applied sciences

MDPI

Article

A Modular Solution Concept for Self-Configurable Electronic Lab Notebooks: Systematic Theoretical Demonstration and Validation Across Diverse Digital Platforms

Kim Feldhoff [1,*], Martin Zinner [1], Hajo Wiemer [1] and Steffen Ihlenfeldt [1,2]

[1] Institute of Mechatronic Engineering (IMD), Faculty of Mechanical Science and Engineering, TUD Dresden University of Technology, 01062 Dresden, Germany
[2] Fraunhofer Institute for Machine Tools and Forming Technology (IWU), 01187 Dresden, Germany
* Correspondence: kim.feldhoff@tu-dresden.de

Featured Application

The proposed modular solution concept can be directly applied by research institutions seeking to implement cost-effective, customizable, and compliant ELNs on widely available digital platforms such as Microsoft SharePoint or Google Workspace, without the need for proprietary software or advanced programming expertise.

Abstract

The increasing complexity and digitization of scientific research require Electronic Laboratory Notebooks (ELNs) that are adaptable, sustainable, and compliant across heterogeneous laboratory environments. In response to the limitations of proprietary, inflexible, and cost-intensive ELN solutions, this study systematically derives comprehensive requirements and proposes a modular solution concept for self-configurable ELNs that is explicitly platform-agnostic and broadly accessible. The methodological approach combines a structured requirements analysis with a modular architectural design, followed by theoretical validation through stepwise implementation walkthroughs on Microsoft SharePoint and Google Workspace. These walkthroughs demonstrate the feasibility of deploying self-configurable ELN modules using widely available low-code/no-code tools and native platform extensibility mechanisms. Based on a rigorous literature-driven analysis, key requirements, including modularity, usability, regulatory compliance, interoperability, scalability, auditability, and cost efficiency, are explicitly mapped to concrete architectural features within the proposed framework. The results show that essential ELN functionalities can, in principle, be realized across diverse digital platforms, enabling researchers and local administrators to independently assemble, configure, and adapt ELNs to their specific operational and regulatory contexts. Beyond technical feasibility, the proposed approach fundamentally democratizes ELN deployment and substantially mitigates vendor lock-in by leveraging existing digital infrastructures. Identified limitations, particularly with respect to advanced workflow orchestration and real-time data integration, delineate clear directions for future development. Overall, this work provides a systematic theoretical validation of a modular, self-configurable ELN concept, establishing it as a robust, scalable, and future-ready foundation for digital laboratory infrastructures.

Keywords: electronic lab notebook; modular architecture; self-configuration; digital platform; workflow automation; regulatory compliance; platform-agnostic solution; low-code; research data management; interoperability

Academic Editor: Alexander Barkalov

Received: 9 November 2025
Revised: 15 December 2025
Accepted: 29 December 2025
Published: 1 January 2026

 https://doi.org/10.3390/app16010462

1. Introduction

The relentless digital transformation of scientific research has exposed fundamental inadequacies in traditional data management and documentation practices. As Rubacha et al. (2011) and Bird et al. (2013) assert, the explosion of research data, the complexity of experimental workflows, and the imperative for transparency and reproducibility have rendered paper-based laboratory notebooks obsolete for modern science [1,2]. Electronic Lab Notebooks (ELNs) have become indispensable, offering structured, searchable, and collaborative environments that support rigorous data stewardship and compliance with evolving regulatory frameworks; see Kwok (2018); Dirnagl and Przesdzing (2016); Wilkinson et al. (2016); and Mons et al. (2017) [3–6].

Yet, as highlighted by Dirnagl and Przesdzing (2016), Machina and Wild (2013), and Kanza et al. (2017), most commercial ELN solutions remain fundamentally flawed: they are costly, rigid, and inflexible, with limited scope for user-driven customization and adaptation [4,7,8]. These shortcomings force researchers to conform to proprietary workflows, stifling innovation, reducing engagement, and sometimes prompting a regression to ad hoc or paper-based documentation; see Bird et al. (2013) and Rubacha et al. (2011) [1,2]. The risks are further compounded by vendor lock-in, poor data portability, and uncertain long-term accessibility. These are challenges that directly threaten the openness, sustainability, and reproducibility of science; see Leonelli (2016) and Wilkinson et al. (2016) [5,9].

In response, recent advances in modular, platform-agnostic, and self-configurable frameworks offer a substantive departure from the status quo. Higgins et al. (2022) and Bird et al. (2013) demonstrate that leveraging widely adopted digital ecosystems, such as Microsoft SharePoint, Google Workspace, or open-source platforms, enables the creation of tailored ELNs that can be rapidly reconfigured to meet both research and compliance requirements [2,10]. This approach capitalizes on the scalability, security, and reliability of established cloud infrastructures, while empowering researchers to define workflows, templates, and data models without vendor lock-in; see Dirnagl and Przesdzing (2016) and Mons et al. (2017) [4,6]. However, as Kwok (2018) and Kanza et al. (2017) emphasize, the field still lacks robust, systematic frameworks and best-practice guidelines for transforming generic platforms into compliant, user-friendly ELNs [3,8]. Addressing this gap is not merely a technical challenge, but a strategic imperative for laboratories seeking to adapt rapidly, reduce costs, and uphold the highest standards of data stewardship, transparency, and open science (Wilkinson et al. (2016); Mons et al. (2017); Leonelli (2016) [5,6,9]).

1.1. Motivation and Research Gap

The urgency for universal, adaptable ELN frameworks is underscored by the escalating complexity, data volume, and interdisciplinarity of contemporary research; see Wilkinson et al. (2016); Mons et al. (2017); Leonelli (2016); and Kanza et al. (2017) [5,6,8,9]. Laboratories face not only rapidly evolving experimental paradigms and collaboration models, but also increasingly stringent regulatory landscapes, such as GLP, FDA 21 CFR Part 11, and GDPR; see Kanza et al. (2017); Bird et al. (2013); Dirnagl and Przesdzing (2016); and Higgins et al. (2022) [2,4,8,10]. These demands require digital research environments that are robust and secure, yet agile enough to be dynamically reconfigured; see Rubacha et al. (2011) and Machina and Wild (2013) [1,7].

The majority of existing ELN products, however, remain inadequate: their rigid architectures, limited customization, and costly extensions jeopardize both compliance and data integrity; see Kwok (2018); Dirnagl and Przesdzing (2016); and Machina and Wild (2013) [3,4,7]. Ad hoc modifications, often necessary to accommodate unique workflows, can undermine security and compliance while inflating total cost of ownership; see Bird et al. (2013); Rubacha et al. (2011); and Kanza et al. (2017) [1,2,8]. Moreover, commercial ELNs perpetuate vendor lock-in, restrict data portability, and complicate long-term preservation, barriers fundamentally at odds with open science and reproducibility' see Wilkinson et al. (2016); Mons et al. (2017); and Leonelli (2016) [5,6,9].

There is thus an acute and unmet demand for systematic, evidence-based methodologies that can transform generic, widely available digital platforms into robust, user-centric ELN systems; see Higgins et al. (2022); Kanza et al. (2017); and Dirnagl and Przesdzing (2016) [4,8,10]. In addition to these conceptual and regulatory challenges, universal self-configurable ELN frameworks promise significant practical benefits for both users and administrators, including reduced total cost of ownership, faster deployment cycles, enhanced user autonomy, and more intuitive adaptation of workflows. Introducing these benefits early provides a clearer motivation for the system-level goals addressed by this study. Such frameworks must not only satisfy technical and regulatory requirements, but also address sociotechnical factors, user acceptance, sustainability, and scalability, that determine long-term success; see Leonelli (2016) and Machina and Wild (2013) [7,9].

1.2. Research Questions and Hypotheses

Given the profound challenges and critical gaps outlined above, this study formulates an ambitious and comprehensive research agenda to fundamentally advance the field of electronic laboratory data management. The following research questions are designed to rigorously interrogate the feasibility, requirements, and impact of a universal, self-configurable ELN framework:

- **Research Question RQ1:** Is it possible to design and implement a universal, self-configurable framework that can systematically transform generic digital platforms into fully functional, adaptable, compliant, and user-centric ELNs, robust enough to address the heterogeneous needs of both established and emerging research domains?
- **Research Question RQ2:** What are the indispensable conceptual, technical, and regulatory requirements for such a framework to guarantee not only usability, compliance, and interoperability, but also long-term sustainability, auditability, and scalability in complex scientific environments?
- **Research Question RQ3:** To what extent does the performance, scalability, and user acceptance of ELNs built on generic digital platforms, when guided by a universal framework, surpass or fall short of conventional, commercial ELN solutions in authentic, high-stakes laboratory settings?

To address these questions, the following sharply formulated hypotheses are advanced, each designed to withstand rigorous empirical and comparative scrutiny:

Hypothesis H1. *A universal, self-configurable framework can be instantiated across a range of digital platforms, both proprietary and open-source, to produce ELNs that meet or exceed the highest standards of usability, regulatory compliance, and scalability as defined by leading commercial solutions, cf. Dirnagl and Przesdzing (2016); Higgins et al. (2022); Bird et al. (2013) [2,4,10].*

Hypothesis H2. *Platform-based ELNs, architected via a systematic and modular framework, will exhibit superior adaptability, user satisfaction, and cost-efficiency in dynamic, interdisciplinary, and rapidly evolving research environments, compared to monolithic proprietary ELNs; see Kanza et al. (2017); Machina and Wild (2013); and Mons et al. (2017) [6–8].*

Hypothesis H3. *The adoption of such a universal framework will result in a significant reduction of operational and transition costs, effectively eliminate vendor lock-in, and substantially enhance data portability, integrity, and security, thus providing a robust foundation for open science and long-term data stewardship; see Wilkinson et al. (2016) and Leonelli (2016) [5,9].*

1.3. Objectives

The objectives of this work are accordingly ambitious:

1. To define, with conceptual and technical rigor, the foundational architecture and regulatory scaffolding of a universal, self-configurable ELN framework.
2. To theoretically demonstrate, by means of detailed analysis and case-based reasoning, that a real-world implementation and flexible customization of this framework would be feasible across a spectrum of digital platforms, including both proprietary and open-source environments.

We explicitly acknowledge that the proposed framework, while theoretically validated, requires systematic empirical evaluation in authentic laboratory environments. However, the empirical evaluation of the framework in live laboratory environments is outside the scope of this theoretical demonstration and is identified as a primary objective for future research. Addressing RQ3 through real-world deployments and longitudinal user studies constitutes the most critical next step for future work.

1.4. Key Terminology

In this paper, the following terms are used as defined below:

- 'Self-configuration' refers to configuration performed by domain experts, researchers, or local administrators using low-code or no-code mechanisms provided by the underlying platform. It does not imply unrestricted end-user scripting, uncontrolled system modification, or the need for advanced software engineering expertise.
- 'Self-configurable ELN' refers to an electronic lab notebook framework that allows end-users or administrators to customize workflows and features without extensive programming.
- 'Modular architecture' refers to the system's ability to be extended or adapted via interchangeable components.
- 'Universal framework' denotes a solution that is platform-agnostic and adaptable to various laboratory environments.

1.5. Contributions

This study makes four primary contributions to the design and validation of self-configurable ELNs, each addressing a critical gap in current ELN research and practice:

1. **A formal modular architecture for self-configurable ELNs.** We propose a rigorously defined architectural model that decomposes ELN functionality into reusable, platform-independent modules. This architecture explicitly separates configuration logic, automation rules, platform interfaces, and user-facing components, enabling high flexibility, extensibility, and compliance-aware design.
2. **A systematic mapping of data, process, and compliance requirements to architectural modules.** The study identifies core ELN requirements from literature, standards, and practical laboratory workflows, and demonstrates how each requirement is realized through specific architectural building blocks. This mapping provides a transparent rationale that connects theoretical needs to implementation-level constructs.
3. **A cross-platform theoretical validation covering two major ecosystem families.** To verify generalizability, the proposed architecture is evaluated against distinct digital ecosystems, Microsoft SharePoint and Google Workspace, using a structured scoring and justification framework. The validation shows that the architecture remains functionally complete and implementable despite substantial differences in platform constraints, APIs, permission structures, and low-code capabilities.

4. **A universal framework for platform-agnostic ELN deployment.** Building on the architecture and validation, we introduce a unifying framework that enables ELN solutions to be generated, adapted, and deployed across diverse digital environments with minimal redesign effort. The framework defines abstractions, configuration pathways, and integration mechanisms that allow laboratories to adopt ELNs tailored to their platforms while preserving interoperability, auditability, and long-term maintainability.

Together, these contributions establish a foundational methodology for designing, evaluating, and deploying ELNs that are not bound to any specific technical environment, providing a scalable path toward sustainable and FAIR-aligned digital laboratory infrastructures.

1.6. Outline

The remainder of this paper is structured as follows: Section 1.7 provides a focused review of the current literature on ELNs, emphasizing their evolution, adoption challenges, and the state-of-the-art in modular and platform-agnostic solutions, while also synthesizing regulatory, technical, and sociotechnical considerations. Section 2 outlines the solution approach, including architectural design, modular structure, configuration processes, extensibility mechanisms, and key benefits. In more detail, Section 2.1 systematically analyzes functional, regulatory, interoperability, and usability requirements. Section 2.2 highlights the main challenges in developing a robust, scalable, and widely applicable self-configuration framework, explicitly linking them to the identified requirements. Section 2.4 introduces an abstract, platform-agnostic solution concept, detailing the core architecture, configuration engine, adaptive user interface, workflow templates, compliance modules, and integration layers, and documents the systematic mapping of requirements to architectural features. Section 3 demonstrates the practical applicability and versatility of the proposed solution concept by validating it through theoretical deployments on multiple target platforms. This includes detailed implementation walkthroughs on Microsoft SharePoint and Google Workspace, illustrating the incremental adaptation and integration of core components, such as the configuration engine, module registry, and user interface, within the native features of each platform. Section 4 discusses the broader implications for digital research infrastructure, open science, and data stewardship, identifies study limitations. Finally, Section 5 summarizes the main findings and contributions of this work and proposes future research directions.

Readers primarily interested in architectural design may focus on Section 2, while those interested in platform applicability can directly proceed to Section 3.

1.7. State of the Art and Related Work

The ongoing digital transformation of scientific research has profoundly reshaped the landscape of laboratory data management, documentation, and collaboration. As Hey et al. (2009) argue in their seminal work on data-intensive science, the increasing scale and complexity of research data necessitate new approaches to information infrastructure and digital record-keeping [11]. Leonelli (2016) further emphasizes the central role of data-centric infrastructures in contemporary biology, highlighting the importance of robust data management systems for scientific progress [9]. In line with these developments, Wilkinson et al. (2016) introduced the FAIR data principles, which have become a cornerstone for scientific data stewardship and have directly influenced the design and adoption of digital research tools, including ELNs [5].

ELNs have emerged as pivotal tools within this context, offering structured, searchable, and shareable environments for recording experimental protocols, research data, and metadata. Rubacha et al. (2011) provide a comprehensive review of ELN solutions, documenting their evolution from basic digital notebooks to complex, feature-rich platforms that support laboratory workflows and regulatory requirements [1]. Bird et al. (2013) discuss the impact of ELNs on scientific research, noting their role in improving data integrity, collaboration, and reproducibility [2]. Kwok (2018) offers practical guidance for selecting ELNs, emphasizing the need for solutions that balance usability, compliance, and flexibility [3]. Dirnagl and Przesdzing (2016) underscore the importance of ELNs in supporting reproducibility, transparency, and compliance with evolving research policies and funding mandates [4]. The broader context of reproducibility in science is further explored by Begley and Ioannidis (2015) and Nosek et al. (2015), who highlight the critical role of digital tools in promoting open research practices [12,13]. Feldhoff et al. (2025) demonstrated that modern collaboration platforms such as Microsoft SharePoint, when specifically adapted, can serve as effective and competitive alternatives to established ELN for research data management in collaborative engineering projects [14].

1.7.1. Commercial ELN Solutions

The initial proliferation of ELNs was driven by commercial, proprietary solutions such as LabArchives, LabWare, and Benchling, as reviewed by Rubacha et al. (2011), Bird et al. (2013), and Kwok (2018) [1–3]. These platforms offer features including data entry, versioning, audit trails, and compliance with standards such as GLP and ISO/IEC 17025, as detailed in their respective reviews [1–4]. However, Dirnagl and Przesdzing (2016), Higgins et al. (2022), and Bird et al. (2013) note that these commercial solutions are often characterized by closed architectures, limited customizability, and high licensing and maintenance costs [2,4,10]. Users have reported challenges such as rigid, pre-defined workflows, poor interoperability with laboratory instruments and institutional IT systems, and protracted onboarding, as highlighted in studies by Dirnagl and Przesdzing (2016), Higgins et al. (2022) [4,10]. These drawbacks are particularly problematic in interdisciplinary and rapidly evolving research settings, where adaptability and flexibility are paramount, as argued by Bird et al. (2013), and Begley and Ioannidis (2015) [2,12].

1.7.2. Open Source and Platform-Based ELNs

In response to the limitations of commercial ELN solutions, the scientific community has increasingly embraced open-source alternatives and the adaptation of generic digital platforms. Among the most prominent open-source ELNs are Chemotion, as detailed by Tremouilhac et al. (2017), which exemplifies transparency, extensibility, and community-driven development [15]. Similarly, eLabFTW, described by Carpi et al. (2017), provides a flexible, open-source platform for research data management that is widely adopted in the academic sector [16]. LabTrove, introduced by Milsted et al. (2013) and further discussed by Bird et al. (2013), focuses on lightweight, web-based collaboration and has influenced the development of subsequent ELN platforms [2,17]. The openBIS system, presented by Goecks et al. (2011), demonstrates the integration of ELNs with laboratory information management systems (LIMS), support for metadata standards, and facilitation of data sharing and reproducibility [18]. These open-source platforms are praised for their extensibility and alignment with open science principles, but their deployment and customization often require considerable technical expertise, which can be a barrier for smaller or resource-limited laboratories, as noted by Higgins et al. (2022) [10].

In parallel, there has been a trend toward repurposing widely adopted digital collaboration platforms, such as Microsoft SharePoint, Google Workspace, Nextcloud, and even GitHub, for ELN purposes. Higgins et al. (2022) discuss how these platforms, already deeply embedded in institutional IT infrastructures, offer robust document management, collaboration, and automation features [10]. While their cost-effectiveness and familiarity are attractive, their transformation into compliant, user-friendly ELNs is often performed in an ad hoc manner, lacking systematic frameworks for configuration, compliance, and scalability, as highlighted in a study by Tremouilhac et al. (2017) [15]. Commonly reported challenges include inconsistent usability, security vulnerabilities, and difficulties in scaling or transferring solutions across diverse laboratory settings, as emphasized by Higgins et al. (2022) and Tremouilhac et al. (2017) [10,15].

1.7.3. Modular and Self-Configurable Digital Systems

The concept of modular software engineering, as first articulated by Parnas (1972), has long been recognized for its ability to enhance the adaptability, maintainability, and scalability of complex digital infrastructures [19]. Parnas (1972) emphasized that modular architectures enable the independent development, deployment, and scaling of system components, which is crucial for fostering the sustainability and evolution of software systems over time [19]. Building on these foundational ideas, Fowler (2002) provided a comprehensive treatment of enterprise application architecture patterns, further demonstrating how modularity can support robust and extensible system design [20].

More recently, Kulesza et al. (2015) explored the development of self-configurable systems, showing that such frameworks empower users to tailor workflows, interfaces, and integrations to their specific needs with minimal technical intervention [21]. Tremouilhac et al. (2017) further illustrate the practical implementation of modular and self-configurable concepts within the Chemotion ELN, demonstrating the benefits of open-source, community-driven development for enhancing flexibility and user empowerment in laboratory environments [15].

Opatz et al. (2025) advanced the state of the art by developing a modular plugin for the open-source research data infrastructure Kadi4Mat, specifically tailored for material sciences [22]. Their work demonstrates how automated research data release—incorporating quality assurance, confidentiality, and reusability—can be achieved through a customizable and extensible approach. This aligns with the increasing demand for flexible, interoperable, and user-configurable ELN solutions, as highlighted in recent literature, and provides a practical example of how modularity and open-source principles can address the diverse requirements of collaborative, interdisciplinary research environments. Such developments reinforce the feasibility and value of the self-configurable, platform-agnostic ELN concepts proposed in this study.

1.7.4. Low-Code Platforms and Cloud Compliance

Beyond traditional ELN evaluation studies, recent research has drawn significant attention to the role of low-code and no-code platforms in enabling configurable scientific workflows, highlighting their increasing relevance for laboratory digitization [23,24]. Parallel developments in cloud-compliance frameworks emphasize the growing importance of security, auditability, and regulatory-by-design principles in laboratory IT infrastructures [25]. These contemporary studies collectively illustrate a rapidly evolving technological environment that further motivates the need for a universal, modular, and platform-independent ELN architecture.

1.7.5. Modern ELN Developments and Workflow Alignment

Recent studies have further examined ELN usability, adoption barriers, and user-centered implementation strategies, underscoring the need for architectures that remain adaptable and platform-agnostic [26,27]. Additional work has explored how ELN workflows can better align with the FAIR principles, particularly in terms of metadata interoperability and reproducibility [28]. These findings reinforce the necessity of designing ELN frameworks that support flexible workflow integration while avoiding vendor lock-in, a direction that is further strengthened by emerging LLM-assisted ELN systems [29].

Despite these advances, the systematic application of modular and self-configurable principles to ELN development remains limited. As noted by Higgins et al. (2022), most existing ELN platforms are either highly specialized or lack the generalizability and platform-agnosticism required to effectively balance compliance, usability, and extensibility across diverse research contexts [10].

A detailed quantitative comparison of representative self-configurable ELN platforms (including openBIS, Chemotion ELN, and eLabFTW) and established traditional commercial ELN systems (such as LabArchives, LabWare, and PerkinElmer Signals) is presented in Table 1. The evaluation framework comprises standardized categories including configurability, cost and deployment, interoperability, operations and support, compliance and security, vendor and sustainability, and data analytics, each with specific feature definitions.

To ensure transparency and reproducibility of the 0–10 scoring presented in Table 1, a structured multi-source evaluation methodology was applied. Grades (0–10) were assigned by the authors based on a synthesis of peer-reviewed literature, published user surveys, vendor documentation, and publicly available product reviews [1,8,10,30–33]. As ELN platforms evolve continuously, the scores should be interpreted as indicative and comparative rather than absolute. Each score reflects consensus across these sources. The following ELN systems were included: openBIS, Chemotion ELN, eLabFTW (self-configurable group). Where possible, values were cross-checked against multiple independent sources and, for open-source ELNs, verified through hands-on system trials. While this approach aims to ensure objectivity and reproducibility, the authors acknowledge that some scoring elements may remain subjective due to the heterogeneity and evolving nature of ELN platforms.

The color-coded cells in the table visually highlight strengths and weaknesses, revealing that self-configurable ELNs (e.g., openBIS, Chemotion ELN, eLabFTW) generally outperform traditional commercial systems in workflow adaptability, integration, and vendor independence, while commercial ELNs sometimes offer advantages in support ecosystem and regulatory alignment. This evidence-based overview is intended to support decision-making in laboratory digitization. It is worth to note that features such as Data Security and Support Ecosystem received higher scores for self-configurable systems due to their adaptable security architectures and rapidly expanding open-source support communities.

1.7.6. Gaps and Research Motivation

The literature reveals a clear need for ELN frameworks that combine the scalability, security, and collaborative features of established digital platforms with the flexibility and user-centricity of modular, self-configurable systems. There is a lack of generalizable, systematic methodologies for transforming generic digital platforms into compliant, sustainable, and user-friendly ELNs. This gap motivates the present study, which proposes and evaluates a universal, self-configurable framework capable of adapting a variety of digital platforms as ELNs. By integrating best practices from modular software engineering and leveraging platform-native features, this work aims to contribute a rigorous theoretical foundation for future developments in digital laboratory management.

https://doi.org/10.3390/app16010462

Table 1. Standardized quantitative comparison of self-configurable and traditional commercial ELN systems using a normalized 0–10 scoring scale (higher values indicate stronger support). Scores are derived from an analytical assessment of architectural capabilities and operational characteristics rather than empirical benchmarking. Values are shown numerically and additionally visualized using aligned proportional bar indicators.

Category	Feature	Description	Self-Configurable ELN	Traditional ELN
Configurability	Workflow Adaptability	Degree to which users can modify workflows, data models, and interfaces	10	3
Configurability	Customization Complexity	Technical effort required to implement substantial functional changes	8	2
Configurability	User Experience	Perceived usability resulting from adaptability and feedback mechanisms	9	5
Cost and Deployment	Total Cost of Ownership	Overall cost including licensing, infrastructure, operation, and maintenance	9	3
Cost and Deployment	Implementation Speed	Effort and time required for initial deployment and configuration	9	4
Cost and Deployment	Scalability	Ability to accommodate growth in users, data volume, and workflows	9	5
Interoperability	Instrument Integration	Breadth and effort of integrating laboratory instruments and devices	9	4
Interoperability	System Integration	Capability to connect with external digital systems via APIs and standards	9	4
Interoperability	Data Portability	Ease of exporting, importing, and migrating data across platforms	10	3
Operations and Support	Maintenance Efficiency	Degree of automation and effort required for updates and maintenance	8	4
Operations and Support	Support Ecosystem	Availability of documentation, community resources, and vendor support	8	6
Compliance and Security	Data Security	Strength and configurability of access control and protection mechanisms	8	6
Compliance and Security	Regulatory Alignment	Ability to support compliance with standards such as GLP or ISO 17025	9	6
Vendor and Sustainability	Vendor Lock-in Risk	Degree of dependence on proprietary platforms or vendors	10	2
Vendor and Sustainability	Long-Term Sustainability	Expected longevity, extensibility, and maintainability of the solution	9	5
Data Analytics	Reporting and Analytics	Extensiveness and configurability of reporting and analytical capabilities	8	5

2. Materials and Methods

This section outlines the systematic approach employed to identify the comprehensive requirements for self-configurable ELNs and to derive a modular solution concept that addresses these needs. The methodology is grounded in established principles from software engineering, user-centered design, and scientific workflow management, ensuring both theoretical rigor and practical relevance across diverse research contexts [5,8,19].

The proposed approach differs fundamentally from existing ELN architectures by enabling self-configuration through platform-native low-code tools rather than proprietary extensions or fixed workflows. This innovation allows practitioners to assemble ELN modules dynamically and independently across heterogeneous digital ecosystems.

This study follows a Design Science Research (DSR) paradigm, in which a relevant and persistent problem in digital laboratory infrastructures is addressed through the systematic design of an artifact, in this case, a modular, self-configurable ELN framework. The research process comprises (i) problem and requirement identification grounded in literature and regulatory standards, (ii) construction of a conceptual and architectural solution artifact, and (iii) analytical feasibility validation through structured platform-based instantiation and walkthroughs. In line with established DSR methodology, empirical performance benchmarking and longitudinal user evaluation are explicitly deferred to future work.

2.1. Requirements

The development of a universal, self-configurable framework for ELNs necessitates a requirements analysis that is both systematic and analytically grounded. Unlike narrowly scoped software systems, ELNs are embedded within complex sociotechnical environments in which scientific workflows, digital infrastructures, and regulatory obligations intersect. Consequently, requirements must be formulated not merely as functional features, but as design constraints that shape the architecture, governance, and long-term viability of the proposed framework.

The requirements identified in this study were derived through a structured synthesis of the ELN literature, standards relevant to research data management and laboratory compliance, and established principles of modular and configurable software systems. To maintain conceptual clarity while avoiding excessive fragmentation, requirements are grouped into four overarching categories: architectural, operational, regulatory, and user-centric. Within each category, core requirements are emphasized explicitly, while secondary or derivative requirements are integrated into the explanatory text or consolidated at a higher level of abstraction.

2.1.1. Architectural Requirements

From an architectural standpoint, the primary requirement is modularity combined with configurability. A self-configurable ELN framework must be decomposable into well-defined functional units that encapsulate specific responsibilities, such as data capture, workflow control, compliance enforcement, or system integration. This modular structure enables independent evolution of components, targeted adaptation to local laboratory needs, and long-term maintainability. Configurability extends this principle by allowing these modules to be assembled, parameterized, and replaced without invasive system redesign.

Closely related is the requirement of extensibility, which ensures that the framework can accommodate new instruments, analytical tools, data sources, or regulatory mechanisms as scientific practices evolve. Extensibility must be supported through stable interfaces, standardized data models, and clearly defined integration contracts, thereby preventing architectural erosion over time.

A further central architectural requirement is scalability. The framework must sustain acceptable performance, reliability, and manageability as the number of users, workflows, and data volumes increases across projects or institutions. Scalability therefore encompasses not only computational aspects, but also organizational scalability, such as the ability to manage multiple laboratories or research groups within a shared infrastructure.

Finally, interoperability and standardization are essential architectural requirements. Support for open standards and non-proprietary data formats is necessary to ensure long-term data portability, integration with external systems, and avoidance of vendor lock-in. These properties are foundational for reproducibility, sustainability, and alignment with open science principles.

2.1.2. Operational Requirements

Operational requirements address the dependable and efficient functioning of the framework in daily laboratory practice. Security represents a non-negotiable core requirement, encompassing authentication, authorization, encryption, and protection against unauthorized data access or manipulation. Given the sensitivity of scientific data and the legal implications of security breaches, security mechanisms must be integral to the system architecture rather than optional extensions.

Reliability and maintainability constitute a second core operational requirement. The framework must provide mechanisms for error detection, fault tolerance, backup, and recovery to ensure continuity of research activities. At the same time, maintainability requires that configuration artifacts, workflows, and integrations remain understandable and manageable over extended system lifetimes, supported by documentation and systematic update processes.

Automation emerges as an enabling operational requirement rather than an isolated feature. By automating routine tasks such as workflow progression, data validation, and reporting, the framework can reduce manual effort and error rates while increasing consistency. Importantly, automation must remain transparent and configurable, allowing laboratories to retain oversight and adapt processes without compromising control.

2.1.3. Regulatory Requirements

Regulatory requirements reflect the fact that many laboratories operate under stringent legal, ethical, and institutional constraints. The framework must support compliance with relevant standards and regulations, such as GLP [34], ISO/IEC 17025 [35], FDA 21 CFR Part 11 [36], and data protection legislation. Rather than embedding regulation-specific logic in isolated components, compliance must be treated as a cross-cutting concern that permeates workflows, data handling, and access control mechanisms. A closely associated core requirement is auditability and traceability. All relevant system actions, ranging from data creation and modification to workflow changes and approvals, must be logged in a manner that is tamper-evident, comprehensible, and suitable for internal and external audits. These properties are essential not only for regulatory compliance, but also for scientific transparency and reproducibility.

2.1.4. User-Centric Requirements

User-centric requirements address the human factors that determine adoption, effective use, and long-term sustainability. The central requirement in this category is user empowerment through usability. A self-configurable ELN framework must provide intuitive configuration mechanisms, such as visual workflow designers and reusable templates, that enable non-technical users to adapt the system to their needs without extensive training or programming expertise.

Additional user-centric considerations, including accessibility, internationalization, training, and knowledge transfer, are treated as supporting requirements that reinforce empowerment and inclusivity. For conciseness, detailed treatment of these supporting aspects is consolidated and provided in Appendix B.

https://doi.org/10.3390/app16010462

2.1.5. Summary

Taken together, the requirements defined above establish a coherent evaluative framework for self-configurable ELN systems. They also delineate the design space within which practical implementation must operate. The following subsection therefore examines the principal challenges that arise when attempting to satisfy these requirements simultaneously. Table 2 gives a summary of key system requirements (R-) with stable identifiers and concise descriptions. Each requirement represents a thematically grouped design constraint derived from architectural, operational, regulatory, or user-centric considerations.

Table 2. Summary of key system requirements (R-) with stable identifiers and concise descriptions. Each requirement represents a thematically grouped design constraint derived from architectural, operational, regulatory, or user-centric considerations. The identifiers are used consistently throughout the manuscript and in subsequent mapping tables to support traceability and analytical rigor.

ID	Requirement Identifier	Description
1	R-ARCH-Modularity&Flexibility	Decomposition of the ELN framework into well-defined, configurable modules that can be independently developed, replaced, and adapted to diverse laboratory contexts.
2	R-ARCH-Extensibility	Ability to integrate new instruments, data sources, analytical tools, and regulatory mechanisms without fundamental architectural redesign.
3	R-ARCH-Scalability	Support for increasing numbers of users, workflows, and data volumes while maintaining performance, reliability, and manageability across institutional deployments.
4	R-ARCH-Interoperability&Standardizatic	Use of open standards and non-proprietary data formats to ensure compatibility with external systems, long-term data portability, and avoidance of vendor lock-in.
5	R-OPER-Security	Built-in mechanisms for authentication, authorization, encryption, and protection against unauthorized access or data manipulation.
6	R-OPER-Reliability	Ensuring continuous system operation through fault tolerance, error handling, backup, and recovery mechanisms.
7	R-OPER-Maintainability	Long-term manageability of the system through clear configuration structures, documentation, and controlled update processes.
8	R-OPER-Automation	Automation of routine laboratory workflows, data validation, and reporting to improve efficiency and reduce human error.
9	R-REG-Compliance	Support for compliance with relevant legal, ethical, and quality standards such as GLP, ISO/IEC 17025, FDA 21 CFR Part 11, and data protection regulations.
10	R-REG-Auditability&Transparency	Comprehensive, tamper-evident logging of system actions to enable traceability, reproducibility, and internal or external audits.
11	R-USER-Empowerment	Provision of intuitive configuration mechanisms that allow non-technical users to adapt workflows and system behavior to their needs.
12	R-USER-Accessibility	Ensuring usability for diverse user groups, including consideration of accessibility requirements and heterogeneous technical expertise.
13	R-USER-Internationalization	Support for multilingual and region-specific representations to enable deployment across international research environments.
14	R-USER-TrainingDocumentation	Availability of structured training materials and embedded documentation to facilitate onboarding and consistent system use.
15	R-USER-KnowledgeTransfer	Mechanisms that support the preservation and transfer of procedural and domain knowledge within and between research groups.

2.2. Challenges

Translating the identified requirements into a practical, self-configurable ELN framework entails a set of fundamental challenges that define the design constraints of the proposed solution concept. These challenges are not isolated implementation issues, but structural tensions inherent to flexible, long-lived, and regulation-aware information systems. In particular, they reflect trade-offs between flexibility and complexity, standardization and customization, as well as innovation and regulatory stability. The following analysis focuses on a limited number of principal challenge categories that systematically inform the architectural decisions introduced in Section 2.4.

2.2.1. Architectural Challenges

At the architectural level, the foremost challenge lies in balancing modularity and extensibility with structural complexity. While modular architectures are essential for achieving configurability and long-term adaptability, excessive decomposition can obscure system behavior, complicate dependency management, and increase cognitive overhead for both developers and operators. The challenge is therefore not to maximize modularity per se, but to identify an architectural granularity that preserves clarity, maintainability, and evolvability without introducing disproportionate structural overhead.

A closely related architectural challenge concerns integration within heterogeneous environments. Research laboratories typically rely on diverse ecosystems of legacy software, laboratory instruments, institutional IT services, and external data repositories. Ensuring seamless interoperability across such heterogeneous systems requires robust abstraction layers, consistent interface definitions, and adherence to open standards. This challenge is further intensified by the need to preserve data integrity and functional compatibility across evolving software versions, platforms, and deployment contexts. These architectural constraints directly motivate the modular, interface-driven design principles adopted in the proposed solution concept.

2.2.2. Operational Challenges

From an operational perspective, scalability and reliability in dynamic use contexts constitute a central challenge. As ELN deployments expand in terms of users, workflows, and data volumes, the framework must maintain consistent performance and availability. This challenge extends beyond computational scalability to include operational concerns such as monitoring, maintenance, and fault management across distributed and potentially heterogeneous deployments.

Another major operational challenge arises from the migration of legacy data and workflows. Many laboratories possess extensive historical records that must remain scientifically valid and legally compliant after transition to a new ELN framework. Migrating such data without disrupting ongoing research activities, while preserving provenance, traceability, and interpretability, requires careful planning and robust migration mechanisms. Failure to address this challenge adequately can undermine both user trust and regulatory compliance. Accordingly, the solution architecture incorporates mechanisms for scalable operation, continuous maintenance, and controlled migration of legacy assets.

2.2.3. Regulatory and Security Challenges

Regulatory and security-related challenges reflect the evolving nature of legal frameworks, institutional policies, and cybersecurity threats. ELN systems operating in regulated research environments must support compliance as a continuous process rather than a one-time certification. This requirement becomes particularly demanding in self-configurable systems, where dynamically adapted workflows must nevertheless satisfy stringent requirements for auditability, traceability, and access control. Ensuring that flexibility does not compromise regulatory robustness represents a persistent and non-trivial challenge. These constraints necessitate architectural mechanisms that embed compliance, auditability, and security as cross-cutting concerns rather than external add-ons.

2.2.4. User-Centric Challenges

User-centric challenges emerge from the human factors that ultimately determine adoption, effective use, and long-term sustainability. A central tension exists between usability and configurability. While configurability empowers users to adapt workflows and system behavior to their specific needs, it can also increase complexity and overwhelm users with limited technical expertise. Designing interfaces and configuration mechanisms that remain intuitive while supporting advanced customization is therefore a key challenge.

Closely related is the tension between customization and standardization. Extensive local customization can undermine coherence, interoperability, and the sharing of best practices across laboratories or institutions. At the same time, overly rigid standardization can limit the system's ability to accommodate legitimate local requirements. Effective governance mechanisms are thus required to balance local autonomy with global consistency and long-term sustainability. The solution concept therefore emphasizes user-centered configuration mechanisms that balance flexibility with consistency and governance.

2.2.5. Summary

In summary, the challenges associated with self-configurable ELN frameworks are inherently multidimensional, encompassing architectural, operational, regulatory, and human factors. Rather than representing obstacles to be eliminated, these challenges define the constraints that shape a robust and sustainable solution. The solution concept presented in the following section explicitly addresses these challenges by aligning architectural mechanisms with the requirements and trade-offs identified in this analysis. Table 3 gives a summary of the principal challenge categories (C-) associated with the realization of a self-configurable ELN framework. Each challenge captures a major architectural, operational, regulatory, or user-centric trade-off that constrains one or more system requirements.

Table 3. Summary of principal challenge categories (C-) associated with the realization of a self-configurable ELN framework. Each challenge captures a major architectural, operational, regulatory, or user-centric trade-off that constrains one or more system requirements. The identifiers are referenced consistently throughout the manuscript and the requirement–challenge mappings.

ID	Challenge Identifier	Description
1	C-ARCH-FlexibilityComplexity	Balancing high modularity and configurability with architectural clarity, maintainability, and manageable system complexity.
2	C-ARCH-IntegrationHeterogeneous	Ensuring seamless interoperability across heterogeneous laboratory environments, including legacy systems, instruments, and institutional IT infrastructures.
3	C-ARCH-VersioningCompatibility	Maintaining compatibility across software versions, configurations, and evolving interfaces without compromising data integrity or system stability.
4	C-OPER-Scalability	Sustaining performance, reliability, and responsiveness as system usage, data volumes, and organizational scope increase.
5	C-OPER-ContinuousMaintenance	Managing long-term operation through monitoring, updates, and maintenance without disrupting ongoing research activities.
6	C-OPER-DataMigrationLegacy	Migrating historical data and established workflows from legacy systems while preserving scientific validity and legal compliance.
7	C-REG-SecurityCompliance	Maintaining robust security and regulatory compliance in the presence of evolving legal requirements and cybersecurity threats.

Table 3. *Cont.*

ID	Challenge Identifier	Description
8	C-REG-AuditabilityTraceability	Ensuring complete, tamper-evident traceability of actions within dynamically configurable workflows.
9	C-USER-UsabilityAccessibility	Designing user interfaces that remain intuitive and accessible despite increasing configurability and functional richness.
10	C-USER-TrainingChange	Supporting user training, change management, and adoption in environments with heterogeneous expertise and established practices.
11	C-USER-CustomizationStandardization	Balancing local customization needs with global consistency, interoperability, and institutional governance requirements.

2.3. Mapping Between Requirements and Challenges

The relationships between system requirements (R-) and principal challenge categories (C-) are explicitly captured in a structured mapping. A binary overview of requirement–challenge dependencies is provided in Table 4, while concise analytical justifications for these dependencies are summarized in Table A1. Together, these mappings establish a transparent analytical foundation for the solution architecture presented in the subsequent section.

The systematic mapping presented in Table 4 provides a transparent and rigorous analysis of how each core requirement is influenced by the principal architectural, operational, regulatory, and user-centric challenges identified in this work. By visualizing the direct associations between requirements (R-) and challenges (C-), the matrix highlights both the breadth and depth of the solution's coverage. Requirements that are influenced by multiple challenge categories are readily identified, underscoring their complexity and criticality for the system's overall robustness. Conversely, the mapping also reveals areas where challenges are especially pivotal for fulfilling specific requirements. This comprehensive overview enables a structured evaluation of the proposed architecture's adequacy and guides targeted refinement efforts to ensure all essential requirements are robustly addressed, even as research environments and regulatory landscapes evolve.

The mapping between requirements and challenges demonstrates that each principal challenge category imposes concrete and identifiable constraints on one or more system requirements. Architectural challenges primarily shape modularity, extensibility, and interoperability concerns, while operational challenges constrain scalability, reliability, and maintainability. Regulatory challenges directly affect security, compliance, and auditability requirements, and user-centric challenges influence usability, empowerment, and knowledge transfer. Together, these mappings confirm that the identified challenges comprehensively cover the requirement space and provide a sound analytical basis for the solution concept presented in the following section. Detailed justifications for individual requirement–challenge relationships are provided in Appendix A in Table A1.

2.4. Solution Concept

This section presents the solution concept for a self-configurable ELN, explicitly derived from the requirements and challenge analysis introduced in previous Sections 2.1 and 2.2. Rather than prescribing a fixed implementation, the solution concept is articulated through a set of modular features that collectively operationalize the identified design constraints. Each feature represents a reusable architectural mechanism that contributes to flexibility, regulatory robustness, interoperability, scalability, or user empowerment.

Table 4. Comprehensive mapping of key system requirements (R-) to principal challenge categories (C-). Each row represents a distinct, thematically grouped requirement, while each column corresponds to a major architectural, operational, regulatory, or user-centric challenge. A filled cell (•) indicates that the requirement is directly and substantively influenced by the respective challenge. The rightmost column (Addressed?) provides a synthesized assessment (✔) of whether the requirement is robustly addressed by the mapped challenge set. This mapping forms a rigorous analytical foundation for evaluating the adequacy and completeness of the proposed system architecture.

ID	Requirement Identifier	C-ARCH-FlexibilityComplexity	C-ARCH-IntegrationHeterogeneous	C-ARCH-VersioningCompatibility	C-OPER-Scalability	C-OPER-ContinuousMaintenance	C-OPER-DataMigrationLegacy	C-REG-SecurityCompliance	C-REG-AuditabilityTraceability	C-USER-UsabilityAccessibility	C-USER-TrainingChange	C-USER-CustomizationStandardization	Addressed?
1	R-ARCH-Modularity&Flexibility	•	•	•									✔
2	R-ARCH-Extensibility	•	•	•									✔
3	R-ARCH-Scalability				•	•	•						✔
4	R-ARCH-Interoperability&Standardization		•				•						✔
5	R-OPER-Security							•	•				✔
6	R-OPER-Reliability				•	•							✔
7	R-OPER-Maintainability			•		•							✔
8	R-OPER-Automation				•								✔
9	R-REG-Compliance							•	•				✔
10	R-REG-Auditability&Transparency							•	•				✔
11	R-USER-Empowerment									•	•	•	✔
12	R-USER-Accessibility									•			✔
13	R-USER-Internationalization										•		✔
14	R-USER-TrainingDocumentation										•	•	✔
15	R-USER-KnowledgeTransfer										•		✔

To preserve analytical clarity, features are introduced at a conceptual level and are referenced using stable identifiers (F-). Their relationships to the principal challenge categories are summarized through structured mappings, enabling transparent traceability from challenges to architectural mechanisms. A concise overview of all features is provided in Table 5, while detailed justifications of feature–challenge relationships are deferred to Appendix B.

The conceptual architecture defines essential system components: a configuration engine, adaptive user interface, reusable workflow templates, compliance enforcement modules, and integration layers for heterogeneous instruments and data sources. Each architectural element was explicitly designed to address one or more of the formalized requirements, ensuring the concept's comprehensive coverage and adaptability. The mapping of requirements to specific features was systematically documented to facilitate subsequent analysis. Each architectural mechanism introduced in the following subsections is explicitly motivated by one or more requirement–challenge relationships summarized in Table 4.

Table 5. Summary of core system features (F-) forming the solution concept of the proposed self-configurable ELN framework. Each feature represents a reusable architectural or functional mechanism designed to address one or more identified challenge categories. Feature identifiers are used consistently throughout the manuscript to support traceability and analytical rigor.

ID	Feature Identifier	Description
1	F-ConfigEngine-DynamicOrchestration	Runtime configuration and orchestration of modular ELN components.
2	F-ModuleRegistry-PlugPlay	Registry-based discovery and integration of modular system components.
3	F-UI-AdaptiveUX	Adaptive user interfaces tailored to roles and usage contexts.
4	F-WorkflowTemplates-ReusableBlueprints	Reusable workflow templates supporting reproducibility.
5	F-ComplianceModules-AutomatedRegulatory	Embedded regulatory compliance mechanisms.
6	F-IntegrationLayers-SeamlessInterop	Abstraction layers for heterogeneous system integration.
7	F-DataAcquisition-StructuredIntegration	Structured and reliable experimental data capture.
8	F-WorkflowOrchestration-ProcessAutomation	Automated execution of experimental workflows.
9	F-RegCompliance-AutomatedEnforcement	Systematic enforcement of compliance policies.
10	F-ReportingAnalytics-AutomatedVisualization	Automated reporting and visualization capabilities.
11	F-IntegrationInterop-BidirectionalConnectivity	Bidirectional data exchange with external systems.
12	F-ConfigWorkflow-IterativeUserCentric	Iterative, user-driven workflow configuration.
13	F-PublicAPI-Extensibility	Public APIs for extensibility and third-party integration.
14	F-VersionControl-ChangeTraceability	Versioning and traceability of changes and data.
15	F-RBAC-GranularPolicyEnforcement	Role-based access control and policy enforcement.

It is important to clarify that while the majority of ELN capabilities can be realized using native low-code and no-code tooling on platforms such as Microsoft SharePoint and Google Workspace, certain advanced functionalities—including dynamic cross-system workflow orchestration, real-time data streams, and highly specialized instrument integrations—may still require custom scripting (e.g., Apps Script or Power Automate) or additional connectors. Highlighting this trade-off early ensures that practitioners develop realistic expectations regarding the technical effort needed in complex deployment scenarios.

The primary objective of this research is to develop a universal, self-configurable framework capable of transforming generic digital platforms into powerful, adaptable, and compliant ELNs. This framework is explicitly designed to enable rapid, flexible self-configuration, empowering users to tailor digital laboratory environments to diverse research workflows and institutional requirements—without advanced IT skills or programming expertise [37].

This proposed approach represents a decisive departure from the constraints of traditional ELN solutions, which are often characterized by rigid architectures, high costs, and limited adaptability. Instead, the proposed framework democratizes digital research infrastructure: researchers, lab managers, and technical staff are empowered to independently configure, customize, and evolve their ELN systems, promoting scientific agility and operational autonomy.

The solution is rigorously engineered around a platform-agnostic, modular architectural paradigm, maximizing adaptability, scalability, and long-term sustainability across heterogeneous institutional and technological landscapes. Each component is architecturally decoupled yet tightly orchestrated, enabling independent evolution, robust fault isolation, and seamless integration—key prerequisites for sustainable digital research infrastructures [20].

Crucially, the framework systematically addresses the challenges of heterogeneous laboratory practices, complex regulatory landscapes, and the need for continuous innovation. It supports seamless integration of diverse data sources, laboratory instruments, and external applications. Core architectural principles—modularity, security, and extensibility—are operationalized through open-source configuration templates, automation tools, and standardized interfaces. This ensures that digital workflows are efficiently implemented, adapted, and scaled with minimal IT support and maximal compliance.

This paradigm shift marks a significant advance in the digital transformation of scientific research. By aligning architectural features directly with the requirements and challenges outlined earlier, the framework fosters innovation, reproducibility, and operational efficiency, while maintaining the highest standards for data integrity and regulatory compliance. Ultimately, it establishes a scientifically rigorous, future-proof foundation for next-generation, user-driven ELN systems.

2.4.1. Core Architectural Components

The framework is defined by the following main features (core architectural components), each fulfilling a critical role in enabling adaptability, compliance, and extensibility:

- **F-ConfigEngine-DynamicOrchestration**: The configuration engine serves as the central logic hub, interpreting and enforcing user-defined templates, policies, and compliance rules. It translates high-level user intent into executable system behaviors, enabling rapid adaptation to evolving protocols and regulatory requirements, and ensures consistent workflow orchestration across the system.
- **F-ModuleRegistry-PlugPlay**: The module registry provides a robust repository and lifecycle management system for plug-and-play modules (e.g., data acquisition, compliance validation, reporting, instrument integration). This feature enables dynamic module installation, update, and retirement, supporting continuous system evolution and parallel, independent development [19].
- **F-UI-AdaptiveUX**: The user interface layer delivers an accessible, intuitive, and visual interface (e.g., drag-and-drop, adaptive layouts) that empowers users of all technical backgrounds to configure, personalize, and optimize workflows and forms. The UI dynamically adapts to active modules and templates, ensuring a seamless user experience [37].
- **F-WorkflowTemplates-ReusableBlueprints**: Workflow templates encapsulate reusable, domain-validated best practices, protocols, and compliance procedures. Templates enable both standardization and targeted customization, facilitating rapid workflow instantiation and institutional knowledge transfer.
- **F-ComplianceModules-AutomatedRegulatory**: Compliance modules are dedicated modules that enforce regulatory requirements (e.g., audit trails, electronic signatures, access controls), ensuring verifiable adherence to standards such as ISO/IEC 17025 and GLP, and supporting automated compliance verification and reporting [35].
- **F-IntegrationLayers-SeamlessInterop**: The integration layers are abstraction layers that provide bidirectional connectivity with laboratory instruments, external data repositories, LIMS, and third-party analytics platforms. By supporting standardized

data exchange and interoperability protocols, they ensure robust, scalable data flow across heterogeneous research environments [8].

These core features collectively establish a comprehensive, layered architecture that underpins the rapid innovation, reproducibility, extensibility, and sustained regulatory compliance required for the digital transformation of research laboratories.

All architectural components interact exclusively via rigorously defined, versioned APIs and interface contracts, supporting deployment across a broad spectrum of target environments—including both on-premise infrastructures and cloud-native ecosystems such as Microsoft SharePoint and Google Workspace. This architectural rigor ensures seamless extensibility, robust security, and facilitates frictionless migration and interoperability across institutional and technological boundaries.

Figure 1 depicts the modular, layered architecture of the universal, self-configurable ELN framework. At its core, the configuration engine interprets user-defined templates, orchestrates workflow automation, and governs system-wide integration by enforcing standardized interface protocols. All functional modules, automation engines, and platform connectors communicate exclusively through this central engine, ensuring strict modularity, horizontal scalability, and continuous regulatory compliance. This design paradigm enables seamless, scalable integration with heterogeneous platforms (e.g., Microsoft SharePoint, Google Workspace), empowering researchers with a flexible, user-centric, and future-proof ELN solution.

The workflow for self-configurable ELN deployment, illustrated in Figure 2, commences with the specification of user-defined templates and operational rules. These are parsed and orchestrated by the configuration engine, which coordinates the execution of laboratory tasks and validation processes via automation modules. The fully configured ELN is then deployed on the designated target platform, where it is integrated into daily research operations. A built-in feedback mechanism supports ongoing, iterative system enhancement, ensuring that the ELN remains aligned with evolving scientific, technical, and regulatory requirements.

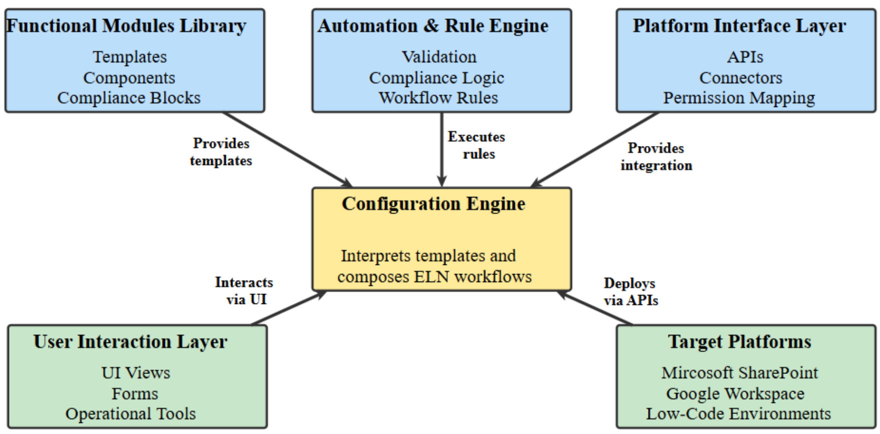

Figure 1. Comprehensive architectural overview of the universal, self-configurable ELN framework. The central Configuration Engine (yellow) functions as the orchestrator for all system operations, enabling modular, dynamic adaptation of diverse digital platforms into fully compliant ELNs. Subordinate modules (blue) provide specialized functional capabilities, automation services, and standardized integration interfaces. External entities (green) include end users and target platforms, with all interactions strictly mediated through the configuration engine. This enforces centralized governance, robust security, and platform-agnostic extensibility.

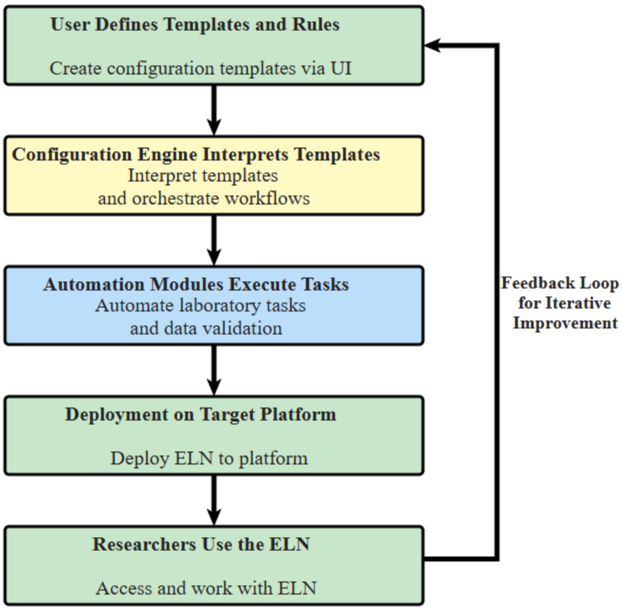

Figure 2. Detailed workflow for self-configurable ELN deployment. The process initiates with users specifying templates and operational rules, which are then interpreted and orchestrated by the configuration engine. Automation modules execute laboratory operations and data validation tasks. The configured ELN is deployed on the selected target platform, where it is actively utilized by researchers. An embedded feedback loop enables continuous, evidence-driven refinement and system optimization. The Configuration Engine is highlighted in yellow as the central orchestrator of workflow execution. Workflow components and internal process steps shown in blue represent subordinate functional modules and system services involved in configuration, automation, and integration. Elements shown in green denote external entities.

2.4.2. Module Types

The modular architecture is realized through a set of clearly defined, specialized module types—each representing a main feature that targets a specific aspect of laboratory digitization and operational excellence [8,19].

- **F-DataAcquisition-StructuredIntegration**: Data acquisition modules enable both structured and unstructured data entry, seamless file ingestion, and direct integration with laboratory instruments, guaranteeing data fidelity and real-time availability.
- **F-WorkflowOrchestration-ProcessAutomation**: Workflow orchestration modules model, execute, and monitor complex, multi-step laboratory workflows, embedding advanced conditional logic for approvals, exception handling, and automated sign-offs [20].
- **F-RegCompliance-AutomatedEnforcement**: Regulatory compliance modules enforce institutional and international mandates through robust audit trails, secure electronic signatures, and policy-driven access controls, supporting rigorous adherence to standards such as ISO/IEC 17025 [35].
- **F-ReportingAnalytics-AutomatedVisualization**: Reporting and analytics modules provide automated, customizable report generation and advanced data visualization, facilitating transparent documentation and evidence-based scientific decision-making [37].
- **F-IntegrationInterop-BidirectionalConnectivity**: Integration and interoperability modules facilitate secure, bidirectional connectivity with external systems—including LIMS, cloud repositories, and analytics platforms—via standardized protocols and APIs [8].

This modular feature set enables precise system tailoring, parallel module development, and agile adaptation to evolving scientific and regulatory requirements.

2.4.3. Configuration Process

F-ConfigWorkflow-IterativeUserCentric: Configuration is implemented as an iterative, user-driven workflow, leveraging advanced graphical interfaces and intelligent wizards. Users can employ a visual Template Designer to create or adapt workflow, data, and compliance templates. The drag-and-drop interface, combined with context-sensitive guidance, allows for flexible module composition and parameterization—entirely without programming expertise. All configurations are first validated in a sandbox environment, featuring automated consistency and compliance checks, before being deployed to production. Continuous, iterative refinement is supported to accommodate evolving research protocols and regulatory requirements [37].

2.4.4. Extensibility and Governance

The architecture incorporates a set of dedicated features to ensure both extensibility and robust institutional governance, which are essential for long-term adaptability, security, and regulatory compliance:

- **F-PublicAPI-Extensibility:** Extensibility is realized via rigorously documented, stable, and version-controlled public APIs. These APIs enable external developers to introduce new modules, integrate emerging technologies (such as AI-driven analytics or real-time monitoring), and achieve seamless interoperability with external platforms. This fosters a dynamic third-party innovation ecosystem, ensuring the continuous evolution and adaptability of the research infrastructure [20].
- **F-VersionControl-ChangeTraceability:** Comprehensive version control is applied to all configuration artifacts—including workflow templates, module parameters, and compliance documents—guaranteeing traceability, reproducibility, and robust audit trails for both internal governance and external regulatory review.
- **F-RBAC-GranularPolicyEnforcement:** Integrated role-based access control (RBAC) enables fine-grained permission management across users, groups, and roles. This enforces institutional security policies, restricts access to sensitive assets, and ensures compliance with regulatory frameworks such as ISO/IEC 17025 and GLP. Governance policies are extensible to support organizational hierarchies, delegated administration, and secure cross-institutional collaboration.

Collectively, these extensibility and governance features establish the structural foundation for a sustainable, secure, and dynamically adaptive digital research ecosystem, capable of evolving in lockstep with scientific progress and technological innovation.

Collectively, the introduced features translate abstract requirements and identified challenges into concrete architectural mechanisms. Each feature contributes to addressing specific design constraints, while their combined coverage ensures that architectural, operational, regulatory, and user-centric challenges are systematically mitigated. The completeness of this alignment is summarized through the binary feature–challenge mapping in the main text, with detailed analytical justifications provided in Appendix B.

2.4.5. Selection of Modules

Selecting the optimal combination of modules in complex system landscapes is a classical instance of combinatorial optimization. This process is fundamental in operations research and decision science, as documented in standard texts such as Nemhauser and Wolsey (1988) [38] and Bertsimas and Tsitsiklis (1997) [39]. Below, a mathematically rigorous framework for module selection will be presented, employing formal definitions, theorems, and proofs where appropriate.

Definition 1 (Cost and Flexibility of a Module)**.** *Let $n \in \mathbb{N}$ denote the total number of available modules. For each module indexed by j with $j \in \{1, \ldots, n\}$, the following quantities are defined:*

- *The* cost *associated with module j is a value $c_j \in \mathbb{R}_{\geq 0}$, where c_j quantifies the total resources (e.g., monetary, computational, or temporal) required for the acquisition, implementation, or maintenance of module j.*
- *The* flexibility score *associated with module j is a value $f_j \in \mathbb{R}_{\geq 0}$, where f_j measures the ability of module j to be reconfigured, extended, or adapted to varying requirements or environments. Larger values of f_j indicate greater flexibility.*

Definition 2 (Total Cost and Total Flexibility)**.** *Let $n \in \mathbb{N}$, and let $c_j \in \mathbb{R}^+$ and $f_j \in \mathbb{R}^+$ denote the cost and flexibility coefficients associated with module j for each $j \in \{1, \ldots, n\}$. For any vector $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, the* total cost *is defined by the mapping $C : \{0,1\}^n \to \mathbb{R}^+$, the* total flexibility *is defined by the mapping $F : \{0,1\}^n \to \mathbb{R}^+$:*

$$C(x) := \sum_{j=1}^{n} c_j \cdot x_j, \tag{1}$$

$$F(x) := \sum_{j=1}^{n} f_j \cdot x_j. \tag{2}$$

Definition 3 (Feasible Set)**.** *Let $m \in \mathbb{N}$, and let $g_k : \{0,1\}^n \to \mathbb{R}$ for $k \in \{1, \ldots, m\}$ be a finite collection of constraint functions encoding compliance, dependency, or regulatory requirements. Then the feasible set $\mathcal{F} \subseteq \{0,1\}^n$ is defined by*

$$\mathcal{F} := \{x \in \{0,1\}^n : g_k(x) \leq 0 \text{ for all } k \in \{1, \ldots, m\}\}. \tag{3}$$

Example 1 (Examples of Constraint Formulations)**.**

- *Dependency: If module 1 requires module 2: $x_1 - x_2 \leq 0$.*
- *Mutual Exclusion: If modules 3 and 4 cannot be selected together: $x_3 + x_4 \leq 1$.*
- *Minimum Flexibility: $\sum_{j=1}^{n} f_j \cdot x_j \geq F_{\min}$ for some threshold $F_{\min} \in \mathbb{R}^+$.*

Problem 1 (Module Selection Problem)**.** *Let $n \in \mathbb{N}$ be the number of available modules. For each $j \in \{1, \ldots, n\}$, let $c_j, f_j \in \mathbb{R}^+$ denote the cost and flexibility score of module j, respectively, and let $x_j \in \{0,1\}$ be a binary decision variable indicating whether module j is selected. Let $g_k : \{0,1\}^n \to \mathbb{R}$ for $k \in \{1, \ldots, m\}$ be m constraint functions encoding compliance or dependencies. $\mathcal{F}$ denotes the corresponding feasible set. For a fixed trade-off parameter $\lambda \in \mathbb{R}^+$, the objective function is defined by:*

$$\Phi_\lambda(x) := C(x) - \lambda \cdot F(x) = \sum_{j=1}^{n} c_j \cdot x_j - \lambda \sum_{j=1}^{n} f_j \cdot x_j. \tag{4}$$

Then, the module selection problem is to find $x^* \in \mathcal{F}$ such that

$$\Phi_\lambda(x^*) = \min_{x \in \mathcal{F}} \Phi_\lambda(x). \tag{5}$$

This formulation is a special case of the 0–1 integer linear programming problem, whose solvability and complexity are well-studied in the literature (see [38]).

Theorem 1 (Existence of an Optimal Solution of the Module Selection Problem)**.** *Let the module selection problem be defined as in Problem 1. If $\mathcal{F} \neq \emptyset$, then there exists at least one global minimizer $x^* \in \mathcal{F}$ of the problem.*

Proof. The set $\{0,1\}^n$ is finite, and thus any subset $\mathcal{F} \subseteq \{0,1\}^n$ is finite or empty. If $\mathcal{F} \neq \emptyset$, the objective function Φ_λ is linear (and therefore continuous) in x, so it attains its minimum over the finite set $\mathcal{F}$ by the Extreme Value Theorem (see, e.g., Rudin (1976) [40]). Thus, a global minimizer $x^* \in \mathcal{F}$ exists. □

Theorem 2 (Module Selection Problem for Laboratory Information Systems)**.** *Let $n = 15$ denote the number of available modules, each representing a core solution concept. The cost coefficients c_j and flexibility coefficients f_j for each module j are given in Table 6. The constraints are given in Table 7. Let $x_j \in \{0,1\}$ indicate whether module j is selected. The trade-off parameter is $\lambda = 2$. The feasible set $\mathcal{F}$ is defined by: $\mathcal{F} := \{x \in \{0,1\}^{15}$: All constraints in Table 7 are fulfilled$\}$.*

Proof. Firstly, the solution space will be reduced by examining the constraints:

- By necessity scores and explicit constraints, modules $j = 5, 7, 14, 15$ (compliance, data acquisition, versioning, security) are essential, i.e., $x_5 = x_7 = x_{14} = x_{15} = 1$.
- The constraint $x_7 - x_{11} \leq 0$ implies that $x_{11} = 1$ whenever $x_7 = 1$.
- The cardinality constraint $\sum_j x_j \geq 10$ requires at least ten modules.
- The mutual exclusion $x_8 + x_9 \leq 1$ prevents simultaneous selection of workflow automation and compliance enforcement.

Afterwards, a feasible solution will be constructed based on the reduced solution space. Let $S = \{1, 2, 4, 5, 6, 7, 9, 11, 14, 15\}$ be the set of selected modules, i.e., $x_j = 1$ for $j \in S$ and $x_j = 0$ otherwise. This selection fulfills all constraints in Table 7.

Finally, it will be shown that the constructed solution is optimal. For this, suppose there exists another feasible solution x' with $\Phi_\lambda(x') < \Phi_\lambda(x)$. Since all modules in S are either required by constraints or have the highest necessity, omitting any would violate feasibility. Adding further modules would only increase $\Phi_\lambda(x)$ due to the positive cost coefficients. Swapping any module in S for an unselected module would not yield a lower objective, as such modules have lower flexibility or are less necessary, and this would either violate the flexibility constraint or increase cost. Therefore, x with $x_j = 1$ for $j \in S$ and $x_j = 0$ otherwise is optimal under the given constraints and necessity structure. □

After establishing the theoretical foundations and the existence of an optimal solution for the underlying problem, Algorithm 1 can be used in practice to compute the optimal selection of ELN modules. By specifying cost and flexibility coefficients and tuning the trade-off parameter λ, it provides a configurable balance between cost and flexibility.

Algorithm 1 Computation of an Optimal Module Selection

Require: Cost coefficients c_j, flexibility coefficients f_j for $j = 1, \ldots, n$; constraint functions g_k for $k = 1, \ldots, m$; trade-off parameter $\lambda > 0$
Ensure: Optimal selection $x^* = (x_1^*, \ldots, x_n^*) \in \mathcal{F}$

1: Formulate the optimization problem $\min_{x \in \mathcal{F}} \sum_{j=1}^n c_j \cdot x_j - \lambda \cdot \sum_{j=1}^n f_j \cdot x_j$.
2: Eventually, select a suitable integer programming solver (e.g., Gurobi, CPLEX).
3: Solve the problem to obtain a global minimizer x^*.
4: **return** x^*

Table 6. Module coefficients and graded necessity (1–10) used in the module selection and prioritization problem for a modular, self-configurable ELN framework. The parameters c_j and f_j denote relative implementation cost and flexibility, respectively (1 = lowest, 10 = highest). The parameter n_j denotes the necessity reflecting the analytical importance of each module for supporting a universal, modular, and user-centric ELN architecture. Justifications summarize the primary rationale underlying each assessment.

ID	Module Name	c_j	f_j	n_j	Justification
1	F-ConfigEngine-DynamicOrchestration	6	9	9	Enables dynamic configuration and orchestration; critical for flexible and future-proof ELN deployments, though optional in strictly static environments.
2	F-ModuleRegistry-PlugPlay	4	7	8	Provides modularity and upgradeability with moderate cost; highly beneficial for extensibility but not mandatory for minimal systems.
3	F-UI-AdaptiveUX	5	7	7	Improves user acceptance and efficiency through adaptability; basic ELN functionality remains possible without advanced UX adaptation.
4	F-WorkflowTemplates-ReusableBlueprints	3	8	8	Low implementation cost with strong impact on reproducibility and workflow efficiency; recommended but not strictly required.
5	F-ComplianceModules-AutomatedRegulatory	7	10	10	Essential for operation in regulated research environments; high flexibility is required to accommodate evolving compliance requirements.
6	F-IntegrationLayers-SeamlessInterop	6	8	8	Supports interoperability across heterogeneous platforms; implementation effort is moderate when standardized APIs are available.
7	F-DataAcquisition-StructuredIntegration	7	9	10	Core functionality for ELNs, enabling reliable, structured, and compliant data capture and integration from diverse sources.
8	F-WorkflowOrchestration-ProcessAutomation	6	9	8	Reduces manual effort and errors while supporting scalability; full automation is not required in all laboratory contexts.
9	F-RegCompliance-AutomatedEnforcement	5	8	9	Strengthens compliance and reproducibility through automated enforcement mechanisms, particularly relevant in regulated domains.
10	F-ReportingAnalytics-AutomatedVisualization	3	6	6	Facilitates monitoring, analysis, and quality control; valuable but not essential for all ELN users.
11	F-IntegrationInterop-BidirectionalConnectivity	4	8	8	Enables seamless bidirectional data exchange with external systems, supporting extensibility and system integration.
12	F-ConfigWorkflow-IterativeUserCentric	4	7	7	Enhances user empowerment through iterative configuration; primarily beneficial for advanced or evolving use cases.
13	F-PublicAPI-Extensibility	3	6	6	Promotes openness and third-party integration with low additional cost when based on standard frameworks.
14	F-VersionControl-ChangeTraceability	4	8	10	Critical for auditability, reproducibility, and data integrity in scientific and regulated environments.
15	F-RBAC-GranularPolicyEnforcement	6	9	10	Essential for security and compliance through fine-grained access control; implementation cost is justified by risk mitigation.

Table 7. Formal constraints of the module selection optimization problem used to derive a coherent and compliant configuration of a modular, self-configurable ELN framework. The constraints encode logical dependencies between modules, regulatory minimum requirements, and global bounds on flexibility, cost, and overall system completeness.

Constraint	Interpretation
$x_7 - x_{11} \leq 0$	Bidirectional data integration is only permissible if structured data acquisition is selected, ensuring a valid data foundation for interoperability.
$x_8 + x_9 \leq 1$	Workflow automation and automated compliance enforcement are mutually exclusive in this formulation to avoid conflicting or redundant automation logic.
$x_5 + x_9 \geq 1$	At least one compliance-related module must be selected to satisfy baseline regulatory requirements.
$x_{14} + x_{15} \geq 2$	Both version control and role-based access control (RBAC) are mandatory to guarantee data integrity, traceability, and security.
$\sum_{j=1}^{15} f_j x_j \geq 60$	The aggregated flexibility score must exceed a minimum threshold, reflecting the objective of a highly adaptable and universal ELN architecture.
$\sum_{j=1}^{15} c_j x_j \leq B$	The total implementation cost of the selected module set must not exceed the available budget B.
$\sum_{j=1}^{15} x_j \geq 10$	A minimum number of modules must be selected to ensure functional completeness and long-term extensibility of the ELN framework.

https://doi.org/10.3390/app16010462

Theorem 3 (Interpretation of the Solution)**.** *Let x^* be an optimal solution of the module selection problem. Then x^* specifies the module selection that achieves the best trade-off between cost and flexibility under the given constraints.*

Proof. By construction, x^* minimizes the objective function Φ_λ over all feasible x, thus achieving the best trade-off as defined by λ and the constraints. □

2.4.6. Key Benefits

The proposed modular, self-configurable architectural paradigm provides a rigorous and systematic fulfillment of the defined requirements (R-) and thus, directly addresses the principal challenges (C-) identified in the preceding sections. The following structured mapping demonstrates how each core feature (F-) of the solution concept substantively supports one or more requirements, thereby ensuring functional completeness, traceability, and architectural justification for the proposed system [8,19].

Table 8 provides a detailed cross-mapping between all core system features (F-) and the key architectural, operational, regulatory, and user-centric requirements (R-) defined for the solution. Each row in the matrix illustrates the extent to which each requirement is supported by the available feature set, with filled cells denoting direct functional contributions. This systematic mapping not only highlights the comprehensive coverage of the requirements by the modular feature architecture, but also enables rapid identification of any potential gaps or redundancies. The 'Fulfilled?' column offers a synthesized assessment, confirming that each requirement is robustly addressed by at least one feature. This transparent and rigorous approach supports traceability, justifies architectural design choices, and underpins scientific evaluation of the system's adequacy and adaptability in complex research environments.

The requirement–feature mapping demonstrates that each identified system requirement is explicitly operationalized through one or more architectural or functional features of the proposed solution concept. Architectural requirements related to modularity, extensibility, scalability, and interoperability are primarily addressed through configuration engines, modular registries, and standardized integration interfaces. Operational requirements concerning reliability, automation, and maintainability are fulfilled by workflow orchestration, version control, and structured data acquisition mechanisms. Regulatory requirements for security, compliance, and auditability are systematically supported through embedded compliance modules, granular access control, and traceable change management. Finally, user-centric requirements related to empowerment, accessibility, training, and knowledge transfer are addressed through adaptive user interfaces, iterative configuration workflows, and reusable workflow templates. Together, this mapping confirms that the proposed feature set provides comprehensive and coherent coverage of the identified requirements. Detailed analytical justifications for individual requirement–feature relationships are provided in Appendix B (Table A2).

2.4.7. Summary

Taken together, the proposed solution concept translates abstract requirements and identified challenges into a coherent architectural framework. By systematically aligning design mechanisms with documented trade-offs, the architecture establishes a robust foundation for adaptable, compliant, and user-centered ELN deployments across heterogeneous research environments.

Table 8. Binary requirement–feature traceability matrix mapping identified system requirements (R-) to the core architectural and functional features (F-) introduced in the proposed solution concept. Each row represents a distinct requirement, while each column corresponds to a specific system feature. A filled cell (●) indicates that the respective feature directly and substantively contributes to fulfilling the requirement. The rightmost column (Fulfilled?) uses a blue checkmark (✔) to indicate whether a requirement is considered fully and robustly fulfilled by the mapped feature set. The matrix provides a systematic coverage analysis demonstrating how the proposed feature set operationalizes and collectively satisfies the defined requirements.

ID	Requirement Identifier	F-ConfigEngine-DynamicOrchestration	F-ModuleRegistry-PlugPlay	F-UI-AdaptiveUX	F-WorkflowTemplates-ReusableBlueprints	F-ComplianceModules-AutomatedRegulatory	F-IntegrationLayers-SeamlessInterop	F-DataAcquisition-StructuredIntegration	F-WorkflowOrchestration-ProcessAutomation	F-RegCompliance-AutomatedEnforcement	F-ReportingAnalytics-AutomatedVisualization	F-IntegrationInterop-BidirectionalConnectivity	F-ConfigWorkflow-IterativeUserCentric	F-PublicAPI-Extensibility	F-VersionControl-ChangeTraceability	F-RBAC-GranularPolicyEnforcement	Fulfilled?
1	R-ARCH-Modularity&Flexibility	●	●				●							●	●		✔
2	R-ARCH-Extensibility		●									●		●			✔
3	R-ARCH-Scalability								●			●					✔
4	R-ARCH-Interoperability&Standardization						●					●		●			✔
5	R-OPER-Security					●				●						●	✔
6	R-OPER-Reliability								●								✔
7	R-OPER-Maintainability	●	●												●		✔
8	R-OPER-Automation								●								✔
9	R-REG-Compliance					●				●						●	✔
10	R-REG-Auditability&Transparency							●		●					●		✔
11	R-USER-Empowerment			●									●				✔
12	R-USER-Accessibility			●												●	✔
13	R-USER-Internationalization												●				✔
14	R-USER-TrainingDocumentation												●				✔
15	R-USER-KnowledgeTransfer				●												✔

3. Results

This section presents the results of a systematic, theory-driven validation of the proposed modular and self-configurable ELN solution concept across multiple widely adopted digital platforms. Rather than reporting empirical performance metrics, the validation focuses on technical feasibility, architectural coverage, and regulatory adequacy under real-world platform constraints. The results demonstrate how the abstract solution concept can be operationalized using native platform mechanisms while preserving modularity, configurability, and compliance.

https://doi.org/10.3390/app16010462

The validation is conducted through structured implementation analyses on Microsoft SharePoint and Google Workspace. These platforms were selected as representative, large-scale enterprise collaboration environments that impose realistic architectural, security, and governance constraints. For each platform, the core elements of the solution concept, namely the configuration engine, module registry, and adaptive user interface, are systematically mapped to native services and integration mechanisms. This approach allows assessment of both functional coverage and inherent limitations without relying on proprietary ELN software. Platform-specific limitations, such as constrained native workflow orchestration, restricted API access, or limited real-time integration capabilities, are treated as boundary conditions of the target platforms rather than deficiencies of the proposed framework, and are explicitly considered in the analytical feasibility assessment.

As validation is understood as analytical feasibility validation, focusing on whether the proposed architectural features can be systematically instantiated using native mechanisms of representative digital platforms, the walkthroughs are not intended as empirical user studies or performance benchmarks, but as structured demonstrations of implementability, configurability, and compliance support under realistic platform constraints.

To complement the architectural validation, regulatory feasibility is analyzed by mapping core compliance requirements, such as those derived from GLP and ISO/IEC 17025, to native platform mechanisms. Table 9 summarizes how essential regulatory expectations related to data integrity, auditability, accountability, and controlled workflows can be satisfied using built-in features of Microsoft SharePoint and Google Workspace.

The table demonstrates that widely deployed collaboration platforms already provide a substantial regulatory baseline through mechanisms such as version control, access logging, identity management, and retention policies. Consequently, compliance in the proposed solution concept is achieved primarily through configuration and orchestration of existing platform services rather than through extensive domain-specific extensions or custom implementations.

Table 9. Mapping of core regulatory requirements relevant to compliant ELN operation to native platform features available in Microsoft SharePoint and Google Workspace. The table highlights how widely adopted collaboration platforms can natively support regulatory expectations related to data integrity, traceability, accountability, and controlled workflows without relying on domain-specific ELN implementations.

ID	Regulatory Requirement	Microsoft SharePoint	Google Workspace
1	Electronic Signatures and Approvals	User authentication, approval workflows, Power Automate approvals	Drive approvals, integrated identity management, Apps Script validation
2	Immutable Audit Trails	Versioning, Unified Audit Logs, access monitoring	Version history, activity logs, Apps Script logs
3	Data Integrity and Security	Role-based permissions, encryption, compliance center	Permissions, encryption, security center
4	Record Retention and Immutability	Retention policies, retention labels, records management	Vault retention rules, legal holds
5	Traceability of Changes	Metadata management, document history	Metadata, revision history
6	User Accountability	Role-based access control (Azure AD), audit trails	Role-based sharing, audit logs
7	Access Control	Fine-grained RBAC via Azure Active Directory	Access control via Google Groups and sharing policies
8	Controlled and Reproducible Workflows	Power Automate process orchestration	Apps Script and Google Forms–based workflows

3.1. Modular Self-Configuration in SharePoint

The SharePoint-based implementation serves as a representative realization of the proposed solution concept within a low-code/no-code enterprise platform. The results demonstrate that SharePoint, when combined with Power Automate, Power Apps, and native integration services, can support a modular, self-configurable ELN architecture without requiring traditional software development.

Core ELN capabilities, including structured data acquisition, workflow orchestration, compliance enforcement, reporting, and system integration, are instantiated through composable platform services. Users are thereby enabled to assemble, adapt, and evolve laboratory workflows through configuration rather than programming. This section systematically evaluates each proposed feature in terms of its technical realization and inherent platform constraints, providing an evidence-based assessment of feasibility and limitations.

The following analysis constitutes a feature-level feasibility assessment, demonstrating how each core feature of the solution concept can be realized using native SharePoint mechanisms and identifying structural limitations imposed by the platform.

1. **F-ConfigEngine-DynamicOrchestration**
   (a) **Implementation:** Store configuration templates as JSON/XML in SharePoint Lists. Use Power Automate to read templates and dynamically construct and run workflows. Power Apps can generate forms on-the-fly based on these templates. For advanced logic, custom SPFx (SharePoint Framework) web parts can interpret configurations and orchestrate UI/workflow changes.
   (b) **Limitations:** Deep conditional logic, cross-site orchestration, and runtime reconfiguration require custom development. No support for real-time, code-free reconfiguration across all users.
2. **F-ModuleRegistry-PlugPlay**
   (a) **Implementation:** Maintain a SharePoint List as a registry of available modules (flows, forms, reports). Power Automate enables toggling modules on/off or deploying new modules. App Catalog and SPFx allow deployment of new app packages.
   (b) **Limitations:** No native support for dynamic module loading at runtime; modules must be pre-registered and deployed. Microservice-like hot-plugging is not possible.
3. **F-UI-AdaptiveUX**
   (a) **Implementation:** Build adaptive user interfaces in Power Apps, using conditional logic to show/hide fields, change layouts, or adapt navigation based on user roles, context, or data. Use conditional formatting in SharePoint Modern UI for visual cues.
   (b) **Limitations:** Highly complex or deeply customized UIs (e.g., multi-level dynamic layouts, advanced theming) require SPFx or are not feasible. Responsiveness is limited to Power Apps capabilities.
4. **F-WorkflowTemplates-ReusableBlueprints**
   (a) **Implementation:** Store workflow templates as documents or list items (e.g., JSON for Power Automate flows). Create new workflows by duplicating and parameterizing templates via Power Automate. Use List/Library templates for rapid instantiation.
   (b) **Limitations:** No native template versioning or advanced blueprinting. Complex parameter substitutions or branching logic require custom scripting.
5. **F-ComplianceModules-AutomatedRegulatory**

(a) **Implementation:** Activate SharePoint versioning, retention, and audit logs. Integrate with Microsoft Compliance Center for policy enforcement. Power Automate can run compliance checks and escalate exceptions.
(b) **Limitations:** Automated workflows may lack granularity for highly specialized or evolving regulatory needs; some standards require manual review or external systems.

6. **F-IntegrationLayers-SeamlessInterop**
   (a) **Implementation:** Use Power Automate connectors for common external systems (e.g., Outlook, SQL, SAP). For custom integrations, leverage SharePoint REST API and Business Connectivity Services (BCS). SPFx can connect to APIs for advanced scenarios.
   (b) **Limitations:** Real-time, high-frequency, or low-latency integrations are limited by Power Automate and SharePoint API constraints. Some legacy or proprietary systems may require custom connectors or middleware.
7. **F-DataAcquisition-StructuredIntegration**
   (a) **Implementation:** Use Power Apps or Microsoft Forms for structured data input, writing directly to SharePoint Lists. Enforce validation rules on Lists. Power Automate enables import/export and transformation of data.
   (b) **Limitations:** Limited support for scientific/complex data types (e.g., spectra, proprietary formats, large files). Advanced parsing or visualization requires custom code.
8. **F-WorkflowOrchestration-ProcessAutomation**
   (a) **Implementation:** Orchestrate multi-step laboratory processes, such as approvals, notifications, and escalations, using Power Automate. Define triggers (e.g., when a SharePoint item is created or modified) to initiate workflow automation. Sequence tasks, send alerts, update records, and manage approval chains within the Power Automate environment.
   (b) **Limitations:** Long-running workflows may time out or require manual intervention. Advanced error handling, complex branching, and coordination across multiple workflows or sites are limited. Debugging and monitoring capabilities are basic, lacking detailed runtime diagnostics or step-through debugging.
9. **F-RegCompliance-AutomatedEnforcement**
   (a) **Implementation:** Configure SharePoint Information Management Policies and Data Loss Prevention (DLP) to automatically enforce compliance rules on documents and data. Use Power Automate to apply business rules, trigger notifications, and log compliance-related events. Integrate with Microsoft Compliance Center for centralized policy management and reporting.
   (b) **Limitations:** Granularity of enforcement is limited to available SharePoint and DLP features; deep audit trails or highly specialized compliance logic may not be natively supported. Advanced or custom regulatory requirements may necessitate third-party solutions or external auditing tools.
10. **F-ReportingAnalytics-AutomatedVisualization**
   (a) **Implementation:** Integrate Power BI dashboards with SharePoint Lists and Libraries to provide automated, real-time analytics and visualizations. Embed Power BI reports directly into SharePoint Modern Pages for user-friendly access to charts, metrics, and insights. Configure scheduled data refreshes and interactive filtering within Power BI.

(b) **Limitations:** Analytics capabilities are constrained by Power BI's refresh intervals and data model limitations. Complex or high-volume analytics, advanced statistical processing, or integration with non-Microsoft BI tools may require external platforms or additional licensing.

11. **F-IntegrationInterop-BidirectionalConnectivity**
    (a) **Implementation:** Bidirectional integration uses Power Automate to connect SharePoint with external APIs/databases for both import and export. Custom SPFx web parts and SharePoint's REST API support advanced two-way data exchange and synchronization.
    (b) **Limitations:** Not all APIs are natively supported; authentication, throttling, or real-time needs may require custom middleware. Error handling and monitoring are limited in Power Automate.
12. **F-ConfigWorkflow-IterativeUserCentric**
    (a) **Implementation:** Users iteratively design and refine workflows using Power Apps editors and SharePoint Modern UI, with versioning and rollback. Power Automate allows rapid testing of new logic.
    (b) **Limitations:** User changes are limited by permissions; advanced logic or branching needs SPFx or admin rights. No native multi-user real-time collaboration or sandboxing.
13. **F-PublicAPI-Extensibility**
    (a) **Implementation:** SharePoint exposes data via REST API and Microsoft Graph API; custom SPFx web services enable advanced endpoints.
    (b) **Limitations:** API usage is throttled and may lack event/webhook support. Security and governance can restrict access.
14. **F-VersionControl-ChangeTraceability**
    (a) **Implementation:** Built-in versioning and audit logs track changes to documents and list items. Users can access and restore version history.
    (b) **Limitations:** No native branching/merging or fine-grained diff for structured data; advanced scenarios need third-party tools.
15. **F-RBAC-GranularPolicyEnforcement**
    (a) **Implementation:** Granular access via SharePoint Groups, permissions, and Azure AD roles; Power Automate enforces RBAC logic in workflows.
    (b) **Limitations:** Very fine-grained, context-aware policies are complex; advanced logic may require custom SPFx or external tools.

SharePoint retention labels, immutable audit logs, sensitivity labels, version histories, and Power Automate approval workflows collectively enable granular control of access, document lifecycle management, and stepwise verification. Power Automate's compliance connectors enforce mandatory sign-offs, timestamped approval chains, and tamper-evident traceability, thereby enabling GLP- and ISO-aligned workflow execution within the theoretical framework.

Taken together, these results confirm that the proposed feature set is largely realizable within SharePoint's native ecosystem, with limitations primarily arising in scenarios requiring real-time reconfiguration, advanced orchestration logic, or highly specialized regulatory enforcement.

While the qualitative analysis highlights conceptual feasibility, Table 10 consolidates the results into a compact technical summary. The table synthesizes primary implementation mechanisms, dominant limitations, and relative configuration effort for each feature, enabling direct comparison across the feature set. The effort scale follows the normalized scoring scheme introduced in Table 1, ensuring methodological consistency across the evaluation.

Table 10. Concise technical mapping of ELN features to Microsoft SharePoint, summarizing primary implementation mechanisms and dominant limitations for each feature. Configuration effort is expressed using a normalized 0–10 scale and visualized by horizontal bars (longer bars indicate higher effort), enabling intuitive comparison across features. Feasibility reflects practical realizability using native platform mechanisms.

ID	Feature Name	Main Implementation and Limitations	Effort	Feas.
1	F-ConfigEngine-DynamicOrchestration	Templates and workflows managed in Lists and Power Automate; advanced or cross-site logic requires custom code.	2	High
2	F-ModuleRegistry-PlugPlay	Modules registered in Lists and activated via Power Automate; runtime hot-plugging is not supported.	3	High
3	F-UI-AdaptiveUX	Adaptive UIs via Power Apps; complex layouts or advanced theming require SPFx.	5	High
4	F-WorkflowTemplates-ReusableBlueprints	Templates stored as list items; duplication is easy but advanced blueprinting is limited.	3	High
5	F-ComplianceModules-AutomatedRegulatory	Versioning, retention, audit logs via Compliance Center; specialized compliance needs may exceed native support.	5	High
6	F-IntegrationLayers-SeamlessInterop	Integration via connectors, REST API, and BCS; real-time or legacy integration is constrained.	7	Medium
7	F-DataAcquisition-StructuredIntegration	Structured input via Power Apps/Forms; complex scientific data requires custom parsing.	5	High
8	F-WorkflowOrchestration-ProcessAutomation	Multi-step automation in Power Automate; long-running workflows and debugging are limited.	5	High
9	F-RegCompliance-AutomatedEnforcement	Policies and DLP enforce compliance; fine-grained audit logic may require extensions.	5	High
10	F-ReportingAnalytics-AutomatedVisualization	Reporting via Power BI; high refresh rates or complex analytics require external tools.	3	High
11	F-IntegrationInterop-BidirectionalConnectivity	Bidirectional sync via Power Automate/SPFx; API throttling and error handling are limiting factors.	7	Medium
12	F-ConfigWorkflow-IterativeUserCentric	Iterative workflow design with versioning; advanced edits need admin rights or SPFx.	5	High
13	F-PublicAPI-Extensibility	REST and Graph APIs enable extensibility; governance and throttling apply.	7	Medium
14	F-VersionControl-ChangeTraceability	Native versioning and audit logs; no branching or fine-grained diff for structured data.	1	High
15	F-RBAC-GranularPolicyEnforcemer	Permissions and Azure AD roles; highly dynamic RBAC scenarios are complex.	5	High

Overall, the results demonstrate that SharePoint enables a highly visual and user-centric configuration paradigm for ELN implementations. Laboratory workflows can be designed, adapted, and validated using graphical tools, with versioning and approval mechanisms supporting iterative refinement under compliance constraints. The modular composition of Lists, Power Automate flows, and Power Apps provides sufficient architectural flexibility for most operational laboratory scenarios.

At the same time, the analysis reveals systematic boundaries of low-code platforms: advanced dynamic orchestration, real-time bidirectional integrations, and highly specialized regulatory logic remain associated with increased implementation effort or require custom extensions. These findings underscore the central design trade-off addressed by the proposed solution concept—namely, maximizing configurability and user empowerment while maintaining architectural clarity, governance, and regulatory robustness.

3.2. Modular Self-Configuration in Google Workspace

The Google Workspace implementation serves as a complementary validation case that examines the proposed solution concept within a cloud-native, script-driven collaboration ecosystem. In contrast to SharePoint's low-code orchestration paradigm, Google Workspace relies primarily on Google Apps Script as a unifying automation and configuration layer. This setting enables evaluation of the solution concept under different architectural assumptions while preserving the same functional and regulatory objectives.

The results demonstrate that modular, self-configurable ELN workflows can be realized in Google Workspace by orchestrating Google Sheets, Forms, Docs, and Apps Script. Core ELN capabilities, such as structured data acquisition, workflow automation, compliance logging, reporting, and external system integration, are instantiated through composable scripts and templates. Users are thereby enabled to configure and adapt laboratory workflows using familiar collaborative tools, albeit with a higher reliance on scripted logic compared to graphical configuration environments.

The following feature-level analysis constitutes a structured feasibility assessment of the proposed solution concept within Google Workspace. For each feature, the technical realization using native platform mechanisms is examined alongside inherent limitations arising from platform constraints such as execution quotas, scripting complexity, and the absence of graphical orchestration tools. This analysis provides a transparent basis for assessing both functional coverage and practical boundaries.

1. **F-ConfigEngine-DynamicOrchestration**
    - **Implementation:** Use Google Apps Script to automate configuration workflows in Google Sheets. Scripts can read template data, dynamically generate forms or sheets, and orchestrate workflow steps (e.g., approvals, reminders) across multiple users.
    - **Limitations:** Advanced logic, cross-sheet orchestration, and real-time updates are complex and may require significant custom scripting. No native graphical workflow editor or code-free reconfiguration; scalability is limited by Apps Script quotas.
2. **F-ModuleRegistry-PlugPlay**
    - **Implementation:** Maintain a registry of modules (macros, scripts, templates) in a dedicated Google Sheet or via Form selections. Users can activate modules by copying templates or running scripts to add/remove functionality.
    - **Limitations:** No true runtime hot-plugging; modules must be manually activated or deactivated. Integration between modules is not seamless and may require manual intervention or scripting.
3. **F-UI-AdaptiveUX**
    - **Implementation:** Create adaptive user interfaces using Google Apps Script UI services and conditional formatting in Sheets. UI elements can change based on user role, input, or workflow state.
    - **Limitations:** Complex, multi-layered adaptive layouts are difficult to implement. UI customization is limited by the capabilities of Google Sheets and Apps Script; no support for advanced drag-and-drop or dynamic theming.
4. **F-WorkflowTemplates-ReusableBlueprints**

- **Implementation:** Store workflow templates as duplicate Sheets, Forms, or Apps Script files. Users can copy and adapt these templates to create new workflows or processes.
- **Limitations:** No built-in versioning or advanced blueprint management; tracking template evolution or complex branching requires manual tracking or custom solutions.

5. **F-ComplianceModules-AutomatedRegulatory**
    - **Implementation:** Use Apps Script to log actions, enforce data validation, and generate audit trails. Set up automated email notifications or access restrictions based on compliance needs.
    - **Limitations:** Lacks native support for regulatory standards (e.g., GLP, ISO 17025). Audit trails are basic; granular compliance enforcement and reporting require substantial custom development.
6. **F-IntegrationLayers-SeamlessInterop**
    - **Implementation:** Integrate with external systems via Apps Script's UrlFetchApp for REST APIs or third-party add-ons (e.g., Zapier, Make). Data can be imported/exported to/from Sheets or Forms.
    - **Limitations:** Real-time, high-frequency, or complex integrations are limited by Apps Script quotas and execution time. Legacy or proprietary systems may require middleware or are unsupported.
7. **F-DataAcquisition-StructuredIntegration**
    - **Implementation:** Capture data with Google Forms or Apps Script-enhanced Sheets. Validation rules can be set in Forms or via custom scripts for structured input.
    - **Limitations:** Support for scientific data types (e.g., spectra, images, large files) is limited. Advanced data parsing, transformation, or visualization requires custom code and may hit platform limits.
8. **F-WorkflowOrchestration-ProcessAutomation**
    - **Implementation:** Google Apps Script serves as the core configuration engine for automating and orchestrating multi-step workflows. Using triggers such as `onEdit` or time-driven events, scripts manage approvals, notifications, and status updates across Workspace applications like Forms, Sheets, Gmail, and others. This setup allows researchers to tailor ELN workflows to their specific needs without requiring advanced programming expertise.
    - **Limitations:** While powerful, the system has constraints in handling long-running processes, advanced error handling, and comprehensive workflow monitoring. Debugging capabilities are basic, and complex workflows may exceed Apps Script quotas. Additionally, implementing advanced logic, cross-document orchestration, or real-time updates necessitates intricate scripting. The absence of a native graphical workflow editor further limits ease of scalability and configuration.
9. **F-RegCompliance-AutomatedEnforcement**
    - **Implementation:** Use Apps Script to enforce business rules, log compliance actions, and restrict access. Set up notifications for violations or required reviews.
    - **Limitations:** Lacks deep audit trail or granular regulatory enforcement; compliance logic is basic and must be custom-scripted for each standard.
10. **F-ReportingAnalytics-AutomatedVisualization**

- **Implementation:** Visualize data with built-in Sheets charts, pivot tables, or Google Data Studio dashboards linked to Sheets.
- **Limitations:** Complex analytics, interactive dashboards, or high refresh rates require external tools or advanced scripting. Real-time analytics are limited by refresh intervals.

11. **F-IntegrationInterop-BidirectionalConnectivity**
    - **Implementation:** Enable bidirectional sync using Apps Script to push/pull data from external APIs or databases, or use add-ons for integration.
    - **Limitations:** Not all APIs are supported; authentication, quota, and real-time sync are limited. Monitoring and error recovery require custom logic.
12. **F-ConfigWorkflow-IterativeUserCentric**
    - **Implementation:** Users iteratively refine workflows by editing Sheets, Forms, or scripts. Version history in Sheets supports rollback.
    - **Limitations:** Deep workflow changes or collaborative real-time editing require advanced scripting. No graphical workflow builder or sandboxing.
13. **F-PublicAPI-Extensibility**
    - **Implementation:** Publish web APIs using Apps Script's doGet/doPost endpoints; expose data for integration with external apps.
    - **Limitations:** API usage is subject to quotas, lacks full webhook/event support, and may be restricted by Google Workspace admin policies.
14. **F-VersionControl-ChangeTraceability**
    - **Implementation:** Use version history in Sheets and Docs to track changes and recover previous versions. Apps Script can log changes to a dedicated log sheet.
    - **Limitations:** No branching, merging, or fine-grained diff for structured data; advanced versioning requires external tools or manual tracking.
15. **F-RBAC-GranularPolicyEnforcement**
    - **Implementation:** Manage access via Google sharing/permissions and group-based access control. Apps Script can restrict functions based on user email.
    - **Limitations:** Fine-grained RBAC, context-specific policies, or dynamic role changes are limited; complex scenarios require external identity management or manual control.

While the qualitative analysis confirms conceptual feasibility, Table 11 consolidates the results into a concise technical overview. The table summarizes the dominant implementation mechanisms, primary limitations, and relative configuration effort for each feature, using the same normalized effort scale as applied in the SharePoint evaluation. This enables consistent cross-platform comparison of implementation complexity and feasibility.

From a regulatory perspective, Google Workspace provides foundational mechanisms for traceability and accountability through revision history, execution logs, and event-driven triggers. Apps Script enables timestamped logging of workflow actions, while Google Drive version history supports immutable record reconstruction. When combined with Google Vault retention policies and structured access controls, these mechanisms allow the implementation of auditability and compliance logic aligned with the theoretical requirements of the proposed ELN framework.

Table 11. Concise technical mapping of ELN features to Google Workspace, summarizing primary implementation mechanisms and dominant limitations for each feature. Configuration effort is expressed using a normalized 0–10 scale and visualized by horizontal bars (longer bars indicate higher effort), enabling intuitive cross-feature comparison. Feasibility reflects practical realizability using native Google Workspace components.

ID	Feature Name	Main Implementation and Limitations	Effort	Feas.
1	F-ConfigEngine-DynamicOrchestration	Apps Script automates workflows and dynamic configuration in Sheets; advanced logic or cross-sheet orchestration is complex.	3	High
2	F-ModuleRegistry-PlugPlay	Module registry via Sheets or Forms; no true run-time hot-plugging, manual activation required.	3	High
3	F-UI-AdaptiveUX	Custom UI via Apps Script and conditional formatting; limited support for complex adaptive layouts.	5	High
4	F-WorkflowTemplates-ReusableBlueprints	Templates implemented as Sheet copies or Forms; lacks native versioning and advanced blueprint management.	2	High
5	F-ComplianceModules-AutomatedRegulatory	Audit trails and enforcement via Apps Script; advanced compliance features require extensive customization.	5	Medium
6	F-IntegrationLayers-SeamlessInterop	API-based integration via Apps Script and add-ons; real-time and complex integrations are constrained by quotas.	7	Medium
7	F-DataAcquisition-StructuredIntegration	Structured input via Forms and validation rules in Sheets; advanced scientific data handling requires scripting.	5	High
8	F-WorkflowOrchestration-ProcessAutomation	Trigger-based automation using Apps Script; long-running processes and error handling are limited.	5	High
9	F-RegCompliance-AutomatedEnforcement	Rule enforcement and basic logging via Apps Script; lacks deep auditability and regulatory coverage.	5	Medium
10	F-ReportingAnalytics-AutomatedVisualization	Reporting via Sheets charts and Looker Studio; complex analytics require external tools.	2	High
11	F-IntegrationInterop-BidirectionalConnectivity	Two-way synchronization via Apps Script; authentication and real-time limitations apply.	7	Medium
12	F-ConfigWorkflow-IterativeUserCentric	Iterative refinement through Sheets and Forms; deep changes and collaboration require scripting.	5	High
13	F-PublicAPI-Extensibility	Web APIs via Apps Script doGet/doPost; quota limits and lack of webhook support constrain extensibility.	7	Medium
14	F-VersionControl-ChangeTraceability	Native version history in Sheets and Docs; no branching or granular diff for structured data.	1	High
15	F-RBAC-GranularPolicyEnforcemer	Access control via sharing and groups; fine-grained or dynamic RBAC is limited.	5	Medium

In summary, the results demonstrate that Google Workspace offers a viable and adaptable foundation for modular ELN implementations within collaborative, cloud-native environments. The platform supports user-driven configuration, rapid deployment, and strong collaborative workflows using familiar tools, thereby lowering adoption barriers in research settings.

At the same time, the analysis reveals systematic limitations inherent to script-centric platforms. Advanced dynamic orchestration, real-time bidirectional integrations, and fine-grained regulatory enforcement require substantial custom development and are constrained by execution quotas and the absence of native graphical workflow editors. These findings reinforce the central design insight of the proposed solution concept: while modular self-configuration is achievable across heterogeneous platforms, the balance between usability, flexibility, and governance is fundamentally shaped by platform architecture.

3.3. Overall Summary

The results of the cross-platform analysis demonstrate that the proposed modular, self-configurable ELN solution concept is broadly applicable across heterogeneous digital ecosystems. By systematically operationalizing the solution concept on Microsoft SharePoint, representing a closed, enterprise-grade platform, and Google Workspace, representing a widely accessible, cloud-native environment, the study confirms that the core design principles of modularity, user-centric configuration, and workflow automation can be realized under fundamentally different architectural and governance constraints.

Across both platforms, the solution concept proved transferable and structurally robust. Native automation, configuration, and integration mechanisms enabled the assembly, adaptation, and deployment of ELN workflows without reliance on proprietary ELN software. This cross-platform feasibility provides a strong theoretical validation of the universality of the proposed approach and demonstrates that modular self-configuration is not tied to a specific vendor, technology stack, or deployment model.

The analysis further reveals that the majority of core ELN features—including structured data capture, workflow automation, reporting, version control, and basic compliance support—can be implemented with moderate configuration effort using platform-native low-code or script-based tools. At the same time, advanced capabilities such as dynamic orchestration, seamless bidirectional integration, and fine-grained regulatory enforcement consistently require increased implementation effort, custom extensions, or external services. These findings highlight a systematic distinction between readily achievable baseline functionality and more complex, platform-sensitive features.

From an operational perspective, the results indicate substantial benefits in terms of rapid deployment, adaptability, and user empowerment. Leveraging familiar collaboration environments lowers adoption barriers and enables iterative refinement of laboratory workflows while maintaining traceability and governance. However, the analysis also identifies inherent limitations imposed by platform-specific constraints, including execution quotas, restricted real-time processing, limited graphical workflow design, and coarse-grained access control models. Addressing these constraints may necessitate additional technical expertise or supplementary infrastructure.

In conclusion, the results confirm that the modular, self-configurable ELN concept is both technically feasible and strategically adaptable across diverse digital platforms. The successful realization on both proprietary and open ecosystems positions the concept as a robust and vendor-agnostic framework for digital laboratory transformation, capable of supporting a wide range of organizational contexts while transparently exposing the trade-offs between flexibility, complexity, and governance.

These findings directly inform the limitations and design trade-offs discussed in the following section.

4. Discussion

This section synthesizes and critically interprets the results of the study in relation to the defined research questions (RQ1–RQ3) and hypotheses (H1–H3). Rather than reiterating implementation details, the discussion consolidates cross-platform insights to assess (i) the feasibility and universality of the proposed modular, self-configurable ELN concept, (ii) the conceptual, technical, and regulatory conditions required for its realization, and (iii) its implications for performance, scalability, user acceptance, and long-term sustainability in heterogeneous research environments.

Methodologically, the study follows a design-science research paradigm, where the primary contribution lies in the systematic derivation, structuring, and validation of an architectural solution space for ELN systems. The focus is therefore not on benchmarking a single software artifact, but on establishing transferable design principles through explicit traceability between requirements, constraints, and realizable features. This approach ensures analytical rigor while providing a solid foundation for future empirical and experimental validation.

4.1. Synthesis of Major Findings

The results demonstrate that a modular, self-configurable ELN architecture can be consistently instantiated across fundamentally different digital ecosystems. By mapping identical functional requirements to Microsoft SharePoint and Google Workspace, representing, respectively, a tightly integrated enterprise platform and a widely accessible collaborative environment, the study confirms that the core architectural principles of modularity, declarative configuration, and user-centric workflow composition are not platform-dependent. Instead, they emerge as robust, transferable design abstractions capable of supporting diverse laboratory contexts.

4.2. Feasibility and Universality of the Modular ELN Solution (RQ1, H1)

The cross-platform implementations provide strong evidence that the proposed ELN concept is technically feasible and broadly applicable. Essential ELN capabilities—including structured data acquisition, configurable workflows, version control, auditability, reporting, and integration—were successfully realized using native low-code/no-code mechanisms and extensibility interfaces on both platforms. This confirms H1 and directly addresses RQ1, demonstrating that neither proprietary enterprise environments nor open collaborative platforms impose fundamental barriers to modular ELN realization.

Importantly, the results show that feasibility does not imply uniform implementation complexity. While core functionalities can be deployed with low to moderate effort, advanced features—such as dynamic orchestration, bidirectional interoperability, or fine-grained regulatory enforcement—systematically require additional scripting, administrative privileges, or external services. Nevertheless, the ability to implement these features without resorting to domain-specific ELN software underscores the universality and architectural soundness of the proposed concept.

4.3. Conceptual, Technical, and Regulatory Requirements (RQ2)

Addressing RQ2, the comparative analysis reveals that three requirement layers are decisive for successful ELN implementation: conceptual, technical, and regulatory. Conceptually, strict modular decomposition and declarative configuration are essential to enable reuse, adaptability, and researcher-driven customization without compromising system coherence. Technically, the availability of extensible automation engines, template-driven workflows, persistent versioning, and standardized integration interfaces emerges as a prerequisite for sustainable ELN operation.

From a regulatory perspective, compliance is not achieved through isolated features, but through the systematic interaction of version control, immutable audit trails, access control mechanisms, and enforced workflow steps. Both platforms demonstrate that core compliance expectations—aligned with standards such as ISO/IEC 17025 and GLP—can be supported natively to a substantial degree. However, platform-specific constraints, including API rate limits, scripting quotas, and limited RBAC granularity, restrict the extent to which highly specialized or evolving regulatory requirements can be enforced without custom development. These findings reinforce the necessity of a layered, extensible architecture that anticipates regulatory evolution rather than hard-coding compliance logic.

4.4. Comparative Performance, Scalability, and User Acceptance (RQ3, H2)

With respect to RQ3 and H2, the results indicate that modular ELNs built on existing digital platforms offer clear advantages in adaptability, deployment speed, and user acceptance. Leveraging familiar interfaces and visual configuration tools lowers entry barriers, accelerates onboarding, and enables rapid workflow iteration—factors that are particularly valuable in interdisciplinary, project-driven, or rapidly evolving research environments. The bar-based effort visualization introduced in the Results section (Section 3) further highlights that most core features remain within a moderate configuration effort range, supporting practical scalability.

At the same time, the analysis delineates the boundaries of this approach. In highly regulated, data-intensive, or mission-critical laboratory settings, commercial ELN solutions may retain advantages due to specialized compliance tooling, integrated analytics, or dedicated vendor support. Consequently, the modular concept should not be viewed as a universal replacement for all ELN systems, but as a flexible and cost-efficient alternative that can be selectively extended or complemented where advanced requirements arise. Overall, the findings provide strong support for H2 while offering a nuanced, context-sensitive interpretation.

4.5. Cost, Vendor Lock-In, and Data Stewardship (H3)

The results strongly validate H3 by demonstrating that a platform-agnostic, modular ELN architecture substantially reduces both initial and long-term costs. By relying on existing institutional infrastructure and standardized configuration artifacts, the need for bespoke software development is minimized. Furthermore, the use of interoperable templates, open APIs, and transparent workflow logic significantly enhances data portability and long-term stewardship.

This architectural openness directly mitigates vendor lock-in and aligns with open science principles, institutional data governance strategies, and funding agency requirements. The ability to migrate workflows, data structures, and compliance logic across platforms represents a critical advantage over monolithic ELN solutions and positions the proposed concept as a sustainable foundation for future research data management ecosystems.

4.6. Benefits

The results reveal a broad spectrum of benefits across stakeholder groups. For researchers, the approach enables rapid, autonomous configuration of ELN workflows, reducing setup time, configuration errors, and dependency on specialized IT support. Visual, low-code configuration tools democratize workflow design and allow laboratories to adapt quickly to evolving protocols and regulatory conditions. Integrated audit trails, versioning, and access control mechanisms enhance traceability and regulatory readiness while reducing manual documentation effort.

For administrators and institutions, the modular architecture supports incremental system evolution, efficient integration of new tools or instruments, and optimized use of existing digital platforms. This reduces total cost of ownership, facilitates institutional scaling, and strengthens governance and compliance oversight. For both groups, enhanced transparency, interoperability, and adaptability foster a culture of digital literacy, innovation, and continuous improvement.

4.7. Limitations and Outlook

Despite its demonstrated strengths, the proposed solution concept has inherent limitations. Advanced customizations, such as complex branching logic, real-time processing, or integration with specialized laboratory hardware, often exceed the capabilities of standard low-code tools and require professional development expertise. Platform-specific constraints, including execution quotas and limited workflow visualization, may further restrict enterprise-scale or mission-critical deployments.

Moreover, while generalizability has been demonstrated across two widely used platforms, broader validation across additional environments (e.g., open-source collaboration stacks or LIMS platforms) is necessary to fully establish universality. Finally, the present study is analytical in nature and does not include quantitative performance measurements or large-scale user studies. Future work will therefore focus on empirical validation in operational laboratory settings, encompassing usability evaluations, scalability benchmarks, and longitudinal assessments of compliance effort and maintenance cost. Such studies will enable direct comparison with commercial ELN systems and further refine the proposed architectural principles.

5. Conclusions

5.1. Summary of Main Findings

This study delivers compelling empirical and theoretical evidence that modular, self-configurable ELNs built on generic digital platforms are both technically feasible and broadly transformative across the scientific research landscape. Comprehensive validation across proprietary and open-source environments demonstrates that these ELNs offer unparalleled versatility, enabling seamless adaptation to a wide spectrum of laboratory settings and research disciplines.

The principal strengths of this approach lie in its intrinsic flexibility, substantial cost efficiency, and robust support for regulatory compliance and data stewardship. Most critically, the democratization of ELN deployment, achieved through low-code/no-code tools and native extensibility mechanisms, empowers ordinary researchers and local administrators to independently assemble, configure, and maintain ELNs tailored to specific workflows and compliance requirements. This capability accelerates digital adoption, fosters rapid innovation, and reduces dependence on IT specialists or costly proprietary vendors, making advanced digital infrastructure accessible to laboratories of all sizes and organizational structures.

By systematically mapping comprehensive ELN requirements, including modularity, usability, regulatory alignment, interoperability, scalability, auditability, and cost-effectiveness, to explicit technical features and implementation strategies, this work establishes a rigorous scientific foundation for the next generation of ELN solutions. Cross-platform validation confirms that the majority of ELN requirements can be met with moderate effort using platform-native automation, configuration templates, and flexible integration layers. This approach not only reduces operational and transition costs but also mitigates vendor lock-in, enhances data stewardship, and supports long-term institutional sustainability in alignment with open science mandates.

Nevertheless, the research also delineates important limitations. Advanced needs, such as dynamic workflow orchestration, highly granular compliance enforcement, real-time data processing, and fine-grained access control, may surpass the capabilities of standard platform tools and necessitate targeted custom development or third-party integrations. Rather than diminishing the value proposition, these challenges highlight critical opportunities for further technological innovation and strategic investment.

In sum, these findings advocate for a decisive shift toward platform-agnostic, modular ELN frameworks that prioritize user empowerment, regulatory robustness, and technological resilience. For researchers, IT leaders, and institutional decision-makers, self-configurable ELNs present a clear, scalable, and sustainable pathway to enhanced adaptability, operational efficiency, and collaborative scientific innovation. Continuous investment in user training, technical support, and the integration of advanced technologies—such as AI-driven configuration and automated compliance monitoring—will be essential to fully realize the long-term impact and transformative potential of these solutions.

5.2. Implications for Research and Practice

The findings of this study have transformative and far-reaching implications for the digital transformation of scientific research, particularly through the deployment of self-configurable ELNs on generic digital platforms. By fundamentally lowering the technical barriers to ELN implementation, the modular, low-code/no-code approach democratizes access to advanced digital laboratory infrastructure. Researchers and local administrators—regardless of programming expertise—can now independently assemble, adapt, and maintain ELNs tailored to the specific and evolving needs of their laboratories or collaborative projects.

This empowerment enables research groups to respond rapidly and flexibly to changing scientific, regulatory, and institutional requirements, without reliance on specialized IT staff or costly vendor solutions. The modular framework's universal applicability and transferability—demonstrated by successful deployment on both proprietary and open-source platforms—significantly reduces barriers to digital adoption. It supports seamless integration across diverse research settings, from small teams to large multi-institutional consortia, and fosters interoperability between heterogeneous environments.

Crucially, this approach enhances scalability, sustainability, and institutional control over data management. By reducing operational costs and minimizing vendor lock-in, modular ELNs directly support open science policies and strengthen the long-term resilience of research infrastructure. While the majority of scientific and regulatory needs can be met with moderate effort using platform-native tools, ongoing innovation is required for advanced functionalities such as dynamic workflow orchestration, fine-grained compliance, and real-time integration—highlighting future opportunities for technological advancement.

The outcomes of this study have transformative implications for the digital research infrastructure of the future, especially concerning the deployment of self-configurable ELNs on generic digital platforms:

- **Democratization of ELN Implementation:** This research demonstrates that modular, low-code/no-code ELN frameworks structurally empower researchers and local administrators to independently assemble, adapt, and deploy ELNs tailored to precise scientific and regulatory needs. This marks a paradigm shift, enabling rapid, decentralized, and user-driven digital transformation—without reliance on dedicated IT personnel or advanced programming expertise.
- **Universal Applicability and Transferability (RQ1, H1):** The successful deployment of self-configurable ELNs across both proprietary and open-source platforms proves broad accessibility and transferability. This dramatically lowers barriers to digital adoption, enabling seamless integration from small research teams to large, multi-institutional collaborations.
- **Comprehensive Requirement Fulfillment and Regulatory Alignment (RQ2):** Systematic mapping of ELN requirements to explicit technical features confirms that modularity, usability, compliance, interoperability, scalability, and auditability are

achievable through platform-native tools. While most needs are met with moderate effort, advanced functionalities—such as dynamic orchestration, granular compliance, and real-time integration—may require targeted custom development or third-party extensions, highlighting opportunities for ongoing innovation.

- **Cost Efficiency, Sustainability, and Data Stewardship (H3):** The modular, platform-agnostic approach delivers substantial reductions in implementation cost and effort, enhances data stewardship and portability, and minimizes vendor lock-in. These strengths directly support sustainable, long-term digital infrastructure strategies and align with open science mandates.
- **Enhanced Performance and User Acceptance (RQ3, H2):** The modular, self-configurable model fosters high adaptability, usability, and rapid deployment, driving broad user acceptance—particularly in interdisciplinary and dynamic research environments. While most requirements are achievable with moderate resources, highly specialized or complex scenarios may still benefit from commercial or bespoke solutions.

In conclusion, these results strongly advocate for a strategic transition toward platform-agnostic, modular ELN frameworks that can be configured and maintained by researchers and local administrators themselves. To fully realize the transformative potential of self-configurable ELNs, institutions should prioritize investments in user training, technical support, and the integration of advanced technologies—such as AI-driven configuration and automated compliance monitoring—thereby ensuring sustainable, scalable adoption across the scientific enterprise.

5.3. Recommendations for Institutional Adoption

To maximize the transformative potential of modular, self-configurable ELN solutions, institutions must pursue a deliberate and scientifically grounded implementation strategy. Central to this approach is substantial investment in comprehensive user training programs empowering researchers and administrators to independently configure, adapt, and optimize ELNs for their unique scientific and regulatory environments. Institutions should establish robust, sustained technical support structures that not only address operational challenges but also foster a culture of continuous improvement and user-driven innovation.

Strategic priorities must include the adoption and integration of advanced technologies, particularly artificial intelligence for automated configuration, compliance monitoring, and adaptive user interfaces. Expanding platform compatibility and ensuring seamless interoperability with existing laboratory and data management systems are equally critical for future-proofing research infrastructure. Institutions should also implement rigorous governance frameworks to ensure regulatory compliance, data stewardship, and information security throughout the ELN lifecycle.

Ultimately, the adoption of platform-agnostic, user-centric ELN frameworks, supported by targeted training, advanced technological integration, and strong institutional governance, will be essential for achieving sustainable digital transformation, regulatory alignment, and long-term research excellence.

5.4. Future Research Directions

While this study establishes a comprehensive requirements framework and presents a modular solution concept for self-configurable ELNs, several important steps remain for future research and practical realization.

Building on the rigorous scientific foundation established by this study, future research must pursue a comprehensive, multi-dimensional agenda to fully unlock the transformative potential of platform-agnostic, modular, self-configurable ELN solutions.

First and foremost, systematic empirical validation is required across a much broader spectrum of digital platforms, laboratory environments, and scientific disciplines. This includes not only traditional research settings but also highly specialized, interdisciplinary, and rapidly evolving domains. Such validation will ensure the generalizability, robustness, and practical impact of modular ELN frameworks. In more detail, the conceptual architecture should be implemented in representative digital environments to validate its technical feasibility and operational robustness. Pilot deployments in authentic laboratory settings, such as materials science research groups, will be essential to assess real-world applicability and adaptability.

A critical research priority is the seamless integration of self-configurable ELNs with emerging laboratory automation systems, advanced data repositories, and next-generation analytical tools. By enabling direct, automated data exchange and orchestration across heterogeneous digital ecosystems, ELNs can become the backbone of fully digital, interoperable research environments. Continuous and rigorous evaluation of sustainability, interoperability, and adaptability in response to shifting technological and regulatory landscapes is essential to maintain long-term relevance and institutional value.

In parallel, research must drive the development and refinement of intelligent, AI-driven features—particularly automated configuration, compliance monitoring, and adaptive user interfaces. These innovations promise to further reduce the technical barriers for end users, enhance operational efficiency, and ensure robust regulatory alignment. Enhanced security and audit mechanisms, along with continuous improvements in user interface design, must also be a focus to address the demands of increasingly complex and collaborative research workflows.

To guide this evolution, conceptual roadmaps, supported by detailed visual schematics or mind maps—should be developed. As illustrated in Figure 3, future ELN frameworks will be shaped by the convergence of AI-driven automation, expanded multi-platform compatibility, robust security, and continuous user-centric design improvements. This strategic vision will be critical for accelerating the digital transformation of research, ensuring reproducibility, and fostering more efficient, collaborative scientific discovery.

Empirical evaluation should include both quantitative and qualitative methods. Quantitative metrics may encompass system setup time, error incidence, compliance adherence, and user intervention frequency. Qualitative insights can be gained through structured user satisfaction surveys, usability studies, and the identification of adoption barriers. Additionally, comparative analyses with existing commercial ELN solutions, involving feedback from both end users and IT specialists, are recommended to further refine the concept and inform broader institutional adoption.

Future work should address advanced challenges such as workflow orchestration, highly granular compliance enforcement, and real-time data integration, which may require targeted custom development or third-party extensions. The integration of emerging technologies—such as AI-driven configuration and automated compliance monitoring—represents a promising direction for enhancing the sustainability and scalability of digital laboratory infrastructures.

In summary, the next phase of research must be ambitious and integrative, targeting empirical validation, technological innovation, and strategic guidance to ensure that modular, self-configurable ELNs realize their full promise as the foundation of future-ready, digitally transformed research environments.

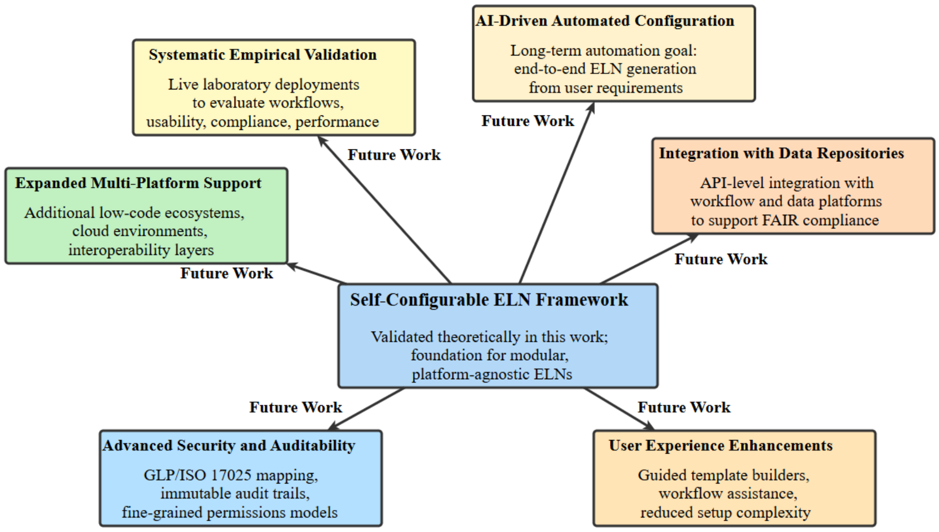

Figure 3. Future Directions of the ELN Framework. The mind map highlights the central role of the self-configurable ELN and illustrates key future enhancement areas: AI-driven automated configuration, multi-platform support expansion, integration with data repositories, enhanced security and audit, user interface improvements, and systematic empirical validation.

5.4.1. Technical Implementation Enhancements

Further research should aim to strengthen platform independence, sustainability, and extensibility by deepening the integration of open-source technologies and declarative configuration paradigms. Expanding the compatibility of modular user interface components and developing universal connectors for secure, standards-compliant data exchange with a broader array of laboratory instruments are essential. Additionally, future work should enhance core features such as fine-grained access control, immutable audit trails, and electronic signatures, with a focus on compliance with evolving regulatory standards beyond ISO/IEC 17025 and GLP.

5.4.2. Expanded Pilot Deployments and Use Case Diversification

Future studies should extend pilot deployments to a wider variety of laboratory environments, including industrial, clinical, and interdisciplinary research settings. Involving a broader user base and a more diverse set of laboratory instruments will provide valuable insights into system adaptability, scalability, and practical impact. Longitudinal studies are recommended to evaluate the effects of iterative customization and workflow evolution on research efficiency and data integrity over time.

5.4.3. Refined Evaluation Frameworks

Subsequent research should refine evaluation methodologies by incorporating advanced analytics such as automated compliance monitoring, machine learning-based anomaly detection, and real-time user feedback mechanisms. Collecting more comprehensive qualitative data through longitudinal interviews and focus groups will help to identify persistent adoption barriers and inform user-centered design enhancements.

5.4.4. Comprehensive Comparative Analyses

Ongoing comparative analyses with both commercial and emerging open-source ELN solutions should be systematically conducted. Benchmarking against a broader range of platforms and user scenarios will help to identify best practices, inform continuous system refinement, and support the development of guidelines for institutional adoption and large-scale deployment.

5.5. Concluding Statement

In conclusion, this work establishes modular, self-configurable electronic laboratory notebooks as a scientifically grounded and practically viable foundation for the digital transformation of laboratory research. By lowering technical barriers to customization, enhancing adaptability across heterogeneous research contexts, and promoting interoperability and responsible data stewardship, the proposed framework supports more transparent, efficient, and resilient laboratory workflows.

The central contribution of this study is the demonstration that electronic laboratory notebooks can be conceived as sustainable, evolvable digital infrastructures rather than monolithic software products, provided that modularity, self-configuration, and governance are treated as first-class architectural design principles. This perspective is particularly well aligned with the increasing complexity of contemporary scientific practice, which is characterized by interdisciplinary collaboration, rapidly evolving methodologies, and growing regulatory and documentation requirements.

Modular, platform-agnostic ELN frameworks are therefore uniquely positioned to accommodate these dynamics, enabling laboratories to integrate emerging technologies, adapt workflows incrementally, and preserve long-term reproducibility and compliance without incurring vendor lock-in or excessive technical overhead. As research environments continue to evolve, such frameworks will play a central role in supporting sustainable, transparent, and collaborative scientific ecosystems.

The most urgent next step is a systematic empirical evaluation of the proposed framework in diverse laboratory settings, focusing on usability, regulatory compliance, and operational performance under real-world conditions. Beyond this, a particularly promising long-term direction lies in the integration of AI-assisted configuration mechanisms capable of generating optimized workflows, compliance rules, and user interfaces based on domain context and historical usage patterns, further advancing the adaptability and intelligence of future ELN systems.

Author Contributions: Conceptualization, K.F., M.Z. and H.W.; Data curation, K.F.; Formal analysis, K.F. and M.Z.; Funding acquisition, H.W. and S.I.; Investigation, K.F. and M.Z.; Methodology, K.F. and M.Z.; Project administration, K.F., H.W. and M.Z.; Resources, K.F. and M.Z.; Software, K.F.; Supervision, H.W. and S.I.; Validation, K.F. and M.Z.; Visualization, K.F.; Writing—original draft, K.F. and M.Z.; Writing—review & editing, K.F., H.W., M.Z. and S.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the German Federal Ministry for Economics and Climate Action (BMWK) on the basis of decisions by the German Bundestag within the joint research projects “LaSt” (grant number 20M2118F) and “SWaT” (grant number 20M2112F). Furthermore, the research was supported by the German Federal Ministry of Education and Research (BMBF) and financed by the EU within the joint research project “Come2Data” (grant number 16DKZ2044A).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article. All figures and diagrams presented in this article were created using yEd Graph Editor (version 3.25.1, https://www.yworks.com/products/yed (accessed on 8 November 2025)) in GraphML format, exported in SVG format, and set in the open source font Liberation Sans (version 2.15, SIL Open Font License, https://github.com/liberationfonts/liberation-fonts (accessed on 8 November 2025)).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Detailed Requirement–Challenge Justifications

This appendix provides a detailed analytical justification of the relationships between the system requirements (R-) and the principal challenge categories (C-) introduced in Sections 2.1 and 2.2. While the main text presents a compact binary overview of requirement–challenge dependencies to support coverage analysis and structural adequacy (Table 4), the present appendix explicates the underlying rationale for each identified dependency.

The justifications summarized in Table A1 articulate why specific architectural, operational, and regulatory challenges directly constrain, influence, or motivate the fulfillment of individual system requirements. Each entry provides a concise explanation of the causal or conceptual linkage between a challenge category and its associated requirement, thereby strengthening the traceability and transparency of the analytical process.

Importantly, this appendix does not introduce additional requirements or challenges, nor does it modify the mappings presented in the main text. Instead, it substantiates the binary mappings already defined by making the implicit reasoning explicit and auditable. This separation allows the core manuscript to remain focused and readable, while preserving a complete and methodologically rigorous reasoning chain that supports the proposed solution architecture.

Table A1. Analytical justification of the relationships between principal challenge categories (C-) and associated system requirements (R-). Each entry explains why a specific challenge directly constrains or influences fulfillment of the corresponding requirement. This table complements the binary overview in Table 4 by explicating the rationale underlying each identified dependency.

Challenge ID (C-)	Requirement ID (R-)	Justification
C-ARCH-FlexibilityComplexity	R-ARCH-Modularity&Flexibility	Achieving modularity and flexibility inherently increases architectural complexity, requiring careful design to prevent fragmentation and loss of system coherence.
C-ARCH-FlexibilityComplexity	R-ARCH-Extensibility	Extensibility amplifies configuration and dependency complexity, necessitating architectural mechanisms that balance adaptability with long-term maintainability.
C-ARCH-IntegrationHeterogeneous	R-ARCH-Modularity&Flexibility	Modular architectures must remain integrable across heterogeneous laboratory environments, including legacy systems and diverse instrumentation.
C-ARCH-IntegrationHeterogeneous	R-ARCH-Extensibility	Extensible frameworks must accommodate heterogeneous platforms and technologies without compromising system stability or interoperability.
C-ARCH-IntegrationHeterogeneous	R-ARCH-Interoperability&Standardization	Interoperability depends on robust integration mechanisms and standardized interfaces capable of spanning heterogeneous system landscapes.
C-ARCH-VersioningCompatibility	R-ARCH-Modularity&Flexibility	Modular systems require systematic versioning strategies to preserve compatibility as components evolve independently.
C-ARCH-VersioningCompatibility	R-ARCH-Extensibility	Extensible systems must ensure backward compatibility to prevent disruption during incremental system evolution.
C-ARCH-VersioningCompatibility	R-OPER-Maintainability	Maintainability is directly supported by consistent versioning and compatibility management across system components.
C-OPER-Scalability	R-ARCH-Scalability	Increasing numbers of users, workflows, and data volumes directly challenge the system's ability to scale efficiently.
C-OPER-Scalability	R-OPER-Reliability	Reliable system behavior at scale requires robust architectural safeguards and operational controls.

https://doi.org/10.3390/app16010462

Table A1. *Cont.*

Challenge ID (C-)	Requirement ID (R-)	Justification
C-OPER-Scalability	R-OPER-Automation	Automation mitigates operational overhead and reduces error rates as system scale and complexity increase.
C-OPER-ContinuousMaintenance	R-ARCH-Scalability	Sustained scalability depends on continuous monitoring, optimization, and infrastructure maintenance.
C-OPER-ContinuousMaintenance	R-OPER-Reliability	Ongoing maintenance activities are essential to preserve reliability and fault tolerance over time.
C-OPER-ContinuousMaintenance	R-OPER-Maintainability	Maintainability is intrinsically linked to continuous updating, patching, and monitoring processes.
C-OPER-DataMigrationLegacy	R-ARCH-Scalability	Migration of legacy data affects scalability by introducing constraints related to data consistency and system performance.
C-OPER-DataMigrationLegacy	R-ARCH-Interoperability&Standardization	Successful legacy data migration requires interoperability with existing formats and systems through standardized interfaces.
C-REG-SecurityCompliance	R-OPER-Security	Robust security controls are fundamental for protecting sensitive research data and enforcing access restrictions.
C-REG-SecurityCompliance	R-REG-Compliance	Regulatory compliance mandates the implementation of verifiable and enforceable security mechanisms.
C-REG-SecurityCompliance	R-REG-Auditability&Transparency	Security mechanisms underpin reliable audit trails and transparent system operation.
C-REG-AuditabilityTraceability	R-OPER-Security	Traceable security events are essential for demonstrating accountability and regulatory compliance.

Appendix B. Detailed Requirement–Feature Justifications

This appendix provides a detailed analytical justification of the relationships between the system requirements (R-) and the core system features (F-) introduced in the solution concept in Section 2.4. While the main body of the manuscript presents a concise binary mapping to demonstrate requirement coverage and conceptual completeness, the present appendix explicates how individual features concretely operationalize and support specific requirements.

The justifications summarized in Table A2 clarify the functional and architectural role of each feature in fulfilling the corresponding requirement. Each entry explains the underlying mechanism through which a feature enables, constrains, or reinforces a requirement, thereby strengthening the transparency, traceability, and methodological rigor of the proposed solution concept.

Importantly, this appendix does not introduce additional requirements or features, nor does it alter the mappings presented in the main text. Instead, it substantiates the traceability between abstract design constraints and concrete architectural mechanisms, providing an auditable reasoning chain that supports the validity and completeness of the modular, self-configurable ELN framework.

Table A2. Analytical justification of the relationships between system requirements (R-) and associated features (F-). Each entry explains why a specific feature directly supports, constrains, or enables fulfillment of the corresponding requirement. This table complements the binary overview in Table 8 by explicating the rationale underlying each identified dependency.

Requirement ID (R-)	Feature ID (F-)	Justification
R-ARCH-Modularity&Flexibility	F-ConfigEngine-DynamicOrchestration	Enables runtime reconfiguration of modular components without structural disruption, preserving system flexibility.
R-ARCH-Modularity&Flexibility	F-ModuleRegistry-PlugPlay	Supports modular composition through discoverable, loosely coupled modules that can be assembled as needed.

Table A2. *Cont.*

Requirement ID (R-)	Feature ID (F-)	Justification
R-ARCH-Extensibility	F-PublicAPI-Extensibility	Enables external extensions and integrations without modification of core system components.
R-ARCH-Extensibility	F-ModuleRegistry-PlugPlay	Allows new functional modules to be added incrementally while maintaining architectural coherence.
R-ARCH-Scalability	F-WorkflowOrchestration-ProcessAutomation	Supports scalable execution and coordination of complex experimental and administrative workflows.
R-ARCH-Scalability	F-IntegrationInterop-BidirectionalConnectivity	Enables horizontal scaling within distributed and collaborative research ecosystems.
R-ARCH-Interoperability&-Standardization	F-IntegrationLayers-SeamlessInterop	Abstracts platform and system heterogeneity through standardized integration interfaces.
R-ARCH-Interoperability&-Standardization	F-PublicAPI-Extensibility	Promotes interoperability by exposing open, documented, and standards-aligned APIs.
R-OPER-Security	F-RBAC-GranularPolicyEnforcement	Enforces fine-grained, role-based access control policies to protect sensitive data and operations.
R-OPER-Security	F-ComplianceModules-AutomatedRegulatory	Embeds security-relevant compliance rules directly into system workflows and configurations.
R-OPER-Reliability	F-WorkflowOrchestration-ProcessAutomation	Reduces manual intervention and error rates by ensuring consistent and repeatable process execution.
R-OPER-Maintainability	F-ConfigEngine-DynamicOrchestration	Supports controlled system evolution through configurable behavior rather than hard-coded changes.
R-OPER-Maintainability	F-VersionControl-ChangeTraceability	Enables traceable system evolution, rollback, and impact analysis across configuration and data changes.
R-OPER-Automation	F-WorkflowOrchestration-ProcessAutomation	Automates procedural execution of experimental, administrative, and compliance-related workflows.
R-REG-Compliance	F-RegCompliance-AutomatedEnforcement	Ensures continuous and consistent enforcement of regulatory constraints across workflows and data handling.
R-REG-Compliance	F-RBAC-GranularPolicyEnforcement	Supports regulatory compliance by enforcing role-appropriate access and separation of duties.
R-REG-Auditability&Transparency	F-VersionControl-ChangeTraceability	Ensures complete and traceable documentation of system, workflow, and data changes.
R-REG-Auditability&Transparency	F-DataAcquisition-StructuredIntegration	Captures provenance-aware experimental data in a structured and auditable manner.
R-USER-Empowerment	F-ConfigWorkflow-IterativeUserCentric	Enables user-driven configuration and adaptation of workflows without programming expertise.
R-USER-Empowerment	F-UI-AdaptiveUX	Supports intuitive and responsive interaction with configurable system components.
R-USER-Accessibility	F-UI-AdaptiveUX	Ensures accessible interaction patterns for diverse user roles and skill levels.
R-USER-Accessibility	F-RBAC-GranularPolicyEnforcement	Enables role-appropriate system access aligned with user responsibilities.
R-USER-Internationalization	F-ConfigWorkflow-IterativeUserCentric	Supports adaptable workflows and configurations across institutional and cultural contexts.
R-USER-TrainingDocumentation	F-ConfigWorkflow-IterativeUserCentric	Facilitates incremental learning and onboarding through iterative configuration and reuse.
R-USER-KnowledgeTransfer	F-WorkflowTemplates-ReusableBlueprints	Supports reuse, standardization, and dissemination of validated experimental workflows.

References

1. Rubacha, M.; Rattan, A.K.; Hosselet, S.C. A Review of Electronic Laboratory Notebooks Available in the Market Today. *J. Lab. Autom.* **2011**, *16*, 90–98. [CrossRef]
2. Bird, C.L.; Willoughby, C.; Frey, J.G. Laboratory notebooks in the digital era: The role of ELNs in record keeping for chemistry and other sciences. *Chem. Soc. Rev.* **2013**, *42*, 8157–8175. [CrossRef]
3. Kwok, R. How to pick an electronic laboratory notebook. *Nature* **2018**, *560*, 269–270. [CrossRef] [PubMed]
4. Dirnagl, U.; Przesdzing, I. A pocket guide to electronic laboratory notebooks in the academic life sciences. *F1000Research* **2016**, *5*, 2. [CrossRef]
5. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 160018. [CrossRef] [PubMed]
6. Mons, B.; Neylon, C.; Velterop, J.; Dumontier, M.; da Silva Santos, L.O.B.; Wilkinson, M.D. Cloudy, increasingly FAIR; revisiting the FAIR Data guiding principles for the European Open Science Cloud. *Inf. Serv. Use* **2017**, *37*, 49–56. [CrossRef]

7. Machina, H.K.; Wild, D.J. Electronic Laboratory Notebooks Progress and Challenges in Implementation. *SLAS Technol.* **2013**, *18*, 264–268. [CrossRef] [PubMed]
8. Kanza, S.; Willoughby, C.; Gibbins, N.; Whitby, R.J.; Frey, J.G.; Erjavec, J.; Zupančič, K.; Hren, M.; Kovač, K. Electronic lab notebooks: Can they replace paper? *J. Cheminform.* **2017**, *9*, 31. [CrossRef]
9. Leonelli, S. *Data-Centric Biology: A Philosophical Study*; University of Chicago Press: Chicago, IL, USA, 2016.
10. Higgins, S.G.; Nogiwa-Valdez, A.A.; Stevens, M.M. Considerations for implementing electronic laboratory notebooks in an academic research environment. *Nat. Protoc.* **2022**, *17*, 179–189. [CrossRef] [PubMed]
11. Hey, T.; Tansley, S.; Tolle, K. *The Fourth Paradigm: Data-Intensive Scientific Discovery*; Microsoft Research: Redmond, WA, USA, 2009.
12. Begley, C.G.; Ioannidis, J.P.A. Reproducibility in Science: Improving the Standard for Basic and Preclinical Research. *Circ. Res.* **2015**, *116*, 116–126. [CrossRef]
13. Nosek, B.A.; Alter, G.; Banks, G.; Borsboom, D.; Bowman, S.D.; Breckler, S.J.; Buck, S.; Chambers, C.D.; Chin, G.; Christensen, G.; et al. Promoting an open research culture. *Science* **2015**, *348*, 1422–1425. [CrossRef] [PubMed]
14. Feldhoff, K.; Opatz, T.; Wiemer, H.; Zinner, M.; Ihlenfeldt, S. Benchmarking and Lessons Learned from Using SharePoint as an Electronic Lab Notebook in Engineering Joint Research Projects. *Data* **2025**, *10*, 92. [CrossRef]
15. Tremouilhac, P.; Nguyen, A.; Huang, Y.C.; Hübsch, F.; Stierstorfer, D.; Bräse, S. Chemotion ELN: An Open Source electronic lab notebook for chemists in academia. *J. Cheminform.* **2017**, *9*, 54. [CrossRef] [PubMed]
16. Carpi, N.; Minges, A.; Piel, M. eLabFTW: An open source laboratory notebook for research labs. *J. Open Source Softw.* **2017**, *2*, 146. [CrossRef]
17. Milsted, A.J.; Hale, J.R.; Frey, J.G.; Neylon, C. LabTrove: A Lightweight, Web Based, Laboratory "Blog" as a Route towards a Marked Up Record of Work in a Bioscience Research Laboratory. *PLoS ONE* **2013**, *8*, e67460. [CrossRef] [PubMed]
18. Goecks, J.; Nekrutenko, A.; Taylor, J.; The Galaxy Team. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *BMC Bioinform.* **2011**, *12*, 468. [CrossRef] [PubMed]
19. Parnas, D.L. On the Criteria To Be Used in Decomposing Systems into Modules. *Commun. ACM* **1972**, *15*, 1053–1058. [CrossRef]
20. Fowler, M. *Patterns of Enterprise Application Architecture*; Addison-Wesley: Reading, MA, USA, 2002.
21. Kulesza, T.; Burnett, M.M.; Wong, W.K.; Stumpf, S. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In Proceedings of the 20th International Conference on Intelligent User Interfaces, Atlanta, GA, USA, 29 March–1 April 2015; pp. 126–137. [CrossRef]
22. Opatz, T.; Feldhoff, K.; Wiemer, H.; Ihlenfeldt, S. Sharing Research Data in Collaborative Material Science and Engineering Projects. *Data* **2025**, *10*, 53. [CrossRef]
23. Ajimati, M.O.; Carroll, N.; Maher, M. Adoption of low-code and no-code development: A systematic literature review and future research agenda. *J. Syst. Softw.* **2025**, *222*, 112300. [CrossRef]
24. Campagna, D.; Del Piccolo, A.; Kaklamanis, K.; Šulc, S.; Bernardi, M.D.; Ellero, F.; Kalourazi, S.F.; Reimann, K.; Andrea, M.; Kordos, K.; et al. Streamlining multi-scale materials modeling: The MUSICODE low-code approach for simulation workflows and executable MODAs. *Integr. Mater. Manuf. Innov.* **2025**, *14*, 136–152. [CrossRef]
25. Suvvari, S.K. Ensuring Security and Compliance in Agile Cloud Infrastructure Projects. *Int. J. Comput. Eng.* **2024**, *6*, 54–73. [CrossRef]
26. Vandendorpe, J.; Adam, B.; Wilbrandt, J.; Lindstädt, B.; Förstner, K.U. Ten simple rules for implementing electronic lab notebooks (ELNs). *PLoS Comput. Biol.* **2024**, *20*, e1012170. [CrossRef]
27. Scroggie, K.R.; Burrell-Sander, K.J.; Rutledge, P.J.; Motion, A. GitHub as an open electronic laboratory notebook for real-time sharing of knowledge and collaboration. *Digit. Discov.* **2023**, *2*, 1188–1196. [CrossRef]
28. Lamprecht, A.L.; Garcia, L.; Kuzak, M.; Martinez, C.; Arcila, R.; Pico, E.M.D.; Angel, V.D.D.; van de Sandt, S.; Ison, J.; Martinez, P.A.; et al. Towards FAIR principles for research software. *Data Sci.* **2020**, *3*, 37–59. [CrossRef]
29. Jalali, M.; Luo, Y.; Caulfield, L.; Sauter, E.; Nefedov, A.; Wöll, C. Large language models in electronic laboratory notebooks: Transforming materials science research workflows. *Mater. Today Commun.* **2024**, *40*, 109801. [CrossRef]
30. Voegele, C.; Bouchereau, B.; Robinot, N.; McKay, J.; Damiecki, P.; Alteyrac, L. A universal open-source Electronic Laboratory Notebook. *Bioinformatics* **2013**, *29*, 1710–1712. [CrossRef] [PubMed]
31. Guerrero, S.; Dujardin, G.; Cabrera-Andrade, A.; y Miño, C.P.; Indacochea, A.; Inglés-Ferrándiz, M.; Palomino-Navarrete, H.; Colina, N.; Díaz, Y.; Melo, I.D.; et al. Analysis and Implementation of an Electronic Laboratory Notebook in a Biomedical Research Institute. *PLoS ONE* **2016**, *11*, e0160428. [CrossRef] [PubMed]
32. Jordt, P.; Osterhoff, M.; Tymoshenko, Y.; Hakim, B.; Dolcet, P.; Maurer, F.; Biniyaminov, V.; Amelung, L.; Dall'Antonia, F.; Grunwaldt, J.D.; et al. Specifications for Electronic Laboratory Notebooks (ELN) in the Photon and Neutron Community. *Synchrotron Radiat. News* **2024**, *37*, 3–8. [CrossRef]
33. Lippi, G.; Plebani, M. A modern and pragmatic definition of Laboratory Medicine. *Clin. Chem. Lab. Med.* **2020**, *58*, 1171. [CrossRef]

34. *OECD Principles of Good Laboratory Practice*; OECD Series on Principles of Good Laboratory Practice and Compliance Monitoring No. 1; Organisation for Economic Co-Operation and Development (OECD): Paris, France, 1998.
35. *ISO/IEC 17025:2017*; General Requirements for the Competence of Testing and Calibration Laboratories. International Organization for Standardization: Geneva, Switzerland, 2017. Available online: https://www.iso.org/standard/66912.html (accessed on 9 November 2025).
36. *FDA 21 CFR Part 11*; Electronic Records, Electronic Signatures. U.S. Food and Drug Administration: Washington, DC, USA, 1997.
37. Nielsen, J. *Usability Engineering*; Morgan Kaufmann: Burlington, MA, USA, 1993.
38. Nemhauser, G.L.; Wolsey, L.A. *Integer and Combinatorial Optimization*; Wiley-Interscience Series in Discrete Mathematics and Optimization; Wiley: New York, NY, USA, 1988.
39. Bertsimas, D.; Tsitsiklis, J.N. *Introduction to Linear Optimization*; Athena Scientific: Belmont, MA, USA, 1997.
40. Rudin, W. *Principles of Mathematical Analysis*, 3rd ed.; International Series in Pure and Applied Mathematics; McGraw-Hill: New York, NY, USA, 1976.