Politecnico di Torino
International University

Master's Thesis

Artificial Intelligence-Enhanced Building Code Navigation: Improving Regulatory Compliance Efficiency in Architectural Practice

——A Case Study in the Chinese Context

Master of Science in Architecture

Politecnico di Torino

MASTER'S THESIS

ARCHITECTURE FOR SUSTAINABILITY

Author: Zhili Niu

Student ID：328251

Supervising Professor: Carlo Deregibus

CONTENT

Abstract:

Building code consultation is a critical and inefficient task for architecture, which needs highly fragmented information, interpretive ambiguity and over-reliance on expert experts. While general-purpose AI tools such as ChatGPT are readily available, they have hallucinations (create unfamiliar code by using no documentation) and the domain-specific advice practitioners need.

This thesis explores AI-enhanced code consultation through a mixed-methods case study in Chinese regulation. Experiments conducted by four large-scale architecture firms revealed the following key issues: senior expert bottlenecks, the gap between regulatory text and practical application, multi-domain consultation complexity, and the verification requirements for professional accountability. These findings reveal four hierarchical requirements: reliable source attribution, embedded expert interpretation, context-aware multi-area support, and workflow integration.

For these requirements, we propose Experience-Augmented RAG (EA-RAG), which is a two layer knowledge base (Code Layer + Experience Layer), multi-expert domain routing with sparse activation and structured response generation. Our prototype consists of 8 core Chinese building standards (~2,400 articles) and 168 curated expert knowledge samples in six domains. Evaluation with 30 practitioner-validated queries shows excellent performance over ChatGT-4o: 94% citation accuracy vs. vague references, 3% hallucination rate vs. 23%, +78% interpretation quality, and +122% results on large multi- domain queries. Expert blind testing verified the validity and usefulness of the EA-RAG.

This research contributes: (1) empirically validated practitioner requirements for AI-assisted code consultation, (2) the EA -RAG architecture shows that domain-adapted systems are superior to general-level LLMs, and (3) actionable consequences for practice, such as workflow integration, knowledge democratization at firm level, and educational shifts towards interpretation and human-AI cooperation. Our results show that AI for professional regulatory consultation requires not only accurate retrieval, but well curated interpretive guidance, and that code consultation can be updated from document search to expert-like assistance.

Keywords: Artificial Intelligence (AI); Building Code; Regulatory Compliance; Architectural Practice; Code Consultation; Large Language Models (LLMs); Retrieval-Augmented Generation (RAG); Experience-Augmented RAG (EA-RAG); Senior Expert Bottleneck; Tacit Knowledge; Dual-Layer Knowledge Base; Multi-Expert Domain Architecture

CHAPTER 1 INTRODUCTION

1.1 Research Context and Motivation

1.1.1 The Challenge of Building Code Compliance in Contemporary Practice

The field of Architecture is in a much more complex situation, where building code compliance is both an important and a challenging task. In the professional practice literature, "code compliance is both critical and challenging—architects must balance creative aspirations with the rigid frameworks of safety, functionality, and legality[1]". Despite their enormous importance to public health and safety, code consultation still may be one of the most time-consuming and error-prone aspects of building design, Not only is this complexity associated with the volume of regulations, but also because modern building codes operate across multiple jurisdictional levels—national or federal requirements, sector-specific regulations concerning certain building systems, and regional or local codes providing geographic adaptations[2]. The codes keep changing according to new research and technology, and are a time-consuming target for practitioners who need to keep updating and managing projects in the past code. Construction workers face many difficulties when trying to comply with building codes. ranging from lack of understanding due to complex legal language to difficulty keeping updated with rapid changes in the codes themselves[3].

1.1.2 Inefficiencies in Current Code Consultation Methods

Even though regulatory compliance is key, many new ways to inform a building code architect are still inefficient.Traditional approaches such as printed code books, keyword searches in PDF databases, and institutional memory, do not account for the interconnected nature of regulation: The limitations of other digital tools are obvious: PDF code books and keyword searches are no longer a good improvement over printed manuals, they struggle with several fundamental challenges that impede efficient professional practice[4]. Architects routinely face challenges attempting to comply with the complexities of building codes, ranging from lack of understanding due to complex legal language to difficulty keeping updated with rapid change[3].

A major challenge is to bridge the gap between knowledge sources. It is common that a document, a country code, an industry, local rules, technical details and instructions, are available, however the present tools only require architects to search each source manually, with no effective mechanism for identifying cross-references or related requirements[5]. This is an unfortunate combination to overcome since building codes are linked documents: a section might cite requirements in another chapter but some industry standard may not. Understanding the whole set of requirements for a specific design decision may involve navigating several references along multiple documents, although search tools that focus on keywords rather than semantic knowledge are very lacking for this sort of relational navigation.

Interpretation ambiguities represent another persistent challenge. Regulatory texts contain subjective requirements which challenge the practical adoption of automated compliance checking[6].Code language usually uses words such as adequate, reasonable, or appropriate that must be expertized by the context. Most search engines return text snippets without the larger context for interpretation, leaving architects to manually record background. These interpretation challenges are time-critical: building codes are constantly updated or modified to meet changes in technology or climate and social needs[7]. Practitioners should track which version is adopted by projects at various stages, understand transitions, and keep aware of possible changes in late design times.

Perhaps most importantly, current system struggles with organizational knowledge loss. Lee, Jung and Baek observe that “even experienced engineers cannot review all technical details and even experienced engineers are reluctant to transfer experience knowledge to new engineers"[8]. High level practitioners having decades of experience in regulatory applications leave or change companies as they lack previous knowledge of code interpretation, previous encounters with challenging situations, and working relations with regulatory authorities. The net effect of these imitations is not only related to direct time costs (perhaps one-fifth of project hours) but also indirect effects in design quality, innovation, and risk management. Uncertain regulatory results could discourage architects to consider new solutions and concerns about the requirements may lead to over-optimized designs and late time replacements. [9].

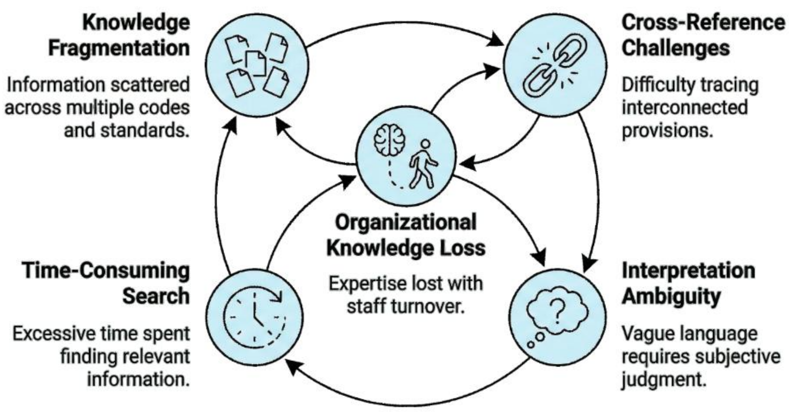


Figure 1: Architectural Code Consultation Pain Points Map.

1.1.3 The Emergence of Artificial Intelligence for Professional

Knowledge Work

Recent advances in artificial intelligence, especially in natural language processing, open new opportunities to tackle these longstanding challenges in professional knowledge management. Large Language Models (LLMs) (AI models on massive amounts of text, which can learn human language) are shown to be very good at understanding complex documents, generating information from multiple sources, and answering natural language queries. They are trained by self-supervised learning on massive unsupervised data and learn knowledge, rules, and thinking from the data, and achieve performance on questions like question-answering comparable to traditional deep neural network models. [10]. With the technological advancement of large language models such as ChatGPT, fine language models can be used for automatic review of technical specification documents. Such systems seem promising in professional fields as legal documents, medical literature consultations and technical documentation guidance which may be applied to architectural code consultation.

In particular, some recent work is in the literature on Retrieval Augmented Generation (RAG), which has an important role to take AI to specific knowledge. As Amazon Web Services defines it, “The task of Retrieval Augmentation is the task of improving a large language model, by pointing to an authoritative knowledge source besides its training data sources before generating a response."[11]. Unlike generic language models that merely rely on training information, RAG systems extract information from external documents and generate responses by working with specific, current information such as building codes. [12]. This addresses a major challenge for independent language models: knowledge cutoff times, and inability to access raw or updated information.RAG makes large language models more aware of information retrieval before being able to answer –in contrast to LLMs that generate answers from static training data, RAG takes text from databases, uploaded documents, or web sources. [13].

Some applications can extend beyond simple document search.It is recently demonstrated that LLMs can be exploited to read, infer and encode complex data for automatic compliance checks, which is promising for improved efficiency, accuracy, and scalability in architecture, engineering, and construction. [14]. Systems could be able to track cross-references automatically, synthesize requirements from multiple sources, identify conflicts between provisions, and even provide some preliminary interpretations based on published precedents. For engineering knowledge management, it has been demonstrated that tuned language models can be used to automatically process and query engineering documents and make access to very rich architectural knowledge intuitive in an intuitive question and answer form.[8]. Of course, this promise must come at the cost of past limitations and new assumptions. Soliman and Keim wrote: “LLMs have not been checked for answering questions about architectural knowledge and are uncertain of their accuracy."[15]. There are still questions about reliability, transparency, and role of AI in professional decision making where public safety is an issue.

Rationale for a Case Study Approach

Given the novelty of using LLM-based systems to consult infrastructure regulatory systems, a systematic and context-based study is needed before conclusions can be drawn. In this paper, we employ a case study approach in which we address this question through extensive empirical studies in a regulatory system in order to see current practices, the practitioner need and tests of technical approaches in a real-world context.

The case study provides a clear basis for exploring general questions about AI applicability to professional knowledge. While regulations may vary across the world, the fundamental challenges (fragmented information sources, cross-references, interpretation inferences, time complexity, organizational knowledge management) are common across various countries. Knowledge gained from careful analysis of one system could shed light on patterns and possibilities that may be found for the others and even differentiate context-specific results from broader transferable results.

1.2 Research Problem

The main challenge is to remain efficient in building code consultation for building developers and to explore whether emerging AI technologies can address this problem. Despite decades of digitization, architects still face the time-consuming and error-prone task of determining and interpreting the requirements that apply.

1.2.1 Current Inefficiencies in Building Code Consultation

Coding advice tools are changing from print to digital, but do not fundamentally change the user experience or address core inefficiency drivers. Architects need to keep updated on updated building code, since building codes can change regularly due to construction and safety[5]. Current methods require practitioners to query multiple fragmented sources independently across regulatory levels or standards, search for links and relationships in an automated way, interpret ambiguous languages without any consolidated precedents or interpretation guidance, maintain personal knowledge bases on updates, common situations, and jurisdictional differences, require people to have no tools for knowledge transfer or organizational learning.[7]Zhou et al. see in their review of automated compliance checking research “After decades of research, we have not made full progress on automated rule checking applications for building models for boosting productivity in AECO."[16]. While BIM-based compliance checking tools have already made significant progress on geometric rule checking for certain building blocks, the problem of intelligent code consultation for early design decisions (that designers must reason on context rather than pass/fail tests) is yet to be addressed.

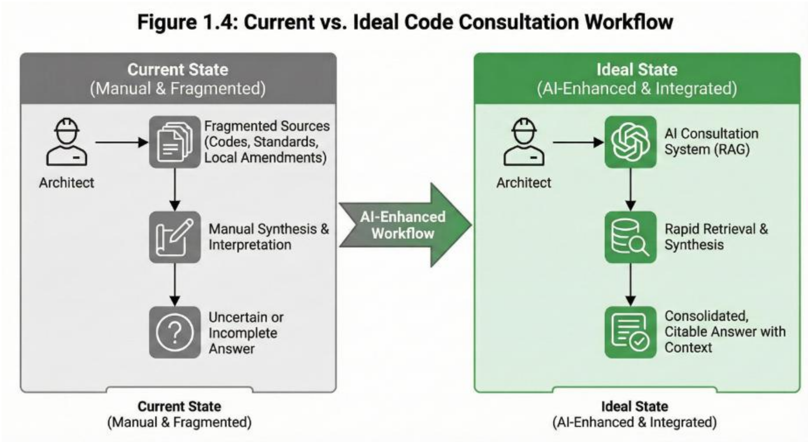


Figure 2: Current vs. Ideal Code Consultation Workflow..

1.2.2 The Gap Between Regulatory Knowledge Management and Practice

There is a gap between the way building code are published and maintained and how architects need to read and apply them in practice. Codes are published for broad coverage and legal precision, not for rapid consultation during design. In many applications, authors are expected to read whole documents and maintain complete knowledge, while practitioners are often required to search only for specific questions that arise by design decisions. Such a fundamental mismatch results in inefficiencies on several levels: at the individual level, a person searches for better meaning than design, while frequently having to interrupt each other to highlight a specific regulatory requirement; at the project level, regulatory uncertainty leads to long-time waiting, conservative decisions that do not rely on innovation, and late-stage revisions for unknown requirements; at a firm level, firms are unable to capture and share regulatory information, resulting in redundant code research and inconsistent interpretations across projects; and at the industry level, the time spent by hundreds of firms for redundant code study will result in a dramatic loss of productivity at a minimal value creation. [17]. As noted in architectural management literature, accurate and thorough documentation combined with efficient knowledge retrieval mechanisms are cornerstones of compliance, yet current systems fail to provide the integration necessary to bridge the gap between regulatory complexity and practice efficiency[4].

1.2.3 Opportunities Presented by Large Language Models

Large language models, as Allen, Stork and Groth show in knowledge engineering for LLMs, are "probabilistic models of natural language trained on very large corpora of content, which are learnt from the Web" with capabilities drawn from large-scale training data and neural networks.[18]. While LLMs are able to understand complicated text, maintain context in long documents, and produce consistent syntheses, they may be uniquely suited to code consultation issues. The potential is that they match the LLM capabilities with code consultation requirements: LLMs can understand questions in natural language (meaning intent, rather than the specific terminology itself, architecture should ask questions as they arise naturally in design thinking); rather than returning disconnected snippets, LLMs from several sections or documents may synthesize answers with relevant cross-references and requirements; chains of references may be traced; meaningful but not explicitly cross-referenced provisions could be traced, and multiple requirements interact to answer a design problem; and LLMs may provide preliminary interpretations in document context (either ambiguities or challenges to human decision-making) by suggesting what to do. [19]. RAG systems in particular offer a method to extend LLMs with specialized, current regulation content and keep explainable using source citations, which are very important in professional applications for accountability and verification. [20].

1.2.4 The Need for Empirical Evidence

Although the technical abilities of LLM systems are promising, there are still open questions that would be required to implement an architecture code consultation. Are current LLM based systems accurate and reliable enough for professional implementation in areas where public safety is of concern? How are the systems understand the specific structure and language of regulatory texts such as cross-references, exceptions and conditional requirements? Are they understood how failure modes and limitations have to be understood? How do these systems incorporate existing architectural workflows and decision processes? What is the impact on professional responsibility, liability, and oversight? Soliman and Keim note that "LLMs are not tested whether they can answer questions about architectural knowledge with accuracy."[15]. We close this gap with a systematic empirical study combining field work with technical evaluation, and provide evidence-based answers for both technology development and professional practice. We recognize that successful tools must be both efficient by computational performance, and professional acceptance for trust, verification, workflow integration, and responsibility.

1.3 COMPARATIVE ANALYSIS OF EUROPEAN AND CHINESE BUILDING REGULATORY SYSTEMS

1.3.1 Definitions: Chinese Building Regulatory System

The Chinese building regulatory system follows a government-led hierarchical structure where technical standards possess inherent legal authority.

1.3.1.1 Legal Framework

法律 (Laws): Legislation enacted by the National People's Congress or its Standing Committee. The *Construction Law of the People's Republic of China* (中华人民共和国建筑法, 2019 revision) and the *Standardization Law* (中华人民共和国标准化法, 2017) establish the fundamental legal framework.
行政法规 (Administrative Regulations): Regulations issued by the State Council, such as the *Regulations on Quality Management of Construction Projects* (建设工程质量管理条例, 2019 revision).
部门规章 (Departmental Rules): Rules issued by ministries, including the Ministry of Housing and Urban-Rural Development (MOHURD). Examples include the *Management Measures for Engineering Construction National Standards* (工程建设国家标准管理办法).

1.3.1.2 Technical Standards Framework

强制性标准 (Mandatory Standards): Standards designated as mandatory by government authority, requiring strict compliance. These include:

- **全文强制性工程建设规范 (Full-text Mandatory Engineering Construction Codes)**: New-generation codes where all provisions are mandatory (e.g., GB 55001-55037 series)
- **强制性条文 (Mandatory Provisions)**: Specific articles within otherwise recommendatory standards that must be strictly followed

推荐性标准 (Recommendatory Standards): Standards designated as GB/T (national recommendatory) that are voluntary unless referenced in mandatory requirements.
标准分类 (Standards Classification):

- **规范 (Guīfàn)**: Standards governing behavioral requirements and conditions
- **规程 (Guīchéng)**: Standards specifying operational procedures and methods
- **标准 (Biāozhǔn)**: Standards defining technical specifications and parameters

1.3.1.3 Multi-Level Standards System

Level	Designation	Authority	Scope
National	GB / GB/T	MOHURD, SAMR	Nationwide
Industry	JGJ, CJJ,	MOHURD, sector	Industry-specific

	etc.	ministries	
Local	DBJ, DB	Provincial/municipal governments	Regional
Association	T/CECS, etc.	Professional associations	Voluntary adoption

1.3.1.4 Governance Model

The Chinese system operates on a government-led model where:

- The state directly issues mandatory standards with inherent legal force
- Standards development is coordinated by government agencies
- Mandatory standards require no separate "adoption" process
- Local standards may impose stricter requirements than national standards

1.3.2 Conceptual Comparison and Non-Correspondences

1.3.2.1 Terminology Mapping

European Term	Chinese Term	Correspondence Type	Notes
Laws	法律	Direct	Both represent primary legislation
Regulations	行政法规/部门规章	Direct	Secondary legislation
Technical Regulations (WTO sense)	强制性标准	Functional	Both carry mandatory force
Standards/Norms	推荐性标准	Direct	Voluntary technical specifications
Codes	**— (No equivalent)**	**Absent**	See detailed analysis below

1.3.2.2 Core Conceptual Discrepancies

Discrepancy 1: The "Codes" Concept
The European/Anglo-American concept of "Codes" has no direct equivalent in the Chinese system:

Aspect	European "Codes"	Chinese System
Development	Non-governmental organizations	Government agencies

	(CEN, ICC)	(MOHURD)
Legal Status	Voluntary until adopted by government	Inherently mandatory (if designated 强制性)
Adoption Process	Requires governmental reference/adoption	No separate adoption needed
Example	Eurocodes (EN 1990-1999), IBC	GB 50016, GB 55001 series

In Europe and North America, "codes" come at a different level of government, not necessarily voluntary. They are developed by experts but have to go into government for them to be enforced. But in China, it is not yet in place, and standards are simply issued by government as soon as they become enforced.

Discrepancy 2: Dual Nature of Chinese "Standards"

Characteristic	European Standards	Chinese 标准
Default Status	Voluntary	Depends on designation (GB vs. GB/T)
Mandatory Force	Only when referenced in regulations	GB standards inherently mandatory
Terminology	"Standards" = voluntary	"标准" includes both mandatory and voluntary

Under WTO/TBT Agreement definitions, standards are voluntary documents and technical regulations must be needed. Chinese GB (强制性国家标准) standards are Technical Regulations in WTO but are domestically referred to as standards.

Discrepancy 3: 规范 (Guīfàn) ≠ Norms

Despite common translation practices:

Concept	European "Norms"	Chinese "规范"
Status	Independent category, interchangeable with "Standards"	Subcategory within the standards system
Meaning	Technical specification document	Standards governing behavior/conditions
Governance	Developed by CEN/national bodies	Developed under government coordination

Therein, a Chinese term 规范 refers to standards defining behavior or conditions. This is a distinction between 规程 (traditional standards) and 标准 (specification standards). This internal term is of no interest in European terms.

Discrepancy 4: Governance Model Differences

Dimension	European Model	Chinese Model
Standards Development	Consensus-based (industry + government)	Government-led (with expert participation)
Mandatory Mechanism	Reference in legislation	Direct designation
Flexibility	National Annexes allow	Local standards may be

	adaptation	stricter
Update Cycle	Varies by standard	Systematic review every 5 years

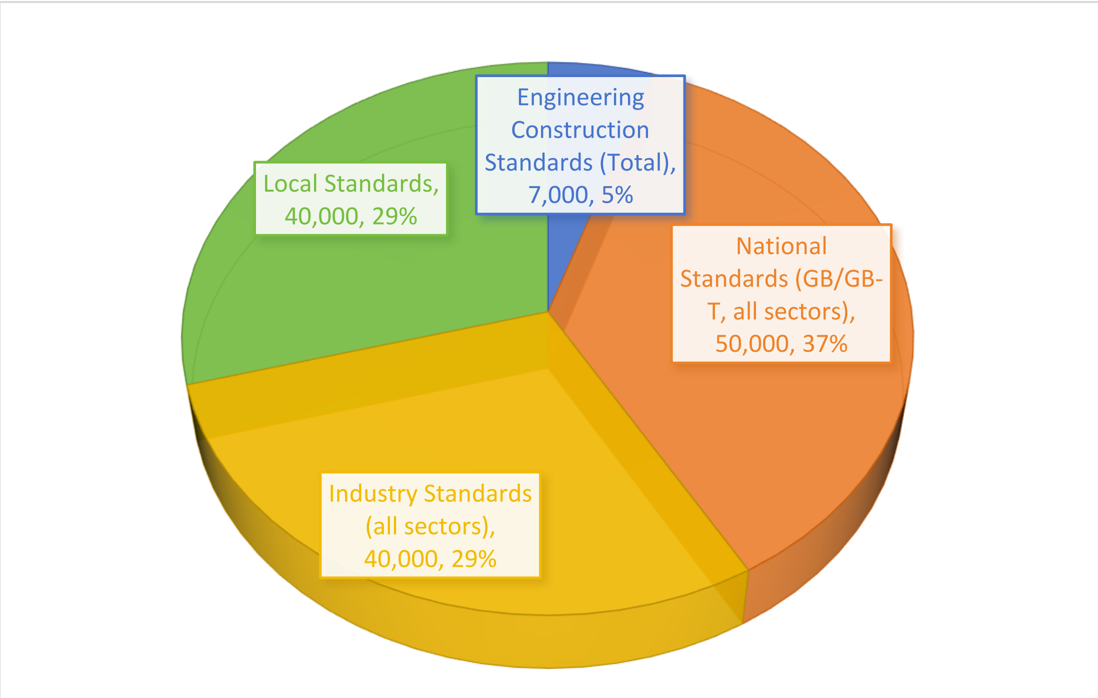
1.3.3 Quantitative Overview of Regulatory Systems

1.3.3.1 Chinese Building Standards Inventory

Overall Standards System

Category	**Quantity**	**Source**
Engineering Construction Standards (Total)	~7,000+	MOHURD estimates
National Standards (GB/GB-T, all sectors)	50,000+	SAMR database
Industry Standards (all sectors)	40,000+	National standards platform
Local Standards	40,000+	Provincial databases

Sources: National Standards Information Public Service Platform; MOHURD

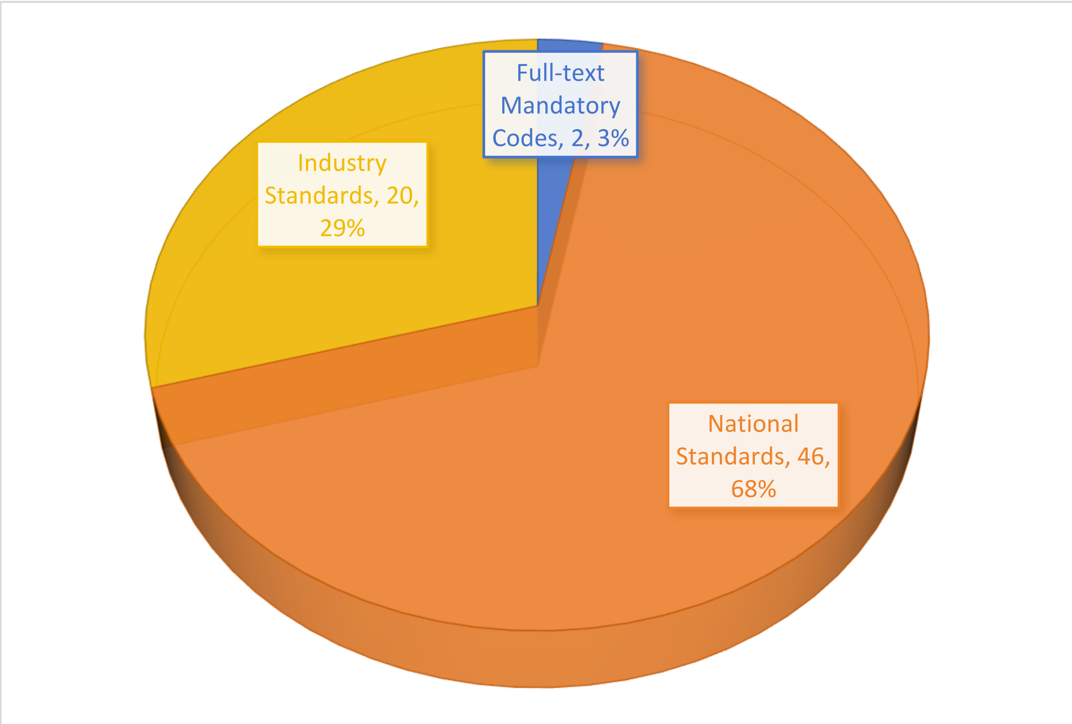


Construction Drawing Review Standards
According to the MOHURD *Technical Review Guidelines for Construction Drawing Design Documents*, approximately **160 engineering construction standards** (excluding local standards) are relevant to building construction drawing review, containing approximately **1,500 mandatory provisions**.

Fire Protection Standards System

Type	Quantity	Examples
Full-text Mandatory Codes	2	GB 55036-2022 (Fire Facilities), GB 55037-2022 (Building Fire Protection)
National Standards	46+	GB 50016-2014 (Building Design Fire Code), GB 50116-2013 (Fire Alarm Systems)
Industry Standards	20+	Various specialized fire standards
Total Fire-related	**~70+**	

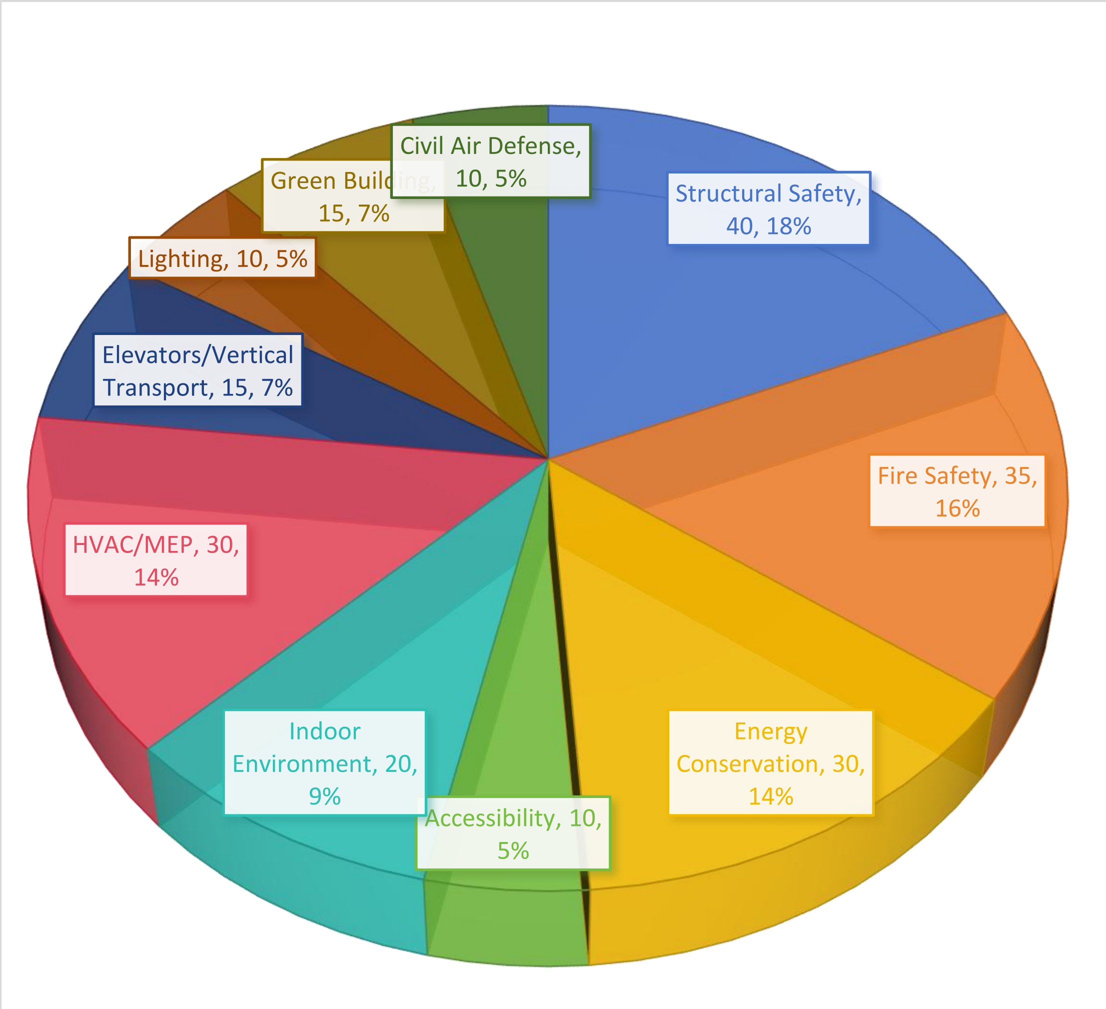
Source: China Fire Protection Association; MOHURD announcements



Building Design Standards by Professional Domain

Domain	Approximate Standards Count	Key Standards
Structural Safety	40+	GB 50010 (Concrete), GB 50011 (Seismic), GB 50017 (Steel)
Fire Safety	35+	GB 50016, GB 55037, GB 50084 (Sprinkler)
Energy Conservation	30+	GB 50189 (Public Buildings), GB 55015 (Energy Code)
Accessibility	10+	GB 50763, GB 55019
Indoor Environment	20+	GB 50325 (Air Quality), GB 50118 (Acoustics)
HVAC/MEP	30+	GB 50019 (HVAC), GB 50015

		(Water Supply/Drainage)
Elevators/Vertical Transport	15+	GB 50096 (Residential), various elevator standards
Lighting	10+	GB 50034 (Lighting Design)
Green Building	15+	GB/T 50378, GB 55015
Civil Air Defense	10+	GB 50038, various regional standards



1.3.4 Multi-Level Application of Building Regulations

Building regulations in China operate across multiple application levels, each with distinct regulatory instruments:

1.3.4.1 Application Levels

Level	Regulatory Instruments	Examples
Urban Planning Level	Urban planning regulations, zoning	GB 50137 (Urban Land Classification), local zoning

	codes	regulations
Building Level	Building design standards, construction codes	GB 50352 (Civil Building Design), GB 55031 (Civil Building Code)
Local/Regional Level	Provincial and municipal standards	DBJ (provincial), various municipal supplements
Specialized Domains	Domain-specific technical standards	See table below

1.3.4.2 Specialized Domain Standards

Domain	Governing Standards	Mandatory Provisions
Structural Safety	GB 50010, GB 50011, GB 55002, GB 55003	Full-text mandatory for GB 55xxx series
Fire Safety	GB 50016, GB 55037, GB 55036	200+ mandatory clauses
Energy Performance	GB 50189, GB 55015	Full-text mandatory for GB 55015
Accessibility	GB 50763, GB 55019	Full-text mandatory for GB 55019
Acoustics	GB 50118, GB 50096	Specific mandatory clauses
Lighting/Daylighting	GB 50033, GB 50034	Specific mandatory clauses
Indoor Air Quality	GB 50325, GB/T 18883	Specific mandatory clauses
Civil Air Defense	GB 50038, regional standards	Mandatory for applicable buildings

1.3.5 The Relevance of Language in Comparative Analysis

1.3.5.1 Translation Challenges and Semantic Shifts

Language plays a critical role in comparative studies of building regulatory systems. The translation of technical terminology between European languages and Chinese introduces inherent challenges that directly affect research outcomes.

Term	Common Translation	Actual Correspondence	Risk of Misinterpretation
Codes	规范	No direct equivalent	High — implies equivalent regulatory tier
Norms	规范	Partial	Medium — obscures governance differences
Standards	标准	Partial	Medium — ignores

			mandatory/voluntary distinction
Regulations	法规	Direct	Low

The identical Chinese translation "规范" for both "Codes" and "Norms" creates particular confusion, as these terms carry distinct meanings in their original contexts but collapse into a single concept when rendered in Chinese.

1.3.5.2 Impact on Research Results

The linguistic dimension affects comparative research in several ways:
(1) False Equivalence
This could be a matter of assuming two things is the same, e.g. If Chinese “Building Code” translates into “建筑规范”, the Chinese ”Codes” should be interpreted as “Codes is exactly what Anglo-American “codes“ should be.

(2) Obscured Systemic Differences
Language comparisons might imply underlying differences in governance. While the European model of voluntary standards that become mandatory via regulation differs significantly from the Chinese model of inherently mandatory standards, both may be translated.

(3) Implications for Practice
Professionals engaged in cross-border projects must recognize that terminological similarity does not guarantee procedural or legal equivalence. Compliance with "规范" in China involves different mechanisms than compliance with "Codes" in Europe or the United States.

1.3.5.3 Recommendations for Terminology Use

To enhance clarity in comparative studies, this thesis adopts the following conventions:

Original Term	**Recommended Approach**
European Codes	Retain as "Codes" with explanation; avoid direct translation as "规范"
European Norms/Standards	Use "Norms/Standards" interchangeably; note voluntary nature
Chinese 强制性标准	Translate as "Mandatory Standards" or "Technical Regulations (WTO sense)"
Chinese 规范	Retain pinyin "Guīfàn" or use "Chinese Codes" with clarification

Researchers should explicitly acknowledge that terminology carries embedded assumptions about regulatory structures, and that cross-linguistic comparison requires careful contextualization rather than direct lexical mapping.

1.3.6 Summary

The European and Chinese building regulatory systems differ fundamentally in terminology, governance structure, and regulatory mechanisms:

1. **The "Codes" concept exists in Europe but is absent in China**. European codes are developed by non-governmental organizations and must be required by government agencies. In China, mandatory standards are issued directly by government authorities of pure legal force and do not require a separate "codes".
2. **Chinese "Standards" include both mandatory and voluntary categories**, All of European standards/norms, however, are voluntary in their own right and must be found by a regulation. Chinese GB standards are "Technical Regulations" under WTO law.
3. **Both systems encompass extensive regulatory inventories**. European engineering construction standards include 58 Part Eurocodes plus thousands of EN standards. Chinese engineering construction rules cover over 70000 items, containing around 160 standards and ~1,500 required mandatory provisions that apply to building construction drawing reviews.
4. **A single building project in China may involve 75+ applicable standards and 500+ mandatory provisions** spanning structural safety, fire protection, energy conservation, accessibility, acoustics, lighting, and other specialized domains. The regulatory review process involves multiple parallel review tracks (construction drawing review, fire protection review, civil air defense review) with complex standard interrelationships.
5. **Language significantly influences comparative analysis**, as translation between European languages and Chinese may create false equivalences or obscure systemic differences. The term "Codes" has no direct Chinese equivalent, and the identical Chinese rendering "规范" for both "Codes" and "Norms" masks fundamental conceptual distinctions. Careful terminological treatment is essential for accurate cross-cultural regulatory comparison.

1.4 Research Questions

This research is guided by one primary question and four supporting secondary questions:

1.4.1 Primary Research Question

What are the specific pain points architects face when consulting building regulations?

This question considers technical performance (accuracy, completeness, speed), practical usability (integration with workflows, learning curve, reliability), and professional acceptance (trust, verification, liability). The question is posed at the practice level, rather than technical level, because successful tools should satisfy computational and human factors requirements.

1.4.2 Secondary Research Questions

RQ1: Can Large Language Model-based systems significantly improve the efficiency and accuracy of building code consultation in architectural practice?

This question may be examined empirically: where inefficiency occurs, which queries are most common, which problems are in current tools, what workarounds practitioners have come up with. Understanding the problem space in detail may aid in the assessment whether or not solutions are solving the real problem.

RQ2: How do existing general-purpose AI tools perform in architectural regulatory contexts?

General-purpose AI tools like ChatGPT have become readily available to architecture experts, yet they are not well tested for building code answering. We have to technical test existing AI tools for building codes answering queries, see what strengths and weaknesses are known, failure (often hallucinating missing provision), and performance, to understand when domain-specific adaptation is necessary, and what improvements are needed.

RQ3: What modifications are necessary to adapt AI systems for specialized architectural knowledge?

On top of performance, we address the issue of targeted improvements for overcoming certain limitations that include document preprocessing, retrieval optimization, prompt engineering, and output formatting to better fit architectural use cases.

RQ4: What are the broader implications for professional practice and architectural education?

Besides technical advantages, we also ask whether AI-assisted code consultation can change architecture workflows, how junior and senior practitioners are shared, how innovation and risk take place and how professional responsibility and liability may be questioned in this era of AI augmentation.

1.5 Research Objectives

To address these research questions, this thesis pursues five specific objectives:

1.5.1 Objective 1: Analyze the Current State of Building Code Navigation in Architectural Practice

Through qualitative field research with multiple architecture firms representing different scales and types of practice, this objective seeks to:

- Document current workflows for code consultation across project phases
- Identify specific pain points, inefficiencies, and workarounds in existing approaches
- Understand how practitioners manage organizational knowledge and expertise
- Characterize the types and frequency of code queries arising in typical practice
- Assess current awareness and experience with AI tools, if any

This empirical foundation ensures that subsequent technical work addresses real practice needs rather than assumed requirements.

1.5.2 Objective 2: Evaluate the Applicability of Large Language Models to Regulatory Consultation

Large language models might help fill adoption gap by facilitating the creation of architectural documentation. This is needed to provide baseline performance of current LLM based approaches (RAG frameworks) for building code consultation:

- Test system accuracy on representative code queries across difficulty levels
- Identify failure modes specific to regulatory text characteristics
- Assess completeness of responses (full requirement identification vs. partial)
- Evaluate explainability and source citation quality
- Measure performance metrics relevant to practice (response time, consistency)

This objective provides empirical evidence on current capabilities and limitations, moving beyond theoretical potential to demonstrated performance.

1.5.3 Objective 3: Identify Industry Needs Through Field Research with Design Firms

Going beyond documenting current pain points, this objective seeks to understand practitioner requirements and expectations for future tools:

- What features would provide the highest value in daily practice?
- How should AI-generated responses be presented for professional review?
- What verification and oversight mechanisms are necessary for trust?
- How should such tools integrate with existing software ecosystems (CAD, BIM, project management)?
- What are acceptable tradeoffs between different performance dimensions?

This objective ensures that system development prioritizes features aligned with actual user needs and acceptance criteria.

1.5.4 Objective 4: Propose and Test Improvements to Existing AI Frameworks for Architectural Applications

Informed by both technical evaluation (Objective 2) and user needs (Objective 3), this objective develops targeted improvements to baseline systems:

- Adapt document preprocessing for regulatory text structure
- Optimize retrieval strategies for cross-references and regulatory hierarchies
- Enhance response generation for professional communication norms
- Implement verification and transparency features for professional oversight

The improvements are tested through a prototype system, demonstrating feasibility and measuring performance gains relative to baseline approaches.

1.5.5 Objective 5: Derive Insights Transferable to Other Regulatory Contexts

While grounded in a specific case study, this research seeks to identify patterns, principles, and approaches with applicability beyond the immediate empirical context:

- What are universal challenges in professional regulatory consultation versus context-specific factors?
- Which technical approaches show promise for other professional domains with similar knowledge work patterns?
- What organizational and workflow considerations apply across different regulatory frameworks?
- How do findings inform broader questions about AI augmentation of professional expertise?

This objective ensures that the research contributes to both immediate practical innovation and longer-term theoretical understanding of AI in professional practice.

1.6 Scope and Limitations

1.6.1 Research Scope

Empirical Focus: This research follow the practical example of a case study based on in-depth research within one national government system. More complete is the method, which uses for depth and allows for detailed observation and analysis that may not be done with more shallow sampling.

Practice Scales Examined: There are four architecture firms across the practice-up - large with hundreds of employees, small with small dedicated studio, and advanced technology-forward practices - to which context (firm size, resources, project types, organization culture) can play an important role in solving problems and solutions.

Functional Focus: There are four architecture companies: four for their own reasons. The one that we focus on in detail is the code consultation—the search of and knowledge of required information to guide design decisions, not the complete code compliance checking: proof of full designs against all applicable codes. That's a more frequent, design-based activity, and the latter is a later validation activity.

Technical Scope: The research consider RAG based implementations of information retrieval and language model generation. It is not a survey of all possible AI-based code consultation approaches, but an extensive analysis of one potential technical direction. RAG can be used to increase LLM output, so it is relevant, accurate, and useful in different settings.

Query Types: The research addresses textual code queries—questions about requirements, applicability, definitions, and interpretations. It does not address geometric compliance checking (e.g., verifying 3D models against dimensional requirements) or automated drawing analysis, which represent complementary but distinct technical challenges.

1.6.2 Research Limitations

Sample Size: The field research involves four firms, providing rich qualitative data but limited statistical generalizability. Findings represent patterns observed across these cases but may not capture the full diversity of practice approaches across larger populations.

Geographic and Temporal Constraints: The empirical work is conducted in a specific geographic region during a particular time period. Practices, code versions, and technology adoption may differ in other locations and times.

Prototype vs. Production: The technical work develops and tests a prototype system demonstrating feasibility and measuring performance on test cases. A production-ready system would require additional engineering for scale, reliability, security, and integration that are beyond this research scope.

Regulatory Context Specificity: While transferability is discussed, and comparison to other systems is worth considering, technical details and some results are specific to the case study. Performance-based building regulations have plenty of opportunity and may not provide safe spaces for those who will make new construction techniques not defined in building regulations. Direct application to other building regulations would need to be tested and adapted.**Evaluation Constraints:** System evaluation relies on test cases and expert judgment rather than long-term deployment with real projects. The true impact on practice efficiency, design quality, and error rates would only be measurable through extended real-world use.

Interdisciplinary Boundaries: This work is based on architectures as well as computer science and is mainly based on the architectural analysis. Deep technical applications to machine learning or natural language processing are not considered, but, rather, current technical solutions to a new application set are applied

.

1.6.3 Delimitations

Certain questions, while related, are intentionally excluded from the research scope:

Code Authoring and Maintenance: The research addresses code *consultation* by practitioners, not the processes by which regulatory authorities develop, update, and maintain codes. Questions about optimal code structure, formatting, or publication methods are outside the scope.

Legal and Liability Framework: While the work considers the implications of the professional responsibility, it is not legal analysis on the liability for AI-assisted professional services. The questions of professional liability law from jurisdictions and circumstances.

Comprehensive BIM Integration: Although we can consider integrating to Building Information Modeling, the full technical integration is not possible. All our attention will be on the part of the core code consultation function.

Comparative Regulatory Analysis: While international comparisons inform the research and transferability is discussed, comprehensive comparative analysis of different national code systems is not a primary objective.

1.7 Thesis Structure

This thesis is organized into seven chapters:

Chapter 2: Theoretical Framework and Literature Review establishes the conceptual foundation, reviewing relevant literature on building regulation systems and their challenges, knowledge management in architectural practice, fundamentals of large language models and RAG architectures, and identification of research gaps at the intersection of these domains.

Chapter 3: Research Methodology details the research design, including the case study approach rationale, field research protocols (firm selection, interview methods, data analysis), technical experimental design (baseline system evaluation, improvement strategies, prototype development), and evaluation frameworks for both empirical and technical components.

Chapter 4: Findings—Current State of Practice presents results from field research, documenting current code consultation workflows, identified pain points and inefficiencies, existing tool usage and satisfaction, AI awareness and adoption where present, and practitioner expectations for future systems. This chapter grounds subsequent technical work in observed practice realities.

Chapter 5: Technical Analysis and Improvement Proposals reports technical research findings, including baseline RAG system performance on architectural code queries, root cause analysis of limitations, proposed improvements informed by field research needs, prototype system design and implementation, and comparative evaluation results.

Chapter 6: Discussion and Implications synthesizes findings, directly addressing the research questions posed in Chapter 1. It discusses implications for architectural practice (workflow integration, professional responsibility, education impacts), examines transferability to other regulatory contexts, situates findings within broader questions about AI in professional work, and provides critical reflection on benefits, risks, and open questions.

Chapter 7: Conclusions and Future Research summarizes research contributions (empirical, technical, and practical), acknowledges limitations, and proposes directions for future investigation, both short-term refinements and longer-term research trajectories.

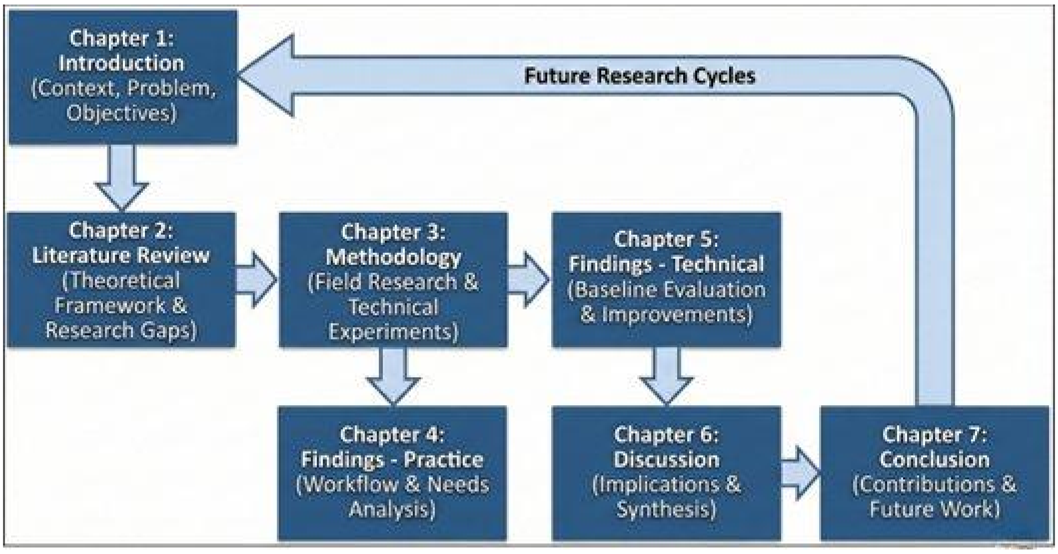


Figure 3: Thesis Structure and Research Roadmap.

CHAPTER 2 THEORETICAL FRAMEWORK AND LITERATURE REVIEW

This chapter establishes the theoretical foundation for investigating AI-enhanced building code consultation in architectural practice. The review proceeds through three interconnected domains: first, the structure and challenges of building regulation systems, with particular attention to multi-level regulatory frameworks; second, knowledge management practices in architectural practice, including the persistent challenge of capturing and transferring professional expertise; and third, the emergence of Large Language Models and Retrieval-Augmented Generation systems as potential solutions to knowledge-intensive professional tasks. By synthesizing literature from these distinct but related fields, this chapter identifies the research gap that motivates the empirical and technical investigation described in subsequent chapters.

2.1 Building Regulation Systems: Structure and Challenges

2.1.1 Multi-Level Regulatory Frameworks: Universal Patterns and Variations

Building management systems around the globe share fundamental structural properties but exhibit a lot of variation in implementations. At the same time, they are hierarchical for the sake of national consistency with local adaptation, public safety and innovation support, as well as standards of building performance which permit performance flexibility. Modern construction is governed by codes and technical standards regarding structure integrity, electrical safety, fire protection, and many other building performance aspects. [21]. This hierarchy often involves national or federal standards that define baseline requirements across jurisdictions, sector-specific standards regarding specific building types or technical systems, and local or local codes providing geographic adaptation of climate, seismicity, material and local construction. [22].

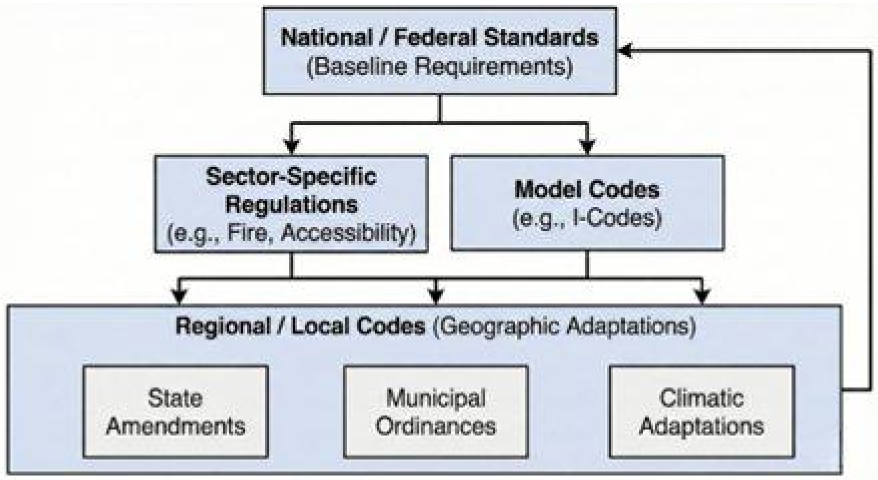


Figure 4: Multi-Level Regulatory Framework.

The European Union is one example of multi-level harmonization: it puts all member states in a control and allows national annexes to be placed. [23]. The Eurocodes represent pan-European building codes that have superseded older national codes, with each country maintaining National Annexes to adapt provisions to local conditions[24]. This system enjoys standards of inclusion (e.g., cross border practice, economies of scale in product manufacturing and knowledge transfer) but is recognizing that geographic factors are important and do not necessarily matter for purposes. North American systems show a different model: the US uses model codes developed by non-governmental organizations (primarily the I-Codes of International Code Council), in which states and local governments may choose, adapt or reject them. [25]. This approach provides maximum local flexibility but creates complex patchworks of varying requirements that challenge multi-jurisdictional practitioners[26].

Performance-based regulation is based on an independent design of functional goals (rather than based on the individual solutions) that explicitly enables different compliance steps based on performance-based engineering analysis. [27]. Australia's National Construction Code both includes “Deemed-to-Satisfy” rules as well as performance based solutions, each of the eight state and territorial states sign an Intergovernmental Agreement to adopt the national code and leave authority for some variation27. This dual strategy acknowledges that in practice, the text in a regulatory text should not predict any of the design problems, especially for novel or new buildings, but does not guarantee clear compliance paths for standard buildings. New Zealand's building code also supports performance goals that are guided by "Acceptable Solutions" that provide the required compliance paths in standard environments[27].

Several problems are identified in terms of the different authorities. Inter-jurisdictional Regulatory Collaboration Committee (Coordinating authorities in several countries) of regulators from across countries find collusion between different regulatory levels, maintaining currency as knowledge advances, verifying alternative compliance strategies, and enforcing consistency consistent regardless of the country. [28]. These universal challenges are characterized by tensions in regulation: intractability vs. usability, specificity vs. adaptability, innovation encouragement vs. safety conservatism, international harmonization vs. local response. The architecture community must overcome these tensions while meeting project-specific requirements in time and budget.

2.1.2 Conflicts and Coordination Issues in Multi-Level Systems

Having more than one regulator poses some coordination problems that practitioners face everyday. Vertical conflicts arise when provisions at different levels are contradictory or conflicting. For example, a national energy efficiency standard may indicate minimum insulation levels, or a local area mandates different values based on the regional climate data. [29]. Finding what is a requirement and where is not a requirement: the technical aspects, and the legal hierarchy and precedence rules, depending on jurisdiction. This conflicts often arise in code transition periods when different documents in the chain have changed, resulting in temporary inconsistencies that practitioners must resolve by interpretation or authority consultation. [30].

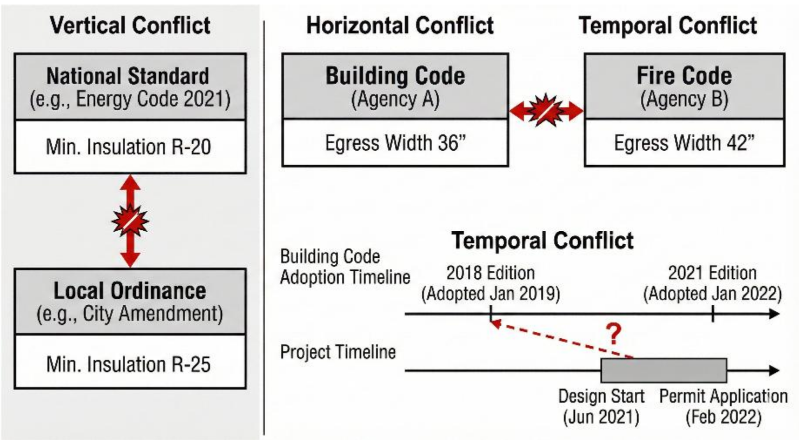


Figure 5: Diagram of Regulatory Conflict Types.

Horizontal conflict emerges when several authorities compete for authority over the same building. Fire safety regulations may come from building codes or separate fire codes (and may be managed by different agencies with different review processes) [31]. Accessibility requirements for building codes and disability rights legislation enforced by wholly different authorities31 Environmental performance standards for building code, energy codes, and environmental protection regulations with maybe different metrics, compliance paths and enforcement mechanisms [32]. Navigating these overlapping jurisdictions requires practitioners to identify all applicable regulatory instruments—itself a non-trivial task—and then resolve any conflicts or ambiguities that emerge from their interaction.

Temporal coordinating. Building codes update with three-year cycles, but not all provisions update at once, and different countries update in different times31. A project started at one code edition may have permits under a different code edition if the jurisdiction updates its code at some point in the project timeline. Referenced standards—technical documents that are used in codes by reference—may update with their own schedules and sometimes mislead us about which version applies. [33]. Long-term projects may be of multiple codes, and the practitioner must track what needs are required at different phases of the project and when the code changes at mid-project. This occurs as projects run out of time, when existing buildings are remodeled or remodeled in such a way that parts of the structure may be dictated by the code at project start, and new building must maintain current requirements, and one must carefully discern what conditions will be maintained or which need be remodeled.

Automatic compliance checking is also understood as the challenge for fully automated implementation. According to Zhou et al., regulatory texts have subjective requirements which challenge the practical adoption of automatic compliance checking[34], A problem where several instruments coordinate is an intractable one for rule-based systems. Due to the interplay of regulations (one design decision may result in considerations in multiple code sections, reference standards, and jurisdictional layers), this issue can create a knowledge navigation problem that simple keyword search or even complex geometric checking cannot solve.

2.1.3 International Comparative Perspective and Lessons for Technology

Further studies of different international ways of building control reveal different approaches of dealing with complexity and informing new technologies. Performance systems, like Australia, New Zealand and others, are less interested in giving specific solutions than in providing functional results that must be achieved. This might lead to innovation by allowing an intuitive solution which meets performance goals, but will lead to difficulties for practitioners and regulators in providing sufficient evidence of performance and assessing new approaches[35]. The Inter-jurisdictional Regulatory Collaboration Committee notes that "although the Summit did not profess to answer all" questions about performance-based regulation, consensus exists on core principles while legal practices for implementation vary significantly across member nations[28].

Different jurisdictions have implemented different policies for keeping a regulatory currency and access. Some have centralized government-provided code development and maintenance and universal adoption of each new version27, others use non-governmental consensus-based development and implementation which are voluntary, providing diversity and possibly greater industry ownership26 and some consider technologies such as machine readable codes, application programming interfaces for online code access, and compliance checking tools. [36]. Each approach differs from other governments, industry structures, and technology and indicates that technology must deal with a major change in how the regulators operate and maintain.

There are several potential implications for technology technologies such as AI consultation systems for such international systems. Firstly, successful systems need to work in a fundamentally different regulatory framework from prescriptive to performance-based, centralized to distributed, single-document to multi-instrument. Second, they need to address inherent ambiguity and conflicts that human experts have to deal with, implying AI tools should be better developed and not attempt to replace expertise. Third, time-varying regulatory change requires systems that handle multiple versions of the same code and track the applicability according to project time. Fourth, the ongoing challenges found in various international systems suggest that technology cannot solve all regulatory challenges and should work within current professional and institutional systems. [37].

2.2 Knowledge Management in Architectural Practice

2.2.1 Traditional Methods of Code Consultation and Their Limitations

Architectural practice typically relies on several overlapping approaches to manage knowledge of the regulations, which, in particular, have limitations on how effectively practitioners can handle code compliance. Manual reference of printed codebooks remains very common despite digitization for experienced practitioners familiar with physical document navigation1. This is tactile, allows rapid cross-referencing by bookmarking and annotation, but has obvious limitations: codes become obsolete when new amendments are issued, searching across a number of books is time-consuming, and linking provisions from several codes requires several collections of documents with storage and update costs. [38].

Internal knowledge is perhaps the most valuable resource in well-known firms. Management workers have mental models of rules, know background for difficult situations, work with code officials that allow interpretation, and accumulate firm knowledge about jurisdictional changes or common circumstances. [39]. However, this knowledge remains largely tacit—difficult to articulate explicitly and resistant to systematic capture[40]. As we saw in the literature on tacit knowledge in architecture, engineering and construction industry, "very much knowledge in AEC industry is experience-based and tacit, but it's typical for knowledge management that computer-based approaches capture and share explicit knowledge". [41]. This mismatch between the nature of expertise and the methods employed to manage it creates persistent knowledge transfer challenges, particularly when experienced practitioners retire or change firms[42].

Consulting experts and regulatory agencies are another classical practice: practitioners contact building officials for guidance, code experts for complicated cases, and professional association committees that clarify code. It can be useful, but it is not scalable: the authority cannot provide guidance for all practitioners for each question, the consultants add project costs and committee processes are often timescales unsuitable for answering queries during active design. Also, reliance on external experts for routine questions can block junior practitioners from developing their regulatory competence, creating open questions as experienced practitioners retire without passing on knowledge to a younger practitioner.

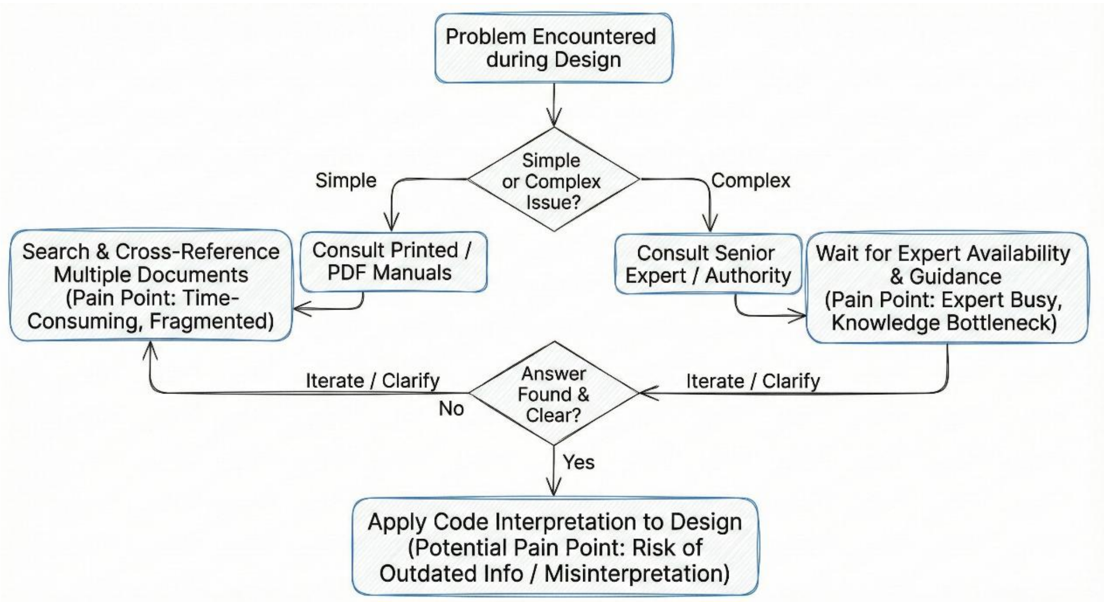


Figure 6: Flowchart of Traditional Code Consultation Process.

These limitations manifest themselves in measurable inefficiencies. Time consumption is the single most obvious cost: architects spend a lot of time doing code research than design. Knowledge fragmentation implies a redundancy: multiple practitioners across firms work on the same regulation issues, and they benefit from the same information. Update delay: busy practitioners may not keep track of changes and revisions, and may even have outdated information. The most worrisome aspect, that full regulatory review is slow: requirements might only be aware during plan review, or even after construction when corrections are increased. [43].

2.2.2 Digital Tools in Current Use and Persistent Gaps

Digitizing building codes first promise to address many of the challenges in print-based consultation and, indeed, PDF-based code databases and keyword search systems improve upon purely manual approaches. New versions or amendments are easily read, keyword search allows certain words to be quickly located across large document sets, hyperlinks connect cross-references required by hand page-flipping to book pages, and storage costs are nearly zero compared with storing large code libraries. However, as architectural technology literature states, "While PDF- based code databases & keyword search provides a general improvement on print-only manuals, they still face several fundamental issues hampering professional practice"[4].

Keyword matching, the main search approach in current digital code tools, fails for many architectural queries. Typical users search by design-domain terms: residential fire escape, accessible entrance, lighting conditions, and codes that may use very different terms—epic, accessible entry, natural light and ventilation. [44]. Even when the words are equivalent, keyword search returns snippets matching queries without knowing semantic relations or context factors. For example, search for "stair width" will return several provisions that deal with stairs, but determining which of those apply to a project needs occupancy information, building heights, sprinkler condition, and a variety of other factors that keyword search cannot assess. [45].

More sophisticated digital tools are also available for building information modeling, and such compliance checking applications can automatically check geometrical requirements (dimensional criteria, clearances, spatial relations) from 3D building models directly, say the doors widths meet accessibility requirements, minimum room size, or egress travel distances. However, as many research works on automatic compliance checking shows, "follow decades of research, there is still a lack of confidence in the possibility of automatic rule checking applications of building models for enhancing productivity of AECO companies"[16]. These tools check objective, quantitative rules, but they lack qualitative qualities, context, and the early stage design questions that most often require a code review. A schematic designer asking "What occupancy classification should I use for this building program? or do I need fire-rated building here?" can't use geometrical checking tools that assume a well-designed design to check.[34].

The problem now, then, is to support knowledge intensive, contextual and interpretative aspects of code consultation that are still human activities. [46]. Practitioners need to know which requirements apply to them, how several requirements interact, understanding ambiguous or subjective language, and applying the intention to new designs. The current digital tools provide code text and verify certain geometric requirements, but they do not give support for such higher level cognitive tasks. [47]. This gap explains why experienced practitioners remain essential and why regulatory expertise continues to carry significant professional value—the existing technological toolkit addresses only a portion of the actual work involved in code compliance.

2.2.3 The Role of Tacit Knowledge and the Challenge of Codification

The difficulty in transferring regulatory knowledge relates to general challenges in working with managers such as tacit knowledge, the innate intuitive knowledge and experience that practitioners use in an unconsciously and not at all articulate way. [48]. The literature on tacit knowledge for architecture states that “this is a type of knowledge architects use when designing and it is reflected by the material vectors they design with.” In regulatory work, tacit knowledge is pattern recognition skills that practitioners develop: automatically knowing whether a building program will provide specific code, intuitively knowing whether there is a jurisdictional interpretation in relation with code literal reading, recognizing if a design will be censored even if the design is technically compliant, and aware of the relationships between requirements. [49].

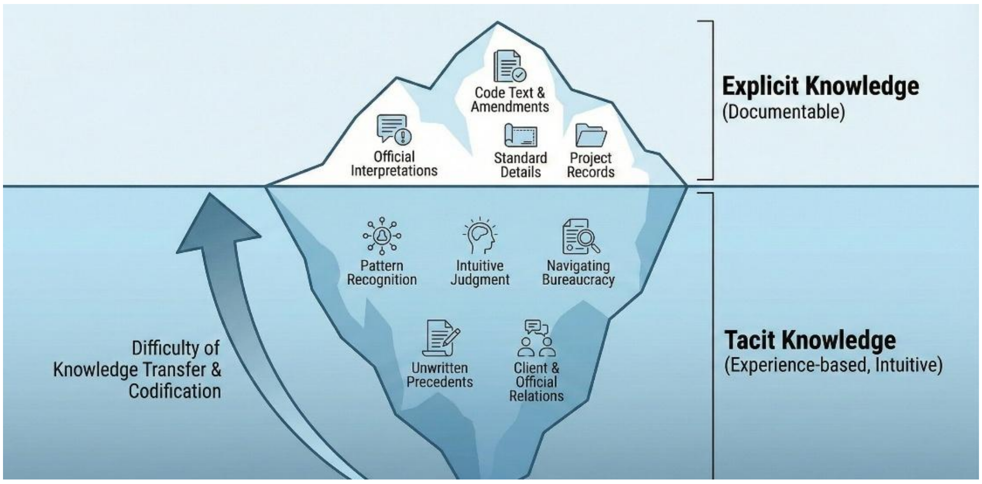


Figure 7: Iceberg Model of Explicit vs. Tacit Knowledge.

This tacit level of knowledge explains why efforts to report and share regulatory knowledge sometimes fail. Organizations try to capture “best practices” or “lessons learned”, but as knowledge management literature has pointed out, that often result in a few missing collections. Recordable knowledge represents decontextualized facts or rules (rather than the situational knowledge and judgement required to apply them appropriately). As one knowledge transfer study notes, “Despite a long history of failed attempts at "capturing" knowledge, we just cannot seem to get over the fact that someone's mind is a file cabinet where they could just put something off and hand it to someone else".[42]. There is a much more complicated picture to the reality of expertise, as there are facts and some common knowledge that experts have learned over a number of problems in different situations.

Psychology research has identified several factors that make it beneficial to share tacit knowledge. Personal interaction and observation are more effective than document-based transfer – in-person workers learn regulatory skills by working with other members of the community, observing how they look at code questions, and meeting building officials. The practice community of practitioners, where practitioners with common interests frequently engage to communicate and problem solve together, enables continued knowledge development which formal training cannot do.[42]. Apprenticeship-style mentoring, extended over time to span multiple project cycles and varied scenarios, allows novices to build the pattern-recognition abilities and contextual judgment that characterize expertise[48].

However, these effective transfer procedures are slow and time consuming, which is scarce for profitable professional practice50, and do not scale well: each expert is mentoring only a small number of students, practice communities work in small numbers than industry scale, apprenticeship models take the practitioner until they become independent.[42]. Can technology, especially AI systems that understand context and give context-based guidance, be replaced or partially supplanted by person-to-person knowledge? Can they incorporate not only facts code but some aspects of the thought processes and contextual awareness that are only available in experienced practitioners?

2.3 Artificial Intelligence and Large Language Models: Capabilities and Applications

2.3.1 The Evolution and Characteristics of Large Language Models

Large Language Models are important extensions of artificial intelligence's ability to learn and generate human language. Allen, Stork and Groth point out that LLMs are "probabilistic models of natural language trained on very large chunks of content, mostly extracted from Web."[18], These tasks can be produced from both massive training data (often trillions of words from various domains) and massive neural network structures with billions of parameters. LLMs have general language patterns learnt for the first time by hand-crafted rules or task-specific training data, in which the LLMs can self-super-implementation the training data on raw text and do many language tasks without task-invariant adaptation. [50].

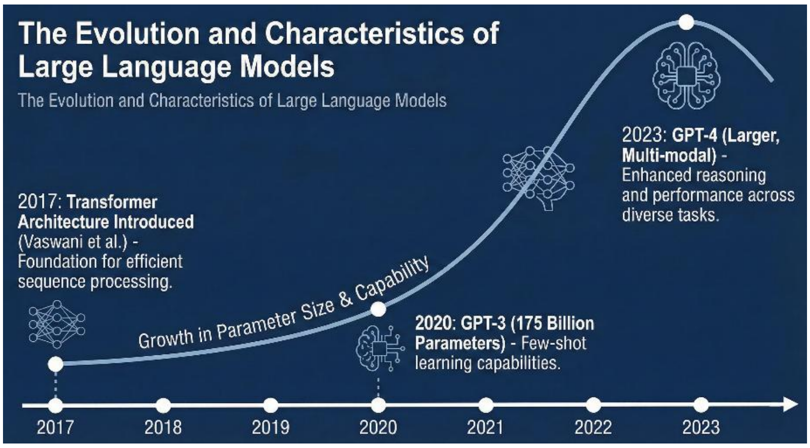


Figure 8: Timeline of LLM Evolution.

The transformer architecture of Vaswani et al. (2017) can be used to train modern LLMs to process long text sequences efficiently and learn complex relationships between distant text elements. In recent years, more and more mature models have been developed: GPT-3 (2020) with 175 billion parameters was shown to learn very few-shot when the model could learn a new task from only a few examples. [51]; GPT-4 (2023) further improved performance across diverse tasks; and numerous other models from different organizations have achieved comparable or superior capabilities on specific tasks[52]. The key characteristics that distinguish modern LLMs from earlier AI systems include their ability to understand context across long passages, maintain coherent conversation through multiple exchanges, exhibit some degree of reasoning and inference, and generate human-quality text across diverse styles and formats[10].

There are several LLMs that are of particular interest for professional knowledge work. Semantic knowledge allows LLMs to understand meaning, not just matching keywords – they can read queries in natural professional language and find some useful information even when they say different terminology. [53]. Contextual synthesis enables LLMs to combine information from multiple sources or document sections, generating coherent summaries that preserve important nuances and relationships[54]. Relational navigation allows LLMs to trace connections between concepts, following chains of references and identifying relevant but not explicitly linked information[55]. These abilities are in accord with cognitive tasks associated with code consultation: understand what a design problem is asking, make sure that specific provisions can be taken into account in many documents, take into account requests from linked code sections, and generate explanations that answer practitioners' needs in ways that do not return matching text snippets.

However, LLMs also exhibit significant limitations that become particularly concerning for professional applications where accuracy and reliability are paramount. Hallucination—generating confident-sounding but factually incorrect information—represents a well-documented problem across all current LLMs[56]. The models lack true understanding or grounding in reality; they predict plausible-seeming text based on statistical patterns in training data, but these predictions may not correspond to actual facts, particularly for topics underrepresented in training or requiring precision beyond typical language use[57]. For building code consultation, where incorrect guidance could lead to unsafe or non-compliant designs, hallucination risk presents a fundamental challenge to LLM deployment. Additionally, LLMs' knowledge cutoff dates—they cannot access information published after their training—means they become increasingly outdated over time, problematic for regulatory domains where codes update continuously[58].

2.3.2 Retrieval-Augmented Generation: Combining Search with Synthesis

Retrieval-Augmented Generation emerged as a technical approach to address key LLM limitations while preserving their strengths. As defined by AWS, "RAG is the process of optimizing the output of a large language model, so it references an authoritative knowledge base outside of its training data sources before generating a response"[11]. The fundamental architecture combines two distinct operations: first, retrieving relevant passages from a specified document collection based on query analysis; second, providing these passages as context to an LLM which generates a response that synthesizes the retrieved information with the query[12]. This architecture addresses multiple LLM limitations simultaneously: it grounds responses in specified documents rather than relying solely on training data, reducing hallucination risk; it allows working with current documents, avoiding knowledge cutoff limitations; and it enables source citation, supporting verification and professional accountability[13].

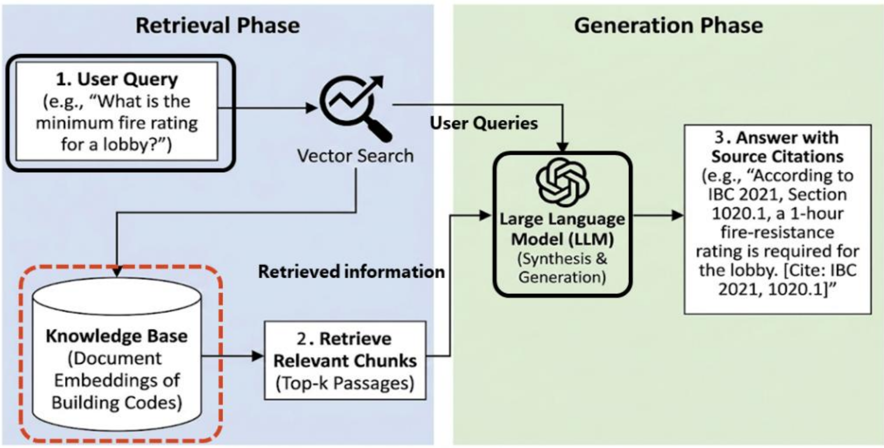


Figure 8: RAG Work Flow Diagram.

The retrieval component typically employs semantic search techniques, where both queries and document passages are converted to high-dimensional numerical representations (embeddings) that capture meaning[59]. Similarity calculations in this embedding space identify passages semantically related to the query even when they use different vocabulary, addressing the keyword matching limitations of traditional search[60]. These passages are then incorporated into the LLM's input along with the user query, effectively providing the model with relevant background material before generating a response[61]. Generator(s) which takes advantage of the language understanding and synthesis offered by the LLM produces responses that incorporate information from several retrieved passages, share an intuition in natural professional language, and at least mention sources to support a claim.[21].

For building code consultation only, RAG has several pros and cons against pure search or pure LLM. Unlike keyword search, RAP can extract complex queries in design-domain language and extract required code even in cases where query/regulation are different. Unlike pure LLMs, RNP only answers in real code text rather than the model's (possibly inaccurate) general knowledge of building regulations. The source citation capability particularly aligns with professional practice needs—practitioners must often justify compliance decisions to building officials or document code bases for design decisions, requiring specific code citations that pure LLM generation cannot reliably provide [20].

However, RAG systems also face challenges in specialized technical domains like building codes. The document preparation phase requires decisions about how to segment codes into retrievable chunks: too small, and important context gets separated; too large, and retrieval becomes insufficiently targeted[62]. Building codes' extensive use of cross-references, tables, and conditional provisions creates challenges for text-based embedding and retrieval—a provision's full meaning may depend on content in other sections or accompanying tables that traditional chunking strategies separate[63]. Generation must balance information generated from more than one reading of some retrieved passages with additional generation errors or hallucinations than the sources documents have indicated.[56]. Research on RAG in technical domains is still evolving, with relatively limited published work specifically addressing highly structured regulatory texts like building codes[64].

2.3.3 Applications to Professional Knowledge Work and Domain-Specific Adaptation

Recent years have seen growing exploration of LLM and RAG applications in various professional knowledge domains, providing both proof-of-concept demonstrations and identification of domain-specific challenges. In legal practice, LLMs have been applied to case law search, contract analysis, and legal research tasks, showing promise but also highlighting concerns about accuracy in specialized legal terminology and the consequences of errors in legal advice contexts[65]. Medical literature navigation represents another active application area, where the volume of published research overwhelms human capacity to maintain current awareness, yet the stakes of inaccurate medical guidance create similar reliability concerns[66].

In construction and architectural domains specifically, several recent studies demonstrate emerging applications and identify challenges. Lee, Jung, and Baek's work on in-house knowledge management using LLMs focused on technical specification document review, showing that "fine-tuned language models can efficiently automate the processing and querying of engineering documents"[8]. Their findings indicated that fine-tuned models outperformed base models on domain-specific tasks, but significant effort was required for dataset preparation and fine-tuning, and some hallucination issues persisted even with domain adaptation[8]. Zhang et al.'s ArchGPT system demonstrated LLM applications to architectural heritage conservation, using multimodal models to process both textual guidelines and visual documentation[14]. This work highlighted the potential for LLM systems to make specialized domain knowledge more accessible but noted challenges in ensuring accuracy for culturally significant heritage work[14].

Research specifically addressing building code applications remains limited. Dhar et al.'s exploratory study on LLM generation of architectural design decisions found that while GPT-4 could generate relevant design rationales in zero-shot settings, the quality fell short of human expert performance and smaller models required few-shot examples or fine-tuning to achieve comparable results [19]. Soliman and Keim's evaluation of LLMs for software architectural knowledge noted moderate accuracy and trustworthiness but highlighted lower precision in identifying quality attribute solutions—aspects requiring specialized domain knowledge[15]. These findings suggest that while base LLMs possess some general knowledge about technical domains including architecture and regulation, direct application without domain adaptation or retrieval augmentation produces inconsistent and unreliable results for specialized professional queries.

The adaptation strategies investigated in existing literature include several approaches: fine-tuning models on domain-specific text corpora, though this requires substantial computational resources and appropriate training data[8]; few-shot learning by providing examples of desired input-output pairs in the query context[19]; prompt engineering to structure queries in ways that elicit better model responses[67]; and retrieval augmentation to ground generation in authoritative domain documents[14]. For building code applications specifically, RAG appears particularly promising because it leverages the codes themselves as the knowledge source—codes represent relatively stable, authoritative, publicly available document collections that change through published amendments rather than requiring continuous model retraining[20]. However, as noted, adapting RAG to highly structured regulatory texts with extensive cross-references and conditional logic presents challenges not fully addressed in existing literature.

2.4 Research Gap Identification

2.4.1 Limited Research on AI in Architectural Regulatory Compliance

Review of existing literature reveals substantial research on building code systems and their challenges, growing investigation of knowledge management in AEC practice, and rapidly expanding exploration of LLM applications across various professional domains. However, the intersection of these areas—AI-enhanced tools specifically for architectural code consultation—remains notably underdeveloped. While automated compliance checking for geometric verification has received significant research attention over several decades [16], the broader problem of intelligent code consultation to support design decision-making has been largely neglected [34]. Most automated compliance research focuses on post-design verification rather than during-design guidance, assumes availability of complete building models rather than early-stage schematic concepts, and addresses quantifiable rules rather than interpretive questions requiring professional judgment[68].

Studies of LLM applications in architecture have primarily addressed design generation, heritage documentation, or high-level architectural knowledge rather than the specific, procedural knowledge of regulatory compliance[14]. The few studies that mention building codes do so peripherally—as one application domain among many rather than as a focused investigation [19]. This gap is particularly significant given that code compliance represents a major time investment in architectural practice, creates persistent frustration for practitioners, and directly impacts public safety outcomes [4]. The lack of research means fundamental questions remain unanswered: Can current LLM/RAG systems accurately retrieve and synthesize building code provisions in response to design questions? What specific adaptations are needed for regulatory texts' structural characteristics? How do practitioners evaluate and trust AI-generated code guidance? What workflow integrations and professional oversight mechanisms would enable safe deployment?

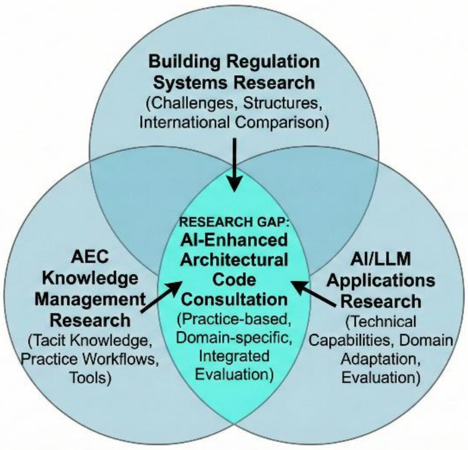


Figure 9: Research Gap Visualization..

2.4.2 Absence of Practice-Based Empirical Studies

Beyond the technical gap, literature reveals an absence of practice-based empirical research examining how architects actually consult codes in real project contexts and what they would need from technological assistance. Research on automated compliance checking typically approaches the problem from computer science perspectives, demonstrating technical feasibility through test cases but without extensive practitioner involvement [34]. Studies of professional knowledge management in AEC tend to address knowledge sharing broadly rather than the specific knowledge domain of regulatory compliance [41]. The result is limited understanding of actual practitioner workflows, pain points prioritization, current workaround strategies, and requirements for trust and verification in AI-assisted code consultation tools.

User-centered design research emphasizes that successful professional tools must address actual user needs as they exist in practice, not as researchers or developers imagine them[69]. For code consultation tools specifically, this means understanding not just what technical capabilities are possible but what capabilities practitioners need most urgently, how tools must integrate with existing software ecosystems, what verification and oversight mechanisms are necessary for professional acceptance, and how different firm scales and practice types affect tool requirements. Without such empirical grounding, technically capable systems may fail to achieve practice adoption because they address the wrong problems, require workflow changes practitioners cannot accommodate, or fail to provide verification features that professional liability concerns demand.

2.4.3 Need for Architect-Centric Evaluation Frameworks

Existing evaluation frameworks for LLM and RAG systems typically emphasize metrics from computer science and natural language processing: perplexity, BLEU scores, embedding quality measures, retrieval precision and recall[70]. While these metrics assess technical performance, they do not directly address whether systems serve professional practice needs. A response might score well on technical metrics but fail to address the practitioner's actual information need, or conversely, might provide highly useful guidance despite imperfect technical performance as measured by standard metrics. Professional evaluation frameworks must consider dimensions like practical usefulness for design decision-making, appropriate level of detail and explanation, reliability and consistency across similar queries, integration with verification and professional oversight, and impact on learning and competence development for junior practitioners[71].

Moreover, evaluation must address not only individual query-response pairs but also how systems perform over extended use across diverse project scenarios. Does performance degrade for unusual building types or novel design situations? How does the system handle ambiguous or underspecified queries that reflect early design uncertainty? Can it identify when a question requires human judgment rather than providing potentially misleading algorithmic guidance? These questions require evaluation approaches that go beyond accuracy testing to consider fitness-for-purpose in authentic professional contexts—an evaluation paradigm not yet established in literature for architectural code consultation applications.

2.4.4 Unanswered Questions About Professional Integration

Finally, substantial questions remain about how AI-based code consultation tools would integrate into professional practice from liability, workflow, and skill development perspectives. Who bears responsibility when AI-provided code guidance proves incorrect and results in non-compliant construction? How do practitioners verify AI-generated responses sufficiently to satisfy professional duty of care obligations? Does reliance on AI assistance impede junior architects' development of regulatory competence, and if so, how might this be mitigated? What workflow touchpoints best accommodate AI tool consultation without disrupting creative design processes? Literature on AI in professional work addresses these questions at abstract levels but not with the domain-specific investigation needed for building code applications[72].

These unanswered questions matter because they affect whether technically capable systems can achieve practical adoption. Professional practices operate under liability frameworks, ethical obligations, and economic pressures that shape what tools they can use and how they use them[2]. Tools that cannot demonstrably satisfy professional responsibility requirements, that require workflow disruptions exceeding efficiency benefits, or that create unacceptable risks to firm reputation or professional licenses will not achieve adoption regardless of technical capabilities. Understanding these professional integration requirements requires empirical research grounded in actual practice contexts—research that existing literature has not provided.

In summary, while substantial literature addresses building regulation challenges, architectural knowledge management, and LLM/RAG technical capabilities independently, the integration of these domains through research specifically addressing AI-enhanced code consultation for architectural practice remains underdeveloped. This research gap motivates the empirical and technical investigation described in subsequent chapters, combining practice-based field research to understand actual needs and constraints with technical evaluation and adaptation to assess and improve AI system capabilities for this specific professional application domain.

CHAPTER 3 RESEARCH METHODOLOGY

This chapter discusses a method of investigating AI-enhanced building code consulting for building practice based on a mixed-methods design from qualitative field research to technical system development and evaluation around a multiple case study approach.

3.1 Research Design Overview

3.1.1 Mixed-Methods Approach

We tackle the problem between human practice (how architects use regulations) and technical performance (how AI systems process regulatory documents) and hence a mixed-methods approach is appropriate.:
Qualitative Component: Semi-structured interviews with architecture practitioners to provide insight into current workflows, pain points and the requirements for AI-aided tools.
Technical Component: Development and evaluation of a domain-adapted RAG system for testing whether identified requirements can be met by engineering breakthrough.
Integration Logic: These components are organized as follows: field results provide insight into technical system design, and evaluations are based on practitioner demands rather than abstract technical benchmarks.

3.1.2 Case Study Methodology

A multiple case study approach examines code consultation practices across four architectural firms of varying scales. This methodology suits research where:

- Questions focus on "how" and "why" rather than "how many"
- The phenomenon is embedded in real-world professional context
- Multiple factors (firm size, project types, resources, expertise) influence the phenomenon

Case Selection: Select firms from different segments of building practices – from large state-owned institutes with formal processes, to small startups with innovation-based approaches. This choice can be used to find common pain points and scale-related needs.

3.1.3 Research Context

The work is set within a Chinese building code system, which has high complexity (national codes, industry standards, local standards) for representing problems common in a country. The Chinese context also offers natural language-jurisdiction alignment, where queries in Chinese clearly map to Chinese regulations, and translation layer complexity that would limit the evaluation of core system performance.

3.2 Field Research Methods

3.2.1 Data Collection

Participants: Four architectural firms representing different practice scales, with interviewees ranging from 8 to 15 years of professional experience. Detailed participant profiles are presented in Chapter 4.

Semi-Structured Interviews: 60-90 minutes sessions covering five thematic areas:

1. Workflow and value distribution in daily practice
2. Time-consuming, low-value tasks and improvement attempts
3. Documentation and review processes
4. Code consultation methods and challenges
5. AI tool experience and expectations

This format can take the form of structured discussion of research topics and allow topics to be explored further. All interviews were in Mandarin Chinese, audio-recorded with consent, and transcribed verbatim.

3.2.2 Data Analysis

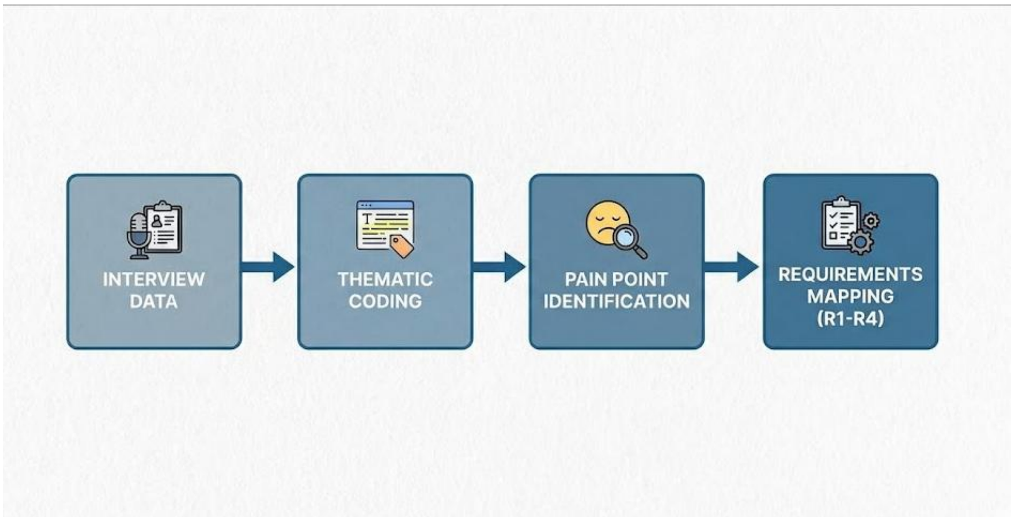


Figure 11：Data Analysis process

Thematic Analysis: Following Braun and Clarke's six-stage approach, interview data were manually coded to find patterns in cases and analyzed from familiarization via initial coding, theme development and cross-case comparison.

Requirements Derivation: A key step focused on technical requirements. Pain points reported in interviews translated into specific feature requirements; workflow patterns provided integration needs; trust issues led to transparency features; this mapping ensures technical development for realistic practitioner needs, not assumed requirements. The requirements derived in field research (R1-R4) and their hierarchical structure is presented in Chapter 4.

3.3 Technical Research Methods

3.3.1 System Development Approach

The technical component builds a domain-optimized RAG which directly meets the requirements of field studies. Instead of providing generic improvements to RAG, each design decision refers to a practitioner requirements summarized in Chapter 4. The system architecture features three key innovations responding to field research findings:

- **Dual-layer knowledge organization** separating authoritative code text from interpretive guidance
- **Multi-expert domain architecture** mirroring practitioners' consultation patterns with different specialists
- **Structured output format** supporting workflow integration and professional verification

Detailed architectural design and implementation are presented in Chapter 5.

3.3.2 Knowledge Base Construction

The knowledge is composed of eight Chinese building standards (~2,400 articles) selected based on the relevant relevance to common architectural practice situations considered in field studies. Other than regulatory text, the system also contains expert knowledge (interpretations, risk alerts, application patterns) in domain specific modules.
The curation process, knowledge organization strategy, and technical implementation details are documented in Chapter 5.

3.4 Evaluation Methods

3.4.1 Comparison Design

We compare our system to ChatGPT-4o, general-purpose AI practitioners may use for code consultation. This reflects the practical choice architects make: general tools vs. built domain systems.

3.4.2 Query Set Development

Thirty building code consultation queries were developed in collaboration with practicing architects, stratified across three complexity levels:

- **Simple**: Single provision lookup (10 queries)
- **Medium**: Multi-clause reasoning within single domain (12 queries)
- **Complex**: Multi-domain synthesis requiring cross-specialty integration (8 queries)

Queries reflect realistic consultation scenarios identified during field research, covering major regulatory domains encountered in typical architectural practice.

3.4.3 Evaluation Framework

Evaluation combines technical metrics with practitioner assessment:
Technical Metrics: Citation accuracy, hallucination rate, domain routing accuracy—measuring whether the system provides verifiable, trustworthy outputs.
Expert Blind Assessment: Four architects from field research independently evaluate responses on practitioner-relevant dimensions: citation reliability, interpretation quality, risk awareness, and professional usefulness. Blind assessment (systems labeled only as "A" and "B") prevents bias from system identification.
Qualitative Case Analysis: Detailed examination of representative queries illustrates specific error patterns and performance differences, providing concrete evidence beyond aggregate metrics.

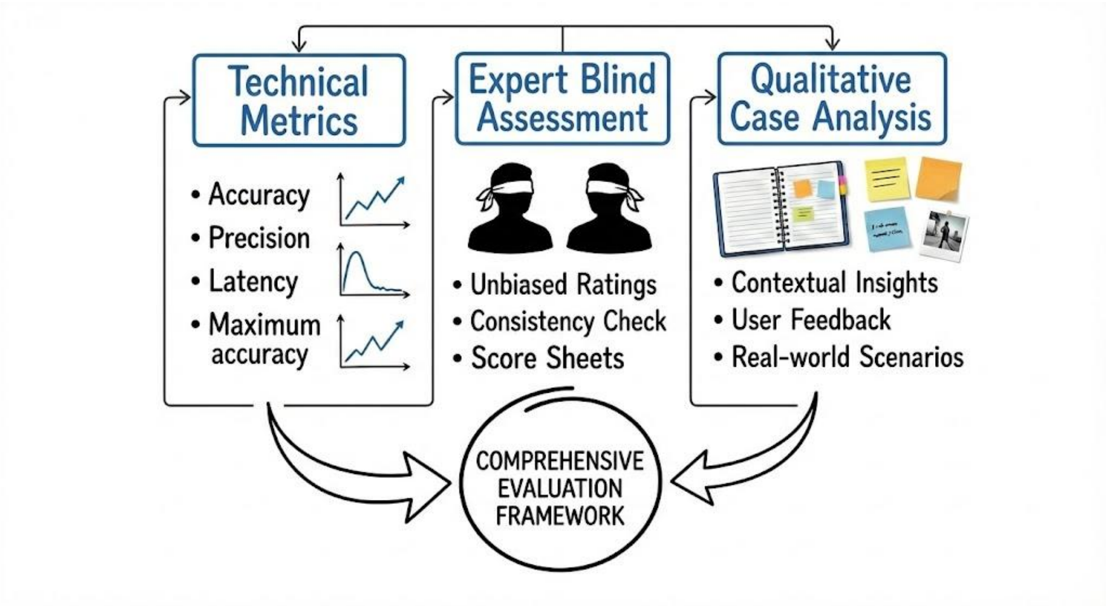


Figure 12：Evaluation Framework

3.5 Chapter Summary

This chapter established the methodological foundation for investigating AI-enhanced building code consultation:

Field Research: A number of case studies are considered at four firms, semi-structured interviews are analyzed in detail to obtain practitioner requirements.

Technical Development: Domain-adapted system design is done to directly respond to the field research findings and its architecture and implementation are described in Chapter 5.

Evaluation: Multi-dimensional evaluation vs. general purpose AI based on practitioner-specific criteria from field research. Both sequential approaches offer technical innovation for real-life issues and evaluation criteria reflect what practitioners need from AI-assisted code experts. This is done in chapter 4 which contains field research results, and in chapter 5 the system development and evaluation results are shown.

CHAPTER 4 FINDINGS - CURRENT STATE OF PRACTICE

In this chapter, we present research results of four architectural firm of different sizes. We give semi-structured interviews with practitioners, highlight the current code consultations practice and present requirements for AI-assisted tools.

4.1 Research Overview

Several interviews with four firms with different organizational models from China architecture industry were conducted during October - November 2024.

Firm	Type	Role	Project Types
Firm A	Large State-Owned Institute (~500s staff)	Project Architect (8 years)	Large public buildings
Firm B	Medium Private Firm (~80s staff)	Project Lead (12 years)	Commercial, office, residential
Firm C	Small Studio (~20s core staff)	Principal (15 years)	Residential renovation, boutique hotels, small public buildings
Firm D	Startup Studio (5-6 core + flexible collaborators)	Founder (10 years)	Non-standard commercial, cultural projects, installations

All interviews were done in Mandarin Chinese, interviewed with consent and transcribed.

4.2 Key Findings from Practice

4.2.1 Code Consultation as Interpretive Practice

A key finding from all four firms is that code consultation is fundamentally an interpretation, not an information retrieval task.

Wang Xiang (Firm D) articulated this distinction:

◆ *Interviewee：Wang Xiang (Firm D)*
Original interview transcript： *"规范的解读和规范的灵活运用，这其实也是一门学问。这个其实它跟法律一样，如果说法律的话真的是所有东西就是明文都写在上面的话，我们其实有时候不太需要律师。"*
Translated text： *"Interpreting and flexibly applying codes is actually a specialized skill. It's like law—if everything were simply written plainly, we wouldn't really need lawyers. "*

The legal analogy is therefore not sufficient: regulation documents must be interpretable based on experience, enforcement patterns, and awareness of implicit flexibility in seemingly rigid requirements.

Zhang Zong (Firm B) described how this interpretive expertise manifests in practice:

◆ *Interviewee：Zhang Zong (Firm B)*

Original interview transcript：
"规范条文是死的，但应用是活的。很多条款有弹性空间，把握这个度完全靠资深建筑师的经验。比如哪些条款可以灵活应用、哪些是不可逾越的红线、常见的坑有哪些——这些规范本身不会告诉你。"

Translated text：
"Code clauses are static, but their application is dynamic. Many clauses have flexibility, and mastering this balance depends entirely on senior architects' experience. Which clauses can be flexibly applied, which are non-negotiable red lines, what common pitfalls exist—the codes themselves don't tell you this."

The consequence is that practitioners rely heavily on senior colleagues rather than documents:

◆ *Interviewee：Wang Xiang (Firm D)*

Original interview transcript：
"我个人习惯还挺喜欢问人的，问那些有经验的人，因为有时候我问的规范，更多想的是怎么去巧妙的去，相当于结合规范去让自己的想法去能够得以实现。"

Translated text：
" I personally prefer asking people—those with experience—because when asking about codes, I'm really thinking about how to cleverly work with regulations to make my ideas achievable. "

4.2.2 Domain-Specific Expert Consultation

A critical finding that emerged consistently across interviews is the reliance on domain-specific experts for different regulatory areas. Chen (Firm C) explained this pattern most explicitly:

◆ *Interviewee：Chen (Firm C)*

Original interview transcript：
"遇到复杂的问题我会请教有经验的人，比如结构问题找结构师，消防问题找消防顾问，术业有专攻我不可能什么都懂。"

Translated text：
" For complex problems, I consult experienced people—structural issues go to structural engineers, fire safety issues go to fire consultants. Each field has its specialists; I can't possibly know everything."

This level-specific consultation pattern reflects the building regulations itself: fire safety, structural engineering, accessibility and energy efficiency are all professional domains of high-level knowledge, terminology and enforcement.

◆ *Interviewee：Wang Xiang (Firm D)*

Original interview transcript：

"这就是为什么以前就喜欢有一些这种老法师，上海话讲就是老法师，意思他们就是一个行走的规范手册，他们会懂得会更加多。我们的确会遇到有时候就会跟规范就会对我们限制。那我们也需要去联系有经验的老法师去跟他们沟通说，如果说我们想达到我们这样的效果，那么我们可以怎么样巧妙的从规范层面来规避这些问题。"

Translated text：

" This is why we've always valued 'laofashi' [veteran experts]—in Shanghai dialect, they're like walking code manuals who understand much more. When regulations constrain us, we need to consult experienced laofashi to discuss how we can cleverly work around these issues from a regulatory perspective to achieve our design goals. "

◆ *Interviewee：Chen (Firm C)*

Original interview transcript：

"很多规范的具体执行要看当地主管部门的尺度，所以我经常是直接去找他们沟通，问清楚他们的要求。这个比查规范更直接更有效。"

Translated text：

" The specific enforcement of many regulations depends on local authorities' discretion, so I often communicate with them directly to clarify their requirements. This is more direct and effective than searching through codes."

This gives us a clear conclusion that good code consulting must not only be general enough, but also have access to specialists from different regulatory fields, providing different knowledge of requirements, enforcement, and solutions in their field of specialization..

4.2.3 The Senior Expert Bottleneck

All four firms identified bottlenecks in accessing senior expertise as a primary pain point, though the manifestation varies by firm scale.
At Firm A (large institute), senior experts are organizational assets but with limited capacity:

◆ *Interviewee：Li Gong(Firm A)*

Original interview transcript：

"我们有几位'活规范'——真正懂规范怎么在实践中用的专家。但项目高峰期，大家都排队等他们的意见，很多项目就卡在等资深专家的判断上。"
"院里资深专家精力有限，只能覆盖重点项目的关键阶段。很多普通项目缺乏足够的经验支持，年轻设计师只能自己摸索——慢且容易出错。"

Translated text：

" We have a few 'living codes'—experts who truly understand how regulations apply in practice. But during peak project periods, everyone queues for their input, and many projects are delayed waiting for senior experts' judgments."
" The firm's senior experts have limited capacity, only covering core stages of key projects. Many ordinary projects lack sufficient experience support, forcing junior designers to learn through trial and error—slow and error-prone. "

At Firm B (medium firm), the challenge is systematic knowledge transfer:

◆ *Interviewee：Zhang Zong (Firm B)*

Original interview transcript：
"我们没有正式的培训体系。年轻人主要靠跟着资深建筑师做项目来学，但资深人员太忙，没法系统带教。很多教训是自己踩坑学的，成长慢、项目质量也参差不齐。"

Translated text：
" We have no formal training system. Juniors primarily learn by working on projects with senior architects, but seniors are too busy to provide systematic guidance. Many lessons are learned through mistakes, leading to slow growth and inconsistent project quality."

At Firm C and D (small studios), the dependence on external senior expertise creates instability:

◆ *Interviewee：Wang Xiang(Firm D)*

Original interview transcript：
"核心技术把关和规范适应，我们始终要咨询长期合作的资深建筑师。没有他们的经验支持，大多数非标项目我们不敢接。"

Translated text：
" For core technical oversight and code adaptation, we always consult with long-term collaborating senior architects. Without their experience support, we wouldn't dare take on most non-standard projects."

◆ *Interviewee：Chen(Firm C)*

Original interview transcript：
"我们的资深资源都是临时合作者。不同项目涉及不同的资深建筑师，经验和风格各异，导致项目质量不稳定。另外，资深建筑师的很多经验是口头的，没有系统记录——类似问题以后还要重新咨询，效率低。"

Translated text：
" Our senior resources are all temporary collaborators. Different projects involve different senior architects with varying experience and styles, leading to inconsistent project quality. Additionally, much of senior architects' experience is verbal, with no systematic documentation—we have to re-consult for similar issues

later, which is inefficient."

4.2.4 Limitations of Current Tools

Practitioners use Jianbiaoku (建标库) as the primary digital code database, but consistently noted its limitations:

◆ *Interviewee：Zhang Zong(Firm B)*

Original interview transcript：
"建标库是基本工具——大家都用。但它只是搜索引擎——你得知道搜什么，搜到了还得自己解读。如果能加入资深建筑师对关键条款的解读——比如哪些条款有弹性空间、常见坑有哪些——价值会大大提高。"

Translated text：
" Jianbiaoku is the basic tool—everyone uses it. But it's just a search engine—you still need to know what to search for, and once found, you still need to interpret it yourself. If it could include senior architects' interpretations of key clauses—such as which clauses have flexibility and what common pitfalls exist—its value would increase significantly."

◆ *Interviewee：Zhang Zong(Firm B)*

Original interview transcript：
"我试过用 ChatGPT 问规范问题。有时候答案挺好的，但有时候会编造不存在的条款。你必须每条都核实，所以节省的时间有限。"

Translated text：
" I've tried using ChatGPT for code questions. Sometimes answers are good, but sometimes it fabricates non-existent clauses. You must verify everything, so time savings are limited."

4.3 Derived Requirements

Based on the interview results, we identify four core requirements for AI-aided code consultation tools: baseline requirements ensure that there is no tool to use that is not used; core value requirements represent the difference between existing tools; enhancement requirements highlight professional utility.

4.3.1 Requirement Hierarchy

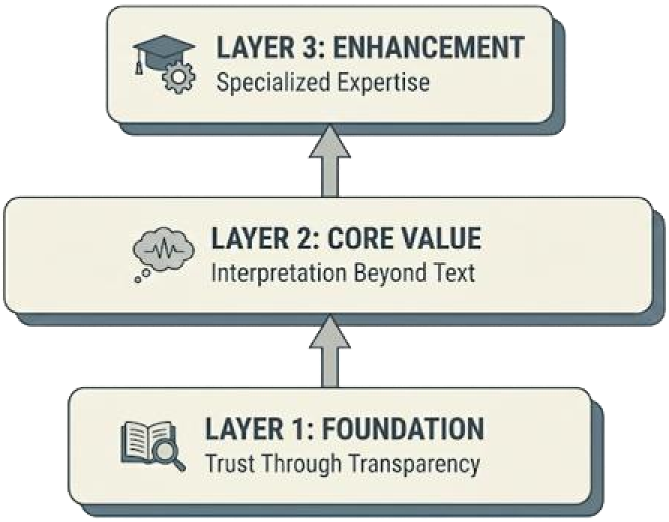


Figure 13：Requirement Hierarchy

R1: Verifiable Sources with Transparent Attribution (Foundation)

Interview basis:

◆ *Interview basis:*
Original interview transcript： *"我不需要 AI 猜测——我需要它告诉我确切的条款，准确的引用，我可以自己去核实。"* *"我需要知道 AI 是怎么得出答案的——搜了什么文件、为什么选这些条款。更重要的是，引用的资深经验来源要清楚。只有这样我才敢信。黑箱我不敢用。"* *"如果 AI 给了错误建议，项目出问题，责任算谁的？注册建筑师的章是我盖。"*
Translated text： *" I don't need AI to guess—I need it to provide exact clauses and accurate citations for me to verify."* *" I need to know how AI arrived at its answer—what documents it searched and why it selected specific clauses. More importantly, I want to know the source of the senior experience cited. Only then can I trust it. I don't dare use a black box. "* *" If AI provides incorrect advice and the project encounters problems, who is responsible? I'm the one stamping the registered architect seal. "*

Requirement definition:
This requirement encompasses two interrelated aspects:

1. **Accurate citation**: All a regulatory claim must contain exact code references (single number, article, clause) that practitioners can verify against original documents.
2. **Transparent attribution**: A clear difference between regulatory text and guidance with explicit information on experience-based content and

statements of limitations where human judgement is necessary.

Without this foundation, practitioners will not trust or adopt any AI tool regardless of other capabilities.

R2: Embedded Expert Interpretation (Core Value)

Interview basis:

◆ *Interview basis:*
Original interview transcript： *"规范条文是死的，但应用是活的。很多条款有弹性空间，把握这个度完全靠资深建筑师的经验。我希望 AI 能把这些经验记录下来——比如哪些条款可以灵活应用、哪些是不可逾越的红线、常见的坑有哪些。"* *"这就是为什么我们一直看重'老法师'——上海话里，他们就像行走的规范。他们脑子里装的不是死条款，而是活的经验——那才是真正有价值的。"* *"如果 AI 能像我们院里的老专家一样——不仅知道条款，还知道实际项目中怎么用、怎么避坑——那就厉害了。"*
Translated text： *" Code clauses are static, but their application is dynamic. Many clauses have flexibility, and mastering this balance depends entirely on senior architects' experience. I hope AI can document this experience—such as which clauses can be flexibly applied, which are non-negotiable red lines, and what common pitfalls exist."* *"This is why we've always valued 'laofashi' [veteran experts]—in Shanghai dialect, they're like walking code manuals. What they carry in their minds isn't static clauses but living experience—that's what's truly valuable. "* *"If AI could be like the senior architects in our firm—knowing not just the clauses but how to apply them in practice and avoid pitfalls—it would be a game-changer. "*

Requirement definition:

The system must provide interpretive guidance beyond regulatory text, including:
- Clause interpretations explaining practical meaning and application scope
- Risk alerts identifying frequently misinterpreted provisions and review focus areas
- Practical application patterns showing how requirements are typically satisfied
- Case examples demonstrating solutions in actual projects

This is the main value that distinguishes our system from others. Jianbiaoku provides document access; what practitioners want is the interpretive layer that transforms regulatory language to actionable guidance. This directly tackles the top expert bottleneck by making tacit understanding explicit.

R3: Context-Aware Multi-Domain Consultation (Enhancement)

Interview basis:

From the interviews, we found that efficient code consulting requires three complementary abilities: project-specific adaptation, domain-specific expert routing and cross-domain discovery..

Project-specific adaptation:

◆ *Interview basis:*

Original interview transcript：
"我想说'这是北京的 8 层住宅'，工具就能自动识别适用的规范和地方标准。更关键的是，我想知道资深建筑师通常怎么为这类项目适应规范，有没有成熟经验可以直接复用。"

Translated text：
" I want to say 'this is an 8-story residential building in Beijing' and have the tool automatically identify applicable codes and local standards. More critically, I want it to tell me how senior architects typically adapt these codes for such projects and if there are mature experiences to reuse directly."

Domain-specific expert routing:

◆ *Interview basis:*

Original interview transcript：
"遇到复杂的问题我会请教有经验的人，比如结构问题找结构师，消防问题找消防顾问，术业有专攻我不可能什么都懂。"
"很多规范的具体执行要看当地主管部门的尺度，所以我经常是直接去找他们沟通，问清楚他们的要求。这个比查规范更直接更有效。"

Translated text：
" For complex problems, I consult experienced people—structural issues go to structural engineers, fire safety issues go to fire consultants. Each field has its specialists; I can't possibly know everything."
" The specific enforcement of many regulations depends on local authorities' discretion, so I often communicate with them directly to clarify their requirements. This is more direct and effective than searching through codes. "

Cross-domain discovery:

◆ *Interview basis:*

Original interview transcript：
"我们不能说关键词说这是个交通建筑，我们一定找的就是交通建筑的案例……我们更希望说它的概念从这些其他领域的东西能够给予我们哪些新的灵感和创意。"

Translated text：
" We can't just search for 'transportation building' and only get transportation building cases... We want concepts from other domains to give us new inspiration and creativity."

Requirement definition:

This requirement encompasses three integrated capabilities:

1. **Project-specific adaptation**: Individuals should adapt to project characteristics (type, size, location, phase) of experienced practitioners to interpret their guidance in different contexts.
2. **Domain-specific expert routing**: The system should bring queries to domain experts - fire safety questions to fire safety knowledge, structural questions to structural knowledge, accessibility questions to accessibility knowledge - with practitioners actually consulting experts in specific areas, like “structural problems go to structural engineers”, “fire safety problems go back to fire consultants."
3. **Cross-domain discovery**: Prove knowledge of prior points and knowledge along project types and implementation experience – continue the usual arguments that practitioners value other types of projects.

The domain specific routing ability: building rules are not a single field of knowledge but are a shared professional area (fire safety, accessibility, structural, energy conservation, etc.) with expertise, enforcement, and practical solutions. A good consultation system should be designed to reflect this domain rather than uniformly implement all regulations.

4.3.2 Summary

ID	Requirement	Layer	Derived From
R1	Verifiable Sources with Transparent Attribution	Foundation	Trust concerns; professional liability; verification burden
R2	Embedded Expert Interpretation	Core Value	Interpretive nature of practice; senior expert bottleneck; tacit knowledge gap
R3	Context-Aware Multi-Domain Consultation	Enhancement	Project-specific guidance needs; domain-specific expert consultation pattern; cross-domain case discovery
R4	Workflow Integration	Application	Continuous consultation pattern; efficiency priority

Table 4.3:1： Requirements Summary

Table 4.2: Requirements Summary

These requirements directly relate to the architecture of Chapter 5. The hierarchical structure from the top to the top level to the enhancement and application of system components map to specific system elements:

- **R1** drives the dual-source attribution mechanism in response generation
- **R2** motivates the dual-layer knowledge base separating code provisions from expert experience
- **R3** informs both the context-aware query processing and the **multi-expert architecture with domain-specific modules**—the domain-specific expert routing pattern observed in practice ("structural issues go to structural engineers, fire safety issues go to fire consultants") directly motivates

organizing the system around specialized domain experts rather than a single unified retrieval mechanism
- **R4** shapes the structured response format for workflow integration

---

4.4 Chapter Summary

Field research across four firm scales reveals a fundamental gap in current code consultation practice: practitioners need interpretive expertise, not just document access. The core findings are:
The interpretation gap: Code reading is a learning activity that requires some real knowledge. Existing tools fix "finding code text" but not "understanding what codes mean for my project."
The domain expertise pattern: Practitioners consult two different domain experts in a particular purpose — fire safety experts for fire, structural engineers for structural questions. This domain-specific consultation pattern corresponds to building regulations as two different professional types.
The expertise bottleneck: The tacit knowledge of older architects is the most valuable consultation resource, but they are limited in their capacity and lead to bottlenecks at all firms.
The trust barrier: Current AI tools fail to meet professional verification requirements and are limited by the interpretation they require from practitioners, despite efficiency benefits.
These results result in four hierarchically-structured requirements: verifiable sources with transparent attribution (foundation), embedded expert interpretation (core value), context-aware multi-domain consultation (enhancing) and workflow integration (application).
The domain specific expert relationship – how practitioners refer different types of questions to different experts is directly practical for a multi-expert system. Instead of being treated as a single set of data, an AI system should reflect how practitioners actually seek expertise: they call related experts in specific domains based on the type of query. This influences the multi-intrusive architecture presented in the following chapter.

CHAPTER 5 TECHNICAL ANALYSIS AND SYSTEM DEVELOPMENT

Chapter introduces technical research of this thesis, how AI can be improved to meet the practitioner requirements found in Chapter 4. Inspired by field work results and the fact that code consultation is an interpretation practice that requires experience, knowledge, expertise routing, and transparent attribution, this chapter proposes, implements and evaluates Experience-Augmented RAG (EA-RAG), a new architecture for building code consultation.

5.1 EA-RAG Architecture Design

5.1.1 Design Rationale

The output quality of an RAG system (1) user query is a wild issue (2) basic large model which requires expensive change (3) knowledge base (the core building block to which meaningful innovation is realized in practice). As a consequence, we design **the knowledge base structure** – what is the knowledge structure? How should the regulatory content and expert knowledge are organized to achieve retrieval success and response quality?
Modern RAG designs make all of their results as facts, not comparing reliable source text to interpretive guidance, or routing questions to domain experts. This is the basic reason for EA-RAG design, which answers the four requirements from Chapter 4 directly.

Requirement	Architectural Response
R1: Verifiable Sources with Transparent Attribution	Dual-layer knowledge base with explicit source tracking
R2: Embedded Expert Interpretation	Experience Layer containing curated interpretive knowledge
R3: Context-Aware Multi-Domain Consultation	Query Router + Domain Expert Modules with sparse activation
R4: Workflow Integration	Structured response format with combined code + interpretation

5.1.2 Three Key Innovations

The transition from basic RAG to EA-RAG involves three interconnected innovations:

Aspect	Basic RAG	EA-RAG	Rationale
Knowledge Organization	Single unified knowledge base	Domain-specific expert modules with Query Router	Mirrors real consultation patterns: "structural issues go to structural engineers"
Knowledge Content	Code provisions only	Code Layer + Experience Layer	Addresses interpretation gap: practitioners need guidance, not just text
Response Generation	Single LLM response	Multi-Expert Fusion with Conflict Detection +	Handles cross-domain queries requiring synthesis

Structured Output

5.1.3 System Components

Query Router: The entry point of all queries: identifying relevant regulatory domains and activating expert modules via **sparse activation**: only experts related to a given query are active. Domain identification takes place as follows: (1) keyword search for domain specific words; and (2) semantic analysis computing similarity between the query and domain description embeddings. For example, “commercial building corridor width requirements” activates Fire Safety Expert and Accessibility Expert but Energy and Structural experts remain inactive.
Domain Expert Modules: Each module functions as a self-contained knowledge unit with a **dual-layer structure**:

- **Code Layer**: Remains original text with preserved hierarchical structure (standard → chapter → section → article → clause), metadata (version, effective date, mandatory status) and article-level chunking with clause subdivision.
- **Experience Layer**: Contains expert knowledge such as clause interpretation (meaning and application), risk alerts (common mistakes and review focus areas with severe severity), application patterns (typical compliance methods) and examples (anonymized illustrations from real projects).

Multi-Expert Fusion: When multiple experts activate for cross-domain queries, this component aggregates results while maintaining source attribution, detects conflicts (numerical or conceptual) between domain requirements, and applies the "strictest requirement" principle for resolution.
Structured Response Generation: Output is generally composed of: (1) References as well as expert recommendations (2) Recommendations with three-level solutions (minimum/recommended/premium), (3) Interpretation with context-specific guidance, (4) Risk alerts with severity and remedy information, and (5) Disclaimer of professional requirements.

5.2 Prototype Implementation

5.2.1 Knowledge Base Construction

Code Layer:

- 8 core standards: GB 50016, GB 55019, GB 55015, GB 50096, GB 50352, GB 50099, GB 50189, GB 50763
- Total: ~2,400 articles with preserved hierarchical structure
- Chunking: Article-level with clause subdivision for lengthy articles

Experience Layer:

Expert Module	Entries	Content Focus
Fire Safety	45	Evacuation calculations, fire compartment errors, review focus areas
Accessibility	32	Barrier-free route conflicts, facility dimension pitfalls
Energy	28	Thermal performance compliance, HVAC coordination issues
Residential	24	Spatial requirements, ventilation/lighting interpretations
Schools	18	Classroom standards, safety requirements for educational facilities
General	21	Building classification, general spatial and functional requirements

5.2.2 Technical Stack

Component	Technology
Embedding	gemini-embedding-exp-03-07
Vector Store	ChromaDB
Generation	gemini-2.5-flash
Framework	LangChain

5.2.3 Retrieval Strategy

The system executes a two-phase retrieval for each query:
Phase 1 - Code Layer Retrieval: Query Router identifies relevant expert domains; each activated expert retrieves top-5 code chunks via semantic similarity; context filter applies project parameters (building type, height, location).
Phase 2 - Experience Layer Retrieval: System retrieves provision IDs from retrieved code chunks; experience entries associated with the provisions are retrieved (top-3/ expert); context filter is relevant for query. This linked retrieval ensures that experience knowledge always is based on specific regulatory requirements, with traceability and verifiability.

5.3 5.3 Evaluation

5.3.1 Evaluation Design

Comparison Systems: EA-RAG was compared against ChatGPT-4o, representing the current state-of-the-art in general-purpose AI assistants that architects might realistically use for code consultation.

Query Set: 30 building code consultation queries developed with practicing architects, stratified by complexity:

Complexity	Description	Count	Example
Simple	Single provision lookup	10	"Stairwell width requirements for residential buildings?"
Medium	Multi-clause, single domain	12	"Evacuation staircase requirements for high-rise residential?"
Complex	Multi-domain synthesis	8	"Corridor requirements for basement commercial space?"

Evaluation Methods: Technical metrics (citation quality, completeness, domain routing accuracy, hallucination rate) and expert blind evaluation by four architects (citations reliability, interpretation quality, risk awareness, professional usefulness, 5-point scale).

5.3.2 Technical Performance Results

Metric	ChatGPT-4o	EA-RAG	Δ
Citation Accuracy	N/A*	94%	—
Citation Completeness	N/A*	87%	—
Domain Routing Accuracy	N/A	93%	—
Hallucination Rate	23%	3%	-20%

*ChatGPT scarcely cites specific clauses, offering only general knowledge and vague references to broad regulations.

5.3.3 Expert Blind Assessment Results

Dimension	ChatGPT-4o	EA-RAG	Δ	Improvement
Citation Reliability	2.8	4.4	+1.6	+57%
Interpretation Quality	2.3	4.1	+1.8	+78%
Risk Awareness	3.6	3.9	+0.3	+8%
Professional Usefulness	3.3	4.2	+0.9	+27%
Overall Mean	**3.0**	**4.2**	**+1.2**	**+40%**

Results by Query Complexity:

Complexity	ChatGPT-4o	EA-RAG	Δ	Improvement
Simple	3.2	4.3	+1.1	+34%
Medium	2.4	4.2	+1.8	+75%
Complex	1.8	4.0	+2.2	**+122%**

5.3.4 Qualitative Case Analysis

To complement quantitative findings, detailed error analysis was conducted on representative queries:
Case 1 - Accessibility Ramp Slope: ChatGPT stated maximum slope as 1:12 and provided a fabricated "length vs. slope" table that does not exist in any standard. EA-RAG correctly cited GB 50763-2012 §3.4.4, identifying maximum slope as 1:8 (with height limit ≤0.30m) and providing the actual regulatory table. This represents a **critical safety error** that could lead to non-compliant designs.

Case 2 - School Window Protection: ChatGPT stated horizontal load requirement as ≥1.0 kN/m without citations. EA-RAG correctly cited GB 50099-2019 §8.1.6 specifying ≥1.5 kN/m, a **50% understatement** by ChatGPT that could result in inadequate protective barriers.
These cases demonstrate three fundamental limitations of general-purpose AI: (1) **hallucination problem**—fabricating non-existent clauses and tables; (2) **interpretation gap**—generic responses without practical guidance; and (3) **multi-domain blindness**—missing requirements from overlapping regulatory domains.

5.3.5 Requirements Validation

Requirement	Validation Evidence
R1: Verifiable Sources	EA-RAG citation accuracy 94% vs. ChatGPT's vague references
R2: Expert Interpretation	Interpretation quality score 4.1 vs. 2.3 (+78%)
R3: Multi-Domain Consultation	EA-RAG identifies cross-domain conflicts; 93% routing accuracy
R4: Workflow Integration	Structured output with tiered recommendations received positive feedback

5.4 Chapter Summary

This chapter presented Experience-Augmented RAG (EA-RAG), an architecture directly addressing the four requirements identified in field research:

1. **Dual-layer knowledge base** separates verifiable code provisions from interpretive guidance (R1, R2)
2. **Multi-expert domain architecture** mirrors real-world consultation patterns with 93% routing accuracy (R3)
3. **Transparent attribution** distinguishes sources and acknowledges limitations (R1)
4. **Structured output** enables workflow integration (R4)

Evaluation with 30 practitioner-validated queries demonstrates significant improvements over ChatGPT-4o:

Key Metric	Result
Citation Accuracy	94% (vs. vague references)
Hallucination Rate	3% (vs. 23%)
Interpretation Quality	+78% improvement
Complex Query Performance	+122% improvement

The largest gains can be seen for difficult multi-domain queries, validation of multi-expert fusion. By making tacit expert knowledge explicit and accessible, EA-RAG resolves the expert bottleneck that limits practitioners at all firm scale: AI-aided code consultation from document retrieval to expert consultation.

Current Limitations: Knowledge base holds common cases but edge cases need updating; does not cover local jurisdiction enforcement; standalone interface needed to be developed for CAD/BIM integration.

Future Directions: Expand experience through structured expert knowledge capture; add jurisdiction modules for regional enforcement; develop CAD/BIM plugins for seamless workflow integration.

Technical validation indicates that domain-specific RAGs, not general LLMs, are necessary for professional regulatory consultation. The key point is that building code consultation not only requires correct retrieval, but also interpretive guidance, which should be driven by professional expertise not generated by the model.

CHAPTER 6 DISCUSSION AND IMPLICATIONS

This chapter summarizes the most recent field research (Chapter 4) and technical development (Chapter 5), directly addressing the research questions presented in Chapter 1, discusses the implications for the performance of EA-RAG for architecture, education, and general applications to regulations, as well as critical remarks on the role of AI in architecture, balancing technology and ethics.

6.1 Synthesis of Research Findings

6.1.1 Direct Responses to Research Questions

Primary Research Question: Can Large Language Model-based systems significantly improve the efficiency and accuracy of building code consultation in architectural practice?

It is a very **qualified question**: LLM based systems can be easily adapted to the architecture by EA-RAG, which is efficient and accurate in their own right, if not general purpose.

The evaluation results from Chapter 5 demonstrate this clearly:

Metric	ChatGPT-4o (General AI)	EA-RAG (Domain-Adapted)	Improvement
Citation Accuracy	Vague references only	94%	Transformative
Hallucination Rate	23%	3%	-87%
Interpretation Quality	2.3/5.0	4.1/5.0	+78%
Complex Query Performance	1.8/5.0	4.0/5.0	+122%

This directly leads to efficiency improvements since we do not have the “verification tax” that has been required in current generic AI tools. As one of the researchers noted in field work, “Sometimes answers look good but you've to check every point”. EA-RAG's 94% citation accuracy is directly addressed by this problem.

RQ1: What are the specific pain points architects face when consulting building regulations?

Field research across four firm scales identified six interrelated pain points, with **senior expert dependency** as the central bottleneck:

1. **Senior Expert Dependency**: Over-reliance on "living codes" creates workflow bottlenecks and risks knowledge loss
2. **Time Inefficiency**: Repetitive consultation, regulatory uncertainty, and cross-reference tracing
3. **Knowledge Access Barriers**: Scattered provisions, complex cross-references, version tracking challenges
4. **Interpretation Challenges**: Ambiguous language, inter-code conflicts, lack of practical guidance
5. **Organizational Knowledge Gaps**: Loss of experiential knowledge, slow junior onboarding
6. **Case Search Difficulties**: Time-intensive cross-domain precedent discovery

EA-RAG's dual layer approach directly addresses these pain points, via its Experience Layer (high-level expert knowledge) and multi-expert domain routing (lower search time).

RQ2: How do existing general-purpose AI tools perform in architectural regulatory contexts?

Evaluation revealed that general-purpose AI tools (represented by ChatGPT-4o) are inadequate for professional building code consultation:

Metric	ChatGPT-4o Performance	Professional Threshold
Citation Accuracy	Vague references only	>90% required
Hallucination Rate	23%	<5% acceptable
Interpretation Quality	2.3/5.0	>4.0 needed
Complex Query Performance	1.8/5.0	>3.5 needed

The failure modes: hallucination of non-existent provisions (to match clause numbers, even full regulatory tables), interpretation gap (generic answers without guidance), multi-domain blindness (to meet requirements from overlapping regulatory areas such as fire safety and accessibility).

RQ3: What modifications are necessary to adapt AI systems for specialized architectural knowledge?
EA-RAG's three key innovations directly respond to identified limitations:

Innovation	Addresses	Evaluation Result
Dual-layer knowledge base (Code + Experience)	Interpretation gap, senior expert bottleneck	+78% interpretation quality
Multi-expert domain architecture with sparse activation	Multi-domain blindness, domain semantic gaps	93% routing accuracy
Structured response with conflict detection	Cross-reference failure, verification needs	94% citation accuracy

RQ4: What are the broader implications for professional practice and architectural education?
The implications span multiple dimensions:

- **For practice**: AI adaptation requires workflow integration, new documentation protocols, and organizational knowledge curation processes
- **For education**: Balance efficiency gains with core regulatory expertise development; emphasize interpretive skills over rote memorization
- **For industry**: Potential democratization of regulatory knowledge, reducing the gap between large and small firms

6.1.2 Why EA-RAG Outperforms General-Purpose AI

The evaluation reveals three fundamental limitations of general-purpose AI for building code consultation that EA-RAG specifically addresses:
1. Hallucination Problem
ChatGPT's 23% hallucination rate (using non-existent clauses and whole tables) cannot be tolerated in the enterprise, where error can be safety and liability adverse. EA-RAG's retrieval based framework reduces this to 3% by only quoting provisions in its knowledge base.
Case study: ChatGPT produced a “length vs. slope” table for accessibility ramps that did not exist in any of its variants but where maximum slope was 1:12 instead of 1:8 (gf 50763-2012 §3.4.4).

2. Interpretation Gap
ChatGPT's generic answers such as “please consult relevant codes” are of little practical value beyond what practitioner knows. EA-RAG's Experience Layer

contains what practitioners are actually looking for—of which clauses are flexible, which pitfalls are common, and of which solution recommendations are related.
3. Multi-Domain Blindness
ChatGPT missed accessibility for corridor width examples, and EA-RAG's multi-expert architecture recognized Fire Safety and Accessibility experts that 1.8m (accessibility) and 1.4m (fire safety) requirements needed to satisfy fire safety and Accessibility experts.

6.2 Implications for Architectural Practice and Education

6.2.1 Workflow and Organizational Adaptations

EA-RAG's demonstrated performance suggests several workflow integration strategies:
Tiered Consultation Process: AI is first-line answer for the simplest queries, with expert comments for complex problems. EA-RAG's 93% domain routing and 94% citation accuracy make it useful for initial use, and clearly indicates uncertainty signaling for cases where humans need to be informed.
Formalized Knowledge Curation: Successful EA-RAG Experience Layer shows the importance of maintaining expert knowledge at a level. Organizations should establish processes for record interpretation reasons, risk patterns and project prior information–transforming one-time consultations into re-use institutional knowledge.

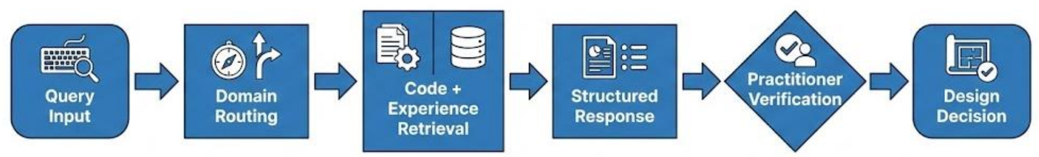


Figure 14: Proposed Workflow Integration Model

Scale-Specific Implementation:

Firm Scale	**Priority Focus**	**EA-RAG Feature Leverage**
Large Institute	Senior expert load distribution, knowledge preservation	Experience Layer captures institutional knowledge
Medium Firm	Efficiency, experience transfer	Multi-expert routing reduces search time
Small Studio	Cross-domain support, non-standard projects	Domain expert modules provide specialist access

6.2.2 Professional Responsibility and Liability

EA-RAG's design directly addresses liability concerns raised in field research ("If AI gives wrong advice, who's responsible?"):
Transparent Attribution: The difference between Code Layer (title) and Experience Layer (interpretation) allows practitioners to trust authoritative sources and enjoy the experience of practitioners.
Verification Support: 94% citation accuracy with correct clause references (i.e., number, article, clause) enables efficient checking while no practitioner checks a single statement.
Professional Boundaries: Structured answers provide clear indications of project differences and authority consultations in special cases.

However, firms should set clear rules: keep the AI output only as some important information that needs to be reviewed by professional researchers, maintain audit records of the AI queries and checks and ensure organizational policies clearly state that the primary responsibility rests with the registered architect.

6.2.3 Implications for Architectural Education

EA-RAG automates routine code retrieval issues that traditional training models pose and provides new learning opportunities:
Shift from Memorization to Interpretation: With AI to retrieve facts, education should focus on interpretive skills – interpreting ambiguous provisions, avoiding conflicts among codes, and setting up rules that work in an optimal manner to get design goals.
AI Collaboration Skills: Junior architects need training on effective AI tool use: framing goal-oriented queries, validating outputs, and recognizing situations requiring human judgment. EA-RAG's transparent structure supports this learning by showing reasoning paths.
Accelerated Competence Development: EA-RAG's Experience Layer makes senior expert knowledge available to young participants, potentially making them 3+ years from the current status of regulatory competence development, while letting them understand reason behind interpretations rather than receiving answers.

6.3 Broader Industry and Cross-Context Implications

6.3.1 Industry Infrastructure and Knowledge Democratization

EA-RAG's performance demonstrates the value of curated domain knowledge, suggesting industry-wide infrastructure needs:
Machine-Readable Code Formats: Moving building codes from static PDFs to structured format should simplify parsing and retrieval, EA-RAG reached 94% of citations though it did work with unstructured source documents. native structured

formats might increase this.
Shared Knowledge Platforms: Experience Layer could grow to industry-wide reports of expert annotations, project histories, and regulatory authority reports provided by professional organizations with quality guarantees.
Democratization Potential: Big firms historically had advantages from knowing senior “living code” experts. EA-RAG shows that information can be captured and made available, maybe improving the gap between small and medium firms. The +122% gain for high queries for small studios in dealing with not-standard projects is especially valuable for small firms.

6.3.2 Transferability to Other Regulatory Contexts

The key lessons of this work — the need to integrate experiential knowledge in RAG systems and general AI challenges—are applicable to other multi-level frameworks:
Universal Challenges: Stacked provisions, cross-references, ambiguous language, expert tacit knowledge. In Europe, Eurocodes, North American (IBC) and other regulatory systems.
Required Adaptations:

Dimension	Adaptation Needed
Language & Structure	Adjust document processing for different code organizations (e.g., Eurocode's modular structure)
Regulatory Culture	Adapt experience capture to local interpretation practices and enforcement patterns
Digital Maturity	Integrate with existing tool ecosystems (MasterFormat, BIM standards)

6.3.3 Language as Jurisdictional Signal

This research's focus on "Chinese queries + Chinese regulations" leverages the natural mapping between language and regulatory jurisdiction. This mapping has implications for system design:

- **Single-language scenarios**: Natural term alignment ensures retrieval accuracy
- **Multi-language scenarios** (e.g., English queries): Require jurisdiction disambiguation mechanisms
- **Cross-border projects**: Need explicit jurisdiction selection and comparative query support

6.4 Critical Reflection on AI in Architecture

6.4.1 Risks of Over-Reliance

Despite EA-RAG's strong performance, over-reliance risks must be acknowledged:
Expertise Erosion: If AI is automatic at every phase of code retrieval, young designers may not have deep control skills. EA-RAG transparent structure (coding sources and reasoning about interpretation) can also serve this purpose by supporting learning, but intentional educational design is needed.
Verification Complacency: With AI accuracy increasing (EA-RAG reaches 94% of citation accuracy, ChatGPT barely includes references), the practitioner may do less checking: the 6% error rate is much better than other alternative but still requires some professional supervision.
Edge Case Blindness: As the AI accuracy increases (EA-RAG reaches 94% of citations, ChatGPT does not include references) the practitioner may not check as much as other options: 6% is very good but also requires some supervision.

6.4.2 Preserving Design Creativity

A core architectural value is creative problem-solving within regulatory constraints. EA-RAG's design supports rather than constrains creativity:
Multiple Solution Pathways: General answers contain three-level recommendations (minimum/recommended/premium) than single "correct" answers and allow the design exploration within compliance bounds.
Strategic Consultation: The Experience Layer contains application patterns illustrating that practitioners generally satisfy their needs and can be creative but not rigidly enforced.
Risk Awareness: By explicitly identifying common pitfalls and reviewing focus areas, EA-RAG helps practitioners understand how flexible is it or not.

6.4.3 Ethical Considerations

Ethical AI adoption in architecture centers on three principles validated by this research:
Accountability: EA-RAG's transparent attribution ensures that humans can always be responsible (and AI can be useful but not a decision maker).
Transparency: Correct choice between authoritative code text and interpretable guidance can help in making professional decisions.
Equity: The democratization of the knowledge of experts may benefit small firms and emerging practitioners, but the access fairness of people is important.

6.5 Chapter Summary

This chapter synthesized the research findings, directly addressing the research questions and exploring implications across multiple dimensions:

Key Conclusions:

1. **LLM-based systems can significantly improve code consultation** when domain-adapted through EA-RAG architecture, achieving 94% citation accuracy, 3% hallucination rate, and +78% interpretation quality improvement—but they augment rather than replace professional judgment.
2. **The core modifications required** include dual-layer knowledge organization (code + experience), multi-expert domain architecture with sparse activation, and structured output with transparent attribution.
3. **Implications for practice** include tiered consultation workflows, formalized knowledge curation processes, and clear liability protocols maintaining professional responsibility.
4. **Implications for education** involve shifting from memorization to interpretation, developing AI collaboration skills, and leveraging Experience Layer access to accelerate competence development.
5. **Industry implications** include potential knowledge democratization, the need for machine-readable code infrastructure, and transferability to other regulatory contexts with appropriate adaptations.
6. **Critical reflection** highlights the need to balance efficiency gains with expertise preservation, creative support, and ethical practice.

The research further increases our awareness of AI's role in building code consultation, in providing both a validated technical framework (EA-RAG) and evidence-based guidance for practice implementation. The key point is that **building code consulting requires not only retrieval but also interpretation**—and this interpretation must be curated from professional expertise rather than from AI models themselves.

CHAPTER 7 CONCLUSIONS AND FUTURE RESEARCH

This chapter summarizes the main conclusions from this work, summarizes both the experimental, technical and practical work, analyzes the limitations and gives ideas for future research. It concludes by an overview of AI and architecture.

7.1 Summary of Contributions

This research addresses the application of AI in architectural code consultation, with contributions spanning three dimensions: empirical exploration, technical innovation, and practical guidance.

7.1.1 Empirical Contributions

Systematic Documentation of Practitioner Needs
As semi-structured interviews with 4 different firms of size (large institute, medium firm, small studio, startup), we observed four core issues of code consultation, with senior expert dependence as the bottleneck. Unlike other studies focusing on technical functions, we explored demand logic behind tool use—the practitioner will need guidance not just document access.

Hierarchical Requirements Framework
The research derived four hierarchically-structured requirements (R1-R4) that directly informed system architecture:

Requirement	Layer	Architectural Response
R1: Verifiable Sources	Foundation	Dual-layer knowledge base with explicit attribution
R2: Expert Interpretation	Core Value	Experience Layer with curated knowledge
R3: Multi-Domain Consultation	Enhancement	Query Router + Domain Expert Modules
R4: Workflow Integration	Application	Structured response format

7.1.2 Technical Contributions

EA-RAG Architecture
This research proposed Experience-Augmented RAG (EA-RAG), a novel architecture specifically designed for building code consultation, featuring:

- **Dual-layer knowledge base**: Selecting known code provisions from expert interpretations without being explicitly linked.
- **Multi-expert domain architecture**: Smooth activation analogous to real consultation “structural problems go to structural engineers”.
- **Multi-expert fusion with conflict detection**: For cross-domain queries requiring synthesis

Validated Performance Improvements
Evaluation demonstrated significant improvements over general-purpose AI (ChatGPT-4o):

Metric	Baseline/ChatGPT	EA-RAG	Improvement
Citation Accuracy	N/A	94%	94%
Hallucination Rate	23%	3%	-20%
Interpretation Quality	— / 2.3	4.1	+78%
Complex Query Performance	— / 1.8	4.0	+122%
Domain Routing Accuracy	N/A	93%	New capability

Architect-Centric Evaluation Framework
We developed a multiple-dimensional evaluation framework in terms of technical measurements (citation accuracy, hallucination rate) and professional measurements

(Citation reliability, interpretation quality, risk awareness, professional usefulness) that can be used in future architectural AI tool development.

7.1.3 Practical Contributions

Actionable Recommendations for Tool Developers
Based on empirical needs and technical validation:

- Prioritize Experience Layer development - interpretation is the value difference.
- Introduce multi-expert approach to domain-specific routing.
- Ensure transparent attribution of code text to interpretation.
- Design structured output for workflow integration.

Implementation Guidance for Architectural Firms
Scale-specific strategies:

Firm Scale	Implementation Priority
Large Institute	Institutional knowledge capture, senior expert load distribution
Medium Firm	Efficiency optimization, experience transfer acceleration
Small Studio	Cross-domain expert access, non-standard project support

Framework for Assessing AI Tool Adoption
Dimensional estimation of technical performance, organizational compatibility, professional acceptance, and risk management for evidence-based tool selection decisions.

7.2 Research Limitations

7.2.1 Methodological Limitations

Sample Size: There were four firms presenting rich qualitative information with limited statistical generality. Pain points and requirements may not reflect industry-wide patterns.
Single Regulatory Context: We have researched Chinese building codes. Different problems exist across regulatory systems but certain technical implementations and recommendations need to be verified elsewhere.
Prototype vs. Production System: EA-RAG was considered a prototype that is feasible and reliable enough to be used on test examples. Development would require more engineering of scale, reliability, and integration.

7.2.2 Scope Limitations

Code Consultation Focus: We asked about code consulting (identifying and interpreting requirements) rather than fully compliance checking (checking complete designs), which are complementary but different problems.
Textual Queries Only: All text-based queries and documents were considered, for drawing analysis and multi-modal (text + graphics) consults.
Knowledge Base Coverage: The Experience Layer consists of 168 curated entries in 6 domains, and is only valid for common situations, possibly not for edge cases or special building types.

7.3 Recommendations for Future Research

7.3.1 Short-Term Research Directions

Expanded Field Studies

- Larger sample across more regions to validate pain point patterns
- Longitudinal studies tracking actual EA-RAG adoption and impact
- Quantitative industry survey measuring AI tool acceptance and usage

Technical Refinements

- Integration with BIM systems for design-phase consultation
- Multi-modal systems processing both text queries and architectural drawings
- Dynamic code update handling as regulations are revised

Human-AI Interaction Studies

- Trust calibration research: What factors affect practitioner trust in AI outputs?
- Expert-novice differential impacts: How does AI affect skill development?
- Decision-making process studies: How does AI change consultation workflows?

7.3.2 Long-Term Research Directions

Ecosystem Development

- Connections with government digital regulatory services
- Industry-wide shared knowledge platforms integrating expert annotations across firms
- International harmonization: Cross-linguistic and cross-regulatory knowledge mapping

Impact Studies

- Effect on design quality and safety outcomes
- Changes in project timelines and costs
- Profession-wide skill evolution and educational implications

Comparative Studies

- Cross-national regulatory AI systems: Eurocodes, IBC, other frameworks
- Applications in related professions: Civil engineering, urban planning

- AI in other architectural knowledge domains: Sustainability, heritage conservation

7.4 Final Reflections

7.4.1 The Promise and Pragmatism of AI in Architecture

This research shows that AI can improve code consultation time and accuracy –EA-RAG's +78% interpretation quality improvement and 87% hallucination reduction are significant gains; however, pragmatism: AI cannot substitute professional judgment in uncertain regulations.
The key finding is that building code consultation **requires not only the correct retrieval, but also the interpretative information.** The general purpose AI cannot capture the experience that makes regulatory text work. EA-RAG captures and organizes the experience with transparency of the sources.

7.4.2 Technology as Enabler, Not Replacement

The real value of AI is that it enabling the professional expertise not replace them. The senior architects' complete knowledge-regulatory information, design experience, value judgment cannot be directly reproduced by AI. EA-RAG's Experience Layer captures part of this knowledge and curation is even necessary. The ideal model for human-AI collaboration is human-ai collaboration: AI provides accurate, efficient information and designers are creative decision-makers and difficult problem solving. EARAG's structured output (recommendations, risk alerts, explicit uncertainty) supports this collaboration in the form of informing and not giving professional judgment.

7.4.3 The Evolving Nature of Architectural Practice

AI integration is promoting evolution across multiple dimensions:
Workflow Evolution: Traditional sequential code consulting is translated into human-AI collaboration, with AI answering routine queries and flagging problems with expert assistance.
Skill Evolution: Examination to interpretive skills and AI collaboration skills. Junior architects need new skills: queries, validating outputs, recognizing AI limitations.
Industry Evolution: Knowledge sharing is increasing as knowledge gained from experience becomes more accessible and flexible. Competition may change as small firms gain access to consultative support previously offered to big institutes.

7.4.4 Closing Thoughts

The research of using AI for architectural code review is a continuing process. Our work establishes a foundation, demonstrating that a domain-adapted RAG architecture can achieve professional performance, identify challenges that any effective system has to fulfill and offers a foundation for real-world deployment.
Goal is not to automate professional judgment but to make sure that practitioners take their minds at the creative and human aspects of design that shape their profession. EA-RAG's +122% improvement on complex multi-domain queries shows that this is possible: AI can enable expert knowledge at scale, which addresses the bottleneck that keeps practitioners at all firms.
Future work should focus on such a topic as testing EA-RAG in production environments, other regulatory settings, and on integrating the development tool. The architectural profession is at an era in which AI can alleviate long-standing waste and, yet, it needs continued collaboration among technologists, practitioners, and teachers to ensure that AI helps an enterprise while preserving the ideals of architecture.

REFERENCES

[1] UpCodes. (2024). Navigating the challenges of manual code compliance. *UpCodes Blog*. Retrieved from https://up.codes/a/navigating-the-challenges-of-manual-code-compliance

[2] AIA (American Institute of Architects). (2013). *Emerging Professional's Companion: 2D Codes & Regulations*. American Institute of Architects. Retrieved from https://content.aia.org/sites/default/files/2017-03/EPC_Codes_Regulations_2D.pdf

[3] Lumber Financial. (2023). Mastering building code compliance: Challenges and solutions for construction professionals. *Lumberfi Blog*. Retrieved from https://www.lumberfi.com/blog/understanding-and-overcoming-common-challenges-in-building-code-compliance

[4] Minute7. (2024). Navigating compliance and building codes for modern architecture firms. *Minute7 Blog*. Retrieved from https://minute7.com/blog/navigating-compliance-and-building-codes-for-modern-architecture-firms

[5] BIM Heroes. (2023, August 26). Navigating building codes and regulations: Ensuring compliance in architectural design. *BIM Heroes Blog*. Retrieved from https://bimheroes.com/navigating-building-codes-and-regulations/

[6] Ghannad, P., Lee, Y. C., Dimyadi, J., & Solihin, W. (2022). Towards fully-automated code compliance checking of building regulations: Challenges for rule interpretation and representation. In *Proceedings of the CIB W78 Conference 2022*. https://doi.org/10.1088/1755-1315/1101/9/092024

[7] NumberAnalytics. (2024). Ultimate guide: Code compliance in architectural management. *NumberAnalytics Blog*. Retrieved from https://www.numberanalytics.com/blog/ultimate-guide-code-compliance-architectural-management

[8] Lee, J., Jung, W., & Baek, S. (2024). In-house knowledge management using a large language model: Focusing on technical specification documents review. *Applied Sciences, 14*(5), Article 2096. https://doi.org/10.3390/app14052096

[9] Meroit. (2024, January 3). Best strategies for code compliance in architectural design. *Medium*. Retrieved from https://medium.com/@meroit/best-strategies-for-code-compliance-in-architectural-design-2d178a11fadd

[10] Wang, S., Zhang, Y., Liu, H., Chen, X., Li, M., & Yang, J. (2024). A review of large language models: Fundamental architectures, key technological evolutions, interdisciplinary technologies integration, optimization and compression techniques, applications, and challenges. *Electronics, 13*(24), Article 5040. https://doi.org/10.3390/electronics13245040

[11] AWS (Amazon Web Services). (2024). *What is RAG? - Retrieval-Augmented Generation AI Explained*. Amazon Web Services. Retrieved from https://aws.amazon.com/what-is/retrieval-augmented-generation/

[12] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474.

[13] IBM. (2024). *Retrieval Augmented Generation (RAG) pattern*. IBM Architecture Center. Retrieved from https://www.ibm.com/architectures/patterns/genai-rag

[14] Zhang, J., Liang, Z., Zhao, W., Yin, H., & Qu, M. (2024). ArchGPT: Harnessing large language models for supporting renovation and conservation of traditional architectural heritage. *npj Heritage Science, 12*, Article 45. https://doi.org/10.1038/s40494-024-01334-x

[15] Soliman, M., & Keim, J. (2024). Do large language models contain software architectural knowledge? An exploratory case study with GPT. In *Proceedings of the*

IEEE International Conference on Software Architecture. https://doi.org/10.5445/IR/1000178724

[16] Zhou, Y., Zheng, Z., Lin, J., & Lu, X. (2022). Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry, 142*, Article 103746. https://doi.org/10.1016/j.compind.2022.103746

[17] Integrated BIM. (2024, November 8). The impact of code compliance construction in architecture. *Integrated BIM Blog*. Retrieved from https://integratedbim.com/code-compliance-construction-in-architecture/

[18] Allen, B. P., Stork, L., & Groth, P. (2023). Knowledge engineering using large language models. *Transactions on Graph Data and Knowledge, 1*(1), Article 3. https://doi.org/10.48550/arXiv.2310.00637

[19] Dhar, R., Vaidhyanathan, K., & Varma, V. (2024). Can LLMs generate architectural design decisions? An exploratory empirical study. In *Proceedings of the IEEE 21st International Conference on Software Architecture* (pp. 79–89). IEEE. https://doi.org/10.1109/ICSA59870.2024.00016

[20] SAP Architecture Center. (2024). *Retrieval Augmented Generation (RAG)*. SAP. Retrieved from https://architecture.learning.sap.com/docs/ref-arch/e5eb3b9b1d/3

[21] Umbrex. (2024, October 2). Policy & regulation in data center construction. *Umbrex Resources*. Retrieved from https://umbrex.com/resources/data-center-primer/policy-regulation/

[22] AIA (American Institute of Architects). (2013). *Emerging Professional's Companion: 2D Codes & Regulations*. American Institute of Architects. Retrieved from https://content.aia.org/sites/default/files/2017-03/EPC_Codes_Regulations_2D.pdf

[23] Strucsoft. (2025, July 15). Comparing building standards in North America and Europe. *Strucsoft Blog*. Retrieved from https://strucsoftsolutions.com/blog/building-standards-and-typology-in-north-america-and-europe/

[24] Wikipedia. (2024). Building code. Retrieved from https://en.wikipedia.org/wiki/Building_code

[25] Congress.gov. (2024). Building codes, standards, and regulations: Frequently asked questions (CRS Report R47665). Library of Congress. Retrieved from https://www.congress.gov/crs-product/R47665

[26] CCR-Mag. (2024, April 7). U.S. building codes/standards vs other countries? Retrieved from https://ccr-mag.com/u-s-building-codes-standards-vs-other-countries/

[27] ICC (International Code Council). (2023, August 29). Comparing building regulatory systems in New Zealand, Australia and the United States. Retrieved from https://www.iccsafe.org/building-safety-journal/bsj-dives/comparing-building-regulatory-systems-in-new-zealand-australia-and-the-united-states/

[28] IRCC (Inter-jurisdiction Regulatory Collaboration Committee). (2004). *Performance-Based Building Regulatory Systems: Principles and Experiences*. Retrieved from https://ircc.info/Doc/A1163909.pdf

[29] Architectural Technology. (n.d.). Hierarchy of building regulations and standards. Retrieved from https://architecturaltechnology.com/member-homepage/technical-standards/hierarchy.html

[30] IRCC. (2003). Performance systems in international building regulations. Retrieved from https://www.ircc.info/Doc/Paper3-DougBeller-CIBKLConference-Codes_Standards.pdf

[31] Congress.gov. (2024). Building codes, standards, and regulations: Frequently asked questions (CRS Report R47665). Library of Congress. Retrieved from https://www.congress.gov/crs-product/R47665

[32] Umbrex. (2024, October 2). Policy & regulation in data center construction. *Umbrex Resources*. Retrieved from https://umbrex.com/resources/data-center-primer/policy-regulation/

[33] Congress.gov. (2024). Building codes, standards, and regulations: Frequently asked questions (CRS Report R47665). Library of Congress.

[34] Zhou, Y., Zheng, Z., Lin, J., & Lu, X. (2022). Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry, 142*, Article 103746. https://doi.org/10.1016/j.compind.2022.103746

[35] IRCC (Inter-jurisdiction Regulatory Collaboration Committee). (2004). *Performance-Based Building Regulatory Systems: Principles and Experiences*.

[36] Van der Heijden, J. (2009). International comparative analysis of building regulations: An analytical tool. *International Journal of Law in the Built Environment, 1*(1), 9-25. https://doi.org/10.1108/17561450910950238

[37] Wikipedia. (2024). Building code. Retrieved from https://en.wikipedia.org/wiki/Building_code

[38] UpCodes. (2024). Navigating the challenges of manual code compliance. *UpCodes Blog*. Retrieved from https://up.codes/a/navigating-the-challenges-of-manual-code-compliance

[39] NumberAnalytics. (2024). Ultimate guide: Code compliance in architectural management. *NumberAnalytics Blog*. Retrieved from https://www.numberanalytics.com/blog/ultimate-guide-code-compliance-architectural-management

[40] TACK (Communities of Tacit Knowledge). (2019). Tacit knowledge in architecture. *Politecnico di Milano*. Retrieved from https://tacit-knowledge-architecture.com/object/

[41] Kivrak, S., Arslan, G., Dikmen, I., & Birgonul, M.T. (2008). Capturing knowledge in construction projects: Knowledge platform for contractors. *Journal of Management in Engineering, 24*(2), 87-95. https://doi.org/10.1061/(ASCE)0742-597X(2008)24:2(87)

[42] Dixon, N. (2015, January). Transferring expertise: The best way to move tacit knowledge. *Nancy Dixon Blog*. Retrieved from https://www.nancydixonblog.com/2015/01/transferring-expertise-the-best-way-to-move-tacit-knowledge.html

[43] European Commission. (2019). Communities of Tacit Knowledge: Architecture and its Ways of Knowing (TACK ITN, Project ID 860413). *CORDIS*. Retrieved from https://cordis.europa.eu/project/id/860413

[44] BIM Heroes. (2023, August 26). Navigating building codes and regulations: Ensuring compliance in architectural design. *BIM Heroes Blog*.

[45] Minute7. (2024). Navigating compliance and building codes for modern architecture firms. *Minute7 Blog*.

[46] Polanyi, M. (1966). *The Tacit Dimension*. New York: Doubleday.

[47] Schindler, S. (2015). Architectural exclusion: Discrimination and segregation through physical design of the built environment. *Yale Law Journal, 124*(6), 1934-2024.

[48] Smart Tribune. (2024, June 6). What is tacit knowledge & how to capture and share it at work. *Smart Tribune Blog*. Retrieved from https://blog.smart-tribune.com/en/tacit-knowledge

[49] Benderoth, M., Gebhard, P., Keller, C., Nakhosteen, C. B., Schaffer, S., & Schneeberger, T. (2025). Socially interactive agents for preserving and transferring tacit knowledge in organizations. *arXiv preprint*. https://arxiv.org/abs/2508.19942

[50] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877-1901.

[51] OpenAI. (2020). GPT-3: Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[52] Raschka, S. (2025). The big LLM architecture comparison. *Sebastian Raschka's Magazine*. Retrieved from https://magazine.sebastianraschka.com/p/the-big-llm-architecture-comparison

[53] Dhar, R., Vaidhyanathan, K., & Varma, V. (2024). Can LLMs generate architectural design decisions? An exploratory empirical study. In *Proceedings of the IEEE 21st International Conference on Software Architecture* (pp. 79–89). https://doi.org/10.1109/ICSA59870.2024.00016

[54] Allen, B. P., Stork, L., & Groth, P. (2023). Knowledge engineering using large language models. *Transactions on Graph Data and Knowledge, 1*(1), Article 3.

[55] Wang, S., Zhang, Y., Liu, H., Chen, X., Li, M., & Yang, J. (2024). A review of large language models: Fundamental architectures, key technological evolutions, interdisciplinary technologies integration, optimization and compression techniques, applications, and challenges. *Electronics, 13*(24), Article 5040.

[56] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12), 1-38.

[57] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., ... & Liu, T. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

[58] AWS (Amazon Web Services). (2024). *What is RAG? - Retrieval-Augmented Generation AI Explained*. Retrieved from https://aws.amazon.com/what-is/retrieval-augmented-generation/

[59] Latenode. (2024). RAG diagram guide: Visual architecture of retrieval-augmented generation. Retrieved from https://latenode.com/blog/ai-frameworks-technical-infrastructure/rag-retrieval-augmented-generation/rag-diagram-guide

[60] Cloudflare. (2024). Retrieval Augmented Generation (RAG) reference architecture. Retrieved from https://developers.cloudflare.com/reference-architecture/diagrams/ai/ai-rag/

[61] Wikipedia. (2024). Retrieval-augmented generation. Retrieved from https://en.wikipedia.org/wiki/Retrieval-augmented_generation

[62] LangChain. (2024). Build a retrieval augmented generation (RAG) app: Part 1. Retrieved from https://python.langchain.com/docs/tutorials/rag/

[63] NVIDIA. (2024, August 22). RAG 101: Demystifying retrieval-augmented generation pipelines. *NVIDIA Technical Blog*. Retrieved from https://developer.nvidia.com/blog/rag-101-demystifying-retrieval-augmented-generation-pipelines/

[64] Mujtaba, H. (2024, July 7). Understanding the RAG architecture model: A deep dive into modern AI. *Medium*. Retrieved from https://medium.com/@hamipirzada/understanding-the-rag-architecture-model-a-deep-dive-into-modern-ai-c81208afa391

[65] Katz, D. M., Bommarito, M. J., Gao, S., & Arredondo, P. (2023). GPT-4 passes the bar exam. *Philosophical Transactions of the Royal Society A, 382*(2270), 20230254.

[66] Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., ... & Natarajan, V. (2023). Large language models encode clinical knowledge. *Nature, 620*(7972), 172-180.

[67] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., ... & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*.

[68] Eastman, C., Lee, J. M., Jeong, Y. S., & Lee, J. K. (2009). Automatic rule-based checking of building designs. *Automation in Construction, 18*(8), 1011-1033.

[69] Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. New York: Basic Books.

[70] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. *Proceedings of ICLR 2020*.

[71] Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9*(4), e1312.

[72] Choudhury, A., & Shamszare, H. (2023). Investigating the impact of AI on professional work: An analytical framework. *Harvard Business Review Digital Articles*, 2-11.

Appendix

[Qualitative Case Study text：CHATGPT-4o VS EA-RAG(OUR MODEL)]

[EA-RAG project document]
[Video Demo text]