

ESG-as-Code: A Deterministic Rule-Based Framework for Automated ESG Compliance Validation

¹Isaiah Oluwasegun Owolabi

¹ESGine Inc., Washington DC, United States

Abstract

Environmental, Social, and Governance (ESG) compliance has shifted from voluntary best practice to enforceable legal obligation across major global jurisdictions. Frameworks such as the European Union's Corporate Sustainability Reporting Directive (CSRD), the Sustainable Finance Disclosure Regulation (SFDR), the United Kingdom's Sustainability Disclosure Requirements (SDR), and disclosure rules from the U.S. Securities and Exchange Commission (SEC), adopted in 2024 and now under proposed rescission, impose structured reporting duties on a growing range of organizations. Yet dominant compliance methods remain manual, fragmented, and difficult to verify. Static documentation, subjective interpretation, and scoring systems that cannot be traced back to specific legal provisions continue to define current practice.

This paper introduces ESG-as-Code, a deterministic rule-based framework designed to address these limitations by converting ESG regulatory obligations into structured rule logic formally specified for machine implementation. Drawing on established principles from Policy-as-Code, Infrastructure-as-Code, and computational law, the framework provides a structured methodology for encoding jurisdictional ESG rules as conditional logic that can be evaluated systematically against organizational data and disclosures.

Central to the framework is a deliberate separation between probabilistic systems used for document interpretation and deterministic engines used for compliance decision-making. This separation preserves full auditability, allowing every compliance outcome to be traced directly to a specific regulatory provision rather than a statistical inference. Determinism in this sense guarantees reproducibility and rule-level traceability, properties that probabilistic scoring systems cannot offer by design. It does not by itself guarantee that a given outcome is legally correct; that additionally depends on the accuracy of the underlying rule encoding, the currency of the regulatory source, and the governance processes surrounding the rule library. The paper argues that reproducibility and traceability are necessary but not sufficient conditions for regulatory defensibility and proposes deterministic rule-based architecture as the foundation on which the remaining conditions can be built.

Keywords: ESG compliance, regulatory automation, deterministic systems, Policy-as-Code, RegTech, computational law, CSRD, sustainability reporting

1. Introduction

The global regulatory landscape governing Environmental, Social, and Governance (ESG) reporting has undergone a fundamental transformation over the past decade. What was once a voluntary exercise in corporate transparency has become a structured legal obligation enforced by regulators across multiple jurisdictions. The European Union's Corporate Sustainability Reporting Directive (CSRD), which entered into force on 5 January 2023, requires thousands of companies operating within the EU to disclose detailed sustainability information aligned with the European Sustainability Reporting Standards (ESRS) (European Parliament and Council of the European Union, 2022). At the same time, the Sustainable Finance Disclosure Regulation (SFDR) imposes disclosure obligations on financial market participants, while the United Kingdom's Sustainability Disclosure Requirements (SDR) and the International Sustainability Standards Board (ISSB) frameworks continue to shape global reporting expectations (European Parliament and Council of the European Union, 2019). In the United States, the Securities and Exchange Commission adopted climate-related disclosure rules in March 2024, stayed them pending litigation, and proposed their rescission in 2026, adding further complexity and uncertainty for multinational organizations navigating cross-border compliance obligations (Wong and Dewdney, 2023; U.S. Securities and Exchange Commission, 2024, 2026).

Despite the growing weight of these regulatory demands, the tools and processes most organizations use to manage ESG compliance have not kept pace. The dominant approach remains largely manual. Compliance teams rely on spreadsheets, static PDF reports, and consultant-driven assessments to gather, interpret, and report ESG data. These methods are not only time-consuming and expensive but are also poorly suited to the demands of modern regulatory enforcement. Manual processes introduce inconsistency, create audit gaps, and make it difficult for organizations to demonstrate that their disclosures are directly traceable to specific regulatory provisions. For regulators attempting to detect greenwashing or verify the accuracy of sustainability claims, the opacity of current reporting systems presents a significant challenge (Papantoniou, 2022).

The emergence of ESG technology platforms has provided organizations with new tools for managing sustainability reporting and compliance obligations, but it has also introduced additional challenges. Many existing platforms rely on artificial intelligence and machine learning systems to generate ESG scores or sustainability ratings. Although these systems can process large volumes of unstructured data efficiently, their outputs are often probabilistic and may not always be fully explainable or directly traceable to specific legal requirements. In regulatory environments where accountability and auditability are critical, this distinction becomes particularly important (El Harras and Salahddine, 2025). Organizations that cannot explain how a compliance determination was reached are exposed to legal and reputational risk, regardless of what their ESG dashboard reports. Recent literature confirms that RegTech solutions, while improving compliance efficiency, must be designed with transparency and auditability as foundational requirements rather than secondary considerations (Golzarjannat and Gustafsson, 2026).

This paper responds to these challenges by introducing ESG-as-Code, a compliance engineering approach that expresses ESG regulatory obligations as deterministic, rule-based logic rather than probabilistic inference (Owolabi, 2025b). The framework draws on established approaches from Policy-as-Code, Infrastructure-as-Code, and computational law to propose a structured methodology for encoding jurisdictional ESG rules as conditional logic statements that can be evaluated systematically against organizational disclosure data (HashiCorp, 2025; Owolabi, 2025a). Rather than relying on probabilistic inference, ESG-as-Code encodes regulatory requirements as explicit conditional logic where every compliance determination is traceable to a specific rule, and every rule is traceable to a specific legal provision.

The primary contributions of this paper are as follows. First, it provides a formal conceptual definition of ESG-as-Code as a compliance engineering framework distinct from existing ESG technology approaches. Second, it proposes a jurisdictional logic model for encoding cross-border ESG obligations as structured conditional rules. Third, it presents an illustrative walkthrough using CSRD Article 19a to show conceptually how regulatory text could be translated into executable compliance logic without reliance on probabilistic systems. Fourth, it discusses the implications of this approach for regulators, compliance practitioners, and developers working in the RegTech space.

This paper's contribution and evaluation are scoped to the compliance validation and applicability-determination layers of the framework described in Sections 3 and 4; the reporting and audit layer introduced in Section 3.3 is presented as an architectural design target that the validation layer is intended to feed, not as a capability demonstrated in this paper, and generating disclosure-ready output packages, including taxonomy tagging, template population, and submission formatting, remains future work.

The remainder of this paper is structured as follows. Section 2 reviews relevant literature across Policy-as-Code, computational law, and ESG automation. Section 3 presents the ESG-as-Code framework, including its core principles and conceptual architecture. Section 4 provides the jurisdictional logic model and an illustrative walkthrough. Section 5 discusses broader implications and limitations of the framework. Section 6 concludes the paper.

2. Literature Review

2.1 Infrastructure as Code and Policy as Code

The concept of treating operational configurations as version-controlled, machine-readable code originated in software engineering and DevOps practice. Infrastructure as Code (IaC) developed as a practice for managing and provisioning computing infrastructure through machine-readable definition files rather than manual configuration processes or interactive tools (Kumara et al., 2021). The core insight of IaC is that applying software engineering practices such as version control, automated testing, and peer review to infrastructure management produces more consistent, auditable, and scalable systems than manual alternatives (Morris, 2021). This principle, that operational processes can be represented as structured and executable code, underpins the wider family of “as-code” paradigms used across modern software and governance systems.

Policy as Code (PaC) extended this principle into the area of governance and compliance. Instead of relying solely on written documentation or manual enforcement processes, Policy as Code represents organizational and regulatory policies as structured logic that can be tested, versioned, and automatically enforced across systems (HashiCorp, 2025). In practice, PaC frameworks allow compliance checks to be executed programmatically, producing deterministic pass or fail outcomes against defined rule sets. The significance of this approach lies not merely in its

efficiency but in its auditability: when a policy is expressed as code, every enforcement decision can be traced back to the specific rule that produced it. This traceability property is foundational to the ESG-as-Code framework proposed in this paper.

2.2 Computational Law and Rules as Code

The translation of legal text into machine-executable logic has attracted growing scholarly attention under the headings of computational law and Rules as Code. Mohun and Roberts (2020) identified the conversion of regulatory rules into formats readable by both humans and machines as a significant emerging direction for public administration, noting that machine-readable legislation offers potential benefits for regulatory consistency, enforcement efficiency, and citizen-facing service delivery. Wong (2020) characterized this as a multi-level process of digitization, distinguishing between superficial digitization of legal documents and deeper formalization of legal logic into executable computational structures.

Research on translating legal texts into machine-executable logic has explored both the opportunities and the limitations of this approach. Studies have shown that rule-based engines can support the automation of public sector decision-making processes when the underlying legal obligations are clearly defined and sufficiently unambiguous (Bertl, Price and Draheim, 2026). At the same time, scholars have cautioned that the formalization of law into code carries risks of ossification, mistranslation of legislative intent, and challenges to judicial review if automated systems make consequential decisions without adequate human oversight (Legg and Bell, 2024). These tensions between automation and accountability are directly relevant to the design of ESG-as-Code, which explicitly reserves compliance decision-making for deterministic rule engines while keeping human judgment available for interpretation and review.

The Catala programming language, developed to formalize tax and benefits legislation as executable code, represents a practical attempt to bridge the gap between legal drafting and computational implementation (Merigoux, Chataing and Protzenko, 2021). These efforts demonstrate both the technical feasibility and the institutional complexity involved in translating legal obligations into machine-readable logic, providing an important precedent for ESG compliance contexts.

2.3 RegTech and ESG Compliance Automation

Regulatory Technology, or RegTech, is broadly defined as the use of information technology to enhance and support regulatory compliance processes (Golzarjannat and Gustafsson, 2026; Papantoniou, 2022). The field emerged in response to the growing complexity and cost of regulatory compliance following the 2008 financial crisis (Papantoniou, 2022), and has since expanded beyond financial services into sustainability and ESG reporting contexts. A systematic review of RegTech literature identified its core capabilities as including real-time compliance monitoring, automated data validation, structured risk assessment, and automated regulatory reporting (Golzarjannat and Gustafsson, 2026). The same review noted that RegTech solutions must be designed with transparency and auditability as foundational requirements, particularly in regulatory environments where enforcement consequences are significant.

The application of RegTech to ESG compliance presents challenges that distinguish it from earlier financial compliance use cases. ESG regulations are multi-jurisdictional, rapidly evolving, and frequently expressed in qualitative rather than quantitative terms. Cross-border compliance obligations, such as those faced by multinational organizations subject simultaneously to the CSRD, SFDR, and SEC disclosure requirements whose legal status has itself shifted between adoption, stay, and proposed rescission, require compliance systems capable of handling jurisdictional differences in rule logic while maintaining consistency and auditability across reporting processes (Wong and Dewdney, 2023). Manual compliance processes are structurally inadequate for this task, creating demand for automated approaches that can apply jurisdiction-specific rule sets systematically and verifiably.

2.4 Limitations of Probabilistic AI in ESG Compliance

A growing body of research has examined the application of artificial intelligence, particularly natural language processing and large language models, to ESG compliance challenges such as greenwashing detection and disclosure analysis. Studies suggest that AI-assisted analysis can improve the efficiency and accuracy of ESG disclosure reviews by helping identify vague language, missing quantitative targets, and inconsistencies in sustainability reporting (He and Wang, 2025; Sari et al., 2025). AI-based greenwashing detection tools have also shown potential

in cross-validating corporate ESG claims against regulatory standards and third-party data sources (Sari et al., 2025).

However, the probabilistic nature of machine learning systems introduces a fundamental limitation in regulatory compliance contexts. Probabilistic models produce outputs based on statistical inference from training data rather than deterministic evaluation of defined rules. In environments where compliance outcomes must be traceable to specific legal provisions, this distinction has material consequences. An organization that cannot explain how a compliance determination was reached, or that cannot map a compliance outcome directly to the regulatory article it reflects, faces significant legal and reputational exposure regardless of the accuracy of the underlying model. This limitation motivates the central design choice of ESG-as-Code: the separation of probabilistic AI systems, used for document interpretation and data extraction, from deterministic rule-based engines used for compliance decision-making. This separation manages the consequences of probabilistic uncertainty rather than eliminating it; Section 3.4 returns to this point directly.

2.5 Gap in Existing Literature

While the fields reviewed above provide important conceptual foundations, the specific combination of features ESG-as-Code proposes, jurisdiction-versioned rule libraries with explicit temporal and legal-status metadata applied to multi-jurisdictional ESG disclosure obligations, has not been demonstrated as a unified artifact in the literature surveyed for this paper. This is a narrower claim than asserting that no related work exists, and it is worth situating explicitly against several adjacent bodies of work rather than treated as self-evident.

LegalRuleML, an XML-based formal representation for legal norms distinguishing constitutive statements from prescriptive requirements, has been applied to domains including building-code compliance checking, and offers a mature representation language that ESG-as-Code's rule packages could in principle be expressed in rather than the generic notation used in Section 4.5 (Athanasopoulos et al., 2015). The Decision Model and Notation (DMN) standard, maintained by the Object Management Group, provides a widely adopted decision-table formalism for expressing business rules and has informed the notation used in this paper's worked example (Object Management Group, 2021). XBRL, the machine-readable tagging standard already mandated for financial

filings by the SEC and, under the European Single Electronic Format, by ESMA, demonstrates that structured, machine-readable regulatory submission is an established practice in adjacent reporting domains, though XBRL addresses tagging of disclosed data rather than deterministic evaluation of whether an obligation has been satisfied. Compliance-by-design approaches and regulatory ontologies have similarly sought to embed regulatory logic into operational systems, generally within single-jurisdiction financial compliance contexts rather than the cross-jurisdictional ESG setting this paper addresses. Most recently, hybrid neuro-symbolic approaches combining large language model-based extraction with code-generation or rule-execution back ends have been proposed for financial compliance checking, reflecting a broader research trend toward exactly the separation of probabilistic interpretation from deterministic evaluation that this paper's third design principle also adopts, and suggesting that the boundary problem discussed in Section 3.4 is an active area of inquiry rather than a solved one (see, e.g., recent code-generation approaches to compliance checking).

ESG-as-Code's contribution relative to this landscape is therefore best stated as an application and extension rather than a wholly novel category: it applies the traceability and versioning discipline of Policy as Code and LegalRuleML-style formalisms specifically to the cross-jurisdictional structure of ESG disclosure obligations, a domain in which, to the literature reviewed for this paper, no prior work has proposed the combination of a jurisdictional rule library, explicit rule-level temporal and legal-status metadata, and a differentiated non-binary outcome model of the kind presented in Sections 3.3 and 4.5.

This paper also extends the author's own prior work rather than restating it. Table 1 summarizes the distinction.

Table 1. Relationship to prior ESG-as-Code publications

Aspect	Owolabi (2025b), “What is ESG-as-Code™?”	Owolabi (2025a), “Operationalizing ESG-as-Code”	This paper
Primary contribution	Introduces the ESG-as-Code concept and terminology.	Describes a containerized deployment architecture (Kubernetes–OpenStack) for running ESG-as-Code pipelines.	Formalizes a rule representation, outcome model, and jurisdictional logic model, and evaluates a worked example.
Rule formalization	Not addressed.	Not addressed.	Provided (Section 4.5).
Outcome model	Not addressed.	Not addressed.	Provided (Section 3.3).
Temporal/legal-status metadata	Not addressed.	Not addressed.	Provided (Section 4.3, 4.5).
Evaluation	None.	None.	Preliminary internal-consistency evaluation (Section 4.6).
Focus	Conceptual introduction.	Infrastructure and deployment.	Compliance-logic formalization and evaluation.

3. The ESG-as-Code Framework

3.1 Overview and Motivation

In this paper, ESG-as-Code denotes the compliance engineering discipline the framework applies: expressing ESG regulatory obligations as deterministic rule logic formally specified for machine implementation and evaluating organizational disclosure data against that logic to produce an auditable compliance determination. Machine execution is an intended design objective of the broader framework; the artifact evaluated in this paper is a formal specification manually traced through illustrative scenarios rather than an executable implementation. The reporting and audit layer introduced in Section 3.3 would extend this discipline toward regulatory submission, but is presented there as an architectural design target rather than a capability this paper demonstrates or evaluates.

The designation ESG-as-Code® reflects the framework’s UK-registered trademark (UK Intellectual Property Office No. UK00004216322, registered September 2025); the mark appeared as ESG-as-Code™ in earlier materials, including the author’s own prior publications cited throughout this paper (Owolabi, 2025a, 2025b), reflecting its status as an unregistered claim at the time those works were published. The framework is motivated by three converging conditions in the current ESG compliance landscape: the rapid expansion of mandatory ESG reporting obligations across multiple jurisdictions; the structural inadequacy of manual compliance

processes at the scale and pace that modern regulation demands; and the limitations of probabilistic AI systems in environments where compliance decisions must be fully traceable to specific legal provisions.

The name reflects a deliberate conceptual alignment with the as-code paradigm established by Infrastructure as Code and Policy as Code. Just as those approaches brought the rigor of software engineering practices to infrastructure management and governance policy, ESG-as-Code applies the same structural principles to the domain of sustainability regulation. The core premise is that ESG regulatory obligations, once expressed as structured conditional logic, can be systematically evaluated, versioned, tested, and audited in ways that manual compliance processes fundamentally cannot support.

The remainder of this paper relies on five terms used in a specific, consistent sense. A **rule** is a single conditional logic statement tagged to one authoritative legal source, evaluated independently of other rules. An **obligation** is the underlying legal duty a rule is written to test, typically narrower than the full text of the provision it derives from. **Evidence** refers to the disclosure data, whether human-entered, AI-extracted, or system-calculated, that a rule consumes as input. A **condition** is a single testable element within a rule, either a scope condition determining whether the rule applies or a compliance condition determining what it requires once triggered. A **compliance determination** is the output the rule engine produces for a given entity, rule, and reporting period, drawn from the outcome model in Section 3.3. This vocabulary also marks a boundary the paper observes throughout: legal compliance, whether an obligation is satisfied under the governing law, is distinct from completeness checking, whether the required fields were populated, and from disclosure-quality assessment, whether the populated fields meet a qualitative standard such as clarity or specificity. ESG-as-Code's rule engine performs the first of these three; the outcome states in Section 3.3, particularly `INSUFFICIENT_EVIDENCE` and `DATA_QUALITY_FAILURE`, exist precisely to keep the second and third from being silently folded into the first.

3.2 Core Principles

The ESG-as-Code framework is built on four core principles that distinguish it from both traditional compliance management approaches and existing ESG technology solutions.

The first principle is regulatory primacy. Every rule in the ESG-as-Code system originates from a specific legal provision. Rules are not derived from industry conventions, scoring methodologies, or proprietary assessments. Each rule traces directly to the regulatory article, section, or standard from which its obligation arises. This traceability chain, from compliance output back to legal source, is non-negotiable and forms the foundation of the framework's defensibility in enforcement contexts.

The second principle is determinism. Compliance decisions within the ESG-as-Code framework are produced by deterministic rule engines rather than probabilistic models. Given the same inputs, the system produces the same output each time. This property is particularly important in regulatory environments because it helps ensure that compliance outcomes remain reproducible, explainable, and consistent across reporting periods. As enterprise compliance literature has noted, compliance checks must be repeatable and justifiable, and deterministic rule-based systems are generally better suited to providing this level of consistency than probabilistic models (Elementum AI, 2026).

Determinism, however, is a guarantee about consistency, not about correctness. A deterministic engine will reproduce an incorrect result with the same reliability as a correct one if the underlying rule encodes an obsolete provision, misreads a scope condition, or is built on stale source data. Legal defensibility is therefore a broader property than determinism, resting additionally on the accuracy of the authoritative source mapping, ongoing legal review of encoded rules, data lineage back to disclosure inputs, effective-date and version controls, and a defined process for challenge or override. ESG-as-Code is designed to make these additional requirements tractable, principally by keeping every rule traceable to a single, versioned legal source, but determinism alone does not satisfy them.

The third principle is separation of concerns. ESG-as-Code explicitly separates the functions of document interpretation and data extraction from compliance decision-making. Probabilistic AI systems, including large language models and natural language processing tools, may be used in the former role to process unstructured ESG data and extract relevant disclosure information. However, once data enters the compliance validation layer, all decisions are made by deterministic

rule logic. This separation preserves the efficiency benefits of AI-assisted processing while ensuring that no compliance determination depends on a statistical inference that cannot be traced to a defined rule.

3.3 Conceptual Architecture

Figure 1. ESG-as-Code Conceptual Architecture

The first is the data ingestion and interpretation layer. This layer is responsible for receiving raw ESG data inputs, which may include structured datasets, sustainability reports, financial

disclosures, and operational records. Where inputs are unstructured, AI-assisted tools may be applied at this layer to extract relevant data points, normalize formats, and flag missing or ambiguous disclosures. All outputs from this layer are expressed as structured data records that can be evaluated by the rule engine.

The second is the jurisdictional rule engine layer. This is the core compliance processing layer of the framework. It contains a library of jurisdiction-specific compliance rules, each expressed as conditional logic statements derived from specific regulatory provisions. Rules are organized by jurisdiction, regulatory instrument, reporting standard, and applicable entity type. When organizational data enters this layer, the engine evaluates each applicable rule against the data and produces a deterministic outcome for each rule, drawn from a defined outcome model rather than a simple pass/fail/not-applicable result. This outcome model distinguishes SATISFIED and NOT_SATISFIED determinations from outcomes that reflect a problem with the evaluation itself rather than with the entity's underlying compliance: INSUFFICIENT_EVIDENCE, where required data was not supplied; DATA_QUALITY_FAILURE, where supplied data failed a validation check; INTERPRETATION_REQUIRED, where the rule's applicability or satisfaction depends on a qualitative judgment outside the rule engine's competence; MATERIALITY_REVIEW_REQUIRED, where satisfaction depends on a materiality assessment reserved for human judgment; RULE_CONFLICT, where two applicable rules produce contradictory requirements; NOT_YET_EFFECTIVE, where the rule's entry-into-force or first-applicable-period has not yet arrived for the entity in question; NOT_APPLICABLE, where the entity's profile does not meet the applicability or scope conditions for the rule in the reporting period in question; SUPERSEDED, where a later rule version has replaced the one that would otherwise apply; and EXTRACTION_ERROR, where an upstream AI-assisted extraction step failed or produced a result outside expected bounds. This distinction matters because a binary engine that returns only PASS or FAIL implicitly converts every one of these distinct conditions into a false positive or false negative, producing confident-sounding output that misrepresents what the engine actually knows. Every result is tagged with the specific rule that produced it and the regulatory provision from which that rule derives.

Outcome Precedence and Transition Semantics

The outcome states above are not evaluated independently of one another; they form an ordered pipeline in which an earlier stage that produces a terminal or gating result prevents later stages from running. This section specifies that ordering directly, since an unordered outcome model would leave open exactly the ambiguities a reviewer or auditor would need resolved: whether a data problem masks a substantive failure, how a conflict between two applicable rules is detected rather than merely labeled, whether a superseded rule version remains usable for the periods it governed, and when an open human-review item becomes a final answer.

Stage 0, temporal and scope resolution, runs first, before any disclosure data is examined. The engine determines which rule version, if any, governs the entity's reporting period using the temporal metadata described in Section 4.3. **NOT_YET_EFFECTIVE** is returned if no version's effective date has arrived for that period, and no later stage runs. If a version has been superseded by a later one, but the reporting period in question is one the earlier version governed, evaluation proceeds under that earlier version rather than being blocked; the output is tagged with the version actually used, so a determination made under a since-superseded version remains fully valid and independently auditable for the period it governed. **SUPERSEDED** therefore describes which version produced a determination, not a rejection of that determination. If the entity's profile does not meet the applicability conditions of the governing version, **NOT_APPLICABLE** is returned and no later stage runs.

Stage 1, data integrity, runs next, before the rule's substantive logic. **EXTRACTION_ERROR** is checked first, ahead of evidence sufficiency: if an upstream AI-assisted extraction step failed or produced a result outside expected bounds for a required input, the engine cannot yet determine whether the underlying disclosure is actually missing or simply unread, and returns **EXTRACTION_ERROR** rather than treating a pipeline fault as though it were a disclosure gap. Only once extraction is confirmed to have run correctly does the engine check whether required fields are populated at all, returning **INSUFFICIENT_EVIDENCE** for missing fields, and then whether populated fields pass basic validation, returning **DATA_QUALITY_FAILURE** for a field that is present but invalid, for example non-numeric or negative where a positive figure is required.

Stage 2, conflict detection, runs after data integrity is confirmed and before this rule's own compliance logic executes. Before producing a substantive result, the engine checks whether any other rule applicable to the same obligation and reporting period imposes a requirement that directly contradicts this rule's own compliance conditions, for example two applicable rules specifying incompatible disclosure formats for the same figure. **RULE_CONFLICT** is a pre-check on the applicable rule set, not an inference drawn after two rules have separately produced disagreeing SATISFIED and NOT_SATISFIED outputs; contradictions are intended to be caught before either rule is allowed to produce a final substantive result. A complete conflict-detection procedure across the full rule library, rather than the pairwise check specified here, is future work identified in Section 5.5.

Stage 3, human-judgment gates, applies where the rule's satisfaction depends on a determination the deterministic layer is not designed to make. **INTERPRETATION_REQUIRED** applies where the rule's applicability or satisfaction turns on ambiguous or contested scope language, and **MATERIALITY_REVIEW_REQUIRED** applies specifically where satisfaction depends on a materiality determination and no completed, dated materiality record is present for the applicable reporting period. Neither state is resolved by the engine itself: each converts to a final SATISFIED or NOT_SATISFIED determination only when the required human input, a documented interpretation decision or a completed, dated materiality record, is supplied as a new input and the rule is re-evaluated against it. An unconverted gate is treated as an open item for audit purposes rather than defaulted to either a pass or a fail.

Stage 4, substantive evaluation, runs only if no earlier stage has produced a terminal result or an open gate. Only at this stage does the rule's own compliance logic execute, producing SATISFIED or NOT_SATISFIED. The overall precedence is therefore: temporal and scope resolution, data integrity, conflict detection, human-judgment gates, and finally substantive evaluation, with an outcome at any earlier stage taking priority over one that a later stage would otherwise have produced.

The third is the reporting and audit layer, an architectural layer this paper specifies conceptually rather than demonstrates. As designed, this layer would compile the outputs of the rule engine into

structured compliance reports, dashboards, and audit trails, which organizations would use to produce regulatory submissions, internal compliance reviews, and evidence packages for third-party assurance. Because every output generated within this layer can be traced back through the rule engine to its originating legal provision, the framework is designed to support both limited and reasonable assurance requirements under the CSRD and related frameworks (European Parliament and Council of the European Union, 2022). The worked example in Section 4.5 demonstrates the validation layer that would feed this reporting layer, not the reporting layer itself.

3.4 The AI and Deterministic Logic Separation in Practice

The separation between probabilistic AI and deterministic rule logic is the most operationally significant design choice in the ESG-as-Code framework. Current ESG technology platforms frequently blend AI-generated analysis with compliance outputs in ways that obscure the distinction between inference and rule application. A tool that uses a language model to assess whether a disclosure sufficiently addresses a regulatory requirement is making a probabilistic judgment, not a deterministic compliance evaluation. The output of such a tool cannot be reliably mapped to the regulatory provision it purports to address, and the same input may produce different outputs depending on model version, prompt configuration, or training data.

ESG-as-Code treats this as a categorical design flaw for regulatory compliance purposes. The framework uses probabilistic AI only for tasks where statistical inference is appropriate and where its outputs feed into a subsequent deterministic evaluation rather than producing compliance conclusions directly. For example, a language model may be used to extract the number of employees reported in a sustainability disclosure. That extracted figure then becomes an input to a deterministic rule: if reported employees exceed 1,000 and jurisdiction equals EU, then CSRD Article 19a reporting obligation is triggered. The compliance determination is made by the rule, not by the model. This architecture ensures that every compliance outcome is auditable and reproducible regardless of the AI tools used upstream, and that responsibility for the compliance determination itself rests with a rule that can be inspected, versioned, and challenged rather than with an opaque model. Legal defensibility follows from this auditability only when the underlying rule encoding is itself accurate and current, a condition the framework supports through versioning but does not by itself guarantee.

This separation should not be read as implying that uncertainty disappears once data crosses from the interpretation layer into the rule engine. An AI-extracted figure can be incomplete, internally disputed across source documents, or misclassified, and some legal predicates, most notably materiality judgments of the kind discussed in Section 5.4, require human judgment that no rule engine is designed to supply. The framework's contribution is not to eliminate this uncertainty but to govern it: as the provenance fields in the rule package presented in Section 4.5 illustrate, every AI-derived input retains its source type, confidence score, and extraction model version through to the audit output, so that a reviewer can see precisely which inputs carried residual uncertainty and which were directly supplied or independently validated. Final decision authority within the rule engine rests entirely with the deterministic logic; the uncertainty present throughout the upstream pipeline is preserved as visible metadata rather than resolved away and remains available for human review wherever the outcome model in Section 3.3 flags a condition, such as `INSUFFICIENT_EVIDENCE` or `INTERPRETATION_REQUIRED`, that the deterministic layer is not designed to resolve on its own.

3.5 Research Method: Design Requirements and Evaluation Approach

This paper follows a design-science research approach: an artifact, the ESG-as-Code framework and its rule representation, is constructed in response to identified problems in existing ESG compliance practice and is evaluated against explicit criteria rather than presented as a finished, empirically validated system.

The work is organized around three research questions.

- **RQ1:** How can ESG regulatory obligations, which are multi-jurisdictional, subject to frequent amendment, and expressed in a mixture of quantitative and qualitative terms, be represented as structured rules without sacrificing traceability to their legal source?
- **RQ2:** What rule-level information is required for a compliance determination to remain auditable not only at the moment it is produced but also retrospectively, after the underlying regulation has been amended or superseded?
- **RQ3:** Where should the boundary between probabilistic AI assistance and deterministic rule evaluation be drawn so that efficiency gains from AI do not compromise the traceability of the final determination?

These questions yield four design requirements that the framework in Section 3 and the rule representation in Section 4.5 are built to satisfy:

- 1) every rule must carry an explicit, versioned link to its authoritative legal source, satisfying RQ1;
- 2) every rule must carry temporal and legal-status metadata sufficient to determine which version applied to a given entity in a given reporting period, satisfying RQ2;
- 3) the outcome model must distinguish a substantively unmet obligation from an evaluation-layer problem such as missing evidence or an unresolved materiality judgment, also addressing RQ2; and
- 4) AI-derived inputs must retain source, confidence, and model-version metadata through to the audit output without that metadata influencing the deterministic evaluation itself, satisfying RQ3.

Evaluation of the artifact proceeds at two levels appropriate to its current stage of development. First, Section 4.6 presents a preliminary internal-consistency evaluation of a single worked rule package against a small set of synthetic entity profiles, intended to demonstrate that the design requirements above can be satisfied in a concrete instance rather than only asserted in the abstract. Second, Section 5.4 identifies the threats to validity that this limited evaluation does not address, and Section 5.5 sets out the empirical work, larger rule libraries, synthetic entity populations, and comparison against expert-produced determinations, that would be required before claims of real-world effectiveness could be supported.

4. Jurisdictional Logic Model and Illustrative Walkthrough

4.1 The Jurisdictional Logic Model

A central technical contribution of the ESG-as-Code framework is the jurisdictional logic model, a structured approach to encoding ESG regulatory obligations as conditional rule logic organized by jurisdiction, regulatory instrument, entity type, and disclosure requirement. The model is designed to address one of the most persistent operational challenges in multinational ESG compliance: the need to manage materially different regulatory obligations across jurisdictions without sacrificing consistency, traceability, or auditability.

The jurisdictional logic model operates on four structural components. The first is the entity profile, which captures the organizational attributes necessary to determine which regulatory obligations apply to a given entity. These attributes include employee count, net annual turnover,

balance sheet total, jurisdiction of incorporation, jurisdictions of operation, sector classification, and listing status. The second is the rule library, which contains jurisdiction-specific compliance rules expressed as conditional logic statements. Each rule in the library is tagged with the regulatory instrument, article, and section from which it derives. The third is the obligation mapping engine, which evaluates an entity's profile against the rule library to determine which obligations apply and whether they are being met based on available disclosure data. The fourth is the compliance output, which records the result of each rule evaluation, including the rule triggered, the regulatory provision it reflects, the data inputs evaluated, and the outcome produced, drawn from the outcome model described in Section 3.3.

This architecture allows the framework to handle the jurisdictional complexity of the current ESG regulatory environment. An organization operating in both the EU and the United States may be subject simultaneously to CSRD reporting obligations under Directive 2022/2464/EU, SFDR disclosure requirements under Regulation 2019/2088/EU, and SEC climate disclosure rules, adopted in March 2024, stayed pending litigation, and proposed for rescission in 2026. Each of these instruments carries distinct scope thresholds, materiality definitions, and reporting standards. The jurisdictional logic model accommodates this complexity by maintaining separate, independently versioned rule sets for each instrument that can be applied in combination based on an entity's profile without conflating obligations across regulatory regimes.

4.2 Encoding Regulatory Rules as Conditional Logic

The process of translating a regulatory provision into a compliance rule within the ESG-as-Code framework follows a structured methodology. The starting point is the primary legal text of the relevant regulation. The relevant obligation is identified, decomposed into its constituent conditions and requirements, and expressed as a conditional logic statement that maps directly onto the original provision.

This process requires careful attention to the scope conditions embedded in regulatory text. Many ESG obligations are conditional on entity size, sector, jurisdiction, or listing status. The CSRD applies different reporting obligations depending on whether an undertaking qualifies as a large undertaking under the Accounting Directive, whether it is a public interest entity, and whether it

has already been subject to the predecessor Non-Financial Reporting Directive (NFRD) (European Parliament and Council of the European Union, 2022). These scope conditions are not merely administrative details but foundational components of the compliance logic that must be encoded accurately for the system to produce legally defensible outputs.

The Omnibus simplification proposals published by the European Commission in February 2025 highlight the importance of maintaining versioned rule libraries that can evolve alongside changing regulatory requirements. That process moved through several distinct legal stages and produced two separate instruments that should not be conflated. The scope and threshold changes, including the revised employee and turnover thresholds discussed below, derive their legal effect from Directive (EU) 2026/470 of the European Parliament and of the Council of 24 February 2026, which amended the CSRD and related legislation and was published in OJ L, 2026/470, 26.2.2026 (European Parliament and Council of the European Union, 2026). Separately, the European Commission adopted a revised delegated act setting out updated European Sustainability Reporting Standards on 3 July 2026. Adoption did not itself make those revised standards immediately applicable: the delegated act was transmitted to the European Parliament and the Council for scrutiny and applies only after the applicable scrutiny period has ended and the act has entered into force (European Commission, 2026). Under the thresholds established by Directive (EU) 2026/470, the employee threshold for mandatory CSRD reporting was revised from 250 to 1,000 employees, while a new net annual turnover threshold of EUR 450 million was introduced for EU entities. An ESG-as-Code system that cannot accommodate such changes through targeted rule updates, without requiring a wholesale redesign of the compliance architecture, may struggle to remain operationally resilient in real-world compliance environments. Versioned rule libraries, similar to version control practices in software engineering, help address this challenge directly.

4.3 Illustrative Walkthrough: CSRD Article 19a and Scope 3 Emissions Disclosure

The following discussion illustrates, at a conceptual level, how a specific regulatory obligation under the CSRD could be decomposed into the kind of deterministic compliance logic the ESG-as-Code framework proposes. This walkthrough is descriptive rather than a system specification and does not disclose implementation details; it uses the Scope 3 emissions disclosure obligation

to make the framework's core principles concrete, and it deliberately traces that obligation through its full legal chain rather than treating it as arising from a single article.

The obligation does not originate from a single, self-executing provision. Article 19a of Directive 2013/34/EU, as inserted by the CSRD (Directive (EU) 2022/2464), establishes the general obligation for large undertakings to include sustainability information in their management report, covering environmental, social, and governance matters at a level of detail specified elsewhere in the framework (European Parliament and Council of the European Union, 2022). The specific content of that obligation, including the requirement to disclose gross Scope 1, 2, and 3 greenhouse gas emissions, arises not from Article 19a directly but from the delegated act adopted under it: Commission Delegated Regulation (EU) 2023/2772 of 31 July 2023, which sets out the European Sustainability Reporting Standards (ESRS), including ESRS E1 on climate change. ESRS E1 mandates disclosure of gross Scope 3 emissions across significant upstream and downstream value chain categories, limits the use of carbon offsetting or removals in reported figures, and requires the disclosure to be aligned with the double materiality principle established in ESRS 1 and ESRS 2 (Commission Delegated Regulation (EU) 2023/2772, 2023; Anthesis Group, 2026). A rule encoding this obligation must therefore trace to two distinct legal instruments, not one: the directive establishing the reporting duty, and the delegated act specifying its content, each carrying its own enactment date, entry-into-force date, and amendment history.

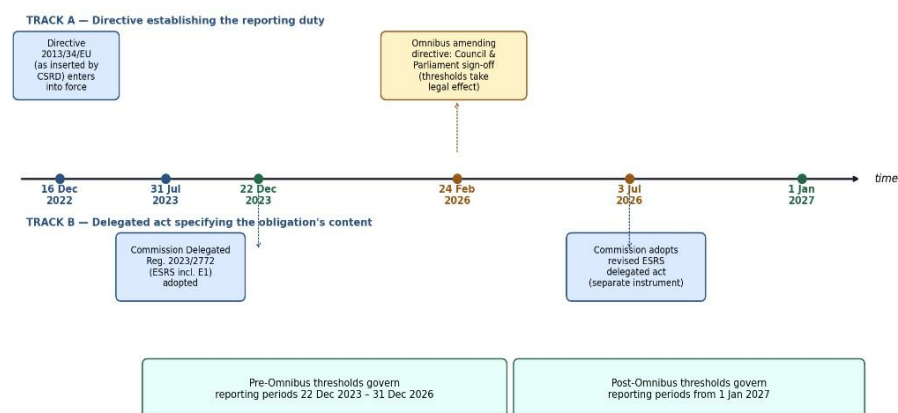
This two-instrument structure has practical consequences for how the obligation must be modeled. Since the ESRS were first adopted in 2023, their regulatory setting has changed through a targeted "quick fix" postponement (July 2025), Directive (EU) 2026/470 amending the CSRD scope and thresholds, and a further round of revised standards adopted by the European Commission on 3 July 2026 (European Parliament and Council of the European Union, 2026; European Commission, 2026). The 3 July 2026 Commission adoption is distinct from legal applicability: the revised ESRS remain subject to European Parliament and Council scrutiny and apply only once the scrutiny period has ended and the delegated act has entered into force. Under Directive (EU) 2026/470, the employee threshold for mandatory CSRD reporting was raised from 250 to 1,000 employees, and a net annual turnover threshold of EUR 450 million was introduced for EU entities, with the relevant post-Omnibus reporting timetable reflected in the rule versioning described below. A rule encoding this obligation as it stood in 2023, without a mechanism for recording

subsequent amendments and their applicable dates, would misstate both who the obligation applies to and when it takes effect for any given entity.

This is why the ESG-as-Code framework treats time and legal status as first-class attributes of every rule rather than incidental metadata. Each rule in the library is designed to carry, at minimum: its authoritative source instrument and article or standard reference; the date that instrument was adopted by the issuing body; its publication date in the Official Journal or equivalent register; its entry-into-force date; where the source is a delegated act, the date any applicable Parliament or Council scrutiny period concluded without objection; its date of transposition where the source is a directive requiring member state implementation; the date from which it is applicable to a given entity, which may postdate entry into force; the first reporting period to which it applies; any transitional or phase-in relief available to specific entity categories; whether it has been superseded, and if so by which subsequent instrument and as of what date; and, where applicable, the entity "wave" to which a given threshold or timeline applies. Critically, adoption by the issuing body is treated as necessary but not sufficient evidence that a rule is applicable: a delegated act can be adopted and not yet in force, in force and not yet applicable to a given entity, or subject to a scrutiny period during which its terms remain provisional. A rule evaluation engine built on this model determines which version of a rule applied to a specific entity during a specific reporting period, rather than applying whatever version of the regulatory text is most current at the time of evaluation. This distinction matters directly for audit purposes: a compliance determination made for financial year 2024 must be evaluated against the rules in force for financial year 2024, even if those rules have since been superseded.

Figure 2. Legal Chronology of the Illustrative Scope 3 Obligation

CSRD Article 19a / ESRS E1, illustrative walkthrough (§4.3)



Note: the amending directive (Track A) gives the revised thresholds legal effect; the Commission's adoption of the revised ESRS delegated act (Track B) is a separate instrument specifying disclosure content, with its own scrutiny and applicability timeline. Superseded rule versions are retained and dated, not deleted, so determinations made under an earlier version remain independently auditable.

At a conceptual level, the resulting rule for Scope 3 disclosure under the post-Omnibus thresholds would therefore be keyed to the relevant legal chain, including Directive (EU) 2022/2464 and Article 19a of Directive 2013/34/EU, Commission Delegated Regulation (EU) 2023/2772, ESRS E1, and the threshold amendments made by Directive (EU) 2026/470, alongside its temporal metadata. The scope conditions determine whether the obligation applies to a given entity in a given reporting period: EU jurisdiction, the applicable employee and turnover thresholds, and the reporting period governed by the relevant rule version. The compliance conditions determine what the obligation requires once triggered: disclosure of gross Scope 3 emissions, coverage of the Scope 3 categories identified as significant by the entity's value-chain and materiality assessment, whether those categories are upstream, downstream, or both, and completion of the required double materiality assessment. Each condition traces to a specific element of Article 19a, ESRS E1, or the applicable amendment, so that a resulting determination can be mapped back to the specific regulatory language and rule version that produced it.

This decomposition illustrates several key properties the ESG-as-Code framework is designed to exhibit. First, every compliance determination is intended to trace directly to a named regulatory provision and a named version of that provision, not merely to the regulatory instrument in general. A FAIL output does not produce a generic non-compliance score but identifies the specific unmet condition, the article and standard that requires it, and the version of that standard in force for the

relevant reporting period. Second, the rule is fully deterministic given a fixed rule version: given the same entity profile, disclosure data, and applicable rule version, the system produces the same output every time. Third, the scope conditions are explicitly encoded and versioned, ensuring that the obligation is not evaluated against the wrong threshold for entities whose reporting period predates or postdates a given amendment. Fourth, the rule is independently versionable: when the Omnibus threshold changes were adopted and transposed, the scope conditions could be updated in the rule library, with the prior version retained and dated, without affecting any other rule in the system or losing the ability to audit determinations made under the earlier version.

4.4 Cross-Jurisdictional Application

The same rule encoding methodology applied in the CSRD example above can be extended to other regulatory instruments within the same framework. For an organization subject to both CSRD and SEC climate disclosure rules, which remain formally adopted but stayed and proposed for rescission as of this writing, the jurisdictional logic model maintains separate rule sets for each instrument and evaluates them independently against the entity's profile. Where obligations overlap, for example where both regimes require disclosure of Scope 1 and Scope 2 emissions, the framework identifies the overlapping requirements and flags them as satisfiable by a single disclosure, reducing duplication while preserving the distinct traceability chains required by each regime. Where obligations diverge, each obligation is tracked and reported independently.

This cross-jurisdictional capability is not achievable through manual compliance processes or through AI scoring tools that produce aggregate ESG ratings without jurisdiction-level granularity. It represents one of the most practically significant contributions of the ESG-as-Code framework to the operational challenges faced by multinational organizations navigating the current regulatory landscape.

4.5 Formal Rule Representation: A Worked Example

To move from conceptual description to a formally specified artifact, this section walks through, in implementation-neutral prose rather than any runnable syntax, one complete rule specification for the Scope 3 disclosure obligation discussed in Section 4.3. Consistent with the framework's design as described in Sections 3.2 and 3.5, this walkthrough is formally specified and manually

traced through a set of illustrative scenarios; it is not an executable implementation, and no claim of automated execution or generated output is made for it here. The proprietary production rule engine that implements this and other rule specifications is described further in the Declarations section.

The specification is keyed to its authoritative source and version history rather than to a single provision. Following the two-instrument distinction established in Sections 4.2 and 4.3, it identifies both the directive establishing the reporting duty, Article 19a of Directive 2013/34/EU as inserted by the CSRD, and the delegated act specifying the obligation's content, Commission Delegated Regulation (EU) 2023/2772, ESRS E1, together with the amending directive that revised the applicable thresholds following the Omnibus process. Two versions of the specification are distinguished: an earlier version governing reporting periods from 22 December 2023 under the pre-Omnibus thresholds, and a current version governing reporting periods from 1 January 2027 under the post-Omnibus thresholds. The earlier version is retained and dated rather than discarded, so that a compliance determination made for a reporting period it governed remains independently auditable after supersession, consistent with the temporal metadata model set out in Section 4.3.

The applicability conditions determine whether the obligation reaches a given entity in a given reporting period: EU jurisdiction, an average headcount exceeding 1,000 employees, net annual turnover exceeding EUR 450 million, and a reporting period beginning on or after 1 January 2027 under the current version. An entity meeting the pre-Omnibus thresholds for a reporting period before that date would instead be routed to the earlier version governing that period, rather than being evaluated against thresholds that were not yet in effect for it.

Once the obligation is determined to apply, the specification requires three categories of disclosure data: the entity's gross Scope 3 emissions figure, a record of which value-chain categories that figure covers, and a dated record confirming that a double materiality assessment has been completed. Consistent with the outcome model in Section 3.3, a missing required disclosure field produces INSUFFICIENT_EVIDENCE rather than a substantive failure, and a Scope 3 figure that is present but non-numeric or negative produces DATA_QUALITY_FAILURE. Where the rule's satisfaction depends on a materiality determination and no completed, dated materiality

assessment record is present for the applicable reporting period, the determination is `MATERIALITY_REVIEW_REQUIRED` rather than `NOT_SATISFIED`, because the deterministic layer does not make the underlying materiality judgment. Once a completed, dated materiality record is supplied, the rule is re-evaluated. Where the entity's disclosure then omits Scope 3 categories that its own value-chain assessment identifies as significant, whether those categories sit upstream or downstream, the determination is `NOT_SATISFIED` on the basis of ESRS E1 paragraphs 44 and 51, which require disclosure of each significant category rather than imposing a categorical requirement that both directions be represented; an entity whose significant categories arise entirely on one side of its value chain can satisfy this condition without reporting the other. Where a completed materiality record and complete coverage of the entity's own significant categories are both present, the determination is `SATISFIED`.

The materiality determination itself is treated as data the specification consumes rather than a judgment the rule logic makes: the underlying double materiality assessment is produced outside the deterministic layer, consistent with the separation-of-concerns principle in Section 3.2, and the specification verifies only that a completed, dated record of that assessment exists for the applicable reporting period. Each required input also carries a provenance tag distinguishing whether it was human-entered, AI-extracted, or system-calculated, and, where AI-extracted, a confidence score and extraction model version, all retained through to the audit output without influencing the rule logic itself, consistent with the AI-derived data handling described in Section 3.4.

This walkthrough illustrates the properties the outcome model in Section 3.3 is designed to support. A substantively unmet obligation (`NOT_SATISFIED`) is distinguished from a condition reflecting incomplete or invalid data (`INSUFFICIENT_EVIDENCE`, `DATA_QUALITY_FAILURE`) and from an obligation the entity is not yet or no longer subject to (`NOT_APPLICABLE`, or routing to a superseded version). This is one illustrative specification rather than a full rule library; the same structure, authoritative-source versioning, applicability conditions, required inputs, provenance handling, and change history, is intended to extend across the framework's other rule specifications, though a comprehensive specification of the full rule library falls outside the scope of this paper.

4.6 Preliminary Evaluation

The specification presented in Section 4.5 was manually walked through against five illustrative entity scenarios, each constructed to exercise a distinct path through the rule logic and outcome model: an entity below the applicability threshold; an entity above threshold with complete, satisfying disclosure; an entity above threshold for which no completed, dated materiality assessment record is available, producing MATERIALITY_REVIEW_REQUIRED; an entity above threshold with a missing required disclosure field; and an entity above threshold reporting for a period governed by the earlier, pre-Omnibus version of the specification. In each scenario, working through the specification by hand produced the outcome anticipated for that scenario, and each outcome carried a distinct basis, the specific ESRS provision or applicability condition, rather than an undifferentiated pass or fail.

This exercise is best understood, in the terms of the FEDS framework for design-science evaluation, as a Phase 1 artificial, formative evaluation, an internal check of logical consistency appropriate to this artifact's current stage of development, rather than the naturalistic, summative evaluation, involving independent expert review of real or realistic disclosure data, that a mature artifact would require (Venable, Pries-Heje and Baskerville, 2016). It is a demonstration that the specification and outcome model are coherently defined with respect to the intended design, rather than as evidence of the framework's real-world effectiveness. Several relevant evaluation criteria remain untested at this stage. Rule coverage, the proportion of a given regulatory instrument's disclosure obligations that can be represented as specifications of this kind, has been assessed for one obligation within one standard, not across CSRD, SFDR, or ISSB in aggregate. Expert agreement, the extent to which the specification's determinations match those a qualified compliance reviewer would independently reach given the same disclosure data, has not been tested, since the entity profiles used are synthetic and no independent reviewer assessment was collected. Maintainability and update effort, how much rework a specification requires when its source regulation is amended, is suggested by the versioning structure in Section 4.5 but has not been measured against an actual amendment cycle.

The principal threats to validity are accordingly threefold. First, construct validity: five illustrative scenarios for a single specification may not represent the range of edge cases, ambiguous

disclosures, and conflicting rule interactions that arise in production compliance environments. Second, external validity: the entity profiles used are illustrative rather than drawn from actual corporate disclosures, so this exercise says nothing about performance against real-world data quality and completeness issues of the kind discussed in Section 5.4. Third, evaluator bias: because the specification, scenarios, and expected outcomes were authored by the same party, this exercise cannot rule out the possibility that the rule logic was implicitly shaped to pass its own scenarios rather than independently verified against the regulatory text by a reviewer with no role in the encoding process. Addressing this threat directly would require the expert-comparison and larger-scale evaluation set out in Section 5.5.

5. Discussion, Implications, and Limitations

5.1 Implications for Compliance Practitioners

The ESG-as-Code framework has direct practical implications for compliance teams operating within increasingly complex regulatory environments. Industry surveys continue to highlight the operational burden associated with manual ESG compliance processes. According to recent assessments, companies spend significant time each year on ESG data collection and reporting, while many still struggle to produce audit-ready documentation and maintain the infrastructure needed for comprehensive ESG reporting, particularly for Scope 3 emissions tracking across complex supply networks (Adherent, 2025). These findings suggest that existing ESG compliance processes remain resource-intensive and operationally challenging for many organizations.

The ESG-as-Code framework addresses this challenge not by replacing human judgment in ESG strategy, but by automating the routine application of clearly defined regulatory rules to organizational data. Compliance officers using a system built on this framework would be freed from the repetitive and error-prone task of manually checking disclosures against regulatory requirements, allowing their expertise to be directed toward the interpretive and strategic dimensions of ESG management that genuinely require human judgment. The framework also produces a continuous, auditable compliance record rather than a point-in-time report, which aligns with the audit-ready documentation requirements of the CSRD and similar frameworks (European Parliament and Council of the European Union, 2022).

5.2 Implications for Regulators

The framework carries significant implications for regulatory bodies and supervisory technology developers. One of the persistent challenges facing regulators attempting to enforce ESG disclosure requirements is the opacity of current reporting systems. When disclosures are submitted in unstructured PDF formats with no machine-readable structure, verification requires manual review that is neither scalable nor consistent (Papantoniou, 2022). ESG-as-Code offers a pathway toward machine-readable compliance submissions in which every disclosed data point is linked to the regulatory provision it satisfies, enabling regulators to verify compliance programmatically rather than relying solely on manual document review. This is discussed here as a design implication of the framework's traceability properties; it extends beyond the validation-layer worked example presented in Section 4.5 and has not itself been implemented or evaluated.

This reflects broader developments in regulatory technology. The CSRD already requires sustainability disclosures to be digitally tagged for submission to the European Single Access Point (ESAP), highlighting the growing shift toward structured and machine-readable reporting systems (European Parliament and Council of the European Union, 2022). ESG-as-Code extends this logic to the compliance validation layer by proposing that the process of checking whether disclosures satisfy regulatory requirements should be structured and deterministic, with rule logic formally specified for machine implementation. Automated machine execution remains an intended design objective of the broader framework rather than an empirical result demonstrated by the manually evaluated artifact in this paper. Regulators that adopt or endorse implementations of such frameworks could potentially improve their capacity to detect greenwashing and enforce disclosure standards at scale, subject to the validation requirements discussed in Sections 5.4 and 5.5.

5.3 Implications for RegTech Developers

For developers working in the RegTech space, the ESG-as-Code framework offers a design approach that distinguishes between the appropriate and inappropriate uses of probabilistic AI in compliance contexts. Separating AI-assisted document interpretation from deterministic compliance decision-making provides a practical architecture that can preserve many of the efficiency benefits of AI tools while maintaining traceability and accountability in compliance

outcomes. Enterprise AI guidance has increasingly emphasized the importance of explainability and reproducibility in high-impact decision-making systems (Elementum AI, 2026).

Developers building on this framework would also benefit from the versioned rule library approach, which provides a structured mechanism for responding to regulatory changes without requiring wholesale system redesign. Given the pace of regulatory change in the ESG space, including the significant threshold and scope revisions introduced through the CSRD Omnibus proposals between 2022 and 2026, a compliance system that cannot be updated through targeted rule amendments rather than complete rebuilds may struggle to remain adaptable over time (Council of the EU, 2026; Deloitte, 2026).

5.4 Limitations

The ESG-as-Code framework carries several limitations that require acknowledgment and that point toward areas for future research.

The first limitation is regulatory ambiguity. Not all ESG regulatory obligations are amenable to deterministic encoding. Many provisions in frameworks such as the CSRD involve qualitative assessments, materiality judgments, and principles-based requirements that resist precise formalization as conditional logic. The double materiality assessment required under CSRD Article 19a, for example, requires organizations to make substantive judgments about which sustainability matters are material to their operations and impacts, judgments that cannot be automated through rule logic alone. ESG-as-Code is most effective for the subset of obligations that are sufficiently specific and quantitative to be encoded deterministically, and practitioners must recognize that a significant portion of ESG compliance will continue to require human judgment regardless of the maturity of rule-based automation tools (Legg and Bell, 2024).

The second limitation is data quality dependency. The accuracy of any compliance determination produced by the ESG-as-Code framework depends heavily on the quality and completeness of the input data against which rules are evaluated. Recent industry surveys have identified ESG data quality as a major challenge for organizations, with 57 percent of respondents citing it as the top challenge associated with ESG data management (Deloitte, 2024). Scope 3 emissions data in particular is often affected by inconsistent supplier reporting formats, reliance on estimates and

proxy data, and limited visibility across tier two and tier three supply chain partners (National Law Review, 2025; Veridion, 2025). As a result, a deterministic rule engine evaluating incomplete or inaccurate inputs may still generate technically consistent compliance outputs that do not fully reflect operational reality. The framework therefore depends on parallel investment in data governance and supply chain data collection infrastructure beyond its immediate scope.

The third limitation is legal interpretation dependency. The process of translating regulatory text into rule logic requires interpretive decisions that carry legal significance. Where regulatory text is ambiguous or where its application to a specific organizational context is contested, the rule encoding process itself becomes a site of legal risk. A rule that incorrectly interprets a regulatory provision could produce systematically incorrect compliance determinations at scale. This limitation highlights the importance of involving qualified legal counsel in the rule development process and maintaining clear documentation of the interpretive choices made when encoding each rule (Legg and Bell, 2024; Mohun and Roberts, 2020).

The fourth limitation is framework novelty and empirical validation at scale. ESG-as-Code is a proposed framework that has been evaluated only through the preliminary internal-consistency check described in Section 4.6 and has not yet been subjected to empirical validation in production compliance environments. That preliminary evaluation demonstrates internal consistency for a single rule package against synthetic data; it does not constitute evidence of the framework's real-world effectiveness, scalability, or legal defensibility in enforcement proceedings. Future research should pursue empirical evaluation of the framework across diverse organizational types, jurisdictions, and regulatory instruments.

5.5 Future Research Directions

The limitations identified above point toward several productive directions for future inquiry. Research is needed on methodologies for handling qualitative and principles-based ESG obligations within a deterministic framework, potentially through hybrid approaches that combine rule-based evaluation of quantitative requirements with structured human-in-the-loop review processes for qualitative assessments. Further research is also needed on data governance standards for ESG compliance inputs, particularly for Scope 3 value chain data, where inconsistent reporting practices and the lack of globally harmonized metrics continue to affect reporting quality across

jurisdictions (National Law Review, 2025). Finally, empirical case studies examining the deployment of rule-based ESG compliance systems in regulated organizational environments would help strengthen the evidentiary basis for the framework's practical claims. In FEDS terms, this defines Phase 2 of the evaluation programme this paper begins: a naturalistic, summative evaluation in which an independent compliance or ESRS specialist with no role in the rule's encoding works through the same or an expanded set of disclosure scenarios and reports agreement or disagreement with the specification's determinations, extending the internal-consistency check in Section 4.6 rather than substituting for it (Venable, Pries-Heje and Baskerville, 2016; Hevner et al., 2004).

6. Conclusion

The rapid expansion of mandatory ESG reporting obligations across major global jurisdictions has created compliance demands that existing tools and processes are structurally ill-equipped to meet. Manual approaches are too slow, too inconsistent, and too opaque to satisfy the traceability and auditability requirements that modern ESG regulation demands.

This paper has introduced ESG-as-Code as a deterministic rule-based framework built on regulatory primacy, determinism, separation of concerns, and jurisdictional modularity (Section 3.2), formalized this into a rule representation carrying explicit temporal and legal-status metadata (Section 4.5), and evaluated a single worked package against that representation to demonstrate internal consistency (Section 4.6). The determinism the framework provides is a guarantee of reproducibility and traceability, not, on its own, of legal correctness; Sections 3.2 and 3.4 set out what additional conditions, accurate rule encoding, ongoing legal review, and governed handling of AI-derived uncertainty, that correctness further requires.

Section 5 has acknowledged the framework's limitations without minimizing them: qualitative and principles-based obligations resist deterministic encoding, data quality constrains any downstream determination, rule encoding carries legal interpretation risk, and the evaluation presented here remains preliminary and internally focused rather than validated at scale. These limitations define the framework's appropriate scope rather than undermine its contribution. ESG-as-Code does not propose to automate ESG compliance in its entirety; it proposes to automate the deterministic,

rule-applicable subset of compliance obligations, while keeping the boundary of that subset, and the uncertainty in what falls outside it, visible rather than hidden.

As ESG regulation matures and enforcement intensifies, the demand for compliance systems that can demonstrate their reasoning rather than merely report their conclusions will only grow. The empirical work set out in Section 5.5, larger rule libraries, comparison against expert-produced determinations, and evaluation across diverse regulatory instruments, represents the next stage of this research, and the invitation extended by this paper is for researchers, practitioners, regulators, and technologists to engage critically with the foundation presented here and contribute to that work.

Declarations

Conflict of interest: The author is the sole founder and Chief Architect of ESGine Inc., which commercializes the ESG-as-Code® framework discussed in this paper. ESG-as-Code® and ESGine® are trademarks held by the author. This commercial interest is disclosed in full; the framework, principles, and limitations discussed in this paper reflect the author's independent academic judgment and include limitations, such as the framework's current lack of large-scale empirical validation, that a purely commercial presentation would be expected to omit.

Funding: This research received no external funding.

Data and code availability: ESG-as-Code is presented in this paper as a conceptual research framework; the walkthrough in Section 4.5 formally specifies, in implementation-neutral prose, one illustrative rule for the Scope 3 disclosure obligation discussed in Section 4.3, and is manually traced through the scenarios in Section 4.6 rather than executed. A separate, proprietary production implementation of the broader framework exists and is commercialized by the author's company, ESGine Inc., consistent with a pending patent application covering that system's technical architecture; that implementation, including its internal schema, rule identifiers, and source code, is not released alongside this paper and is not the artifact evaluated here. No proprietary source code, rule-identifier conventions, or internal schema are disclosed in this paper.

References

Adherent (2025) 'The complete 2025 ESG and sustainability readiness assessment guide'. Available at: <https://www.adherent.com/blog/the-complete-2025-esg-sustainability-readiness-assessment-guide-test-your-compliance-preparedness/>

Anthesis Group (2026) 'CSRD Scope 3 reporting requirements: what's mandatory, what's changed?'. Available at: <https://www.thesisgroup.com/insights/scope-3-reporting-csrd/>

Athan, T., Governatori, G., Palmirani, M., Paschke, A. and Wyner, A. (2015) 'LegalRuleML: design principles and foundations', in Faber, W. and Paschke, A. (eds.) *Reasoning Web. Web Logic Rules: 11th International Summer School 2015, Tutorial Lectures*. Cham: Springer, pp. 151–188. doi: https://doi.org/10.1007/978-3-319-21768-0_6

Bertl, M., Price, S. and Draheim, D. (2026) 'Transforming legal texts into computational logic: enhancing next generation public sector automation through explainable AI decision support', *International Journal of Cognitive Computing in Engineering*, 7, pp. 40–57. doi: <https://doi.org/10.1016/j.ijcce.2025.07.003>

Council of the EU (2026) 'Council signs off simplification of sustainability reporting and due diligence requirements to boost EU competitiveness'. Available at: <https://www.consilium.europa.eu/en/press/press-releases/2026/02/24/council-signs-off-simplification-of-sustainability-reporting-and-due-diligence-requirements-to-boost-eu-competitiveness/>

Deloitte (2023) 'Frequently asked questions about the EU corporate sustainability reporting directive'. Available at: <https://dart.deloitte.com/USDART/home/publications/deloitte/heads-up/2023/csr-corporate-sustainability-reporting-directive-faqs>

Deloitte (2024) '2024 sustainability action report'. Available at: <https://www.deloitte.com/us/en/services/audit-assurance/articles/esg-survey.html>

Deloitte (2026) 'European sustainability reporting: omnibus legislative developments and updates to European sustainability reporting standards'. Available at:

<https://dart.deloitte.com/USDART/home/publications/deloitte/heads-up/2026/eu-sustainability-reporting-omnibus-esrs-updates>

EFRAG (2022) *Draft European Sustainability Reporting Standards: ESRS E1 Climate Change*. European Financial Reporting Advisory Group. Available at: <https://www.efrag.org/en/sustainability-reporting/esrs/sector-agnostic/first-set-of-draft-esrs>

El Harras, M. and Salahddine, M.A. (2025) 'Tracking financial crime through code and law: a review of RegTech applications in anti-money laundering and terrorism financing', *Corporate Law & Governance Review*, 7(3), pp. 73–85. doi: <https://doi.org/10.22495/clgrv7i3p7>

Elementum AI (2026) 'Deterministic vs. probabilistic AI: enterprise workflow guide'. Available at: <https://www.elementum.ai/blog/deterministic-vs-probabilistic-ai>

European Commission (2026) 'Commission adopts revised sustainability reporting standards to reduce administrative burdens'. Available at: https://finance.ec.europa.eu/news/commission-adopts-revised-sustainability-reporting-standards-reduce-administrative-burdens-eu-2026-07-03_en

European Parliament and Council of the European Union (2019) *Regulation (EU) 2019/2088 on sustainability-related disclosures in the financial services sector (SFDR)*. *Official Journal of the European Union*, 9 December. Available at: <https://eur-lex.europa.eu/eli/reg/2019/2088/oj/eng>

European Parliament and Council of the European Union (2022) *Directive (EU) 2022/2464 on corporate sustainability reporting (CSRD)*. *Official Journal of the European Union*, 16 December. Available at: <https://eur-lex.europa.eu/eli/dir/2022/2464/oj/eng>

European Parliament and Council of the European Union (2026) Directive (EU) 2026/470 of 24 February 2026 amending Directives 2006/43/EC, 2013/34/EU, (EU) 2022/2464 and (EU) 2024/1760 as regards certain corporate sustainability reporting requirements and certain corporate sustainability due diligence requirements. OJ L, 2026/470, 26.2.2026. Available at: <https://eur-lex.europa.eu/eli/dir/2026/470/oj/eng>

Golzarjannat, A. and Gustafsson, R. (2026) 'Regulatory technologies for enhancing sustainability compliance: a multivocal literature review', *Journal of Financial Regulation and Compliance*, 34(3), pp. 385–415. doi: <https://doi.org/10.1108/JFRC-04-2025-0074>

HashiCorp (2025) 'Policy as code, explained'. Available at: <https://www.hashicorp.com/blog/policy-as-code-explained>

He, L.-Y. and Wang, L. (2025) 'Can artificial intelligence curb greenwashing? Firm-level evidence based on large language model', *Energy Economics*, 152, 108954. doi: <https://doi.org/10.1016/j.eneco.2025.108954>

Kumara, I., Garriga, M., Romeu, A.U., Di Nucci, D., Palomba, F., Tamburri, D.A. and van den Heuvel, W.-J. (2021) 'The do's and don'ts of infrastructure code: a systematic gray literature review', *Information and Software Technology*, 137, 106593. doi: <https://doi.org/10.1016/j.infsof.2021.106593>

Hevner, A.R., March, S.T., Park, J. and Ram, S. (2004) 'Design science in information systems research', *MIS Quarterly*, 28(1), pp. 75–105. doi: <https://doi.org/10.2307/25148625>

Legg, M. and Bell, F. (2024) 'Rules as code and the rule of law: ensuring effective judicial review of administration by software', *Law, Innovation and Technology*, 16(1), pp. 170–193. doi: <https://doi.org/10.1080/17579961.2024.2313801>

Merigoux, D., Chataing, N. and Protzenko, J. (2021) 'Catala: a programming language for the law', *Proceedings of the ACM on Programming Languages*, 5(ICFP), Article 77, pp. 1–29. doi: <https://doi.org/10.1145/3473582>

Mohun, J. and Roberts, A. (2020) 'Cracking the code: rulemaking for humans and machines', *OECD Working Papers on Public Governance*, No. 42. Paris: OECD Publishing. doi: <https://doi.org/10.1787/3afe6ba5-en>

Morris, K. (2021) *Infrastructure as Code: Dynamic Systems for the Cloud Age*. 2nd edn. Sebastopol: O'Reilly Media.

National Law Review (2025) 'ESG and supply chains in 2024: key trends, challenges, and future outlook'. Available at: <https://natlawreview.com/article/esg-and-supply-chains-2024-key-trends-challenges-and-future-outlook>

Object Management Group (2021) *Decision Model and Notation (DMN), Version 1.4*. Available at: <https://www.omg.org/spec/DMN/1.4/>

Owolabi, I.O. (2025a) 'Operationalizing ESG-as-Code: automating ESG compliance and regulatory reporting pipelines using containerized AI workflows on Kubernetes–OpenStack infrastructure', *Algora*, 2(2), pp. 1–24. doi: <https://doi.org/10.63084/algora.v2i2.60>

Owolabi, I.O. (2025b) 'What is ESG-as-Code™?'. Available at: <https://doi.org/10.2139/ssrn.5384323>

Papantoniou, A.A. (2022) 'Regtech: steering the regulatory spaceship in the right direction?', *Journal of Banking and Financial Technology*, 6(1), pp. 1–16. doi: <https://doi.org/10.1007/s42786-022-00038-9>

Sari, R.C., Sholihin, M., Cahaya, F.R., Ishola, A., Yuniarti, N., Pranesti, A., Sari, A.R. and Haryanto (2025) 'Artificial intelligence-based ESG greenwashing detection: road to net zero carbon and its impact on corporate performance', *Business Strategy & Development*, 8(4), e70228. doi: <https://doi.org/10.1002/bsd2.70228>

Senken (2026) *CSRD Reporting Requirements: A Practical Climate & ESRS E1 Guide*. Available at: <https://www.senken.io/academy/regulation-and-standards/csr-reporting-requirements>

U.S. Securities and Exchange Commission (2024) *Final rule: the enhancement and standardization of climate-related disclosures for investors*. Available at: <https://www.sec.gov/rules-regulations/2024/03/s7-10-22>

U.S. Securities and Exchange Commission (2026) 'SEC proposes rescission of climate-related disclosure rules'. Available at: <https://www.sec.gov/newsroom/press-releases/2026-49-sec-proposes-rescission-climate-related-disclosure-rules>

Venable, J., Pries-Heje, J. and Baskerville, R. (2016) 'FEDS: a framework for evaluation in design science research', *European Journal of Information Systems*, 25(1), pp. 77–89. doi: <https://doi.org/10.1057/ejis.2014.36>

Veridion (2025) '5 big challenges of ESG data management'. Available at: <https://veridion.com/blog-posts/esg-data-management-challenges/>

Wong, M.W. (2020) 'Rules as code: seven levels of digitisation'. Singapore Management University Centre for Computational Law. Available at: https://ink.library.smu.edu.sg/sol_research/3093

Wong, M. and Dewdney, H. (2023) 'Corporate Sustainability Reporting Directive (CSRD): overview and the impact for U.S. companies with EU operations'. Georgeson. Available at: <https://www.georgeson.com/us/insights/csr-d-overview-sep-2023>