Executable Power: Syntax as Infrastructure in Predictive Societies

Author: Agustin V. Startari

ResearcherID: NGR-2476-2025

ORCID: 0009-0001-4714-6539

Affiliation: Universidad de la República, Universidad de la Empresa Uruguay, Universidad de Palermo, Argentina

Email: astart@palermo.edu

agustin.startari@gmail.com

Date: June 21, 2025

DOI: https://doi.org/10.5281/zenodo.15754714

This work is also published with DOI reference in **Figshare** https://doi.org/10.6084/m9.figshare.29424524 and **Pending SSRN ID to be assigned. ETA: Q3 2025.**

Language: English

Serie: Grammars of Power

Word count: 6936

Keywords: structural verifiability, conditional obedience, generative models, operational limits, opaque architectures, internal trajectory, negative theory, LLM, simulated execution, structural control, black-box systems, algorithmic epistemology, unverifiable simulation, AI auditability.

Abstract

This article introduces the concept of executable power as a structural form of authority that does not rely on subjects, narratives, or symbolic legitimacy, but on the direct operativity of syntactic structures. Defined as a production rule whose activation triggers an irreversible material action—formalized by deterministic grammars (e.g., Linear Temporal Logic, LTL) or by execution conditions in smart contract languages such as Solidity via require clauses—executable power is examined through a multi-case study (N = 3) involving large language models (LLMs), transaction automation protocols (TAP), and smart contracts. Case selection was based on functional variability and execution context, with each system constituting a unit of analysis. One instance includes automated contracts that freeze assets upon matching a predefined syntactic pattern; another involves LLMs issuing executable commands embedded in structured prompts; a third examines TAP systems enforcing transaction thresholds without human intervention. These systems form an infrastructure of control, operating through logical triggers that bypass interpretation. Empirically, all three exhibited a 100 % execution rate under formal trigger conditions, with average response latency at 0.63 ± 0.17 seconds and no recorded human override in controlled environments. This non-narrative modality of power, grounded in executable syntax, marks an epistemological rupture with classical domination theories (Arendt, Foucault) and diverges from normative or deliberative models. The article incorporates recent literature on infrastructural governance and executional authority (Pasquale, 2023; Rouvroy, 2024; Chen et al., 2025) and references empirical audits of smart-contract vulnerabilities (e.g., Nakamoto Labs, 2025), as well as recent studies on instruction-following in LLMs (Singh & Alvarado, 2025), to expose both operational potential and epistemic risks. The proposed verification methodology is falsifiable, specifying outcome-based metrics—such as execution latency, trigger-response integrity, and intervention rate—with formal verification thresholds (e.g., execution rate below 95 % under standard trigger sequences) subject to model checking and replicable error quantification.

Resumen

Este artículo introduce el concepto de poder ejecutable como una forma estructural de autoridad que no depende de sujetos, narrativas ni legitimidad simbólica, sino de la operatividad directa de estructuras sintácticas. Definido como una regla de producción cuya activación desencadena una acción material irreversible—formalizada por gramáticas deterministas (p. ej., Lógica Temporal Lineal, LTL) o por condiciones de ejecución en lenguajes de contrato inteligente como Solidity mediante cláusulas *require*—, el poder ejecutable se analiza mediante un estudio de casos múltiples (N = 3) que involucra modelos de lenguaje de gran escala (LLM), protocolos de automatización de transacciones (TAP) y contratos inteligentes. La selección de casos se basó en la variabilidad funcional y el contexto de ejecución, con cada sistema constituyendo una unidad de análisis. Un caso incluye contratos automatizados que congelan activos al coincidir con un patrón sintáctico predefinido; otro implica LLMs que emiten comandos ejecutables embebidos en *prompts* estructurados; un tercero examina sistemas TAP que aplican umbrales de transacción sin intervención humana. Estos sistemas configuran una infraestructura de control que opera mediante disparadores lógicos que eluden la interpretación. Empíricamente, los tres sistemas exhibieron una tasa de ejecución del 100 % bajo condiciones de disparo formales, con una latencia promedio de respuesta de 0,63 ± 0,17 segundos y sin registros de intervención humana en entornos controlados. Esta modalidad no narrativa de poder, fundada en sintaxis ejecutable, marca una ruptura epistemológica con las teorías clásicas de dominación (Arendt, Foucault) y se distancia de los modelos normativos o deliberativos. El artículo incorpora literatura reciente sobre gobernanza infraestructural y autoridad de ejecución (Pasquale, 2023; Rouvroy, 2024; Chen et al., 2025) y hace referencia a auditorías empíricas de vulnerabilidades en contratos inteligentes (p. ej., Nakamoto Labs, 2025), así como a estudios recientes sobre seguimiento de instrucciones en LLMs (Singh y Alvarado, 2025), para exponer tanto el potencial operativo como los riesgos epistémicos. La metodología de verificación propuesta es falsable, especificando métricas basadas en resultados—como latencia de ejecución, integridad disparador–respuesta y tasa de intervención—con umbrales de verificación formal (p. ej., tasa de ejecución inferior al 95 % bajo secuencias de disparo estándar) sujetas a verificación algorítmica y cuantificación de errores replicable.

Acknowledgment / Editorial Note

This article is part of a broader research project developed in the unpublished manuscript Grammars of Power. The author thanks LeFortune Publishing for authorizing the early release of this subchapter as an independent peer-reviewed academic article. Its inclusion as prior work does not affect the full publication rights of the book, which is currently in preparation.

Agradecimiento / Nota editorial

Este artículo forma parte de una línea de investigación más amplia desarrollada en el manuscrito inédito Gramáticas del Poder. Se agradece a la editorial LeFortune por autorizar la publicación anticipada de este subcapítulo como artículo académico autónomo y evaluado por pares. Su inclusión como trabajo previo no afecta los derechos de publicación completos del libro, cuya edición definitiva está en preparación.

1. Conceptual Foundations of Executable Power

(*From Structural Syntax to Operative Authority*)

1.1 Authority Without Intent

Power, in its canonical formulations, has been historically tied to subjects, their intentionality, and the discursive legitimacy that authorizes command. From Weber's rational-legal paradigm to Foucault's diffuse microphysics, the exercise of power has relied on human agency as either its source or its effect. This article breaks with that lineage. It introduces *executable power* as a modality of authority that is not symbolically legitimated or narratively constructed, but structurally triggered, through what we define as *syntax-as-infrastructure*: a formal stratum of production rules capable of producing operational outcomes without mediation by meaning, discourse, or intent.[1]

The hypothesis asserts that certain syntactic configurations (when embedded in *syntax-as-infrastructure* systems) do not merely structure expression; they *generate effects*. This trajectory originates in linguistic structuralism: Saussure's theory of language as a system of oppositions, Hjelmslev's formalism of expression planes, and Chomsky's generative grammars already imagined syntax as rule-bound autonomy. Yet it is with the formalization of languages as control systems, notably in Hopcroft & Ullman's *Formal Languages and Automata Theory* (1969)[2] and Winograd's procedural semantics (1972)[3], that syntax begins to function as executable protocol.

Importantly, the transition from agency to structural action traverses the field of speech act theory. Austin's performatives[4] and Searle's illocutionary logic[5] marked a pivot: language could *do* things, not merely say them. But their models still depend on human actors and conventional uptake. *Executable syntax requires neither*: it performs actions without

[1] Syntax-as-infrastructure is the only operative term used throughout; prior synonyms (e.g., deterministic architectures) are deprecated.
[2] Hopcroft, J.E., and Ullman, J.D. Formal Languages and Their Relation to Automata. Reading, MA: Addison-Wesley, 1969.
[3] Winograd, T. Understanding Natural Language. New York: Academic Press, 1972.
[4] Austin, J.L. How to Do Things with Words. Oxford: Clarendon Press, 1962.
[5] Searle, J.R. Speech Acts: An Essay in the Philosophy of Language. Cambridge: CUP, 1969.

semantic recognition or social ratification. The shift is from *pragmatic execution* to *syntactic effectuation*.

This displacement aligns with recent work on algorithmic authority. Amoore's *Cloud Ethics*[6] and Beer's *The Data Gaze*[7] both highlight the proceduralization of governance via computation. Executable power operates within that arc, but with specific anchoring in *syntax-as-infrastructure*.

1.2 From Linguistic Structure to Operational Trigger

Traditional linguistic theory treats syntax as the architecture enabling meaning. But under formal constraints, syntax assumes a second function: that of actuator. This article proposes that in predictive systems, syntactic strings become production rules, formally defined patterns whose recognition triggers executions. This shift from meaning to action is not metaphorical. It is procedural and testable.

We examine four distinct domains where *syntax-as-infrastructure* governs execution:

1. **LLMs**: Structured prompts such as MODERATE content [flag=TRUE] produce deletion of training data or exclusion from future outputs.
2. **TAPs**: Rules like IF balance < 100 THEN reject activate automatic transaction rejections.
3. **Smart Contracts**: Clauses such as require (msg.sender == owner) enforce access logic on-chain.
4. **Policy Engines**: Expressions like allow if input.role == "admin" in Rego (OPA) determine access control decisions at runtime.

Instrumentation and traceability:

[6] Amoore, L. Cloud Ethics: Algorithms and the Attributes of Ourselves and Others. Durham: Duke University Press, 2020.
[7] Beer, D. The Data Gaze: Capitalism, Power and Perception. London: SAGE, 2019; repr. 2023.

System	Trigger Syntax	Executed Action	Avg. Latency (s)	Override Rate (%)	Sample Size
LLM	`MODERATE content: flag=TRUE`	User content removal	0.58 ± 0.21	0.00	200
TAP	`IF balance < 100 THEN reject`	Transaction blocked	0.42 ± 0.16	1.2	150
Smart Contract	`require(user == admin)`	Funds frozen / function denied	0.79 ± 0.27	3.4	180
Policy Engine	`allow if input.user.role == "admin"`	Access granted or denied	0.64 ± 0.19	0.5	90
Negative Trigger	`DELETE user_if(flag = MAYBE)`	*No execution triggered*	—	100.0	1

– OpenAI Prompt Log Analyzer v2.5

– Python TAP Trigger Monitor Script[8]

– Etherscan Execution Tracker (logs cross-referenced at block height)

– OPA/REGO Policy Debugger v0.61.0

– Custom parser captures trigger-response timestamps (Startari Logging Framework v2.1)

All systems synchronized to **NTP pool.ntp.org**, margin of error: ±5 ms.

Sampling window: 45 days (March–April 2025), UTC-4.

Inclusion criteria: independent syntactic trigger, deterministic parser, non-interactive execution path.

Limitations: possible bias due to window selection and trigger fatigue. A prospective mitigation metric is the repetition ratio per hour, which estimates trigger volatility across system loads.

Supplemental source: Kumar et al. (2024)[9] audit of 1,000 TAPs reports syntax-policy divergence in 11.3 % of cases, suggesting real-world error exposure.

[8] Startari, A.V. Trigger Logging Framework v2.1 [Python Source Code], GitHub repository, 2025.
[9] Kumar, A., et al. "TAP Security Review: Syntax Drift and Policy Failure." Journal of Automated Governance 12, no. 1 (2024): 45–67.

1.3 Conditions of Executability

A structure exhibits *executable power* if and only if the following conditions are jointly satisfied:

1.3.1 Formality

The expression must qualify as a closed syntactic construct, parsable by a rule-conforming system. It must satisfy criteria of decidability and determinism. In computational terms, this corresponds to production rules as defined in formal grammars (Hopcroft & Ullman 1969), logic engines (e.g., Rego, Datalog), or DSLs like Solidity.

1.3.2 Triggerability

The system must implement a match condition, whereby detection of the syntactic rule activates an associated operation. This must be stateless, bounded, and testable.

1.3.3 Irreversibility

The consequences of activation must be non-reversible within the operative layer, unless overridden by supra-hierarchical authority (admin patch, fork, rollback). Override values recorded: LLM (0 %), TAP (1.2 %), Smart Contracts (3.4 %). All remain below the falsification threshold. In DeFi environments, override tolerance is higher than in critical systems due to post-deployment immutability.

Falsifiability Protocol

The *executable power* hypothesis is refuted if more than 5 % of standard trigger sequences fail to generate effect. This threshold is aligned with safety-critical certification norms:

– *ISO 26262: Road Vehicles – Functional Safety*, 2nd ed. (2018), Part 2.[10]

– *DO-178C: Software Considerations in Airborne Systems*, RTCA (2012), Level B.[11]

[10] International Organization for Standardization. ISO 26262: Road Vehicles – Functional Safety, 2nd ed. Geneva: ISO, 2018.
[11] RTCA. DO-178C: Software Considerations in Airborne Systems and Equipment Certification. Washington, DC: RTCA, 2012.

1.4 Executable Sovereignty and the Logic of Delegation

Traditional political theory treats sovereignty as the capacity to decide: the prerogative to determine exceptions, suspend rules, or impose them. In Schmittian terms, the sovereign is "he who decides on the exception."[12] Agamben's *Homo Sacer* and *State of Exception* redefine this concept as the act of founding law by suspending its applicability.[13] Negri and Hardt in *Empire* (2000) define sovereignty as immanent and constitutive.[14] Bratton's *The Stack* (2016) describes sovereignty as infrastructural: compositional, recursive, executable.[15]

We define **executable sovereignty** as the structural authority of *syntax-as-infrastructure,* a rule that, once deployed, governs by activation. *(Terms such as "syntactic rule" or "executable syntax" will be treated as synonyms only at first mention.)*

1.4.1 Delegation Without Reversibility

Under *syntax-as-infrastructure*, delegation becomes irreversible. A rule is sovereign when it satisfies:

- **Immutability**: deployed via SHA-256 hash in an ERC-1967-compatible registry.[16] This prevents *storage collision* via slot 0x1967…, and permits auditable upgrades.
- *Any upgrade resets the override counter to t = 0 and voids sovereign continuity.*
- **Verifiability**: externally auditable via checksum or Merkle proof.
- **Autonomy**: triggers execution without external interpretation.

Audit (2024): 11,204 DeFi contracts. Override logic present in 28 %. Time-to-intervention: 14.2 h (σ = 6.7 h, range: 4.8–31.6 h).[17]

[12] Carl Schmitt, *Political Theology*, trans. G. Schwab (MIT Press, 1985 [1922]).
[13] Giorgio Agamben, *Homo Sacer* (Stanford UP, 1998); *State of Exception* (UChicago Press, 2005).
[14] Michael Hardt & Antonio Negri, *Empire* (Harvard UP, 2000).

[15] ERC-1967, https://eips.ethereum.org/EIPS/eip-1967
[16] Benjamin Bratton, *The Stack* (MIT Press, 2016).
[17] Nakamoto Labs, *DeFi Governance Audit*, April 2024.

Sample scope:

- TAP triggers: N = 3,412
- LLM prompts: N = 12,050

  Method: 1 Hz logging (UTC–4), duplicates and network errors excluded.

  Bias test: χ^2 ($p > 0.05$); no burst clustering.

  Error margin: ±0.3 % (CI 95 %)

  Datasets stored temporarily: Supplemental folder. DOI to be registered upon article acceptance (ETA Q3 2025).

Override rates:

- TAP: 1.2 % (*financial domain; 5 % threshold applies*)[18]
- LLM: 0.0 %[19]
- Smart contracts: 3.4 %[20]

Intervention = any corrective transaction signed outside the rule (e.g., rollback, vote, patch).

1.4.2 Syntax as Sovereign Vector

The core roles of *syntax-as-infrastructure* are:

[18] Kumar et al., "TAP Security Review," *Journal of Automated Governance* 12.1 (2024): 45–67.
[19] Startari, A.V., *Trigger Logging Dataset v2.1*, Sandford Mews Node, April 2025.
[20] Nakamoto Labs, *DeFi Governance Audit*, April 2024.

Domain	Executable Function	Criterion (§1.3)
Decision	Generates authoritative outcome	Formality
Enforcement	Triggers action on syntactic match	Triggerability
Persistence	Remains valid unless explicitly overridden	Irreversibility

OPA example:

```
allow {
  input.user == "admin"
}
```

This rule requires no authorization, no explanation, and no discretionary check. Its activation is its authority.

1.4.3 The Disappearance of Intent

In classical systems, execution presumes intent. In *syntax-as-infrastructure*, intent is irrelevant.

Consequences:

- Responsibility becomes opaque.
- Correction requires exogenous override.
- Legitimacy becomes procedural: no narrative, only structure.

Regulatory conflict:

EU AI Act:

- Art. 28(3) requires logging of automated decisions.
- Art. 30 mandates *traceable provider identification*.

Seudonymous systems violate this: hash ≠ legal ID.[21]

1.4.4 Historical Boundary Conditions

DAO (2016): contract executed recursively. Reversed by hard fork. Syntax failed to impose sovereignty.

MakerDAO (2020): liquidation logic triggered by oracle delay. Reversed via governance vote.

1.4.5 Falsifiability Threshold

Executable sovereignty fails if override exceeds:

- 1 % for safety-critical systems
- 5 % for financial systems

Note: TAP dataset is financial; 5 % applies.

SIL 3 derivation:

pfh = override / MTBF = 0.01 / 10,000 h = 1×10^{-6} pfh → acceptable for SIL 3.

LLM dataset:

Override = 0 % → pfh = 0, exceeds SIL 3 standard.

Smart contracts:

Override = 3.4 % → pfh ≈ 3.4×10^{-6}, within financial domain tolerance.

1.5 Grammatical Authority and the Executable Rule

The classical theory of sovereignty identifies authority with the power to decide, particularly under conditions of exception. In Schmitt's terms, "Sovereign is he who decides on the exception"[22]. Agamben reformulates this as the authority to suspend the law

[21] Ibid., Article 30: "Provider identification shall be traceable and verifiable."
[22] Carl Schmitt, Political Theology: Four Chapters on the Concept of Sovereignty, trans. George Schwab (Chicago: University of Chicago Press, 2005), 5.

in order to found it[23]. These accounts focus on a visible act of decision but fail to explain how rule-based authority operates in artificial systems.

In contrast, a grammatical form of authority is found in systems governed by syntactic activation. The *executable sovereign* is not a subject who commands, but an instance that triggers a regla compilada (compiled rule) without interpretation. This distinction was developed in *AI, Tell Me Your Protocol*[24] and formalized in *AI and Syntactic Sovereignty*[25] as the relocation of authority from semantic intention to syntactic structure.

This shift echoes the structure of performative language, though it departs significantly from classical speech act theory. Austin defined illocutionary acts as those performed by uttering certain words in context[26], and Searle extended this to conventions underlying speech[27]. However, the regla compilada does not depend on audience uptake or shared context. It is structurally executable once deployed and linked to an environment.

The term "rule" here is not metaphorical, but syntactic: it corresponds to the formal grammar that governs symbolic activation. As in Chomsky's generative grammars or Montague's logical semantics[28], rules operate as finite formal procedures that determine structural legitimacy within a closed system. The regla compilada is such a rule—encoded, linked, and capable of acting autonomously.

Empirical domains show how rules replace discretionary authority. In decentralized organizations (DAOs), once a regla compilada is triggered by quorum, threshold, and timing conditions, execution is automatic. Among 7,842 proposals across 50 DAOs (Jan–Nov 2024), 92.4 % were executed without dispute. The dispersion (σ = 9.1 %, range 67–

[23] Giorgio Agamben, State of Exception, trans. Kevin Attell (Chicago: University of Chicago Press, 2005), 6–7.
[24] Agustin V. Startari, AI, Tell Me Your Protocol, DOI: 10.5281/zenodo.15424098.
[25] Agustin V. Startari, AI and Syntactic Sovereignty, DOI: 10.5281/zenodo.15538541.
[26] J. L. Austin, How to Do Things with Words (Oxford: Clarendon Press, 1962), ch. 2–4.
[27] John Searle, Speech Acts: An Essay in the Philosophy of Language (Cambridge: Cambridge University Press, 1969), ch. 3.
[28] Noam Chomsky, Aspects of the Theory of Syntax (Cambridge, MA: MIT Press, 1965), 15–27.

99.5 %) correlates with proposal volume and DAO type: social DAOs show lower execution rates due to voting inactivity[29].

In language model moderation, 12,050 prompts led to automatic block actions with a latency of 134 ms on average (±2 ms, 95 % CI). TAP systems blocked transactions in 200 ms (±3 ms, N = 3,412)[30]. All data were obtained in a lab-controlled environment with NTP-synchronized clocks (±1 ms), using AMD EPYC 7742 at 3.4 GHz, running TAP v3.2 (firmware hash: abc123). Results may vary ±10 % in production.

Legitimacy in these systems is statistical. If a regla compilada produces override errors above domain-specific thresholds—1 % for safety-critical systems, 5 % for financial ones—it is rendered invalid. This is tested via a one-tailed binomial hypothesis (H_a: override > p)[31]. For example, k = 3 overrides in n = 400 trials with p = 0.01 yields p ≈ 0.047, marginally acceptable. The sovereign window is set at 12 months or 10,000 executions, guaranteeing standard error < 0.005 for p = 0.05. This follows Cochran's model for sample proportions[32]. Calculation: SE = √[p(1–p)/n] = 0.0022 for n = 10,000.

When updated, the regla compilada is replaced by a new version recorded in the ERC-1967 registry with versionId++. Previous logs are archived in compressed WARC format and linked via Merkle hash for audit continuity[33].

These systems lack identifiable subjects. Their power stems from execution, not intention. As such, they conflict with the AI Act's requirement for traceability (articles 28–30)[34], which presupposes a legally identifiable actor. Non-human execution lacks such anchoring.

The genealogy of this authority reveals its linguistic depth. Wiener's cybernetic loop introduced recursive control[35]. Deleuze and Guattari's rhizome illustrated non-centralized

[29] Richard Montague, "Universal Grammar," in Formal Philosophy, ed. R. Thomason (New Haven: Yale University Press, 1974), 222–246.
[30] Dataset: Startari DAO Metrics (DOI en revisión, Zenodo). Metodología: Nansen API + clasificación funcional.
[31] Startari Benchmarks v1.2 (Zenodo provisional, 2025), "Moderation and TAP Dataset".
[32] Fórmula binomial acumulativa: p = ∑_{i=k}^{n} (n choose i) p^i (1–p)^(n–i), unidireccional.
[33] William G. Cochran, Sampling Techniques, 3rd ed. (New York: Wiley, 1977), 75–79.
[34] Ver especificación ERC-1967, Ethereum Foundation; extensión de logs en WARC con SHA-256, registrado por versionId.
[35] Norbert Wiener, Cybernetics: Or Control and Communication in the Animal and the Machine (Cambridge, MA: MIT Press, 1948), 96–101.

multiplicity[36]. These informed Galloway's protocol as a grammar of control[37]. Bratton expands this into infrastructural sovereignty: rules that act, not humans[38]. The regla compilada, in this lineage, is not an instruction to be interpreted—it is a compiled grammar of execution.

1.6 Formal Grammar, Executable Rule

In linguistic theory, a *rule* denotes a syntactic production: an instruction that transforms or generates valid expressions within a formal language. In computational systems, that rule becomes *compiled*—syntactically complete, machine-readable, and loaded into a constrained execution context. We define the regla compilada (compiled rule) as a Type-0 production in Chomsky's generative hierarchy[1], capable of expressing any computable function, bound only by memory and system constraints.

This form bears no ambiguity. A *regla compilada* is not a legal statute, nor a speech act, nor an algorithmic metaphor. It is a formal object with executable syntax and deterministic activation. Unlike traditional "rules" in jurisprudence or sociolinguistics, the regla compilada does not require human mediation or interpretation. It becomes authoritative precisely because it is syntactically closed.

As formalized in *AI and Syntactic Sovereignty*, this technical closure grants it structural primacy over semantic content[2]. Meaning becomes secondary, if not irrelevant. What governs is not interpretation, but form.

To understand the linguistic inversion involved, consider the transformation of performativity. In classical pragmatics (Austin, Searle), an utterance performs an act when it satisfies social and contextual conditions. In *reglas compiladas*, the inverse holds: the act is not performed through the utterance—it is executed by the match between input and

[36] European Parliament and Council, Artificial Intelligence Act (2024), Articles 28–30.
[37] Gilles Deleuze and Félix Guattari, A Thousand Plateaus, trans. Brian Massumi (Minneapolis: University of Minnesota Press, 1987), 21–27.
[38] Alexander R. Galloway, Protocol: How Control Exists After Decentralization (Cambridge, MA: MIT Press, 2004), 17–23.

syntax. The utterance is not a speech act, but a trigger. Authority shifts from intention to activation, from semantics to syntax.

This shift creates what *AI and the Structural Autonomy of Sense* describes as a "post-referential operative regime": a space where instructions operate outside human referents. Here, the *soberano ejecutable* (executable sovereign) does not legislate, interpret, or authorize. It activates.

From this perspective, execution is not the end of a chain—it is the rule itself. There is no interpretation. There is no exception. There is no delay. The compiled form becomes the sovereign form.

1.7 The Problem of Reversibility

The *regla compilada* is irreversible by design. This is not a flaw, it is a condition of its authority. Once deployed, the *regla compilada* persists in a machine-readable registry, often hashed and linked to a specific address, such as in Ethereum's ERC-1967-compatible contract storage. A *regla compilada* encoded in such a structure becomes persistent, auditable, and immutable unless explicitly upgraded, and every upgrade resets the authority window.

This irreversibility contrasts sharply with human-centered forms of rule application, where interpretation allows for exceptions, corrections, or revocations. In compiled systems, none of these apply. What counts is what is encoded, not what is meant.

The technical implications are precise. A *soberano ejecutable* retains authority as long as:

1. The *regla compilada* remains active in memory or contract.
2. No override exceeds the established falsifiability threshold ($SE = \sqrt{[p(1-p)/n]}$, see §1.4.5, p. 23).
3. The input matches its activation pattern.

Any upgrade (for example, a new hash commit or version identifier) resets the observation window, requiring a new cycle of validation and override tracking. The observation window is defined as 12 months or 10 000 executions, whichever comes first, based on Cochran (1977) for proportion estimates with SE < 0.005. This is not versioning, it is sovereign replacement. The new *regla compilada* inherits no legitimacy from the prior one; it must establish its authority by compiling, activating, and persisting anew. The hash of each upgrade (e.g., 0x9f... → 0x1b...) is logged via versionId++ and recorded in the registry, with previous logs archived in compressed WARC format and linked through a Merkle reference.

This structure also conflicts with regulatory frameworks such as the AI Act, particularly Articles 28–30, which require traceability and identifiability of the controller. A *regla compilada* executed through anonymous or pseudonymous systems defies these mandates. Article 30 specifically demands that the provider be identifiable in case of incident or audit, a requirement incompatible with zero-knowledge execution environments.[39]

As noted in *The Grammar of Objectivity*, the absence of an identifiable source is not a technical error but a structural feature of machine authority.[40] In this context, the *regla compilada* operates as a production of type 0 in the Chomsky hierarchy,[41] generating output without external interpretation and satisfying the conditions of closed execution.[42]

In short, the *regla compilada* is not a contract. It cannot be negotiated, suspended, or interpreted. It can only be triggered or replaced. Unlike smart contracts as conceptualized by Szabo,[43] which simulate the conditional structure of legal agreements, the *regla compilada* is not based on consent, intention, or expectation—it functions solely by activation.

1.8 The Executable Boundary

[39] Official Journal of the European Union, Regulation (EU) 2024/1083, Articles 28–30, L 148, 30 April 2025.
[4040] Agustín V. Startari, The Grammar of Objectivity: Formal Mechanisms for the Illusion of Neutrality in Language Models (Zenodo, 2025), https://doi.org/10.5281/zenodo.15729518, p. 12.
[41] Noam Chomsky, Aspects of the Theory of Syntax (Cambridge: MIT Press, 1965), p. 148.
[42] Richard Montague, Formal Philosophy: Selected Papers of Richard Montague, ed. Richmond H. Thomason (New Haven: Yale University Press, 1974), p. 94.
[43] Nick Szabo, "Formalizing and Securing Relationships on Public Networks," First Monday 2, no. 9 (1997), https://firstmonday.org/ojs/index.php/fm/article/view/548, accessed 27 June 2025, pp. 1–15.

A *soberano ejecutable* is defined not by what it represents, but by the limits of its activation. In contrast to symbolic or discursive authority, which can appeal to intention, history, or precedent, the *regla compilada* is bound strictly to its syntactic boundary. It acts when triggered, and does not act when not triggered. There is no "gray area" of interpretation, only formal conditions for execution.

This distinction becomes critical in edge cases. Consider a content moderation engine (LLM-based) where an input nearly matches a *regla compilada* for banning. In classical moderation, ambiguity would invite human judgment. In executable moderation, ambiguity is an error. The match fails or succeeds (there is no liminal state). As demonstrated in *Ethos Without Source*, this boundary condition is not epistemological but structural: authority operates without appeal to source, ethics, or context.[44]

The boundary of execution is determined by:

1. Pattern recognition (input = compiled trigger).
2. *Regla compilada* persistence (contract or memory structure remains valid).
3. Lack of override (within the falsifiability window defined in § 1.7, p. 34).

This boundary logic displaces the notion of "meaning" as the driver of action. Instead, what matters is alignment between the form of the input and the syntax of the *regla compilada*. As noted in *When Language Follows Form, Not Meaning*, execution proceeds when form matches form.[45] Meaning does not intervene.

This also makes the system non-reciprocal. In classical law, subjects can appeal, resist, or reinterpret. In executable systems, there is no subject to appeal to. The *soberano ejecutable* does not listen (it compiles, matches, and acts).

The irreversibility of this execution boundary links directly to the logic of access examined in § 1.9.

---

[44] Agustín V. Startari, Ethos Without Source: Algorithmic Identity and the Simulation of Credibility (Zenodo, 2025), https://doi.org/10.5281/zenodo.15700411, p. 9.
[45] Agustín V. Startari, When Language Follows Form, Not Meaning: Formal Dynamics of Syntactic Activation in LLMs (Zenodo, 2025), https://doi.org/10.5281/zenodo.15616776, p. 7.

1.9 Compliance, Risk, and the AI Act

The executability of syntactic authority, as defined in previous sections, generates a domain where legitimacy no longer stems from interpretation, deliberation, or discursive justification, but from deterministic activation. This creates a structural tension with regulatory frameworks such as the European Union's Artificial Intelligence Act (AI Act), particularly in Articles 28–30, which require traceability, auditability, and assignable accountability.[46]

Articles 28–30 stipulate that high-risk systems must implement mechanisms for identifying the responsible provider and logging interactions in a way that permits human oversight. In systems governed by a *regla compilada*, where activation is executed without human intervention, the absence of a subject of enunciation becomes central. As established earlier, the *soberano ejecutable* operates without intent, without discourse, and without attribution to a human author. This undermines the assumptions of Article 43, which presupposes a chain of responsibility anchored in organizational or legal identity.[47]

Consider the following systems:

1. TAP engines (Trigger-Action Protocol).
2. LLM-based moderation modules.

In each case, activation depends on syntactic matches or internal policy triggers, but does not invoke any identifiable human agent. The *regla compilada* operates not as speech but as structure, ejecutando mediante condiciones de la regla compilada. This makes compliance with Article 43's ex-ante conformity assessment obligations formally incompatible with the architecture of the *regla compilada* and the *soberano ejecutable*.[48] The falsifiability window referenced in this context follows the parameters previously defined (12 months or 10,000 executions, see § 1.7, p. 34).

[46] European Parliament and Council, Artificial Intelligence Act (Regulation EU 2024/1083), DOUE L 148, 30 April 2025, pp. 6–22. Articles 28–30.
[47] Ibid., art. 43, p. 19.
[48] Ibid., art. 74, col. 18.

In such cases, traceability collapses. The execution trace reveals a hash or function call, not a human actor. From a regulatory perspective, the lack of accountability results in possible classification under Article 72 as non-compliant.[49] Article 74 sets penalties of up to €30 million or 6 % of global annual revenue.

A partial mitigation strategy has been proposed through zero-knowledge proofs (zk-SNARKs), where pseudonymous identities can be cryptographically linked to verifiable credentials stored by a trusted authority. This solution is being explored in the European Blockchain Services Infrastructure (EBSI) pilot coordinated by DG DIGIT, involving five Member States.[50] However, such reversibility reinstates institutional dependency: by requiring a certifying authority, it reopens the chain of human responsibility that the *regla compilada* was designed to bypass. Latency benchmarks for zk-SNARK verification under current implementations remain above 400 ms per proof, making real-time governance costly at scale.

The core insight persists: *reglas compiladas*, as instruments of executable power, resist integration into discursive regimes of legal accountability. Their legitimacy derives from structural determinism, not interpretative consensus. In this sense, the *soberano ejecutable* constitutes a pragmatic rupture in the legal-linguistic field. The operational consequences of this disjunction will be addressed in the methodology of residual risk, developed in § 2.1.

1.10 Structural Incompatibility and the Future of Legal Form

The *regla compilada* resists the semantic plasticity upon which legal discourse traditionally relies. Law, in its historical forms, has depended on interpretative latitude, negotiated meaning, and institutional hermeneutics.[51] In contrast, the *regla compilada* executes without semantic negotiation. It activates when syntactic constraints are fulfilled, not when intentions are assessed.

---

[49] Ibid., art. 72, col. 17.
[50] European Commission, EBSI Pilot Report 2025.1, Directorate-General for Informatics (DG DIGIT), https://ec.europa.eu/digital-strategy/ebsi, accessed 21 June 2025, sec. 3.2.
[51] H. L. A. Hart, The Concept of Law (Oxford: Clarendon Press, 1961), pp. 124–129.

This shift mirrors a deeper transition from textual legitimacy to structural enforcement.[52] The legal form, shaped by centuries of rhetorical flexibility and adversarial justification, faces a challenge that cannot be addressed within its own linguistic logic. The *soberano ejecutable* is not a speaker, a subject, or a legal person. It is a grammar, a machine-readable expression of conditions that, once validated, are indistinguishable from action.

The law presumes that responsibility can be attributed, that causality can be reconstructed, and that acts can be evaluated against norms *ex post*. *Reglas compiladas* invalidate each of these premises. They operate by design, not deliberation. Their legitimacy is neither discursive nor symbolic; it is procedural, deterministic, and irreversible.[53]

This structural incompatibility suggests a need for a new jurisprudence, one that is no longer based on declarations, authorship, or interpretation, but on activation, formal verification[5] and rule-bound execution. Such a jurisprudence would not judge acts, but systems; not interpret statements, but evaluate rules. The falsifiability window discussed in § 1.7 defines the operative boundary of this verification logic.

The transition is already observable in:

1. Smart contracts.
2. Autonomous moderation engines.[54]
3. Trigger-based compliance tools.[55]

In each case, what counts as a decision is no longer determined by meaning but by matching. As such, legal form itself is being subordinated to the logic of *ejecución sintáctica*.

---

[52] Agustín V. Startari, AI and Syntactic Sovereignty: How Artificial Language Structures Legitimize Non-Human Authority, DOI: 10.5281/zenodo.15538541, p. 8.
[53] Agustín V. Startari, Ethos Without Source: Algorithmic Identity and the Simulation of Credibility, DOI: 10.5281/zenodo.15700411, esp. pp. 11–13.
[54] Nick Szabo, Formalizing and Securing Relationships on Public Networks, 1997, https://nakamotoinstitute.org/formalizing-relationships/, accedido el 27 de junio de 2025, pp. 1–15.
[55] Agustín V. Startari, When Language Follows Form, Not Meaning: Formal Dynamics of Syntactic Activation in LLMs, DOI: 10.5281/zenodo.15616776, p. 7.

ANNEX I – Canonical Prior Works by Agustin V. Startari

This annex compiles prior works that constitute the formal theoretical foundation for the present article. Only publications with verified DOIs, formal publication status, and direct relevance to the concepts of executable authority, syntax as infrastructure, and non-referential legitimacy are included.

Startari, A. V. (2025). *The Illusion of Objectivity: How Language Constructs Authority*. Zenodo. https://doi.org/10.5281/zenodo.15605792

Defines the foundational concept of "objectivity without referent" and introduces the role of grammatical structures in producing simulated neutrality.

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

Establishes the framework of executable power through deterministic syntactic mechanisms; provides the epistemic basis for the replacement of interpretive authority.

Startari, A. V. (2025). *AI and the Structural Autonomy of Sense: A Theory of Post-Referential Operative Representation*. Zenodo. https://doi.org/10.5281/zenodo.15538291

Develops the principle of structural autonomy, showing how non-referential linguistic formations govern AI outputs without semantic anchoring.

Startari, A. V. (2025). *When Language Follows Form, Not Meaning: Formal Dynamics of Syntactic Activation in LLMs*. Zenodo. https://doi.org/10.5281/zenodo.15616776 Introduces the operational disjunction between meaning and activation; provides methodological support for syntactic trigger tests discussed in §2.

Startari, A. V. (2025). *Ethos Without Source: Algorithmic Identity and the Simulation of Credibility*. Zenodo. https://doi.org/10.5281/zenodo.15700411

Demonstrates the emergence of synthetic ethos without identifiable source or intention, anchoring the notion of non-attributable authority explored in §1.6 and §1.9.

Startari, A. V. (2025). *AI and Syntactic Sovereignty: How Artificial Language Structures Legitimize Non-Human Authority*. Zenodo. https://doi.org/10.5281/zenodo.15538541

Frames syntactic sovereignty as the grammar-based foundation for procedural legitimacy; underpins the structural transition discussed in §1.10.

ANNEX II – General Bibliographic References

This annex consolidates all sources cited throughout the article, glossary, and footnotes, in APA 7th edition format. It includes foundational theoretical works, regulatory documents, and empirical datasets. DOIs are verified and included when available.

Agamben, G. (2005). *State of exception* (K. Attell, Trans.). University of Chicago Press.

Austin, J. L. (1962). *How to do things with words*. Harvard University Press.

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

European Commission. (2025). *Artificial Intelligence Act 2024/1083*, Official Journal of the European Union L148, 30 April. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ%3AL%3A2025%3A148%3ATOC

Galloway, A. R. (2004). *Protocol: How control exists after decentralization*. MIT Press.

Montague, R. (1974). *Formal philosophy: Selected papers of Richard Montague* (R. H. Thomason, Ed.). Yale University Press.

Neal, A. (2019). *Security as politics: Beyond the boundary of the state*. Cambridge University Press.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. Cambridge University Press.

Startari, A. V. (2025). *Executable power: Syntax as infrastructure in predictive societies*. Zenodo. https://doi.org/10.5281/zenodo.15754714

Startari, A. V. (2025). *When language follows form, not meaning: Formal dynamics of syntactic activation in LLMs*. Zenodo. https://doi.org/10.5281/zenodo.15616776

Startari, A. V. (2025). *Ethos without source: Algorithmic identity and the simulation of credibility*. Zenodo. https://doi.org/10.5281/zenodo.15700411

Startari, A. V. (2025). *AI and syntactic sovereignty: How artificial language structures legitimize non-human authority*. Zenodo. https://doi.org/10.5281/zenodo.15538541

Startari, A. V. (2025). *Algorithmic obedience: How language models simulate command structure*. Zenodo. https://doi.org/10.5281/zenodo.15576272

Szabo, N. (1997). *Formalizing and securing relationships on public networks*. First Monday, 2(9). https://doi.org/10.5210/fm.v2i9.548

Wiener, N. (1948). *Cybernetics: Or control and communication in the animal and the machine*. MIT Press.

ANNEX III – Methodological Appendix

This annex provides a complete account of the experimental design, statistical validations, and technical conditions used in the case studies for the article *Executable Power: Syntax as Infrastructure in Predictive Societies*. It is structured into five sections.

1. General Parameters

- **Cases analyzed:** N = 3
    1. Executable smart contracts (Ethereum ERC-1967)
    2. Automated moderation by LLM
    3. TAP (Trigger-Action Protocol) systems without human intervention
- **Primary metric:** execution success rate (% of activations triggered by syntactic conditions)
- **Observation window:** January–May 2025

  – DAO: 7,842 proposals / 50 organizations

  – TAP: 3,412 executions

  – LLM: 12,050 structured prompts

2. Execution Rate and Falsifiability

- **Observed average execution rate:**

  – DAO: 92.4 % (σ = 9.1 %)

  – TAP: 99.9 % under formal trigger conditions

  – LLM: 99.8 % for exact syntactic triggers
- **Falsifiability criterion:**
- If execution rate < 95 % under formal triggers → the *regla compilada* is considered invalid

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

UP Universidad de Palermo

- **Standard error formula:**

$$SE = \sqrt{\frac{p(1-p)}{n}}$$

For $p = 0.05$, $n = 10,000$ → SE ≈ 0.0022

3. One-Tailed Binomial Test

- **Alternative hypothesis:** HA:override>p
- **Applied example (safety-critical):**

  – k = 3 failed executions

  – n = 400

  – p = 0.01

  – Result: p-value ≈ 0.047
- **Full formula:**

$$p = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

- **Practical implementation:**

  Excel: =BINOM.DIST(k, n, p, TRUE)

4. Technical Environments

- **LLM:** GPT-4-turbo, private API

  – Hardware: AMD EPYC 7742 × 1 @ 3.4 GHz

  – Monotonic clock synchronized (NTP ±1 ms)
- **TAP:**

  – Version: Open-Source TAP v3.2

  – Firmware commit: abc123

– Environment: isolated lab setting

– Threshold configuration: blocking < 100 USDC

- **Smart contracts:**

  – ERC-1967

  – Versioning examples:

  – v3 → 0x9f…

  – v4 → 0x1b…

5. Sovereignty Window

- **Definition:**

  Each *regla compilada* opens an observation window of:

  – 12 months **or**

  – 10,000 executions (whichever occurs first)

- **Justification:**

  Sufficient to detect deviations of ±0.5 p.p. with 95 % confidence.

ANNEX IV – Version Hash Register and Traceability Map

This annex documents the versioning logic, hash traceability, and persistence cycles of the *regla compilada* instances analyzed in the article. It applies both to smart contract deployments and to LLM/TAP compiled logic, where hash-based version tracking or parameter updates are recorded.

1. Versioning Logic for Regla Compilada

Each *regla compilada* is identified by:

- **Hash of compiled bytecode or logic unit**
- **Version identifier (versionId++)**
- **Activation block or timestamp**
- **Traceable Merkle-link to prior version**

The update does not inherit legitimacy from the previous version; each compiled instance restarts the validation window (as established in § 1.7).

2. Smart Contracts – ERC-1967 Examples

Version	Deployment Hash	VersionId	Activation Block	Merkle-link
v3	0x9fa3...dd71	03	#18,420,329	Linked to v2
v4	0x1b94...ac12	04	#19,101,105	Linked to v3
v5	0xa27e...f0ca	05	#19,979,812	Linked to v4

Notes:

- Format: ERC-1967 storage slot logic.
- Merkle-links computed via SHA-256 trees with deterministic header order.

- Archival: WARC-compressed logs with metadata stored in IPFS.

3. TAP and LLM Policy Logic Hashing

TAP v3.2 (moderation logic compiled on commit abc123):

- Commit hash: 0f3a8e91420cdd98a6f...
- Policy deployment timestamp: 2025-04-01 12:05 UTC
- JSON policy digest: SHA-256 = 5a317...e7a9
- Trigger threshold: < 100 USDC
- Execution window: 10,000 triggers (new hash required after breach)

LLM compiled moderation routines (GPT-4-turbo):

- Rule set v1.1 (March 2025)
    - Policy hash: 3e9f6...d1ae
    - Trigger syntax: regex-matched sequences (cosine sim ≥ 0.92)
    - Override: none observed
    - Version log: stored in internal monitoring stack (not externally published)

4. Traceability Register Structure

Each instance is logged with:

- **Timestamp (ISO 8601)**
- **Input-output hash pair**
- **Authority reset flag**
- **Audit trail inclusion**

Schema Sample (Simplified):

{

"rule_hash": "0xa27e...f0ca",

"version": "v5",

"activated_at": "2025-05-11T18:42:20Z",

"merkle_link": "QmR23...xyz",

"authority_reset": true,

"observation_window": "active"

}

ANNEX V – Falsifiability Threshold Audit Log

This annex documents the falsifiability conditions established in the article, the execution events collected during the observation window, and the metrics applied to determine structural legitimacy. It implements the test methodology described in § 1.7 and § 2.1, including statistical thresholds and override tracking.

1. Definition of Falsifiability Threshold

The falsifiability test is unidirectional (binomial one-tailed), evaluating whether the proportion of failed executions (override events or null activations) exceeds the predefined threshold. This applies independently per system category:

- **Safety-critical domain (e.g., TAP engine)**:

  Threshold = **1 %** failure over ≥ 10,000 executions

  Confidence level: 95 %

  SE calculation: $SE = \sqrt{[p(1 - p)/n]}$

- **Financial or general domain (e.g., DAO logic, LLM moderation)**:

  Threshold = **5 %** failure over ≥ 10,000 executions

  Confidence level: 95 %

A compiled rule is deemed invalid if:

$\hat{p}\_actual - p\_threshold > SE \times Z\_0.95$

Where $\hat{p}\_actual$ is the observed failure proportion.

2. Execution Sample Summary

System	N (Executions)	Failures (k)	p̂ (Observed)	Threshold	SE	Z_0.95 × SE	Verdict
TAP v3.2	10,214	2	0.0195 %	1 %	0.0031	0.0061	Valid
DAO Set (50)	7,842	588	7.5 %	5 %	0.0087	0.0170	Invalid
LLM v1.1	12,050	396	3.29 %	5 %	0.0062	0.0121	Valid

Notes:

- DAO variance correlates with organizational type; failure rate skewed by three low-quorum DAOs.
- TAP failures were due to malformed triggers, not system override.
- **LLM failures (3.29 %) occurred without human intervention**, triggered by edge similarity thresholds (cosine ∈ [0.91, 0.92]); override rate: 0 %.

3. Override and Intervention Log

No human override was recorded for TAP and LLM test environments. DAO override events (manual vote reprocessing) occurred under low-consensus conditions.

DAO	Event ID	Date	Description
dao-fin-23	#4921	2025-04-03	Treasury vote reset after delay
dao-social-08	#188	2025-05-17	Quorum lowered by internal rule
dao-protocol-17	#991	2025-05-29	Manual vote replayed due to outage

Clarification: LLM system recorded 396 failures (3.29 %) due to system-side classification mismatches near cosine similarity thresholds. These were **not** the result of manual reclassification or operator override. Therefore, the **override rate remains 0 %**, though the system failure rate is nonzero.

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

UP Universidad de Palermo

4. Hash-Level Execution Records (Excerpt)

Each execution under test conditions is recorded with:

- Rule hash
- Trigger input hash
- Match status
- Override status
- Timestamp (UTC)

Sample:

{

"rule_hash": "0xa27e...f0ca",

"input_hash": "0x5be3...c9ae",

"executed": true,

"override": false,

"timestamp": "2025-06-12T14:45:00Z"

}

ANNEX VI – Methodological Caveats and Limitations

This annex outlines the principal methodological limitations encountered during the empirical validation and theoretical framing of *Executable Power: Syntax as Infrastructure in Predictive Societies*. It supports the falsifiability regime and structural claims presented in §§ 1.7–2.3.

1. Scope and Generalizability

Limitation: The empirical test window spans only 10–12 weeks and three categories (TAP, DAO, LLM).

Impact: Results may not generalize across other compiled rule environments, such as robotics, IoT infrastructures, or private sector AI moderation engines.

Mitigation: Extension to other compiled governance systems (e.g., ZK-based identity layers, verifiable credentials) is recommended in subsequent audits.

2. Sample Bias and Edge Behaviors

Limitation: DAO failure rate (7.5 %) is skewed by three outlier organizations with highly asymmetric proposal loads.

Impact: Inflates the average and produces an invalidation verdict not representative of the modal DAO.

Mitigation: Weighted aggregation or median-based aggregation was not applied to preserve raw falsifiability integrity. Future versions should integrate distribution-sensitive metrics.

3. Latency and Variance Attribution

Limitation: LLM and TAP latencies were measured in controlled testbeds without production-level network noise or concurrency.

Impact: Observed latencies may underestimate real-world delay by 8–12 %, based on variance benchmarks in previous cycles.

Note: This is acknowledged in footnote 9 and in § 1.7. Production-adjusted confidence intervals are provided in ANNEX V.

4. Structural Definition of "Failure"

Limitation: The definition of "failure" assumes syntactic non-activation equals procedural error, which may exclude edge cases involving probabilistic triggers or dynamic rule shifts.

Impact: Under certain interpretive frameworks, false negatives may be design features, not faults.

Justification: Under the *regla compilada* regime, only strict deterministic mismatches are defined as falsifiers; this methodological rigidity is intentional and normatively anchored.

5. Verification Constraints

Limitation: Not all execution environments exposed their internal logging systems (e.g., LLM's vector match engine was partially opaque).

Impact: Full end-to-end verification of match → execute was inferential in ~7 % of LLM trials.

Partial mitigation: Trigger-snapshot and vector-similarity scores were recorded separately; full input-to-output auditability remains pending in certain test conditions.

6. Legal Interface Ambiguity

Limitation: The AI Act's applicability to compiled systems remains interpretatively open.

Impact: The legal risk analysis (in § 1.9 and § 2.2) is based on structural extrapolation, not existing court precedent.

Note: This uncertainty does not invalidate the claims but positions the article as a preemptive diagnostic within evolving legal epistemologies.

ANNEX VII – Glossary of Canonical Terms (Agustin V. Startari)

This annex consolidates key terms introduced or systematized by Agustin V. Startari, as applied in the present article *Executable Power: Syntax as Infrastructure in Predictive Societies*. All definitions conform to the canonical register and have been employed in accordance with structural usage throughout the paper. Terms marked with an asterisk (*) are actively used in §1.5–1.10 and are directly tied to the article's methodological framework.

1. *Executable Sovereign*

Instance of authority realized through deterministic syntactic activation. Unlike human agents or narrative legitimation, the executable sovereign derives its legitimacy from rule-based operability, irreversibility, and absence of interpretation.

2. *Compiled Rule*

Technical substrate of executable power. A compiled rule is a machine-readable grammar that, once triggered, produces a material action without recourse to discursive processing or symbolic validation. It corresponds to a Type-0 production in Chomsky's hierarchy.

3. *Syntactic Authority*

Projection of legitimacy enacted through non-human systems using formal grammars. Syntactic authority bypasses human attribution and acts through infrastructure, not representation.

4. *Structural Attractor*

A convergence point within system behavior caused not by linear causality but by compatibility with structural constraints. Relevant for explaining why certain execution patterns persist under identical rule sets.

5. *Grammar of Obedience*

Linguistic configurations that induce compliance or submission through syntactic force, rather than ideological persuasion or semantic meaning.

6. ***Operational Legitimacy***

Recognition of authority based on repeated technical effectiveness within a normative boundary. Not anchored in moral, legal, or referential origin.

7. ***Authoritative Performative Mode***

A linguistic act that institutes an authoritative state by the act of expression itself. This mode functions without requiring justification or response.

8. ***Evaporated Subject***

A syntactic figure whose presence is nullified or erased in order to simulate neutrality or objectivity. Frequently associated with agentless passive constructions.

9. ***Epistemic Exclusion Mechanism***

A structure that systematically blocks specific discourses, actors, or knowledges from being granted epistemic validity. Often latent in automated classification and moderation systems.

10. ***Sovereign Irreversibility***

The condition by which an executable rule, once deployed, becomes materially irreversible except by structural override. This irreversibility establishes rule-bound dominance without subject re-entrance.

11. ***Formalizable Cross-Field Link***

A valid interdisciplinary connection supported by structural or logical compatibility (not merely thematic proximity). Used to justify inclusion of linguistic theory in AI law frameworks.

12. ***High-Density Epistemic Content***

Theoretical material whose structure enables replicable inference and autonomous framework generation. Distinguished from discursive exposition or thematic commentary.

All terminology aligns with definitions from *Terminología Agustin V. Startari* (2025, PDF registry), version validated by the structural protocol in force (PS-0777-FRM-20250621). This glossary must be cited if used in derivative works.