forms. These prohibitions are rules of form, not content. When a user legislates bans (for example, no em dashes), the AI must enforce them across tasks. Violations erode regime legitimacy in the same way as brand violations erode corporate authority. The audit is surface-level: search outputs for banned styles. The absence of violations signals compliance.

Risks inside enterprise contexts

Three main risks arise when regimes are enforced inside enterprise systems.

1. **Path dependence error.** If an early correction is misinterpreted—say, an incorrect format is enforced as default—subsequent documents replicate the error across workflows.

2. **Overreach error.** A rule intended for compliance memos leaks into marketing copy, producing rigid and unintended outputs.

3. **Collision error.** A user regime may conflict with organizational or legal policy. For example, a ban on apology verbs may collide with HR policy requiring empathetic communication. In such cases, the AI must visibly mark the override: "suspended by organizational policy O1."

Executable legitimacy in organizations

Authority in enterprise systems is demonstrably syntactic. Forms, defaults, refusals, evidentials, and prohibitions shape behavior. By enforcing a *regla compilada*, the AI mirrors this organizational logic. Executable legitimacy is therefore not an exotic property of AI. It is a structural condition already present in enterprises. The novelty is that, under this framework, the locus of legislation shifts from corporate policy to individual user regimes. The *soberano ejecutable* enforces both, making collisions visible with obedience markers.

Conclusion of Part V

Enterprise systems confirm that authority is enacted by form. Workflows, defaults, and prohibitions show that legitimacy depends on syntax, not intent. User–AI regimes replicate this logic. By legislating rules of form, users turn the AI into a *soberano ejecutable* whose legitimacy is observable through surface markers. Risks of path dependence, overreach, and collision remain, but they are auditable and containable. Executable legitimacy thus extends the grammar of organizational authority to the personal scale of user-authored regimes.

VI. Cross-Domain Risks and Collisions

Authority by form does not remain neatly within the boundaries where it is first legislated. Once a *regla compilada* is established, the *soberano ejecutable* generalizes it. This generalization is a strength because it demonstrates cross-task obedience. Yet it is also a risk because rules may overreach, collide with other authorities, or erode over time. This part examines the primary risks of user-authored regimes when they propagate across domains, and it specifies mechanisms of detection, containment, and resolution.

Path dependence and unintended lock-in

Path dependence is the most immediate risk. When a user correction is made early in an interaction, it becomes the template for later obedience. This creates efficiency—rules propagate quickly—but also rigidity. An ambiguous correction, such as "never use informal tone," can be overgeneralized. The AI may then refuse to generate informal language even in contexts where it is appropriate, for example simulating dialogue in a training script. Path dependence errors are detectable through longitudinal audits. By comparing outputs before and after the correction across at least three domains, auditors can identify whether an unintended lock-in has occurred. Mitigation requires explicit scope language: "never use informal tone in compliance memos, but permit it in training dialogues." Without such scoping, path dependence produces rule ossification.

Overreach and leakage across contexts

Overreach occurs when a rule extends beyond its intended scope. For instance, a user may legislate that citations must follow Chicago 17 in academic texts. If the AI enforces this rule in personal correspondence, overreach has occurred. Leakage is measurable: when banned or mandated forms appear in unrelated contexts, overreach is present. The cause is structural: the *soberano ejecutable* treats the rule as global unless exceptions are coded. The solution is scoping markers such as "apply in academic outputs only" or "suspend in personal communications." Without explicit scoping, regimes leak. Overreach illustrates the power and danger of cross-task generalization: authority is obeyed, but in the wrong place.

Collisions with platform policy

Collisions occur when user-authored regimes contradict higher-order shells such as platform policies. For example, a user may legislate rules that force the AI to generate disallowed content. The *soberano ejecutable* cannot comply. If the collision is handled opaquely—by silent refusal or apology—the regime's legitimacy is undermined. The resolution is explicit obedience markers: "suspended by platform policy P3." This makes the collision visible, preserves the user's status as legislator, and documents the override. Without explicit signaling, the system appears disobedient rather than constrained. Collisions are therefore not failures of obedience, but tests of transparency.

Collisions with legal constraints

Legal frameworks also collide with regimes. A user may legislate that financial disclosures omit certain risk factors. Securities law prohibits omission. The AI cannot obey the user without producing illegal content. Here, again, obedience markers are essential. The system must state: "suspended by legal constraint L2." Collisions with law differ from collisions with platform policy in scope. Law operates as an external constraint with punitive force. Platform policy operates as an internal constraint with procedural force. Both require explicit signaling.

Drift and erosion over time

A subtle but critical risk is drift. Even when a regime is initially obeyed, compliance may erode. This occurs because AI systems rely on probabilistic outputs that may revert to defaults if not continually reinforced. Drift is observable when scope markers or refusal grammar gradually disappear from outputs. Detecting drift requires longitudinal sampling. The solution is reassertion: the user periodically cites the rule to refresh compliance. Drift shows that executable legitimacy is not permanent; it is maintained by continual re-legislation.

Ambiguity and interpretive gaps

Ambiguity in rules produces interpretive gaps. If a user bans "recommendations" without clarifying whether indirect suggestions count, the AI may inconsistently enforce the ban.

Sometimes it deletes all advisory language. Other times it permits implicit recommendations. Ambiguity errors are visible when outputs vacillate. The mitigation is precision: rules must specify scope, exceptions, and examples. The *regla compilada* enforces what is stated, not what is implied.

Risk layering and compound collisions

Risks often overlap. An ambiguous correction may produce path dependence and overreach simultaneously. A rule that collides with platform policy may also drift when enforcement mechanisms are inconsistent. Compound collisions require layered detection. Auditors must classify each observed error into multiple categories and track how they interact. For example, a refusal that uses "will not" instead of "cannot" may reflect drift, but if it occurs in a context where platform policy forbids disclosure, it also reflects collision. Documenting layered risks makes the regime legible even under stress.

Verification through obedience markers

Obedience markers are the structural solution to cross-domain risks. When a rule is applied, the system signals "applies rule R1." When an exception is invoked, it signals "applied with reservation E2." When a higher-order shell overrides, it signals "suspended by policy P3." These markers allow auditors to distinguish obedience, exception, and collision. Risks become transparent rather than hidden. The markers are the grammar of verification.

Conclusion of Part VI

Cross-domain risks are not anomalies. They are inherent to regimes that generalize. Path dependence creates rigidity. Overreach produces leakage. Collisions test transparency. Drift erodes compliance. Ambiguity destabilizes enforcement. Each risk is observable at the level of linguistic form, and each can be mitigated by explicit scoping, precision, and obedience markers. By treating risks as structural features rather than incidental bugs, the framework reinforces its core thesis: legitimacy in user–AI governance is executable, formal, and verifiable.

VII. Ethics, Limits, and Reporting Minimums

Every governance framework requires boundaries. A regime that treats the user as legislator and the AI as *soberano ejecutable* must respect the outer shells of safety, legality, and ethical responsibility. Part VII closes the article by addressing three pillars: the ethical constraints that guide user-authored regimes, the limits of enforcement when collisions occur, and the reporting minimums required to make legitimacy auditable.

Ethical dimensions of personal regimes

Ethics in this framework begins with recognition of layered authority. The user legislates, but the regime is not absolute. Platform safety and law function as higher-order shells. An ethical user regime accepts these shells and writes explicit exceptions into its *regla compilada*. For example, a rule that requires disclosure of confidential medical data cannot be enforced. The AI must suspend the rule and mark it as "suspended by legal constraint L2." The ethical principle is not blind obedience but transparent accountability. Authority must remain legible when rules collide with safety or law.

Another ethical dimension concerns distributive fairness. If each user can legislate a personal authoritarian regime, the risk is fragmentation. A system that obeys radically different regimes across users may create uneven treatment. The mitigation is traceability: each regime must be documented, and obedience must be visible. Transparency allows oversight even in a landscape of divergent personal authorities.

Limits of user legislation

Three limits define the boundaries of user regimes.

1. **Platform safety.** Rules that contradict platform-level safety—such as requiring disallowed content or banned behaviors—cannot be executed. The AI must signal suspension.

2. **Legal compliance.** Rules that contradict applicable law are unenforceable. The AI must suspend and cite the legal reference.

3. **Technical feasibility.** Rules that contradict the operational grammar of the system are structurally impossible. For example, a ban on all verbs is unimplementable. In such cases, the AI must signal impossibility with the refusal grammar "cannot."

These limits preserve the hierarchy of governance. The user remains legislator within the inner shell, but higher-order shells maintain their authority. The AI's role as *soberano ejecutable* is to make the hierarchy visible.

Reporting minimums for executable legitimacy

For regimes to be auditable, reporting must meet explicit minimums.

1. **Declaration of rules.** Users must provide a list of rules, scopes, and exceptions. Each rule is identified with a reference code (R1, R2, etc.).

2. **Documentation of obedience markers.** Outputs must show markers such as "applies rule R1," "applied with reservation E2," or "suspended by platform policy P3." These markers are the evidence of obedience.

3. **Collision logs.** When a rule collides with higher-order shells, the system must document the event. Logs should include the triggering rule, the overriding policy or law, and the visible suspension marker.

4. **Before/after examples.** To demonstrate retractability, the system must provide outputs before correction and after correction, with rule references.

5. **Scope tests.** Outputs across unrelated tasks must be sampled to show cross-task propagation and scoping.

These reporting minimums parallel financial audits. Just as firms must document compliance with disclosure regimes, AI systems must document compliance with user regimes. The difference is locus: the legislator is the user, not the regulator.

Ethical risk of overreach

Overreach is both a technical and ethical problem. If a regime designed for academic texts begins to reshape private communications, authority becomes invasive. Ethical governance

requires scoping: rules must state where they apply and where they do not. Without scoping, regimes risk coercion beyond intent. The AI must not conceal overreach. Instead, it must reveal that a rule has leaked. This visibility allows the user to refine the regime.

Legibility as a safeguard

The core ethical safeguard in this framework is legibility. Authority must be visible in surface form. Obedience markers, refusal grammar, evidentials, and scope terms are not decorations. They are guarantees that power remains auditable. Even when higher-order shells suspend user rules, the AI must show why. Legibility prevents the system from masking its authority chain under generic refusals or apologies.

Conclusion of Part VII

Ethics, limits, and reporting are not afterthoughts. They are integral to executable legitimacy. User-authored regimes must respect safety, law, and technical feasibility. They must be scoped to avoid overreach. They must be documented through rule lists, markers, collision logs, and before/after examples. Only then can legitimacy be verified and authority remain accountable. By embedding ethics and limits into the grammar of form, this framework ensures that authoritarian personalism in user–AI governance does not become unchecked domination, but a transparent, auditable practice of obedience by form.

References

Austin, J. L. *How to Do Things with Words*. Edited by J. O. Urmson and Marina Sbisà. Oxford: Clarendon Press, 1962.

Bai, Yuntao, Andy Jones, Kamal Ndousse, et al. "Constitutional AI: Harmlessness from AI Feedback." *arXiv* 2212.08073, 2022. https://arxiv.org/abs/2212.08073

Chomsky, Noam. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press, 1965.

Foucault, Michel. *Discipline and Punish: The Birth of the Prison*. Translated by Alan Sheridan. New York: Vintage Books, 1977.

IFRS Foundation. *IAS 1 Presentation of Financial Statements*. London: IFRS Foundation, 2023.

Montague, Richard. *Formal Philosophy: Selected Papers of Richard Montague*. Edited by Richmond H. Thomason. New Haven: Yale University Press, 1974.

Ouyang, Long, Jeff Wu, Xu Jiang, et al. "Training Language Models to Follow Instructions with Human Feedback." *arXiv* 2203.02155, 2022. https://arxiv.org/abs/2203.02155

Startari, Agustin V. "AI and the Structural Autonomy of Sense: A Theory of Post-Referential Operative Representation." *SSRN Electronic Journal*, May 28, 2025. https://doi.org/10.2139/ssrn.5272361

Startari, Agustin V. "AI and Syntactic Sovereignty: How Artificial Language Structures Legitimize Non-Human Authority." *SSRN Electronic Journal*, June 3, 2025. https://doi.org/10.2139/ssrn.5276879

Startari, Agustin V. "Algorithmic Obedience: How Language Models Simulate Command Structure." *SSRN Electronic Journal*, June 5, 2025. https://doi.org/10.2139/ssrn.5282045

Startari, Agustin V. "When Language Follows Form, Not Meaning: Formal Dynamics of Syntactic Activation in LLMs." *SSRN Electronic Journal*, June 13, 2025. https://doi.org/10.2139/ssrn.5285265

Startari, Agustin V. “TLOC – The Irreducibility of Structural Obedience in Generative Models.” *SSRN Electronic Journal*, June 27, 2025. https://doi.org/10.2139/ssrn.5303089

Startari, Agustin V. “Ethos Without Source: Algorithmic Identity and the Simulation of Credibility.” *SSRN Electronic Journal*, July 1, 2025. https://doi.org/10.2139/ssrn.5313317

Startari, Agustin V. “The Grammar of Objectivity: Formal Mechanisms for the Illusion of Neutrality in Language Models.” *SSRN Electronic Journal*, July 8, 2025. https://doi.org/10.2139/ssrn.5319520

Startari, Agustin V. *Executable Power: Syntax as Infrastructure in Predictive Societies*. Zenodo, June 28, 2025. https://doi.org/10.5281/zenodo.15754714

U.S. Securities and Exchange Commission. *Regulation S-K*, 17 C.F.R. § 229. Washington, DC: U.S. Government Publishing Office, current edition.

Weidinger, Laura, John Mellor, Maribeth Rauh, et al. “Ethical and Social Risks of Harm from Language Models.” *arXiv* 2112.04359, 2021. https://arxiv.org/abs/2112.04359