|  |  |
| --- | --- |
| *DFBI 2025 aims to encourage the international exchange of innovative ideas between researchers from academia and industry. In addition to knowledge dissemination, the conference offers a valuable platform for professional networking, particularly benefiting university professors, graduate students, and postdoctoral researchers.* | Research Article |
| ET-ACC: A Conceptual Framework for Explainable and Trustworthy Automated Compliance Checking in the AEC Industry |
| Youssef Senousy 1, Edlira Vikaj 2 |
| Birmingham City University 1,2  Correspondence: [youssef.senousy@mail.bcu.ac.uk](mailto:youssef.senousy@mail.bcu.ac.uk) |
|  | Abstract |
| *Copyright: Copyright: © 2026 by the authors.*  *DFBI is an open-access proceedings distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0). View this license’s legal deed at* [*https://creativecommons.org/licenses/by/4.0/*](https://creativecommons.org/licenses/by/4.0/)  *Εικόνα που περιέχει κείμενο, clipart  Περιγραφή που δημιουργήθηκε αυτόματα* | Automated Compliance Checking (ACC) is now a central activity in the Architecture, Engineering, and Construction (AEC) industry. It supports design assurance, regulatory compliance, and digital delivery from early design through to review. Yet many current ACC systems remain opaque. Complex rule sets and AI-assisted reasoning often produce results that are difficult to inspect or explain, so transparency, interpretability, and user trust remain limited. This paper introduces ET-ACC, a multi-layer conceptual framework for Explainable and Trustworthy Automated Compliance Checking. The framework evaluates BIM and IFC models against pre-formalised, machine-readable rules. It is organised into five connected layers. The Data Layer holds semantically enriched project information. The Rule Layer manages computable rules and logic structures. The Reasoning Layer combines symbolic reasoning with AI-based support. The Provenance Layer records how data, rules, and reasoning steps relate, which enables complete traceability. The Explanation Layer then turns these traces into role-aware, human-readable outputs for designers, regulators, and other stakeholders. A Human-in-the-Loop and Trust Evaluation component strengthens accountability through expert review and structured transparency assessment. The framework is built on design-science methodology and literature across ACC, explainable AI, and compliance modelling. It offers a structured foundation for next-generation systems that place interpretability and auditability at their core. Full implementation is planned as future work. The framework sets a clear blueprint for a practical compliance-checking platform, referred to as ACE-X, that will operationalise the ET-ACC architecture and support more transparent, safe, and accountable digital building regulation processes and automated design assessment workflows across diverse project types and scales. |
| *Received: dd/mm/yyyy*  *Revised: dd/mm/yyyy*  *Accepted: dd/mm/yyyy*  *Published: dd/mm/yyyy*  *Volume: xx*  *Issue: xx*  *Pages: pp-pp* |
|  | **Keywords:** Automated Compliance Checking, Explainable AI, IFC-based BIM, Symbolic–AI Reasoning, Provenance Modelling, Human-in-the-Loop. |
| Cite this article:  Senousy, Y., & Vikaj, E. (2026). ET-ACC: A Conceptual Framework for Explainable and Trustworthy Automated Compliance Checking in the AEC Industry. Open Access Proceedings of the Conference on Digital Frontiers in Buildings and Infrastructure (DFBI 2026), Volume (2026), Page range. https://doi.org/xx.xxx/yyyy | Highlights   * A six-layer ET-ACC framework with an executable prototype (ACE-X) * A practical bridge between symbolic checking and learning * A concrete engineering lesson on trustworthiness |

# Introduction

Building projects operate under dense regulatory constraints, and design teams invest substantial effort in interpreting requirements and preparing evidence for approval. Many jurisdictions now link BIM adoption to permitting workflows, and recent reviews describe BIM-based e-permit ecosystems as an active research and practice area with growing interest in automation and AI integration (Sun & Kim, 2026). This shift increases pressure on automated compliance checking (ACC) systems to deliver not only faster checks but also defensible outcomes that stakeholders can inspect and reuse across project stages.

ACC research has progressed from early rule-based checking to workflows that combine BIM data, rule languages, and knowledge representations. Recent studies show a clear move towards semantic pipelines that translate regulatory statements into machine-readable structures and then run checks on BIM-derived representations (Pauwels et al., 2024; Jia et al., 2025). At the same time, newer work integrates language models and deep learning with ontologies to reduce manual effort in rule interpretation and mapping (Chen et al., 2024). These directions reflect an important trend, yet a key barrier remains across the ACC literature. Many systems still struggle to provide explanations that stay faithful to rules, data, and intermediate reasoning steps. Recent work on reasoning-focused machine learning also shows that explicit, stepwise reasoning mechanisms implemented with compact models can outperform large opaque models on structured tasks and keep intermediate reasoning states open to inspection (Jolicoeur-Martineau, 2025).

Opacity appears in two common ways. Rule translation from natural language into executable constraints often involves hidden choices. The chain from regulation text to rule encoding to model query is not always explicit. Studies continue to report difficulty in representing complex constraints, particularly geometric and context-dependent clauses, and this limits both automation coverage and interpretability (Pauwels et al., 2024). Even when systems run successfully, outputs often stop at pass or fail, with little traceability of the data and logic that produced the decision. Newer ACC frameworks call explicitly for transparency and trust features such as traceable decision paths and structured human checkpoints, which shows that explainability is a core requirement for reliable adoption rather than an optional add-on (Purushotham et al., 2026).

Traceability becomes harder when compliance pipelines rely on multiple representations and tools. IFC models are parsed, transformed into graphs or ontologies, checked through rule engines, and then summarised in reports. Each conversion can lose meaning unless the system records what was used and how it was processed. Provenance research in construction already frames trustworthy data exchange as a core need, and BIM provenance models show that recording activities and exchanges supports the reconstruction and validation of decisions across stakeholders (Celik et al., 2023). For ACC, this points to the need for structured links between inputs, such as IFC elements and properties, applied rules, and reasoning outputs.

This paper addresses this gap through ET-ACC, a layered framework for explainable and trustworthy ACC, and through ACE-X, a partial operationalisation of the framework up to the Reasoning Layer. The conceptual design places data extraction, rule execution, and reasoning into connected layers that can later support provenance capture and role-aware explanations.

In ACE-X, the system parses IFC models, constructs a normalised building graph, executes compliance rules, and generates reasoning outputs that explain failures and propose fixes. The current implementation also integrates a trainable component, referred to as a Tiny Recursive Model (TRM), to support learning from compliance patterns and to expose reasoning-oriented outputs, with explicit handling of training data completeness to avoid misleading performance artefacts.

# Conceptual Foundations and Prior Work in Automated Compliance Checking

Recent digital building permitting research conceptualises automated compliance checking as a socio-technical capability embedded within broader regulatory workflows. Rather than operating as a standalone verification tool, ACC is increasingly positioned between regulatory interpretation, information availability in BIM submissions, and the verification procedures applied by permitting authorities. Noardo et al. (2022) provide a comprehensive synthesis of Digital Building Permit initiatives and illustrate how ACC functions as one component within interconnected permitting ecosystems.

Within this context, explainability is discussed in the literature as the ability to justify compliance outcomes through inspectable links between model evidence and computational checks. Regulator-facing studies indicate that adoption of ACC systems depends not only on technical correctness, but also on clarity of terminology, consistency of interpretation, and a balanced distribution of automation and human oversight. These concerns are reported directly through interviews with regulators and ACC specialists (Fuchs et al., 2025).

Provenance and traceability are similarly framed as decision reconstruction capabilities. They support questions such as which model facts were used, which rules were applied, and how intermediate evaluations contributed to a compliance outcome. Digital permitting research emphasises that machine-readable submissions and structured information requirements are prerequisites for repeatable and auditable checks across stakeholders. Heiss et al. (2025) discuss this requirement through extensions to information delivery structures that support permitting processes.

Taken together, these studies show that while ACC is increasingly embedded in digital permitting workflows, existing approaches still treat explainability, provenance, and traceability as secondary or external concerns. They are discussed as adoption conditions or reporting requirements, but they are rarely modelled as integral components of the compliance checking process itself. As a result, current systems struggle to provide compliance outcomes that can be reconstructed, audited, and justified across organisational and regulatory boundaries.

## Existing ACC Frameworks and Technical Paradigms

Recent work on automated compliance checking can be grouped into three main technical strands. The first strand uses open standards to specify information requirements and to check submission quality. IDS-based workflows formalise the expected content of IFC exchanges and allow automatic verification of whether required entities and properties are present. Mendonça and Ferreira (2024) apply IDS to accessibility rules and discuss the benefits and limits of this standards-led route. Fischer et al. (2024) extend IDS towards compliance tasks such as escape route analysis and fire resistance checks, where relationships between elements and filtering logic become essential. These studies show that IDS can support parts of compliance checking, yet they still offer limited support for reasoning traces that connect results back to regulatory intent or intermediate evaluations.

The second strand centres on semantic consistency and constraint formalisation around IFC representations. Research on ifcOWL examines how IFC schema constraints and validation logic can be translated into forms that support automated checking, which strengthens reproducibility and improves confidence in model-derived facts. Wang et al. (2024) study the conversion and validation of IFC “WHERE” rules, which links directly to the reliability of the facts used in downstream checking pipelines. This line of work strengthens semantic integrity, but it gives less attention to how reasoning steps are made visible or reconstructed once validation has completed.

The third strand proposes system-level architectures that combine compliance logic with knowledge graphs and governance mechanisms. Tao et al. (2024) present a blockchain-enabled common data environment that uses knowledge-graph structures to check BIM metadata compliance. Their work illustrates how compliance obligations extend beyond geometry into collaboration and information management, yet explanation and lineage remain largely implicit within the system’s operational logic.

Alongside these strands, recent reviews discuss the role of machine learning in ACC and the trade-offs between rule-grounded precision and learning-based coverage. Alnuzha (2025) surveys machine learning use in automated compliance-related workflows and highlights both where learning supports scalability and where it increases validation and accountability demands.

## Open Challenges and Research Opportunities in ACC

Recent studies point to three gaps that persist, even where standards and semantic checking pipelines have become more mature.

First, many workflows improve rule execution or increase information completeness, but they still fail to support decision reconstruction that links regulatory intent, formal rule structure, and the specific model evidence used during execution. Regulator-focused work identifies this as a practical barrier to adoption, since authorities need to validate how a decision was reached, not just the outcome (Fuchs et al., 2025).

Second, standards such as IDS raise the quality and consistency of what gets submitted, yet compliance results still depend on how checking logic is assembled, how exceptions are managed, and how relationships between elements are evaluated. Recent IDS extensions show progress, but they also expose the difficulty of capturing contextual constraints in a controlled and reusable form (Fischer et al., 2024).

Third, LLM-assisted ACC pipelines introduce a distinct traceability problem. They can help with mapping and alignment between regulatory language and BIM data, but they add processing steps that must be checked and tied back to authoritative sources. Recent work on LLM-based model alignment highlights this tension and argues for explicit validation layers around language-based components (Saluz et al., 2025).

Taken together, these gaps are unlikely to be addressed by simply extending rule execution engines or attaching reporting modules after the fact. They reflect a structural limitation in many current ACC architectures, where reasoning, lineage, and explanation are not treated as core outputs of the compliance process.

## ET-ACC: Conceptual Framework Overview

In response to these structural limitations, ET-ACC is proposed as a layered conceptual framework that embeds reasoning, provenance, and explanation directly within the compliance checking architecture rather than treating them as auxiliary functions. ET-ACC addresses these gaps by structuring ACC as connected layers that separate concerns while keeping links explicit between them. The Data and Rule layers support-controlled inputs and computable checks. The Reasoning layer supports structured justification of outcomes. The Provenance and Explanation layers support decision reconstruction and stakeholder-facing narratives.

This paper operationalises ET-ACC up to the Reasoning Layer through ACE-X, which provides a concrete base for later provenance capture and role-aware explanation generation.

# Methodology

This study follows a design science research methodology. The work addresses a practical weakness in automated compliance checking by designing and instantiating an artefact that makes compliance outcomes inspectable. The research yields two connected outputs. ET-ACC is the conceptual framework that structures explainable and trustworthy automated compliance checking as a layered process. ACE-X is a partial operational instantiation that implements core layers and prepares interfaces for provenance capture and explanation delivery.

The methodology ties conceptual choices to executable behaviour. This alignment allows the study to examine how traceability and explanation arise during compliance execution rather than being attached as post-processing. The current scope focuses on architectural coherence and reasoning transparency, supported by a working prototype and structured runtime artefacts.

## ET-ACC as the Methodological Framework

ET-ACC acts as the organising structure of the methodology and guides both system design and analysis. The framework models compliance checking as connected layers that separate responsibilities while keeping explicit links between inputs, rules, intermediate evaluations, and outputs. It includes six components: Data, Rule, Reasoning, Provenance, Explanation, and Human-in-the-Loop and Trust Evaluation. The layered structure and information flows are shown in Figure 1.

The Data Layer prepares controlled BIM-derived inputs. The Rule Layer holds machine-readable rules that are linked to their originating regulatory statements. The Reasoning Layer produces structured justifications that connect outcomes to evaluated conditions and evidence. The Provenance Layer records how model facts, rules, and reasoning steps relate. The Explanation Layer turns these records into stakeholder-facing accounts. Human oversight and trust evaluation provide checkpoints for inspection, correction, and refinement. In ACE-X, the Data, Rule, and Reasoning layers are implemented directly, while provenance and explanation are prepared through logs, stable identifiers, and interfaces that support later extension without reworking the pipeline.

A diagram of a computer

AI-generated content may be incorrect.

*Figure 1. ET-ACC layered framework for explainable and trustworthy automated compliance checking*

## ACE-X Compliance and Reasoning Pipeline

ACE-X operationalises the ET-ACC logic as an end-to-end pipeline from IFC ingestion to reasoning outputs. The process starts with ingestion of IFC-based BIM models. The system parses IFC entities and extracts relevant geometric and semantic properties, then converts them into a normalised building data graph that represents elements, their properties, and relationships. This unified graph becomes the authoritative input for downstream processing and preserves references to original IFC entities to support evidence traceability.

Compliance checking then evaluates the graph against a set of pre-formalised, machine-readable rules. In this paper, pre-formalised means that relevant regulatory clauses were encoded into structured rule definitions before system execution. Rules specify applicability, conditions, thresholds, and required relations, and each rule is linked to its originating regulatory statement through clause mapping so that regulatory intent remains visible during execution. Rule execution produces deterministic outcomes and records the data elements accessed and the conditions evaluated.

The Reasoning stage builds on these outputs by generating reasoning traces that capture intermediate evaluation steps, dependencies, and affected elements. Symbolic reasoning provides explicit evaluation paths, and controlled AI-based support is used to assist interpretation and explanation without overriding rule outcomes. Runtime artefacts, including rule evaluation logs, reasoning traces, and version identifiers, are captured to prepare provenance-ready records that can support later decision reconstruction and explanation generation.

## Reasoning-Oriented Learning with the Tiny Recursive Model

ACE-X integrates a compact learning component, referred to as the Tiny Recursive Model, to support reasoning by learning patterns from prior compliance checks and exposing inspectable reasoning-oriented outputs. Its operational flow is shown in Figure 2. Training data links three sources: compliance results, IFC-derived element features, and rule specifications. These inputs are combined into feature vectors that retain explicit semantic meaning, including context, element, and rule feature sets.

The model uses compact layers with attention and controlled expansion to keep inference behaviour inspectable. During inference, it produces an advisory label and confidence score at each step, and an iterative refinement loop controls progression through a step counter and explicit convergence checks, as depicted in Figure 2. Outputs include the advisory label, confidence value, number of reasoning steps, and convergence status. Rule execution remains authoritative, and the model outputs support prioritisation, pattern surfacing, and explanation support rather than determining compliance decisions.

A diagram of a diagram

AI-generated content may be incorrect.

*Figure 2. Tiny Recursive Model operational flow within ACE-X, including iterative refinement and convergence control.*

## System Architecture and Human Oversight

ACE-X is implemented as a web-based application that separates processing services from user-facing inspection. The backend provides services for IFC parsing, graph construction, rule execution, reasoning trace generation, and runtime logging. The frontend supports interaction with models and results and provides views for inspecting failing elements, rule references, and reasoning artefacts. This separation allows the pipeline to remain stable while supporting stakeholder-facing inspection and review.

The system maps ET-ACC components to implementable responsibilities. Data ingestion and graph construction support the Data Layer. Rule execution services support the Rule Layer. Reasoning services support trace generation and structured justification. Logging and version identifiers support provenance preparation, and interfaces expose artefacts needed for explanation delivery. Human-in-the-loop oversight is treated as a structured checkpoint. Experts can inspect explanations, refine rules, and review reasoning artefacts, and these actions can be captured to support trace completeness and trust evaluation.

Methodological limitations follow from the conference scope. The implementation prioritises architectural coherence, reasoning transparency, and provenance readiness over large-scale statistical validation. Full automation of provenance reasoning and extended empirical evaluation across broader building types and regulatory regimes is reserved for subsequent work.

A computer screen shot of a computer program

AI-generated content may be incorrect.

*Figure 3. High-level Web Application System Architecture of ACE-X*

# Implementation

## Data Layer

The Data Layer performs a structured set of validation and quality assurance checks during the processing of IFC building models. These checks focus on data integrity, extractability, and physical plausibility rather than regulatory compliance, which is addressed at later stages of the pipeline.

During ingestion, each IFC file is loaded using the *ifcopenshell.open()* interface. Files that cannot be parsed trigger an explicit loading exception, preventing invalid models from entering the processing pipeline. Upon successful loading, the system records the detected IFC schema version, such as IFC4 or IFC2x3, to support traceability and downstream compatibility analysis.

Element extraction routines operate on key building entities, with tailored checks applied per element type. For spatial elements, the extraction process verifies the presence of a valid *GlobalId*, and spaces lacking this identifier are excluded from further processing. Storey assignment is derived through traversal of the spatial containment hierarchy. Floor area values are extracted from quantity property sets, prioritising *NetFloorArea* and *GrossFloorArea* where available. If explicit storey information is missing, the system attempts recovery through alternative containment relationships.

For door elements, the process again enforces the presence of a valid *GlobalId*. Door dimensions are normalised into millimetres using a heuristic conversion rule, where values below or equal to ten are interpreted as metres and scaled accordingly. Spatial connectivity is established by resolving *IfcRelSpaceBoundary* relationships, allowing doors to be linked to adjacent spaces. Boundary types and boundary sides are validated to confirm semantic correctness.

A diagram of a software development

AI-generated content may be incorrect.

*Figure 4. Workflow of Data Layer*

For other generic building elements, property sets are extracted using guarded access patterns. Any extraction failures are captured and logged without interrupting execution. Complex IFC attribute representations, including wrapped values, nominal values, and nested structures, are serialised into JSON-compatible formats to support downstream processing.

Property extraction applies defensive serialisation logic to convert IFC-specific data types, such as enumerations and wrapped values, into standard Python and JSON representations. When optional utility modules are unavailable, the system returns empty property sets rather than raising runtime errors. Numeric properties undergo controlled type coercion, where values that cannot be converted to floating-point representations are explicitly set to null.

A dedicated data validation stage performs basic plausibility checks to identify anomalous or corrupt data values. These checks are non-regulatory and focus on physical reasonableness. For doors, width values are expected to fall within *100* to *5000* millimetres, and height values within *500* to *5000* millimetres. The presence of a name attribute is also verified. For spaces, extracted areas are validated against a range of 0.1 to 100,000 square metres, and type consistency is enforced across all properties.

Validation outcomes are reported at the property level, indicating pass or fail status, identifying missing mandatory attributes, detecting invalid data types, and flagging values outside physically plausible ranges. The Data Layer generates summary statistics to support transparency and debugging. These include counts of extracted spaces and doors, tracking of elements with successfully populated dimensional attributes, and logging of partial extraction cases. Such metrics assist in diagnosing incomplete or degraded model inputs.

The implementation follows several guiding principles. Element-level failures do not halt the overall extraction process, allowing partial but usable models to progress. Optional data absence is handled through graceful fallback mechanisms. Data validation is clearly separated from regulatory compliance logic, which resides in the Rule Layer. All extraction and validation issues are recorded at debug logging level to support reproducibility and detailed post-hoc inspection.

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 4. Screenshots of the Prototype Implemented in Data Layer*

## Rule Layer

The Rule Layer assesses regulatory compliance by applying formalised rules to the building graph produced by the Data Layer. Its role is to evaluate element-level and space-level constraints in a structured and auditable manner, while preserving sufficient contextual detail to support downstream reasoning and explanation.

Each compliance evaluation produces a structured rule result object that records both the outcome and its regulatory context. A rule result includes a unique rule identifier, a human-readable rule name, and the target element type, such as door, space, or wall. It also records the identifier of the evaluated element, the evaluation status, and a severity level indicating the criticality of the outcome. The result contains a concise explanatory message, a reference to the originating regulatory clause, and an extensible details field that stores additional metadata required by the reasoning layer.

A diagram of a rules

AI-generated content may be incorrect.

*Figure 5. Workflow of Rule Layer*

Rules are executed by a central rule engine that iterates over the configured rule set and evaluates each rule against the building graph. Before execution, the engine performs basic structural checks to confirm the presence of required element categories. Rules are evaluated sequentially to preserve deterministic behaviour and traceability. Execution errors are captured and logged, allowing evaluation to continue under a lenient execution mode when required. This design supports both strict validation workflows and exploratory analysis scenarios.

{

"rule\_id": "R1\_MIN\_DOOR\_WIDTH",

"rule\_name": "Minimum door width requirement",

"target\_type": "door",

"target\_id": "door\_001",

"status": "FAIL",

"severity": "ERROR",

"message": "Door width 600mm is less than required 900mm",

"code\_reference": "IBC 2018 §1010.1.1",

"details": {

"width\_mm": 600,

"min\_required\_width\_mm": 900,

"door\_name": "Front Door",

"connected\_space\_ids": ["space\_01"]

}

}

The rule set is organised by target element category, with separate modules dedicated to doors, spaces, and other building elements. Each rule encapsulates its own evaluation logic and operates on the relevant subset of the building graph. This modular structure simplifies maintenance, encourages reuse, and supports incremental extension of the rule base as new regulatory requirements are introduced.

An advanced compliance checking component extends basic rule execution by supporting multi-source value extraction. Regulatory quantities may be obtained from property sets, quantity take-off definitions, or direct element attributes. When preferred sources are unavailable, the system applies controlled fallback strategies to alternative sources, improving robustness against incomplete or inconsistent IFC models. Extracted values are then evaluated against rule conditions using explicit comparison logic.

Rules are loaded dynamically from structured definitions and validated before execution to confirm internal consistency and required parameters. This loading process allows rule sets to evolve without changes to the core engine, supporting configuration-driven compliance assessment. Each rule result contains sufficient semantic and contextual information to serve as direct input to the reasoning layer. Results are aggregated into a flat collection that preserves individual element outcomes while enabling higher-level analysis across rules and element categories.

The Rule Layer follows several core design principles. Missing or incomplete data leads to an explicit non-applicable outcome rather than execution failure. Rules are pluggable and extend a shared base abstraction, allowing consistent behaviour across heterogeneous checks. Each result carries detailed contextual metadata to support explanation and auditability. Severity levels distinguish between critical violations, advisory findings, and informational notices. Regulatory references and source tracking support traceable compliance outcomes suitable for review and governance processes.

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 6. Screenshots of the Prototype Implemented in Rule Layer*

## Reasoning Layer

The Reasoning Layer constitutes the analytical core of the system and is responsible for coordinating rule execution, interpreting non-compliance outcomes, and generating structured, explainable results. It is implemented as a modular set of interacting components, each addressing a specific reasoning function within the compliance evaluation pipeline.

The Reasoning Engine acts as the central coordination mechanism for all reasoning activities. It orchestrates interactions among the various reasoning components, dynamically loads both regulatory and project-specific rules, and manages integration with the Rules Version Manager to ensure consistency and traceability across rule executions. In addition, it exposes a unified application programming interface (API) that supports systematic analysis and aggregation of compliance failures.

The Failure Explainer is responsible for interpreting and categorising detected non-compliance instances. Failures are classified into distinct categories, including missing property, dimensional violation, range violation, and value violation. For each failure, the component generates both concise one-line explanations and more detailed, multi-line descriptions that include contextual information. It links failures to explicit regulatory references (e.g., IBC 2018 §1010.1.1) and extracts relevant contextual data, such as affected elements, required versus actual values, and measurement units, to support transparent inspection and review.

The Impact Analyzer quantifies the practical implications of detected non-compliance. It computes the scope of impact by identifying the total number of affected elements, providing breakdowns by element type, and estimating the percentage of the building impacted. Severity distributions are generated by categorising failures into error, warning, and informational levels. It also identifies the most frequently affected elements and the rules most violated, supporting prioritisation and risk-informed decision-making.

The Recommendation Engine translates analytical findings into actionable remediation guidance. It adopts a tiered recommendation strategy, ranging from quick low-effort fixes to comprehensive and systemic high-effort interventions. Recommendations are configuration-driven and instantiated from predefined templates stored in external configuration files. Template variables (e.g., element identifiers or measured values) are dynamically substituted with project-specific data. Each recommendation includes indicative estimates of required effort, resource allocation, and associated cost, supporting informed remediation planning.

A diagram of a process

AI-generated content may be incorrect.

*Figure 7. Workflow of Reasoning Layer*

The Reasoning Layer integrates an AI-based assistant built on a Tiny Recursive Model (TRM) to complement symbolic reasoning with data-driven inference. The assistant performs feature extraction by constructing high-dimensional feature vectors that combine element attributes, rule characteristics, and failure metadata. These vectors are processed by a lightweight neural network to generate pass/fail predictions accompanied by confidence scores. To maintain explainability, the component produces a structured multi-step reasoning trace that exposes intermediate inference stages and supports inspection of the model’s decision logic.

The Reasoning Layer relies on a set of structured data models to ensure consistency and interoperability across components. These include representations for comprehensive failure explanations, quantified impact metrics, individual recommendations and grouped recommendation sets, and aggregated reasoning results that encapsulate the full analysis outcome.

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 7. Screenshots of the Prototype Implemented in Reasoning Layer*

## Future Directions for Provenance, Explanation, and Trust Layers

Future extensions of the Provenance Layer may focus on deeper and more granular traceability of compliance decisions. This includes the representation of decision graphs and explicit dependency relationships between data elements, rules, and reasoning steps. Enhanced data lineage mechanisms could record the origin of extracted values, their transformation history across layers, and associated confidence measures. Support for rule evolution would allow versioned tracking of regulatory logic over time, enabling retrospective analysis of compliance outcomes under changing regulatory conditions. Provenance could also be extended to explanations themselves, capturing why a specific explanation was generated, which reasoning paths were selected, and which model or configuration contributed to the outcome.

The Explanation Layer may evolve to support multiple explanation modes tailored to different interpretive needs. These include simplified summaries, comparative explanations across design alternatives, contextual explanations linked to regulatory intent, and counterfactual or what-if analyses. Accessibility considerations could be strengthened through multilingual support, text-to-speech output, and presentation formats suitable for users with specific accessibility needs. Interactive features such as drill-down navigation, three-dimensional visualisation, and heat-map overlays may improve user engagement and comprehension. Personalisation mechanisms could adapt explanations to user roles and expertise levels. Quality-oriented extensions may introduce metrics to assess explanation clarity and completeness, alongside the presentation of confidence indicators or alternative explanations to reflect uncertainty.

Future work may extend human-in-the-loop capabilities through structured and fine-grained feedback mechanisms that allow users to evaluate individual system components and provide contextual annotations. Such feedback could support active learning strategies, enabling the system to refine explanation templates, adjust reasoning priorities, and assess the effectiveness of corrective actions. Trust-related metrics may be formalised through explainability trust scores, rule- or element-specific confidence indicators, and longitudinal tracking of user trust. Collaborative workflows could support multi-user review, conflict resolution, and approval processes. Monitoring facilities such as usage analytics, A/B testing, and quality dashboards may inform continuous system improvement. Expert integration mechanisms could route complex or ambiguous cases for specialist review while capturing expert knowledge for future reuse. Finally, comprehensive audit and accountability features may provide full traceability of decisions, support regulatory documentation, and enable systematic error analysis, complemented by embedded training and user support functions.

# Key Findings and Challenges

## Key findings

* The layered design is stable and testable. Phase 1–2 demonstrates that the Data, Rule, and Reasoning layers operate as independent modules yet integrate through clear inputs and outputs. The Data layer extracts multiple element classes from IFC, the Rule layer evaluates a non-trivial rule set against the graph, and the Reasoning layer coordinates explanation, impact analysis, recommendations, and TRM support.
* The system remains operational under imperfect IFC content. Extraction routines skip malformed entities, and rule evaluation uses an explicit NOT\_APPLICABLE outcome when required fields are absent. This behaviour preserves availability on real models and yields partial results rather than hard failure.
* Behaviour is controlled through configuration rather than code. Rule definitions, thresholds, and recommendation templates are externalised in JSON. This design supports jurisdiction changes and reduces regression risk from frequent code edits.
* The graph representation is a useful intermediate artefact. The JSON building graph normalises IFC heterogeneity into a consistent structure that can be consumed by rule evaluation, reasoning, and feature extraction. It also supports extension to new element types without redesign of the core pipeline.
* Rule-informed feature engineering is functioning. The feature pipeline combines element, rule, and context signals into a fixed-size vector, which creates a concrete interface between symbolic checks and TRM learning.
* TRM outputs are suitable for triage. The model returns a binary compliance prediction with a confidence score and a short stepwise trace, which supports escalation of low-confidence cases and structured inspection of model behaviour.
* Recommendations are produced at multiple effort levels. The recommendation engine generates tiered remediation guidance, which links compliance findings to practical interventions with different scopes and implementation costs.
* The investigation confirms a data-governed failure mode. The fixed-accuracy pattern arises when dimensional data collapses to defaults, driving feature vectors towards constant values. This links model behaviour directly to extraction and enrichment quality rather than to model design.

## Key challenges

* IFC incompleteness and inconsistency remain the dominant technical risk. Missing dimensions, weak spatial relationships, and inconsistent identifiers occur in real submissions across schemas, which increases reliance on fallbacks and raises variance in downstream outputs.
* Unit handling is fragile. Heuristic conversions between metres, centimetres, and millimetres can fail on boundary values and are hard to validate when unit metadata is not preserved in the graph.
* Rule execution cost grows quickly with building size and rule count. Sequential evaluation produces multi-minute runs in realistic scenarios with thousands of elements and dozens of rules, which limits interactive use and increases perceived latency.
* Training data volume and coverage are insufficient. The TRM dataset remains small and narrow across building types and jurisdictions, which constrains generalisation and increases sensitivity to distribution shift.
* Explanation coverage is uneven across rule types. Numeric threshold rules map well to templates, yet complex checks and user-defined rules often fall back to generic text, reducing diagnostic usefulness.
* User-facing feedback during long runs is limited. The current frontend pattern blocks on backend completion without progress reporting, streaming partial results, or cancellation, which increases the operational cost of testing iterations.
* Provenance and auditability are not yet implemented. The system does not systematically record rule versions, data sources, transformations, timestamps, or request context, which weakens reproducibility and governance readiness.
* Borderline predictions lack structured treatment. Cases near decision boundaries do not trigger a consistent escalation path or feature-level rationale that explains uncertainty.
* Confidence scores are not empirically calibrated. Softmax probabilities are treated as confidence without a demonstrated link to observed correctness rates, which complicates threshold setting for acceptance and review.

# Discussion

This study set out to address a persistent limitation in automated compliance checking, namely the difficulty of producing compliance outcomes that remain inspectable, reconstructable, and trustworthy across technical and organisational boundaries. The results demonstrate that treating reasoning and traceability as core architectural concerns, rather than as reporting add-ons, leads to more stable and transparent compliance pipelines.

The implementation of ACE-X shows that a layered design can support this objective in practice. The Data, Rule, and Reasoning layers operate as independent modules with clear responsibilities, yet they remain tightly connected through explicit inputs and outputs. This separation supports robustness under imperfect IFC submissions, a condition frequently reported in real permitting workflows. The system’s ability to continue operation despite missing or malformed elements, while making these limitations visible through explicit outcomes and logs, aligns with regulator-facing concerns raised in recent ACC studies.

A key observation concerns the role of intermediate representations. The normalised building graph serves not only as a technical convenience but as a stabilising artefact that reduces IFC heterogeneity and supports consistent rule evaluation, reasoning, and feature extraction. This reinforces arguments in the literature that graph-based or structured intermediate models are central to scalable and auditable ACC pipelines. In this work, the graph also becomes the point at which traceability can later be anchored, since it preserves links to original IFC identifiers while remaining independent of schema-specific complexity.

The results also highlight the importance of data governance in learning-assisted compliance systems. The fixed-accuracy behaviour observed during early experiments did not arise from model design weaknesses but from feature collapse caused by missing dimensional data. Once this issue was identified and corrected, model behaviour varied meaningfully across buildings. This finding underscores a broader point. Learning components in ACC systems amplify upstream data quality issues, and opaque performance metrics can mask such failures if feature construction is not explicit and inspectable. The study therefore reinforces the need to align explainability requirements with data preparation and feature engineering, not only with model outputs.

The integration of the Tiny Recursive Model illustrates how compact learning components can complement symbolic checking without undermining rule authority. By positioning the model as advisory, and by exposing confidence values and stepwise reasoning traces, the system avoids shifting compliance decisions away from formal rules. This design responds directly to concerns in the literature regarding accountability and validation when machine learning is introduced into regulatory workflows. At the same time, the current results confirm that training data coverage remains a limiting factor, and that confidence values require calibration before they can support operational thresholds.

Several challenges remain evident. Rule execution cost grows with model size and rule count, which constrains interactive use in large buildings. Unit handling and heuristic conversions remain fragile where IFC models lack explicit unit metadata. Explanation quality varies across rule types, with complex or contextual checks proving harder to express through templates. Most importantly, provenance capture is not yet implemented as a first-class layer, which limits reproducibility and governance readiness despite architectural preparation.

These limitations do not undermine the core contribution of the study. Instead, they clarify where architectural decisions succeed and where further work is required. The results suggest that explainable automated compliance checking is less a problem of adding interpretive text and more a problem of designing pipelines in which decisions can be reconstructed from first principles.

# Conclusions

This paper introduced ET-ACC, a conceptual framework for explainable and trustworthy automated compliance checking and presented ACE-X as a partial operational instantiation of the framework. The work demonstrates that explainability, reasoning transparency, and traceability can be embedded within compliance architectures through deliberate layering and explicit interfaces between data, rules, and reasoning processes.

The implementation confirms that automated compliance checking can remain operational under real-world IFC imperfections while still exposing the limits of the input data and the logic applied. The study shows that structured reasoning outputs, impact analysis, and tiered recommendations can be generated in a deterministic and inspectable manner when rule execution and reasoning are treated as connected stages rather than isolated functions.

A central contribution lies in the analysis of learning-assisted compliance checking. The identification of a data-governed failure mode, where missing dimensions caused feature collapse and misleading accuracy, provides a concrete lesson for the design and evaluation of hybrid symbolic–learning systems. The results demonstrate that model behaviour in such systems is inseparable from extraction quality and feature semantics, and that explainability must extend into data preparation and training workflows.

While the current implementation does not fully realise the Provenance and Explanation layers, it establishes a clear architectural path for doing so without restructuring the pipeline. This supports future extensions that can capture lineage, rule evolution, and decision reconstruction in a systematic way. The work also provides a foundation for empirical evaluation across broader building types, regulatory regimes, and user roles.

In summary, the study argues that trustworthy automated compliance checking depends less on increasing automation depth and more on making compliance processes legible. By aligning conceptual design with executable artefacts, ET-ACC and ACE-X offer a structured step towards compliance systems that can be inspected, justified, and governed with confidence.

# References

Journal Article  
Sun, H., & Kim, I. (2026). Global advancements in BIM-based building e-permit system adoption: a review. Journal of Civil Engineering and Management, 32(1), 1–20. https://doi.org/10.3846/jcem.2026.24912. Retrieved from [https://journals.vilniustech.lt/index.php/JCEM/article/view/24912](https://journals.vilniustech.lt/index.php/JCEM/article/view/24912?utm_source=chatgpt.com), Last Access: January 22, 2026.

Journal Article  
Purushotham, N., Kailashnath, C., & Mutis, I. (2026). Framework for automated building code compliance checking to improve transparency, trust, validation, and design interpretation. Automation in Construction, 181(Part A), Article 106598. https://doi.org/10.1016/j.autcon.2025.106598. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580525006387, Last Access: January 22, 2026.

Journal Article  
Pauwels, P., van den Bersselaar, E., & Verhelst, L. (2024). Validation of technical requirements for a BIM model using semantic web technologies. *Advanced Engineering Informatics, 60*, Article 102426. https://doi.org/10.1016/j.aei.2024.102426. Retrieved from [https://www.sciencedirect.com/science/article/pii/S1474034624000740](https://www.sciencedirect.com/science/article/pii/S1474034624000740?utm_source=chatgpt.com), Last Access: January 22, 2026.

Journal Article  
Jia, L., Chen, M., Chen, C., Jin, Y. (2025). A Semantic Web and IFC-Based Framework for Automated BIM Compliance Checking. *Buildings, 15*(15), Article 2633. Retrieved from [https://www.mdpi.com/2075-5309/15/15/2633](https://www.mdpi.com/2075-5309/15/15/2633?utm_source=chatgpt.com), Last Access: January 22, 2026.

Journal Article  
Chen, N., Lin, X., Jiang, H., & An, Y. (2024). Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology. *Buildings, 14*(7), Article 1983. <https://doi.org/10.3390/buildings14071983>. Retrieved from [https://www.mdpi.com/2075-5309/14/7/1983](https://www.mdpi.com/2075-5309/14/7/1983?utm_source=chatgpt.com), Last Access: January 22, 2026.

Preprint

Jolicoeur-Martineau, A. (2025). Less is more: Recursive reasoning with tiny networks. arXiv preprint arXiv:2510.04871v1. Retrieved from https://arxiv.org/abs/2510.04871, Last Access: January 22, 2026.

Journal Article  
Celik, Y., Petri, I., & Barati, M. (2023). Blockchain supported BIM data provenance for construction projects. *Computers in Industry, 144*, Article 103768. https://doi.org/10.1016/j.compind.2022.103768. Retrieved from [https://orca.cardiff.ac.uk/152310/7/1-s2.0-S0166361522001646-main.pdf](https://orca.cardiff.ac.uk/152310/7/1-s2.0-S0166361522001646-main.pdf?utm_source=chatgpt.com), Last Access: January 22, 2026.

Conference Paper   
Alves de Mendonça, E., & Ferreira, S. L. (2024). Use case for Automated Code Compliance Checking of Accessibility Rules in BIM models using the IDS standard. Paper presented at the CIB W78 Conference (2024). Retrieved from [https://itc.scix.net/pdfs/w78-2024-paper\_21.pdf](https://itc.scix.net/pdfs/w78-2024-paper_21.pdf?utm_source=chatgpt.com), Last Access: January 22, 2026.

Journal Article

Noardo, F., Harrie, L., Ohori, K. A., Biljecki, F., Ellul, C., Krijnen, T., Eriksson, H., Guler, D., Hintz, D., Jadidi, M., Stoter, J., & others as listed by the journal. (2022). Unveiling the actual progress of Digital Building Permit: A state-of-the-art review. Building and Environment, 217, Article 109069. Retrieved from https://www.sciencedirect.com/science/article/pii/S0360132322001007

, Last Access: January 22, 2026.

Conference Paper (Published- Available Online)

Fuchs, S., [and co-authors as listed on the PDF]. (2025). The challenge of automated compliance checking: Perspectives from regulators and experts. Paper presented at the European Conference on Computing in Construction (EC3 2025). Retrieved from https://ec-3.org/publications/conferences/EC32025/papers/EC32025\_264.pdf

, Last Access: January 22, 2026.

Conference Paper (Published- Available Online)

Heiss, M., [and co-authors as listed on the PDF]. (2025). Extension of the Information Delivery Specification for BIM-based digital building permits. Paper presented at the European Conference on Computing in Construction (EC3 2025). Retrieved from https://ec-3.org/publications/conferences/EC32025/papers/EC32025\_210.pdf

, Last Access: January 22, 2026.

Conference Paper (Published- Available Online)

de Mendonça, E. A., & Ferreira, S. L. (2024). Automated compliance checking using IDS. Paper presented at the CIB W78 Conference (2024). Retrieved from https://itc.scix.net/pdfs/w78-2024-paper\_21.pdf

, Last Access: January 22, 2026.

Journal Article

Fischer, S., [and co-authors as listed on the journal page]. (2024). Extending Information Delivery Specifications for digital compliance checks. SoftwareX, 27, Article 101915. Retrieved from https://www.sciencedirect.com/science/article/pii/S2666165924002412

, Last Access: January 22, 2026.

Journal Article

Wang, C., [and co-authors as listed on the journal page]. (2024). Enhancement and validation of ifcOWL ontology based on IFC schema constraints. Automation in Construction, 159, Article 105248. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0926580524000293

, Last Access: January 22, 2026.

Journal Article

Tao, X., [and co-authors as listed on the journal page]. (2024). Smarter smart contracts for automatic BIM metadata compliance checking in collaborative design. Advanced Engineering Informatics, 60, Article 102627. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S1474034624002751

, Last Access: January 22, 2026.

Journal Article (Open Access PDF)

Alnuzha, M. (2025). The role of machine learning in automated compliance checking. Journal of Information Technology in Construction, 30, 1–xx. Retrieved from https://www.itcon.org/papers/2025\_02-ITcon-Alnuzha.pdf

, Last Access: January 22, 2026.

Conference Paper (Published- Available Online)

Saluz, U., [and co-authors as listed on the PDF]. (2025). Model alignment for automatic-compliance-checking: Large language model-based BIM processing. Paper presented at EG-ICE (2025). Retrieved from https://strathprints.strath.ac.uk/93288/7/Saluz-etal-EG-ICE-2025-Large-language-model-based-building-information-model.pdf

, Last Access: January 22, 2026.

Disclaimer/Publisher’s Note

The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and do not reflect the views of the Architecture, Buildings, Construction and Cities (ABC2) Journal and/or its editor(s). DFBI Journal and/or its editor(s) disclaim any responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.