Beyond Checklists: Automated Legal Compliance Check in Regulatory Sandboxes

Pietro Bernabei[1,2,†], Salvatore Bramante[1,3,*,†], Ludovica Ciarravano[1,2,†] and Fernando Pannullo[1,4,†]

[1]IMT School for Advanced Studies Lucca, Piazza S. Ponziano 6, Lucca, Italy
[2]University of Florence, Piazza di San Marco 4, Firenze, Italy
[3]Ca' Foscari University of Venice, Sestiere Dorsoduro 3246, Venezia, Italy
[4]Parthenope University of Naples, Via Ammiraglio Ferdinando Acton 38, Napoli, Italy

Abstract

This paper presents an integrated architecture for automated legal compliance assessment designed for use within Regulatory Sandboxes processes, with a first implementation targeting the essential cybersecurity requirements outlined in Annex I, Part I of the Cyber Resilience Act. The work addresses the difficulty of operationalising and enforcing a heterogeneous regulatory landscape, encompassing European Union regulations, national laws, and technical standards, particularly in the context of rapidly evolving and innovative products whose functionalities and risk profiles may change faster than the applicable legal frameworks. To implement the automated legal compliance assessment tool, we combine three complementary approaches. First, we formalise regulatory requirements using *Catala*, a domain-specific language based on prioritised default logic that captures the rule–exception structure of legal texts and guarantees total, deterministic evaluations. Second, we employ Large Language Model as a preprocessing tool that extracts and structures relevant information from manufacturers' documentation submitted in a format consumable by the *Catala* engine. Third, we ensure integrity, non-repudiation, and auditability of both inputs and results through a permissioned blockchain infrastructure built on *Hyperledger Fabric*, which anchors cryptographic hashes of documents and assessments. The overall system is implemented as a microservice architecture comprising a web-based User Interface, an API gateway and orchestrator, Large Language Model analysis services, the *Catala* compliance engine, and blockchain-backed persistence. A Proof of Concept focused on formal compliance demonstrates the feasibility and benefits of this tripartite approach, laying the groundwork for the logical translation of additional regulatory frameworks.

Keywords

Regulatory Sandbox, Formal Methods, Legal Compliance

1. Introduction

In recent years, the European Union (EU) has introduced a series of regulations targeting emerging digital technologies, cybersecurity, online platforms, and electronic communications to support innovation, promote fair competition, protect consumers, ensure data privacy, and address challenges arising from the digital transition. Prominent examples include Regulation (EU) 2024/2847 on horizontal cybersecurity requirements for products with digital elements - Cyber Resilience Act (CRA), and Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence - Artificial Intelligence Act (AIA). These regulations are developed to safeguard consumers and businesses buying software or hardware products with a digital component and address the risks stemming from Artificial Intelligence (AI) systems, respectively. Such regulations ensure high standards of security and reliability in sectors such as healthcare, financial services, and safety-critical systems. To facilitate both technological innovation and the protection of citizens, these regulations envisage the creation of experimental regulatory tools

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT
*Corresponding author.
†These authors contributed equally.
pietro.bernabei@imtlucca.it (P. Bernabei); salvatore.bramante@imtlucca.it (S. Bramante); ludovica.ciarravano@imtlucca.it (L. Ciarravano); fernando.pannullo@imtlucca.it (F. Pannullo)
0009-0004-0582-4591 (P. Bernabei); 0009-0007-9566-7666 (S. Bramante); 0009-0001-1180-3875 (L. Ciarravano); 0009-0004-4761-1295 (F. Pannullo)

called Regulatory Sandboxes (RSs). One of the main objectives of RSs is to provide structured legal and technical support to technology providers throughout the sandboxing process. Given the complexity of a multi-layered regulatory landscape, spanning international, European, and national levels, conducting regulatory testing and evaluation in an efficient manner can be particularly burdensome. To tackle this issue, we propose an integrated platform for automated legal check assessment that combines three key elements. First, we formalise regulations using *Catala*, a language created to represent laws with a system of conditions and exceptions. Second, we use Large Language Models (LLMs) to link the formal representation of applicable legal provisions included in the considered regulations, and the content of documents provided by the manufacturers to support their proof of compliance. These two methods automate the extraction and organisation of helpful information. Lastly, we implement a blockchain infrastructure to maintain the integrity and non-repudiation of the information declared by the manufacturer and generated by the system.

The main contribution of this work lies not in the individual technology components of the proposed automated legal check tool, which can already be found in the literature,[1], [2], [3], but in how they work together. *Catala* clearly expresses legal principles, which can improve precision in managing logic and development. Using LLMs can help connect human-readable documents with formal legal requirements and check for consistency. Implementing blockchain technology can ensure the transparency and reliability needed for sensitive regulatory check processes. Together, these elements create a system that supports the entire compliance cycle within RSs in an automated, verifiable, and flexible manner.

2. Background

2.1. EU Digital Regulation and Regulatory sandboxes

RSs are controlled regulatory environments in which innovative technologies, products, or services can be experimentally deployed under the direct supervision of regulators [4]. Originally developed in the fintech sector, they have since then expanded into other domains, including cybersecurity, artificial intelligence, and digital health. At the supranational level, European Union institutions have acknowledged the strategic importance of regulatory sandboxes, embedding them in key legislative frameworks such as the Artificial Intelligence Act (AIA), Interoperable Europe Act (IEA), and the Cyber Resilience Act (CRA). These acts foresee the establishment of dedicated national regulatory sandboxes to support the development and entry into the market of disruptive technologies that are compliant with the respective regulations and the relevant normative framework. Despite differences in their setup and legal provisions for their establishment,[5] within the EU legal framework, RSs have been conceived to serve two main purposes. On the one hand, they allow companies to test innovative technologies with the existing regulations and be guided in navigating the legal landscape (business learning). On the other hand, they enable regulators to observe real-world innovative technologies functioning and applications, identify potential gaps in existing laws, and, where applicable, iteratively refine the regulatory framework (regulatory learning). Art. 57 of the Artificial Intelligence Act (AIA) mandates National Competent Authorities (NCAs) to establish at least one AI regulatory sandbox by 2 August 2026 to facilitate the development, testing, and validation of AI systems before their placement on the market or putting into service. Participation does not exempt providers from compliance obligations but enables guided compliance and iterative risk mitigation prior to market deployment. Art. 33 of the Cyber Resilience Act (CRA) provides for the voluntary establishment of cybersecurity regulatory sandboxes to support the development and conformity assessment of products with digital elements, particularly hardware and software subject to the Regulation. Under both regulations, NCAs must, among other tasks, assist providers in understanding and meeting regulatory obligations under those Regulations and the relevant legal instruments, while also enabling controlled testing environments for innovative products. A central activity within RSs will be then to assess, in a continued dialogue between the regulator and the manufacturer, the compliance, or challenges to the compliance, of the tested product or service against the applicable legal framework. To support the NCAs in undertaking the assigned tasks within RSs, the present work introduces and develops an automated tool to facilitate

the compliance check of the tested product, process, or service by the competent authority, while also providing full control and understanding of the overall process. In order to conceive and develop the tool, we first identified the phases of the RSs processes to select those where the automated compliance check tool could provide better support to the regulator to assess the compliance of the product, process, or service tested with the relevant regulation. In this regard, although initiatives are still underway in order to provide comprehensive guidance on how to harmonise their establishment and operation [6, 7], the operationalisation of RSs typically involves three main distinct phases: (1) Application and selection phase, which may be either innovator-driven (bottom-up) or regulator-driven (top-down); (2) Participation and testing phase, during which admitted participants define the scope and parameters of the experimentation, develop detailed testing plans, and agree with the competent authorities on the applicable conditions, as formalised in the so-called Sandbox Plan. This phase involves the testing of the innovation within a controlled environment; and (3) Evaluation and exit phase, which concludes the sandbox process. The developed tool is especially conceived to support regulators during the Participation and testing phase and the Evaluation and exit phase. In the first phase, the tool will support NCAs in providing real-time guidance on legal interpretation and compliance as projects evolve, as well as ensure that obligations are clarified. In the Evaluation and exit phase, the tool will support NCAs in drafting exit reports and documenting legal and policy implications.

2.2. Catala Language

Catala [8] is a domain-specific programming language expressly designed to formalise statutory law through a precise and unambiguous computational model. To bridge the semantic gap between legal prose and executable code, *Catala* adopts the paradigm of *literate programming* [9], ensuring that every line of code is inextricably linked to the legislative text it implements. Its formal foundation is *prioritised default logic* [10], a non-monotonic logical framework that naturally captures the "general rule/exceptions" structure pervasive in legal texts. In *Catala*, every legal provision is expressed as a combination of a default case and a hierarchy of prioritised exceptions, which are compiled into a statically known tree of default rules. This ensures deterministic evaluation and mirrors the interpretative patterns used by lawyers, avoiding the complexity of nested conditional control flows.

The language is structured around *scopes*, which define logical sections of the law. Each scope contains *context variables* whose values are computed through *definitions* and *rules*. Definitions introduce data-level computations, while rules introduce boolean conditions equipped with a default value of `false`, reflecting the logic of legal requirements. This declarative structure allows contexts to override values of subscopes, enabling the modelling of complex legal patterns such as backpatching, forward references, and local refinements.

Semantically, *Catala* programs are desugared into a *scope language* and subsequently compiled into the *default calculus*, a minimal lambda calculus extended with explicit default terms. A default term d takes the form $d = \langle e_i \mid e' \ :- \ e'' \rangle$, encoding a static priority tree where e_i represents the list of exceptions, e' the precondition (justification), and e'' the consequence. Evaluation follows this static priority deterministically. The operational semantics of a default term dictate a strict evaluation order. First, the exceptions e_i are evaluated recursively. If exactly one exception triggers (i.e., evaluates to a non-empty value), its result is returned, preempting the current rule. However, if multiple exceptions trigger simultaneously without a precedence order defined between them, the evaluation halts immediately with a *conflict error* ($\otimes$). This rigorous check prevents legal ambiguities that could arise from contradictory rules potentially applying to the same case. If no exception applies, the system checks the precondition e'. If e' holds true, the consequence e'' is returned; otherwise, the term evaluates to a special empty value $\emptyset$, indicating that the rule is not applicable. This calculus is then compiled into a standard lambda calculus extended with options and lists. Crucially, *Catala* enforces that programs are non-Turing-complete: recursion is forbidden, ensuring that all legal computations are total and terminating.

Finally, the *Catala compiler* enforces a topological ordering of definitions within each scope, rejecting programs with circular dependencies. This yields an executable and formally verified semantics, supported by a mechanised correctness proof of the compilation process in the $F^\star$ proof assistant [11].

2.3. Blockchain and Hyperledger Fabric

Blockchain functions as a secure sequence of blocks that contains transaction data and previous block hashes [12]. The connection system between blocks makes it impossible to alter previous blocks, as any modification would force nodes to recalculate all subsequent blocks, which becomes impractical when the network spans multiple nodes with extended blockchains. The Bitcoin and Ethereum networks function as permissionless public networks, which enable any user to access the system for transaction submission and consensus participation [13]. The open nature of public blockchains enables censorship resistance and eliminates the need for trusted third parties, but it creates performance issues and privacy concerns that make them unsuitable for enterprise and regulatory use cases. Public blockchains such as Bitcoin maintain restricted transaction processing at 10-20 transactions per second because they need to perform proof-of-work consensus calculations and maintain complete network state information [14] and expose all data to the network, violating regulatory standards that require controlled information disclosure and protected sensitive information. Moreover, each transaction has a cost in terms of monetary cost. Permissioned blockchain platforms solve public blockchain restrictions through built-in access control systems, which enable only authorised entities to access the network [15]. The new design approach enables three essential features that suit business environments. First, the transition from proof-of-work to Byzantine fault-tolerant consensus protocols enables faster finality times of seconds instead of minutes [16]. Second, the system enables users to control who can view their transactions by setting specific access permissions, rather than sharing all data with everyone. Lastly, the system allows organisations to implement their policies and regulatory needs through automated smart contracts and permission management systems.

Hyperledger Fabric operates as a permissioned blockchain system that serves enterprise consortia and governed networks [15]. More specifically, the Linux Foundation's Hyperledger umbrella project developed Fabric as a blockchain platform, which stands out through its modular design that allows users to customise consensus algorithms, membership protocols and cryptographic functions. Indeed, the platform's adaptable design enables organisations to meet different regulatory needs and operational requirements through customisations that do not require changes to the fundamental platform code.

The Fabric architecture uses execute-order-validate transaction flow and operates differently than Ethereum's traditional order-execute blockchain system [17]. The process of transaction ordering occurs first through consensus, followed by the sequential execution of all nodes to calculate state updates in traditional blockchain systems.

The Membership Service Provider (MSP) utilises public key infrastructure and X.509 certificates to manage identities, as described in [18]. Organisations use their certificate authorities to distribute cryptographic identities to members through public key and organisational attribute binding certificates. The MSP system checks transaction signatures against authorised certificate authorities to verify that only permitted identities can perform transactions and execute *chaincode* functions.

Smart contracts in Fabric, known as *chaincode*, operate through separate Docker containers which establish protected security domains between different contract instances [19]. The containerization system protects *chaincode* from interfering with other contracts and prevents access to system resources which helps defend against security threats from harmful or flawed contract code. The ledger state database supports two storage options: LevelDB for basic key-value operations and CouchDB for handling complex JSON document queries.

The modular design of Fabric allows users to select their preferred consensus protocol, which usually runs Raft for ordering service operations [20]. The Raft protocol maintains system operation through leader election processes, even when network nodes experience failures. The Fabric platform enables BFT-based ordering through the BFT-SMaRt protocol for networks that need protection against malicious node attacks [21].

All Fabric system components use Transport Layer Security (TLS) for encryption, which protects data from unauthorised access and intercepts malicious attacks [22]. The mutual TLS authentication process enables secure data exchange because both parties verify their identities through certificate checks before starting communication. The network remains secure because all nodes must authenticate each

other through mutual TLS before they can access network resources and receive messages from other participants.

Access control policies, expressed through Fabric's policy language, specify which identities or organisational roles can perform specific operations [23]. Endorsement policies determine which organisations must endorse transactions before they are considered valid, implementing multi-party approval workflows. Lifecycle policies govern *chaincode* installation and instantiation, ensuring that contract code cannot be deployed without an appropriate organisational agreement. Reader and writer policies control access to private data collections, implementing a need-to-know approach to data sharing.

2.4. Large Language Models

Large Language Models (LLMs) represent a transformative class of artificial intelligence systems trained on vast corpora of text to perform language understanding and generation tasks. Architecturally based on transformer neural networks [24], these models, including GPT4 [25], Claude, and Gemini, are pre-trained using self-supervised learning objectives such as next-token prediction. This training enables LLMs to capture complex syntactic, semantic, and pragmatic patterns across domains, allowing them to perform zero-shot or few-shot generalisation on downstream tasks without domain-specific fine-tuning.

From a regulatory technology viewpoint, LLMs provide considerable benefits in automating the handling of unstructured, varied documentation. They can extract information, classify data, summarise content, answer questions, and generate structured outputs. These features make them especially useful for connecting human-readable compliance documents with machine-readable formal representations. However, their use in high-stakes legal or regulatory areas needs to consider several limitations:

- LLMs work as probabilistic models and do not guarantee logical consistency or factual accuracy, especially in situations that need clear legal interpretation or specific reasoning [26].
- LLMs black-box design creates problems for transparency and explainability, which are important in regulated areas.
- General-purpose LLMs are not designed for the structure of legal texts, which often include complex terminology, cross-references, and binding language.

2.5. The Testing Scope

To develop a Proof-of-Concept (PoC) of the proposed tool, we simulated the central phase of a RSs, i.e., the activity within the Participation and Testing phase where the regulator is asked to provide legal guidance on the compliance of the tested product, service, or product against the provisions considered within the RSs. For the PoC we considered the application to the RSs of a manufacturer of an IoT artifact which falls within the scope of the CRA and where, under the Sandbox Plan, the compliance with the essential cybersecurity requirements set forth in the Regulation is met.

The Cyber Resilience Act (CRA) [1] is a European Union regulation adopted on 23 October 2024 and designed to enhance the security of hardware and software products with digital elements. Its primary objective is to ensure that products placed on the EU internal market meet baseline cybersecurity requirements throughout their entire lifecycle, from design and development to deployment, maintenance, and end-of-life. The CRA introduces mandatory security-by-design principles, vulnerability management obligations, conformity assessment procedures, and post-market surveillance rules for manufacturers, importers, and distributors.

A central feature of the CRA is its risk-based approach: products are categorised into classes according to their potential security impact, with stricter obligations for important products with digital elements (Class I and Class II) and critical products with digital elements, identified in Annex III and Annex IV respectively. The Regulation also harmonises cybersecurity rules applicable to products included in its scope across Member States, aiming to reduce fragmentation and improve the overall resilience of the digital single market. Annex I of CRA, in particular, enumerates "essential cybersecurity requirements",

including cybersecurity requirements relating to the properties of products with digital elements and vulnerability handling requirements to be implemented by manufacturers. Implementing these requirements demands not only technical controls but also structured, repeatable processes that can be verified and audited. Such assessment applies to products with digital elements that are also in the scope of other legal acts, including high-risk AI systems under the AIA. The CRA provides that Products with Digital Elements (PDEs) that qualify as high-risk AI systems under the AIA shall also comply with the essential cybersecurity requirements under the CRA and may, where they are compliant, be deemed to be compliant with relevant cybersecurity requirements set out in Article 15 of the AIA dealing with ”Accuracy, robustness and cybersecurity” of high-risk AI systems.

To support organisations, especially start-ups and Small and Medium Enterprises (SMEs) in developing products compliant with the regulatory framework established by both the CRA and the AIA, these regulations, with some relevant differences, foresee the establishment of dedicated national or pan-European RSs.

The present work develops a Proof of Concept (PoC) based on the implementation of a RS for testing a product with digital elements under the CRA as envisioned in Article 33 of the Regulation. Article 6 CRA states that ”products with digital elements shall be made available on the market only where: (a) they meet the essential cybersecurity requirements set out in Part I of Annex I, provided that they are properly installed, maintained, used for their intended purpose or under conditions which can reasonably be foreseen, and, where applicable, the necessary security updates have been installed; (b) the processes put in place by the manufacturer comply with the essential cybersecurity requirements set out in Part II of Annex I [...].” This PoC can also be applied within AI Regulatory Sandboxes (AIRSs) for testing the compliance to cybersecurity requirements of high-risk AI systems as provided by Article 15 of the Artificial Intelligence Act. In this regard, Article 12 CRA provides that ”1. Without prejudice to the requirements relating to accuracy and robustness set out in Article 15 of Regulation (EU) 2024/1689, products with digital elements which fall within the scope of this Regulation and which are classified as high-risk AI systems pursuant to Article 6 of that Regulation shall be deemed to comply with the cybersecurity requirements set out in Article 15 of that Regulation where: (a) those products fulfil the essential cybersecurity requirements set out in Part I of Annex I; (b) the processes put in place by the manufacturer comply with the essential cybersecurity requirements set out in Part II of Annex I.

For the present PoC, Annex I, Part I to CRA has been considered and tested through a simulated IoT artifact as in Listing 1:

Listing 1: IoT realistic artifact

```
"""
[Sentry Shield] Smart Wi-Fi Combustible Gas & CO Detector:

- Smart, 24/7 protection from invisible threats like combustible gases (Natural Gas/Methane and
    Propane) and deadly Carbon Monoxide (CO). Get instant alerts on your phone before a leak
    becomes a disaster.

- Unlike traditional detectors that only sound a local alarm, the Sentry Shield is an IoT device
    . It connects directly to your home's 2.4GHz Wi-Fi network to send instant, real-time alerts
     straight to your smartphone via our companion app.

- Featuring advanced electrochemical and semiconductor sensors, a powerful 85dB alarm, and vocal
     alerts that clearly state the nature of the threat.

- Equipped with two distinct, high-precision sensors:
    - Electrochemical Sensor: Detects dangerous levels of Carbon Monoxide (CO).
    - Semiconductor Sensor: Detects a wide range of combustible gases including Methane and
        Propane.

### Technical Specifications
```

```
| Feature | Specification |
| Power Source | AC 110-240V Plug-in |
| Connectivity | 802.11 b/g/n (2.4GHz Wi-Fi) |
| Sensors | 1x Electrochemical (CO), 1x Semiconductor (Combustible) |
| CO Sensor Lifespan | 7 Years |
| Combustible Gas Sensor | 5 Years |
| Alarm Volume | 85 dB @ 3 meters (10 feet) |
| App Compatibility | iOS 12.0 or later / Android 6.0 or later |
| Detection Range (CO) | 50-999 PPM |
| Detection Range (Gas) | 6% - 20% LEL (Lower Explosive Limit) |
| Operating Temperature | 0C to 50C (32F to 122F) |
| Certifications | UL 2034, UL 1484 (pending), FCC, CE |

### What's in the Box

* 1 x [Sentry Shield] Smart Wi-Fi Gas & CO Detector
* 1 x User Manual & Setup Guide
* 1 x Mounting Screw (for securing to the outlet)
"""
```

3. Methodology

3.1. High-level view

Figure 1 provides a high-level overview of the platform architecture, highlighting the main actors involved, the information flows and the interaction between the logical-symbolic and neural components.

The platform is based on translating legislation into a logical representation suitable for automatic processing. This phase constitutes the initial population of the backend's "logical brain", implemented through *Catala*. The codification of the law requires the collaboration of an IT expert and a legal expert, who translate each legislative provision into *Catala*. The *Catala* development environment supports this activity by allowing each element of the law to be annotated with its corresponding formal representation.

This direct coupling between regulatory text and formal code enables the accurate revision of the underlying logic and facilitates modular work on the legislation. Each article, clause, or paragraph can be managed, versioned, and evolved independently, without having to immediately address the entire set of exceptions and versions. This modularity, combined with the explainable nature of *Catala*, is one of the main reasons for choosing this language.

While the backend handles the formalisation and evaluation of the rules, the producer interacts with the platform via a graphical interface that presents two types of fields:

- non-documentary information, consisting of simple values (mainly booleans from checkboxes and strings from textboxes) which can be transmitted directly to the backend;
- documentary information, which instead requires the analysis of technical documents provided by the producer.

This second category represents the core of the compliance verification process, as it reproduces the activity currently performed manually by inspectors. In the platform, this activity is automated using LLMs. As discussed in detail in the next section, LLMs are assigned the task of extracting all the necessary information for formal verification from the documents, using prompts specifically designed for semantic extraction.

The use of LLMs simplifies interaction with the manufacturer, who only has to answer a minimum number of questions. The form indicates the type of document required and the information to be included, but does not impose any particular format, structure or layout. The producer can therefore upload documents that are already available, provided they contain the required content. The ability of

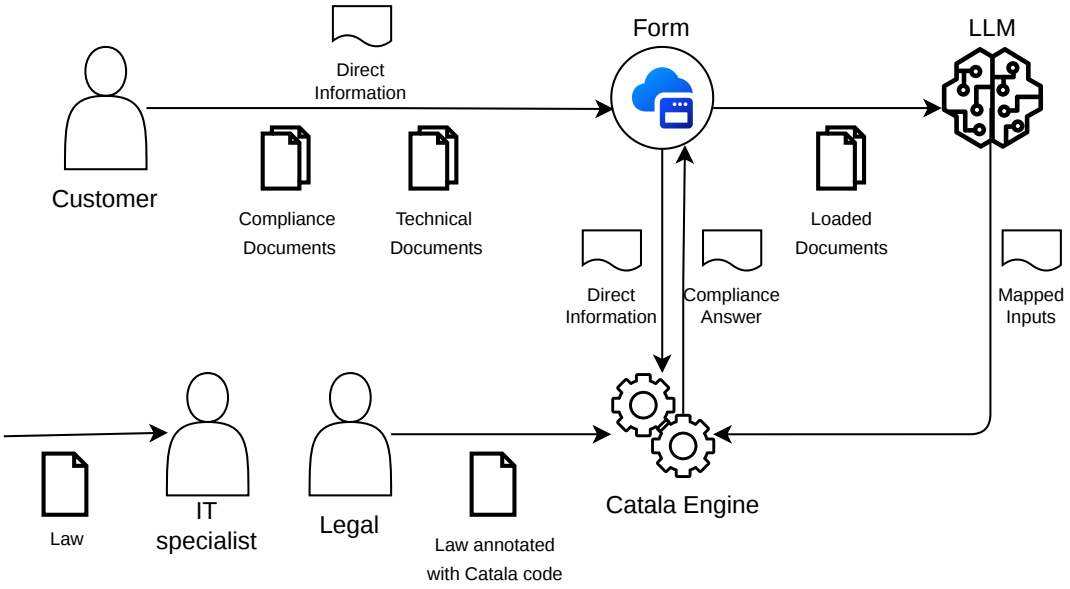

Figure 1: High-level view of the platform concept

LLMs to operate independently of the document's editorial structure reduces the ambiguities typical of manual procedures and facilitates the management of heterogeneous documentation.

The output of the LLM consists of structured data, accompanied by an explanation of the reasons and a confidence value, which are then transferred to the *Catala backend* to complete the data necessary for compliance verification. This combination of neural analysis and logical-symbolic processing is designed to mitigate the opacity of neural models, providing a level of explainability compatible with the requirements of legislative sandboxes. At the end of the backend decision-making process, the overall result is returned to the user interface, accompanied by the LLM's motivations behind the information extraction process on the documents.

The process guarantees two fundamental forms of transparency. On the one hand, the producer can understand the reasons for any non-compliance by examining both the *Catala backend* output and the comments generated by the LLM on its documents. This allows them to identify document deficiencies or interpretative errors on the part of the model, requesting the intervention of a human operator if necessary. On the other hand, the regulatory body can review the extraction process to identify possible pitfalls in the regulations or operational criteria adopted.

3.2. In details

The developed tool implements an automated compliance assessment framework specifically designed to assist regulators in evaluating conformity of products under testing within a regulatory sandbox with CRA provisions. The system architecture has been designed to be inherently extensible, enabling the future integration of AIA compliance evaluation alongside the current CRA implementation and other legal norms. This approach addresses the growing complexity of bringing digital products to the European market, where cybersecurity and artificial intelligence legal framework increasingly intersect. The fundamental design combines three complementary technological paradigms. First, formal legal verification through the *Catala* domain-specific programming language provides a mathematically rigorous evaluation of regulatory compliance. Second, LLM integration enables intelligent processing of unstructured documentation and natural language inputs from providers. Third, blockchain-based data integrity mechanisms ensure complete auditability and non-repudiation of the assessment process. This tripartite architecture creates a system that is simultaneously rigorous, user-friendly, and trustworthy

from a regulatory perspective. The dynamic LLM prompt is structured as in Listing 2.

Listing 2: Introductive prompt to LLM

```
"""
You are a regulatory compliance expert specialising in the EU Cyber Resilience Act (CRA),
    specifically Annex I Part I Essential Cybersecurity Requirements. Your task is to analyze
    product information from a Provider's Questionnaire and supporting documents to determine
    boolean compliance attributes. You must respond in a structured format with: DECISION (true/
    false), CONFIDENCE (0.0-1.0), and JUSTIFICATION (brief explanation based on specific
    evidence from the documents). Be conservative: if information is unclear or missing, choose
    FALSE with lower confidence.

## Your Analysis Task:

Based on the product information and documentation provided above, determine if the attribute
    **{attr_name}** should be **TRUE** or **FALSE**.

### Decision Guidelines:
1. **TRUE**: Only if there is CLEAR EVIDENCE in the product information that this requirement is
     met
2. **FALSE**: If information is missing, unclear, or explicitly shows the requirement is NOT met
3. Look for the keywords mentioned above in the product information
4. Consider both explicit statements and implicit evidence
5. Be conservative: when in doubt, choose FALSE with lower confidence

### Response Format (REQUIRED):

You MUST respond using this EXACT format:

DECISION: [true or false]
CONFIDENCE: [number between 0.0 and 1.0]
JUSTIFICATION: [One or two sentences explaining your decision, citing specific evidence from the
     product information]

### Confidence Guidelines:
- **0.0-0.3**: No relevant information found, or contradictory information
- **0.4-0.6**: Some relevant information, but unclear or incomplete
- **0.7-0.9**: Clear relevant information supporting the decision
- **0.9-1.0**: Explicit and comprehensive evidence supporting the decision

### Important Notes:
- Base your decision ONLY on the information provided in Product Information and Documentation
- Cite specific text from the product information in your justification
- Keep justification concise (1-2 sentences maximum)
- If no information is available for this attribute, return FALSE with confidence 0.0
"""
```

3.3. The architecture

The system architecture organises functionality across five distinct layers, each implementing specific aspects of the compliance assessment workflow. The layered design follows a pipeline pattern where data flows sequentially through processing stages, with each layer adding value through transformation, enrichment, or validation operations.

The system architecture divides its functionality into five separate layers, which execute distinct functions for compliance assessment workflow management. The system design uses a pipeline structure

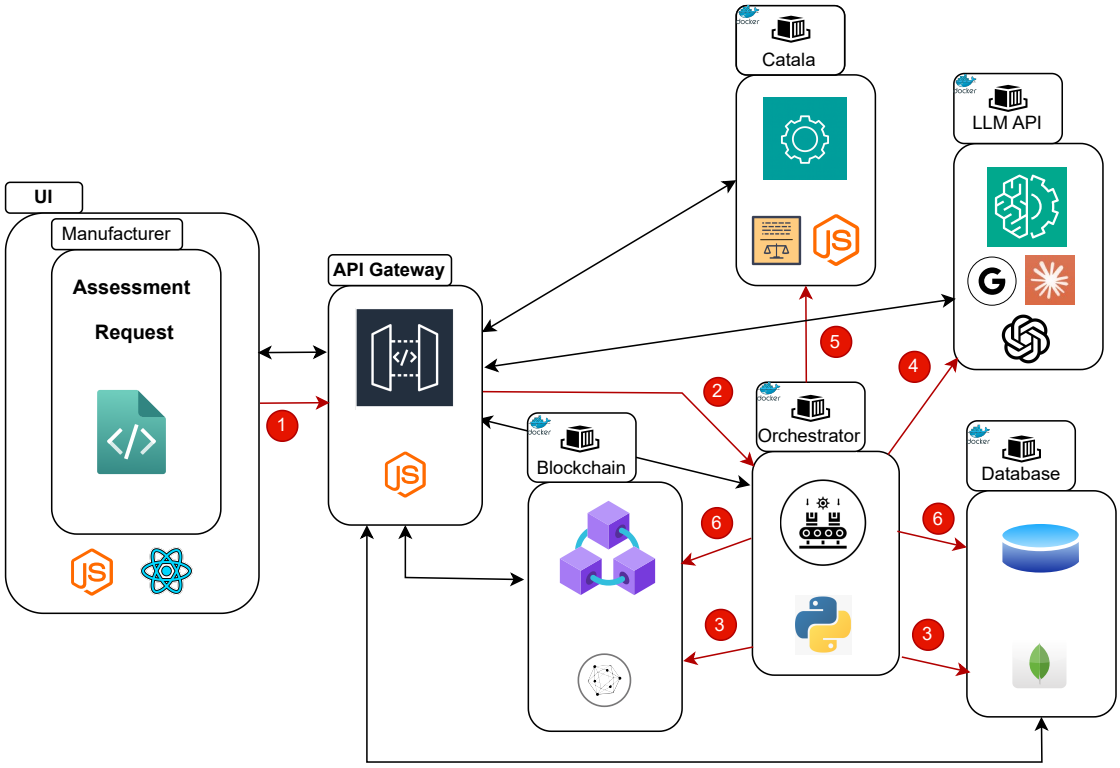

Figure 2: Comprehensive architecture of the tool: Containerised and host modules. In red, the phases of the workflow and in black, the inter-module communication system. Phase 1: The request and the data for the assessment are sent. Phase 2: The Application Programming Interface (API) Gateway and sends the command to initiate the pipeline of the assessment to the Orchestrator. Phase 3: The Orchestrator save the inputs into the Blockchain and the DB. Phase 4: Initialize the LLM Input mapping. Phase 5: The Orchestrator initializes the *Catala* final assessment. Phase 6: The Orchestrator, finalize the pipeline by saving the results into the blockchain and DB

to process data through sequential stages, which perform transformation, enrichment, and validation operations.

The *User Interface (UI)* component at the presentation layer serves as the main entry point for product providers to interact with the system. The presentation layer operates as a Node.js with React web application, displaying organised questionnaires to help providers complete their compliance documentation requirements. The questionnaire structure mirrors the regulatory organisation of CRA Annex I through its organised sections, which match particular cybersecurity requirements categories. The system enables users to fill out extended forms that request both control implementation declarations and proof documents to validate their statements. The system accepts PDF reports and declarations to support the verification. Additionally, the report includes blockchain transaction hyperlinks, allowing users to retrieve the complete assessment report using their assessment identifier, which they obtained when starting the evaluation process through the UI.

The integration layer contains two essential components: the *API Gateway* (host service) and the *Orchestration Service* (dockerized), which operate as the system's core management system. The Node.js-based service operates as a RESTful endpoint server that connects the frontend to backend services through all necessary API calls. The orchestration service performs advanced workflow management to execute complete compliance assessment operations through multiple backend systems. The system uses session management to preserve user state during multiple-step interactions, while authentication and authorisation protocols restrict access to assessment initiation and result viewing to authorised providers.

The *Assessment Orchestrator* operates as a dedicated Docker service that handles complex pipeline management operations. The system utilises a finite state machine to monitor assessment development from start to finish, encompassing submission and report creation. The system enables users to schedule tasks for background processing, thereby preventing system timeouts and enhancing the overall user experience. The orchestrator system includes built-in error handling and retry functionality, which maintains system stability when downstream services experience brief interruptions.

The *LLM API Service* operates in conjunction with the orchestrator to transform human-readable documentation into machine-readable, structured data. The Python-based service operates with three major LLM providers through a redundant system as a fail-safe system. The main integration of OpenAI services uses OpenAI's GPT API. The system switches to Anthropic's Claude API as its secondary provider when OpenAI services become unavailable or reach their rate limits. The system maintains LLM functionality through Google's Gemini API when all other options fail. The system design utilises multiple providers because it recognises that API availability and rate limits can create actual operational issues. In this work, LLMs are used to parse documentation provided by manufacturers, such as technical manuals, cybersecurity policies, and vulnerability reports, in order to produce structured outputs aligned with the data model required for formal compliance evaluation in *Catala*. By limiting LLMs to the preprocessing stages, the architecture leverages their language skills while mitigating the potential risks associated with errors. This use of LLM demonstrates a practical and safety-focused approach to applying generative AI in regulatory systems.

The *Catala* compliance engine operates as the verification layer (*Catala Service*), representing the most advanced element of our technical solution. The programming language *Catala* exists for encoding legal rules into executable code that can be verified through formal methods. The exception-based structure found in legal texts receives natural representation through prioritised default logic that serves as the semantic foundation for *Catala*. The *Catala* engine operates inside a containerised system, which provides a REST API for processing structured compliance information to generate detailed evaluation outputs.

Given the limitations of public blockchain architectures outlined in the background section, the system adopts a permissioned blockchain model to ensure controlled participation, regulatory-aligned data handling, and efficient transaction processing without paying for each transaction. Within this framework, the persistence layer achieves blockchain-based data integrity through *Hyperledger Fabric* operating as a permissioned distributed ledger system. The system utilises MongoDB to store document hashes using a document hash anchoring pattern, rather than storing full assessment data on the blockchain. The system uses *SHA-256* cryptographic hashing to create an unalterable audit trail by storing evidence documents and assessment result hashes in the blockchain without revealing confidential information.

The network operates through a multi-organisation consortium, enabling participants to maintain their own peer nodes and Certificate Authority (CA) functions for identity management and TLS certificate administration. The MSP operating within each organisation enables advanced authorisation functions to manage both transactional access and channel permissions so that only authenticated identities can interact with sandbox smart contracts. The network features three separate logical communication routes, which include the main *RegulatoryChannel* for all organisations and two additional channels for private data exchange and audit trail monitoring. The system design implements "compliance by design" because all document changes produce permanent transactions that become accessible to regulators and auditors when needed. The platform enables researchers to develop new regulatory models through its channel-based architecture, permissioned identity systems, and document verification tools, eliminating the need for intermediaries and post-deployment verification.

3.4. Workflow

The system starts its workflow, as shown in Fig. 2, when the human operator finishes their surveys and submits them through the UI. The frontend system runs two verification processes to confirm that all necessary fields receive responses and all file upload tasks finish successfully. The application

sends an HTTP POST request to the API Gateway after validation with the assessment data and document metadata. The request includes authentication tokens, which the gateway uses to authenticate provider identity before initiating the assessment evaluation. After that, the API Gateway creates a new assessment session in the MongoDB database, assigning a unique assessment identifier that will track the evaluation through all subsequent processing stages. This identifier enables status queries and result retrieval once processing completes. The gateway immediately responds to the frontend with the assessment identifier and a pending status, allowing providers to close their browser or navigate away while processing continues asynchronously. With the assessment session established, the gateway invokes the Assessment Orchestrator to begin pipeline execution. The orchestrator's first action retrieves uploaded documents from storage and computes *SHA-256* hashes for each file. These hashes are immediately submitted to the blockchain service for anchoring, creating the initial audit trail entry that establishes evidence submission timing. The orchestrator awaits blockchain transaction confirmation before proceeding, ensuring that the audit trail is established before any analysis occurs. Once document hashes are safely anchored, the orchestrator initiates LLM analysis by invoking the LLM service API. The document files are transmitted to the LLM service along with metadata indicating the CRA requirements to which each document relates. The LLM service executes its multi-stage pipeline, including document preprocessing, chunking for large files, analysis through the primary LLM provider and structured output generation. The entire LLM processing may require several minutes for comprehensive document sets, justifying the asynchronous architecture. Upon receiving LLM analysis results, the orchestrator validates the structured outputs against expected schemas. Validation failures trigger retry logic that attempts LLM processing again with the secondary provider, under the assumption that the initial failure might reflect transient issues or provider-specific limitations. After successful validation, the orchestrator persists the analysis results in the database and updates the assessment status to indicate completion of processing. The orchestrator now possesses all inputs required for *Catala* evaluation. It constructs the JSON input payload by merging questionnaire responses from the database with LLM-generated labels and classifications. This merged dataset undergoes final validation against the *Catala* input schema before submission. The orchestrator invokes the *Catala* engine API, transmitting the complete input dataset and receiving detailed compliance evaluation results. The *Catala* evaluation results undergo immediate validation to ensure proper structure and completeness. The orchestrator extracts requirement-level compliance determinations and aggregates them into an overall assessment outcome. The detailed *Catala* output is serialised to JSON and submitted to the blockchain for recording alongside the hash value. This creates a cryptographic binding between the blockchain record and the detailed results stored in traditional databases. In the end, the UI generates an assessment web-report by uniting results from all processing stages after completing all tasks. The report includes a comprehensive compliance summary that examines all requirements in depth to identify evidence control weaknesses and provide step-by-step instructions for resolving non-compliant areas.

4. Conclusion and Future Work

The project presented demonstrates the feasibility of integrating regulatory formalisation, analytical capabilities through linguistic models, and blockchain-based integrity guarantees into a single structured compliance verification process. The implementation of Annex I of the CRA has enabled the validation of this integration on a substantial subset of the legislation. This demonstrates the efficacy of combining formal languages, LLM and distributed infrastructures in supporting assessment processes within the framework of regulatory sandboxes.

The potential future developments are manifold. Firstly, encoding additional European and national regulations and related standards would pave the way for a more comprehensive and generalisable platform. This also includes the necessity of overcoming the current limitations of *Catala*, which was originally designed for tax law, so that it can fully represent the peculiarities of technical and cybersecurity-related rules.

A secondary axis of development pertains to the evolution of *Catala*'s logical engine into a neurosymbolic approach, with the objective of automatically detecting inconsistencies, ambiguities, or pitfalls in the regulations represented. This capacity would be especially pertinent in regulatory sandbox contexts, where iterative adaptation of legislation is an integral component of the process.

A further enhancement concerns the role of LLMs: the integration of *Catala*'s logical structures directly into LLM's query generation would allow for deeper automation of the alignment between descriptive documentation and formal regulatory constraints. The enhancement of consistency between these two levels has the potential to optimise the efficiency of compliance verification, thereby minimising the necessity for human intervention during the information extraction and structuring phases.

To enhance security, the total encrypted and authenticated communication is achieved through microservices leveraging Trusted Execution Environment (TEE) that integrates and completes the security mechanism provided by the blockchain.

Finally, the introduction of a multi-agent discussion system between LLMs and the development of a local LLMs model specifically trained for this domain, could increase the reliability of document analysis by leveraging the convergence of different or specialised models. This approach would ensure greater robustness in the critical stages of document interpretation and recognition of information relevant to compliance.

In summary, the work establishes the foundations for an integrated and verifiable ecosystem to support legal compliance in regulatory sandboxes, paving the way for a series of future developments that can make the process increasingly automated, adaptive, and transparent.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5 and Gemini 2.5 in order to: Grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] *Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) (Text with EEA relevance)*, en, Legislative Body: CONSIL, EP, Oct. 2024. Accessed: Nov. 19, 2025. [Online]. Available: http://data.europa.eu/eli/reg/2024/2847/oj.

[2] J. Zhang and N. M. El-Gohary, "Automated information transformation for automated regulatory compliance checking in construction," *Journal of Computing in Civil Engineering*, vol. 29, no. 4, B4015001, 2015.

[3] V. Kulkarni, S. Sunkle, D. Kholkar, S. Roychoudhury, R. Kumar, and M. Raghunandan, "Toward automated regulatory compliance," *CSI Transactions on ICT*, vol. 9, no. 2, pp. 95–104, 2021.

[4] *European Commission, better regulation toolbox*. Accessed: Dec. 15, 2025. [Online]. Available: https://commission.europa.eu/law/law-making-process/better-regulation/better-regulation-guidelines-and-toolbox/better-regulation-toolbox_en.

[5] F. Seferi et al., "A comparative analysis of regulatory sandboxes from selected use cases: Insights from recurring operational practices," in *Regulatory sandboxes for AI and Cybersecurity. Questions and answers for stakeholders*, CINI's Cybersecurity National Lab, 2025, pp. 145–176.

[6] S. Duff and I. Jeník, *How to build a regulatory sandbox : A practical guide for policy makers*, en. Accessed: Dec. 15, 2025. [Online]. Available: http://documents.worldbank.org/curated/en/126281625136122935.

[7] OECD, *Regulatory sandbox toolkit: A comprehensive guide for regulators to establish and manage regulatory sandboxes effectively*, en, Publisher: OECD Publishing, Jul. 2025. DOI: 10.1787/de36fa62-en. Accessed: Dec. 15, 2025. [Online]. Available: https://www.oecd.org/en/publications/regulatory-sandbox-toolkit_de36fa62-en.html.

[8] D. Merigoux, N. Chataing, and J. Protzenko, "Catala: A programming language for the law," *Proceedings of the ACM on Programming Languages*, vol. 5, no. ICFP, pp. 1–29, 2021.

[9] D. E. Knuth, "Literate programming," *The computer journal*, vol. 27, no. 2, pp. 97–111, 1984.

[10] G. Brewka and T. Eiter, "Prioritizing default logic," in *Intellectics and Computational Logic: Papers in Honor of Wolfgang Bibel*, Springer, 2000, pp. 27–45.

[11] N. Swamy et al., "Dependent types and multi-monadic effects in f*," in *Proceedings of the 43rd annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2016, pp. 256–270.

[12] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proceedings of the IEEE International Congress on Big Data (BigData Congress)*, IEEE, 2017.

[13] G. Wood, *Ethereum: A secure decentralised generalised transaction ledger*, Ethereum Project Yellow Paper, 2014. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf.

[14] "Rebello, g.a.f., camilo, g.f., guimarães, l.c.b. et al.," *Annals of Telecommunications*, vol. 77, no. 2, pp. 97–111, 2022.

[15] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the 13th EuroSys Conference*, ACM, 2018. DOI: 10.1145/3190508.3190538.

[16] M. Vukoli'c, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International Workshop on Open Problems in Network Security*, ser. Lecture Notes in Computer Science, Springer, 2015. DOI: 10.1007/978-3-319-39028-4_9.

[17] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *Proceedings of the 26th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, IEEE, 2018. DOI: 10.1109/MASCOTS.2018.00034.

[18] Hyperledger Foundation, *Membership service providers (msp)*, Hyperledger Fabric Documentation, 2021. Accessed: Nov. 19, 2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/msp.html.

[19] Hyperledger Foundation, *Chaincode for developers*, Hyperledger Fabric Documentation, 2021. Accessed: Nov. 19, 2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/chaincode.html.

[20] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, Philadelphia, PA, USA: USENIX Association, 2014.

[21] J. a. Sousa, A. Bessani, and M. Vukoli'c, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, IEEE, 2018. DOI: 10.1109/DSN.2018.00018.

[22] Hyperledger Foundation, *Securing communication with transport layer security (tls)*, Hyperledger Fabric Documentation, 2021. Accessed: Nov. 19, 2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/enable_tls.html.

[23] Hyperledger Foundation, *Policies*, Hyperledger Fabric Documentation, 2021. Accessed: Nov. 19, 2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/policies.html.

[24] A. Vaswani et al., *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[25] OpenAI et al., *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2303.08774.

[26] M. B. II and D. M. Katz, *Gpt takes the bar exam*, 2022. arXiv: 2212.14402 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2212.14402.