

Conversational AI-Powered Multi-Agent System for Mobile Application Accessibility Compliance: A RAG-Enhanced Pipeline Design

Nayeon Kim

Media Research Division

Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea
nyeon@etri.re.kr

Byungjun Bae*

Media Research Division

Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea
1080i@etri.re.kr

Abstract—Mobile application accessibility compliance remains a significant challenge for developers, requiring extensive manual consultation of complex guidelines such as WCAG 2.1 and Korean Mobile Application Content Accessibility Guidelines 2.1 (KS X 3253:2025). This paper presents a novel dual-agent pipeline design leveraging Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) to streamline accessibility analysis and solution guidance. Our proposed system consists of two specialized agents: a Problem Analysis Agent that provides detailed accessibility issue examination and guideline mapping, and a Solution Support Agent that generates contextual remediation strategies. The pipeline integrates vector-based knowledge retrieval from accessibility documentation, potentially reducing the manual effort required for developers and evaluators to consult guidelines and implement accessibility improvements. This work contributes a systematic pipeline architecture that could significantly enhance the efficiency of mobile accessibility compliance workflows.

Index Terms—Mobile Accessibility, LLM Agents, RAG, WCAG 2.1, Pipeline Design

I. INTRODUCTION

Mobile application accessibility compliance has become increasingly critical as digital services expand globally, yet the implementation process remains challenging for development teams. Current accessibility evaluation and improvement workflows require extensive manual consultation of complex documentation, including WCAG 2.1 guidelines and region-specific standards such as Korea's KS X 3253:2025. Developers and accessibility evaluators must frequently navigate between multiple documentation sources, interpret guideline requirements in specific contexts, and manually research appropriate implementation solutions.

The traditional workflow for accessibility compliance involves several time-intensive manual steps: identifying accessibility violations through testing tools, consulting relevant guideline sections to understand requirements, researching platform-specific implementation approaches, and developing context-appropriate solutions. This process often requires significant expertise and can be a bottleneck in development

cycles, particularly for teams without dedicated accessibility specialists.

Recent advances in Large Language Models (LLMs) and multi-agent systems present opportunities to streamline these manual processes through intelligent automation. LLMs demonstrate strong capabilities in document comprehension, contextual analysis, and code generation—core requirements for accessibility guidance workflows. Multi-agent architectures enable task specialization, allowing dedicated agents to focus on specific aspects of the accessibility compliance process.

This paper proposes a systematic pipeline design that leverages conversational AI to reduce the manual burden of accessibility compliance workflows. Our approach introduces two specialized agents enhanced with Retrieval-Augmented Generation (RAG) capabilities to automate the document consultation and solution research processes that currently require extensive human effort.

II. RELATED WORK

A. Current Accessibility Compliance Workflows

Existing accessibility compliance processes rely heavily on manual documentation consultation and expert interpretation. Developers typically use automated testing tools like axe-core or platform-specific scanners to identify potential issues, then manually research appropriate solutions through guideline documentation, developer forums, and implementation examples. This workflow is time-intensive and requires significant domain expertise to interpret guidelines correctly in specific development contexts.

Professional accessibility evaluators follow similar manual processes, combining automated scanning with expert analysis to identify violations and recommend remediation approaches. The evaluation process involves extensive cross-referencing between testing results and guideline documentation to provide accurate compliance assessments and actionable recommendations.

*Corresponding author

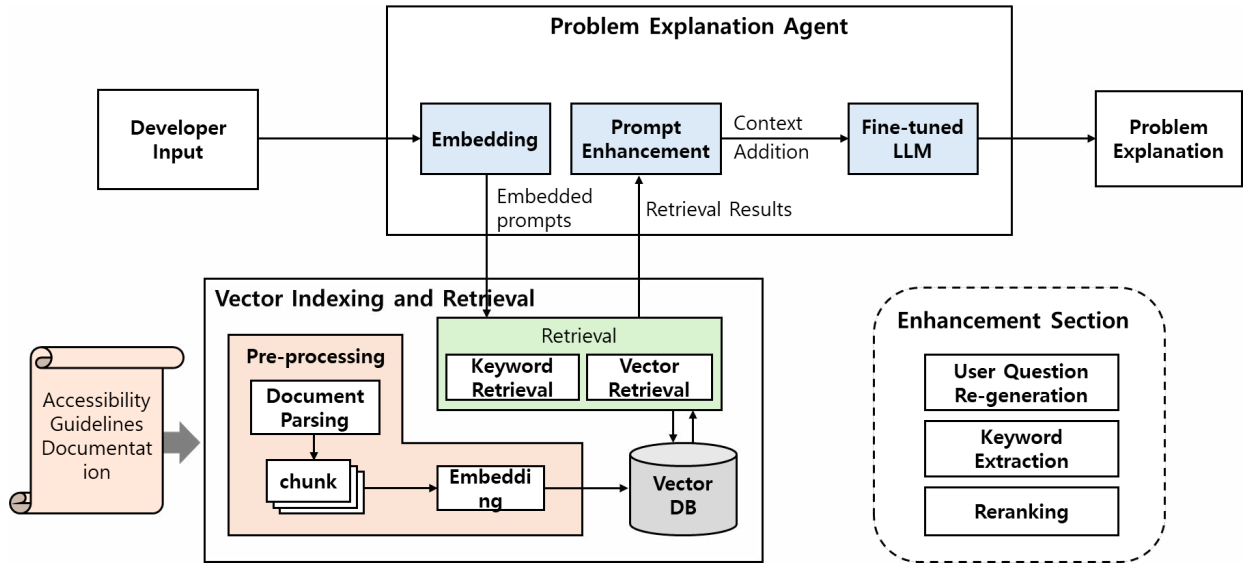


Fig. 1. Problem Analysis Agent Pipeline with RAG Integration

B. LLM Applications in Developer Assistance

Multi-agent LLM systems for software engineering have demonstrated potential for automating complex workflows that traditionally require manual research and analysis. Recent developments show that specialized agent architectures can effectively support code analysis, documentation consultation, and solution generation tasks that parallel accessibility compliance workflows.

RAG-enhanced systems have proven effective for domain-specific developer assistance by enabling LLMs to access current documentation and implementation patterns while maintaining conversational interaction. These approaches demonstrate the feasibility of automating documentation-intensive developer workflows through intelligent retrieval and synthesis.

III. PIPELINE ARCHITECTURE DESIGN

A. Design Philosophy and Objectives

Our pipeline design aims to automate the most time-intensive aspects of accessibility compliance workflows: guideline consultation, issue analysis, and solution research. The system adopts a collaborative dual-agent architecture where each agent specializes in distinct workflow components that currently require extensive manual effort.

The primary design objectives are to:

- Automate guideline consultation and requirement interpretation
- Provide contextual analysis of accessibility issues with relevant documentation references
- Generate platform-specific implementation guidance without manual research
- Reduce the expertise barrier for accessibility compliance

B. Problem Analysis Agent Pipeline

The Problem Analysis Agent is designed to automate the manual process of analyzing accessibility issues and consulting relevant documentation. Currently, this process requires developers or evaluators to:

- Manually identify the nature and scope of accessibility violations
- Research applicable guideline sections across multiple documentation sources
- Interpret guideline requirements in the context of specific implementation scenarios
- Assess the impact and priority of identified issues

Pipeline Components:

- 1) **Issue Input Processing:** Accepts accessibility test results, code snippets, or manual issue descriptions from developers
- 2) **Guideline Retrieval:** Uses RAG to automatically retrieve relevant sections from WCAG 2.1, KS X 3253:2025, and platform-specific documentation
- 3) **Contextual Analysis:** Leverages LLM capabilities to interpret guidelines in the specific context of the reported issue
- 4) **Comprehensive Explanation:** Generates detailed analysis including root cause identification, compliance implications, and user impact assessment

This pipeline could significantly reduce the time developers spend manually consulting documentation and interpreting guideline requirements, particularly for teams without dedicated accessibility expertise.

C. Solution Support Agent Pipeline

The Solution Support Agent targets the manual solution research and implementation guidance process. Traditional workflows require developers to:

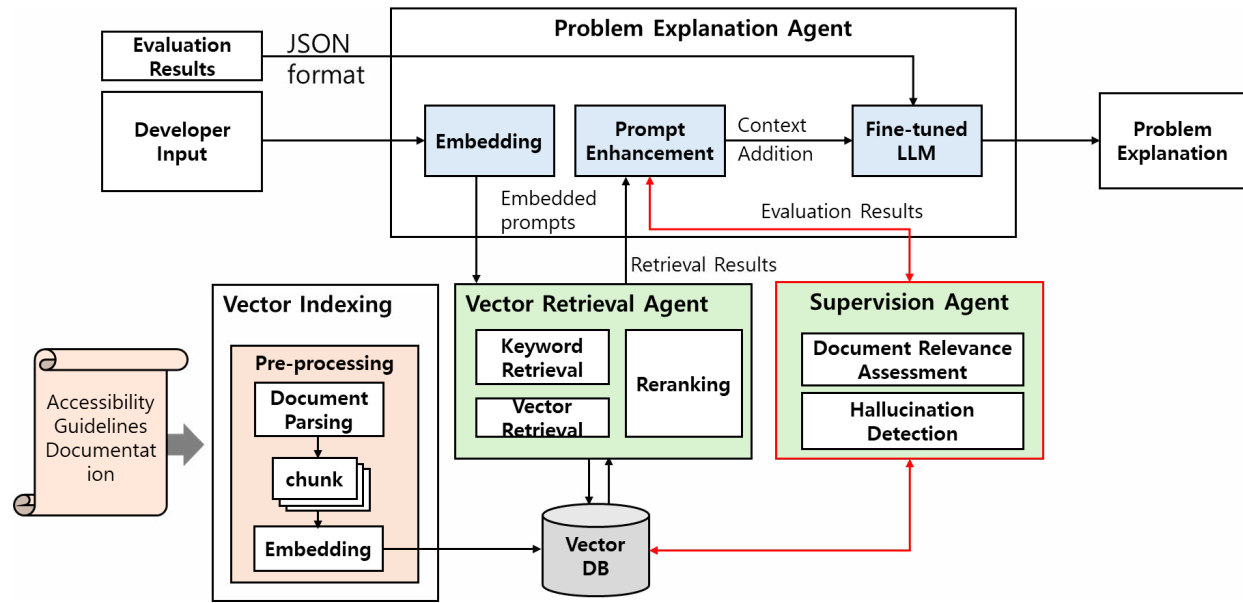


Fig. 2. Solution Support Agent Pipeline with Code Generation

- Research platform-specific implementation approaches across documentation and community resources
- Adapt generic guideline recommendations to specific development contexts
- Locate and understand relevant code examples and implementation patterns
- Develop testing strategies for validating implemented solutions

Pipeline Components:

- 1) **Solution Research:** Automatically retrieves relevant implementation patterns and code examples from curated knowledge base
- 2) **Context Adaptation:** Tailors generic solutions to specific development platforms, frameworks, and application contexts
- 3) **Code Generation:** Provides concrete, executable code examples rather than abstract recommendations
- 4) **Validation Guidance:** Suggests specific testing approaches and validation criteria for implemented solutions

This pipeline could eliminate much of the manual research burden associated with implementing accessibility improvements, enabling developers to focus on integration rather than solution discovery.

D. RAG Knowledge Integration

The RAG system serves as the automated documentation consultation component, replacing manual guideline research with intelligent retrieval. The knowledge base integrates:

Primary Documentation Sources:

- WCAG 2.1 success criteria, techniques, and understanding documents
- Korean Mobile Application Content Accessibility Guidelines 2.1 (KS X 3253:2025)

- iOS VoiceOver and Android TalkBack implementation documentation
- Platform-specific accessibility API references

Implementation Pattern Repository:

- Curated code examples demonstrating accessibility best practices
- Common violation patterns and corresponding solution approaches
- Platform-specific implementation variations and considerations

E. Example Interaction

To illustrate the practical workflow of the proposed pipeline, we present a realistic interaction between a developer and the system.

Developer Input: “They’re telling me my app’s popup has to be explicitly closable. What does that even mean?”

Problem Analysis Agent Output:

- References: **KS X 3253:2025, Guideline 9.2.1 (Execution on User Request)** and **WCAG 2.1 Success Criterion 3.2.1 (On Focus)**.
- Explanation: “When a popup appears, if it cannot be closed by the user or disappears unexpectedly, it reduces accessibility. The guidelines require that users must always have an explicit way to dismiss the popup, such as a visible close button (‘x’ icon).”

Solution Support Agent Output:

- **Practical Fix:** Add a dedicated close button to the popup UI and ensure it is programmatically identifiable by assistive technologies.
- **Validation Guidance:** With TalkBack enabled, the button should be announced as “Close popup, button,” and it should be reachable via keyboard focus.

F. Technical Implementation Notes

Although this paper primarily focuses on the pipeline design, several implementation considerations are worth noting:

Retrieval Strategy:

- Hybrid retrieval combining semantic embedding search (e.g., SBERT, OpenAI embeddings) with keyword-based filtering.
- Ensures both conceptual matches and platform-specific filters.

LLM Model Candidates:

- Lightweight: *Gemma-7B*, *Llama 3.1-8B*.
- Reasoning-heavy: *Qwen-14B*, *GPT-4o*.

Knowledge Base Updating:

- Version-controlled ingestion of WCAG and KS X 3253 guidelines.
- Incremental updates with expert validation.
- Periodic refresh aligned with standard revisions.

G. Innovation over Single-Agent RAG Systems

While single-agent RAG-based chatbots can provide basic documentation lookup, they suffer from shallow analysis and hallucinated fixes. The proposed dual-agent system introduces structural innovations:

IV. INNOVATION OVER SINGLE-AGENT SYSTEMS

This separation of responsibilities provides stronger interpretability, reduces error propagation, and enhances adaptability to evolving accessibility standards.

V. POTENTIAL IMPACT ANALYSIS

A. Workflow Efficiency Improvements

The proposed pipeline design could address several time-intensive aspects of current accessibility compliance workflows:

Documentation Consultation Automation: Currently, developers spend significant time navigating between WCAG 2.1, platform-specific guidelines, and implementation documentation. The RAG-enhanced agents could provide instant access to relevant guideline sections with contextual interpretation.

Solution Research Streamlining: The manual process of researching implementation approaches across developer documentation, forums, and code repositories could be automated through the Solution Support Agent’s curated knowledge base and code generation capabilities.

Expertise Democratization: Teams without dedicated accessibility specialists could access expert-level analysis and recommendations, potentially reducing the knowledge barrier for accessibility compliance.

B. Developer and Evaluator Support

For Developers: The system could transform accessibility implementation from a research-intensive process to a guided workflow, providing immediate access to contextual requirements and implementation patterns without manual documentation consultation.

For Accessibility Evaluators: The Problem Analysis Agent could accelerate the evaluation process by automatically generating comprehensive issue analyses with proper guideline references, allowing evaluators to focus on complex contextual assessments rather than routine documentation lookup.

For Project Teams: Consistent, guideline-aligned recommendations could improve communication between developers, designers, and evaluators by providing shared understanding of accessibility requirements and implementation approaches.

C. Compliance Quality Enhancement

The pipeline design could potentially improve compliance outcomes by:

- Ensuring consistent application of current guideline standards across projects
- Reducing interpretation errors that occur during manual guideline consultation
- Providing comprehensive solution coverage that might be missed in manual research
- Enabling proactive accessibility consideration throughout development cycles

VI. IMPLEMENTATION CONSIDERATIONS

A. Knowledge Base Curation

Successful deployment would require careful curation of the RAG knowledge base to ensure accuracy and currency. This includes:

- Systematic integration of official guideline documentation with proper versioning
- Validation of implementation examples against current platform APIs
- Regular updates reflecting evolving accessibility standards and platform capabilities

B. Agent Specialization

The dual-agent architecture requires careful role definition to avoid overlap and ensure comprehensive coverage:

- Clear handoff protocols between analysis and solution phases
- Consistent terminology and reference frameworks across agents
- Validation mechanisms to ensure agent outputs align with current standards

TABLE I
COMPARISON OF SINGLE-AGENT VS. DUAL-AGENT RAG APPROACHES

Aspect	Single RAG-Chatbot	Dual-Agent System
Task Scope	Unified handling of analysis and solution generation	Specialized roles: diagnosis vs. prescription
Analysis Depth	Often shallow and conflated	Clear separation enables guideline-grounded diagnosis first
Error Propagation	Hallucinated fixes without references	Retrieval-grounded analysis reduces hallucinations
Responsibility	Blurred accountability	Separated responsibilities improve trust
Scalability	Hard to adapt across domains	Modular, extendable independently

VII. FUTURE WORK

A. Implementation and Validation

Future research should focus on implementing the proposed pipeline design and conducting comprehensive evaluation studies. Key areas include:

Prototype Development: Building functional prototypes of both agents using current LLM and RAG technologies to validate the pipeline design’s feasibility.

Document Relevance Monitoring: Supervising how accurately the system retrieves and references appropriate guideline sections for specific accessibility issues.

Hallucination Supervision: Monitoring generated recommendations to ensure alignment with actual guideline requirements and prevent fabricated or inaccurate information.

User Experience Research: Evaluating developer and evaluator adoption patterns, satisfaction levels, and integration challenges in real-world development environments.

B. Advanced Capabilities

Predictive Analysis: Extending the system to identify potential accessibility issues during the design phase, before implementation begins.

Continuous Learning: Implementing feedback mechanisms to improve recommendations based on developer adoption patterns and real-world implementation outcomes.

Multi-Modal Integration: Incorporating visual and interaction analysis capabilities to evaluate accessibility of multimedia content and complex user interface patterns.

C. Broader Applications

Cross-Platform Adaptation: Extending the pipeline design to support web accessibility compliance and other digital accessibility domains.

Organizational Integration: Developing enterprise-level implementations that integrate with organizational development workflows, compliance tracking, and training programs.

Standards Evolution Support: Creating mechanisms to rapidly incorporate evolving accessibility standards and emerging platform capabilities into the knowledge base.

VIII. CONCLUSION

This paper presents a systematic pipeline design for automating key components of mobile accessibility compliance workflows through LLM-based multi-agent systems.

The proposed dual-agent architecture with RAG could significantly reduce the effort required for accessibility issue analysis and guideline interpretation, while the Solution Support Agent pipeline could streamline the implementation research and code development process. By automating these documentation-intensive workflows, the system could make accessibility compliance more accessible to development teams while potentially improving the consistency and quality of implementation outcomes.

The pipeline design contributes a structured approach to applying conversational AI technologies to accessibility compliance challenges, providing a foundation for future implementation and evaluation research. Success in this domain could demonstrate broader applications for AI-assisted compliance workflows across digital accessibility standards.

Future work should focus on implementing and validating the proposed pipeline design through prototype development and comprehensive user studies. Quantitative evaluation of workflow efficiency improvements and qualitative assessment of user experience will be essential for demonstrating the practical value of this approach to mobile accessibility compliance.

ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-02303220, Development of a system for evaluating and validating the UI/UX of digital services for underprivileged groups).

REFERENCES

- [1] W3C Web Accessibility Initiative, “Web Content Accessibility Guidelines (WCAG) 2.1,” W3C Recommendation, 2018.
- [2] Korean Agency for Technology and Standards, “Mobile Application Contents Accessibility Guidelines 2.1 (KS X 3253:2025),” Korean Standards, 2025.
- [3] Deque Systems, “Advancing AI for axe: The next leap in digital accessibility,” Technical Report, 2024.
- [4] H. Chen et al., “LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision and the Road Ahead,” *ACM Transactions on Software Engineering and Methodology*, 2024.
- [5] Microsoft Research, “Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks,” Technical Report, 2024.
- [6] GitHub, Inc., “What is retrieval-augmented generation, and what does it do for generative AI?” Technical Blog, 2024.
- [7] Apple, Inc., “Accessibility Programming Guide for iOS,” Developer Documentation, 2024.
- [8] Google, Inc., “Android Accessibility - Designing for Accessibility,” Developer Documentation, 2024.