



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X
Impact Factor: 7.056

RAG-Based Multi-Agent Legal Decision Support System with Virtual Trial Simulation: Design and Implementation of the Justice Chamber

Jiho Choi¹; Myung-Soo Shin²; Seok-Gyu Choi³; JEONNG WOO Park⁴; Seungjae Lee^{5*}

^{1,2,3,4,5}Department of Computer Science, Sunmoon University, Korea

¹ vpdpdlfl@sunmoon.ac.kr; ² ssuk83158315@gmail.com; ³ tjrrb0313@sunmoon.ac.kr;
⁴ pjw049514@sunmoon.ac.kr; ^{5*} leeko@sunmoon.ac.kr (Corresponding Author)

DOI: <https://doi.org/10.47760/ijcsmc.2026.v15i09.010>

Abstract: Large Language Models (LLMs) have demonstrated considerable potential for legal information processing and decision support. However, their application to the legal domain remains challenging because inaccurate or unsupported information, including fabricated statutes and judicial precedents, can undermine the reliability of legal consultation. This study presents Justice Chamber, an intelligent legal decision support system that combines Retrieval-Augmented Generation (RAG) with stateful multi-agent virtual trial simulation. The proposed system retrieves relevant statutes, Supreme Court precedents, and sentencing information through a vector database and incorporates the retrieved evidence into the reasoning process of multiple legal agents. The system automatically classifies submitted cases into civil or criminal categories and generates supplementary questions when essential case information is missing. It then conducts a multi-round adversarial simulation involving legal advocate agents, followed by parallel adjudication through three judicial perspectives: a Strict Principle Judge, an Equity-Centric Judge, and a Public Opinion Judge. A Master Judge subsequently integrates the resulting perspectives into a final decision report. In addition, a lightweight sentiment analysis pipeline is employed to incorporate public opinion information derived from Korean news articles. The system was implemented using technologies including Azure OpenAI, ChromaDB, LangGraph, Sentence Transformers, FastAPI, and PostgreSQL. For functional validation, two actual Supreme Court cases were used as ground-truth references while withholding their original decisions from the simulation. The system correctly classified both cases, identified their core legal principles, and produced judgment directions consistent with the corresponding Supreme Court decisions. These results demonstrate the feasibility of combining evidence-grounded retrieval and multi-agent adversarial reasoning for interactive legal decision support. The proposed framework provides a foundation for developing more transparent, structured, and context-aware AI-assisted legal consultation systems.

Keywords: Legal Decision Support, Retrieval-Augmented Generation, Multi-Agent Systems, Virtual Trial Simulation, LegalTech, ChromaDB, Sentiment Analysis, Large Language Models



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

I. INTRODUCTION

The rapid development of Large Language Models (LLMs) has expanded the application of artificial intelligence from general information retrieval to specialized professional domains. In the legal domain, LLMs have attracted particular attention because they can process large volumes of legal documents, summarize judicial decisions, generate legal arguments, and provide interactive responses to users. However, legal applications require a substantially higher level of factual reliability than general-purpose conversational systems. A response containing an incorrect statutory provision, an inaccurate interpretation, or a fabricated judicial precedent may directly affect a user's understanding of legal rights and obligations.

One of the major challenges in applying LLMs to legal services is hallucination, in which a model generates information that appears plausible but is not supported by authoritative sources. This problem is particularly significant in legal consultation because statutes and judicial precedents provide the evidentiary foundation for legal reasoning. In addition, the legal environment is continuously updated, meaning that relying exclusively on the parametric knowledge of a pre-trained model may result in outdated or contextually inappropriate responses. Therefore, an effective legal AI system should connect generative reasoning with an external and traceable knowledge source.

Retrieval-Augmented Generation (RAG) provides an effective approach to this problem by combining information retrieval with language generation. Instead of relying solely on information encoded in the model parameters, a RAG system retrieves relevant documents and incorporates them into the generation process. In a legal environment, this architecture can be used to retrieve statutes, judicial precedents, and other authoritative legal materials before generating a response. The proposed system therefore adopts a vector-based retrieval architecture using ChromaDB and semantic embeddings to provide legal agents with case-relevant evidence.

Another limitation of conventional legal chatbots is their reliance on a single-agent interaction model. Legal disputes generally involve multiple parties with competing interests and require the examination of arguments from different perspectives. A single LLM response may therefore provide only one interpretation of a case without explicitly exposing alternative arguments or counterarguments. To address this limitation, the proposed system employs a multi-agent architecture in which different agents assume specialized legal roles. Advocate agents construct opposing arguments through multiple rounds of interaction, while separate judicial agents evaluate the accumulated arguments from different decision-making perspectives.

The proposed system, Justice Chamber, extends this concept into an interactive virtual trial simulation. When a user submits a case description, the system first classifies the case and identifies potentially missing information. Additional questions are generated when necessary to enrich the case context before legal reasoning begins. Relevant legal materials are subsequently retrieved from separate knowledge collections, and the advocate agents conduct a multi-round adversarial debate. The resulting arguments are then evaluated by three judicial sub-agents representing statutory principles, equitable considerations, and public opinion. A Master Judge integrates these outputs into a final decision report.

The system also incorporates public sentiment as an auxiliary information source. Korean news articles are processed through a dedicated sentiment-analysis pipeline, and the resulting sentiment information is provided to the Public Opinion Judge. This component is designed not to replace statutory or precedent-based reasoning, but to represent the potential influence of social sentiment as one of several analytical dimensions within the simulation.

To evaluate the feasibility of the proposed architecture, two actual Supreme Court cases were selected as ground-truth references. The original judgments were withheld from the simulation, and only the factual circumstances of the cases were provided to the system. The resulting outputs were then compared with the corresponding Supreme Court decisions. In the reported experiments, the system correctly classified both cases, identified their principal legal issues, and produced judgment directions consistent with the actual decisions.



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

The main contributions of this study are therefore threefold. First, it presents an evidence-grounded legal decision support architecture that combines RAG with structured legal reasoning. Second, it introduces a stateful multi-agent virtual courtroom that enables adversarial argument generation and multi-perspective adjudication. Third, it demonstrates the feasibility of the proposed architecture through functional evaluation and comparison with actual Supreme Court decisions. Through these components, Justice Chamber aims to provide a more structured and transparent approach to AI-assisted legal consultation and decision support..

II. RELATED WORK

The architecture of intelligent decision support systems in highly specialized domains has evolved significantly through advancements in knowledge anchoring and multi-agent orchestration. The foundational paradigm of Retrieval-Augmented Generation (RAG), formulated by Lewis *et al.* (2020), established that combining parametric memory with non-parametric memory substantially mitigates factual errors in knowledge-intensive natural language processing tasks. In the legal sector, where statutory updates and precise judicial precedents dictate logical outcomes, vanilla foundational models inherently suffer from temporal cutoff boundaries and contextual degradation. To circumvent these limitations, contemporary Legaltech frameworks leverage dense vector retrievers and specialized similarity metrics within systems like ChromaDB to inject immutable legal contexts directly into the model's attention window, effectively suppressing domain-specific computational hallucinations. Parallel to retrieval optimization, the deployment of cooperative and adversarial multi-agent systems has emerged as a robust alternative to single-agent prompting architectures. Rather than relying on a solitary, omniscient model instance, distributing specific behavioral personas—such as prosecutors, defense attorneys, and judges—allows for a complex, multi-turn reasoning workflow. As demonstrated in recent multi-agent dialogue research, structured token-exchange loops among adversarial entities foster collaborative criticism and mutual error correction, yielding superior cross-examination rigor and deeper analytical granularity. Furthermore, integrating real-time socio-legal variables, such as public sentiment, has traditionally imposed heavy computational overhead when processed exclusively through large language models. To resolve this efficiency bottleneck, established literature supports the use of localized, rule-based natural language processing pipelines. Lexicon-based sentiment analysis provides a computationally lightweight method to quantify unstructured textual sentiment into discrete numerical vectors without the inference latency or token expense of generative models. By establishing specialized legal-domain weight patches over standard lexicons, recent systems have successfully bridged the gap between rigid statutory boundaries and the fluid dynamics of public interest, laying the groundwork for comprehensive legal simulation frameworks.

III. PROPOSED SYSTEM

A. System Workflow and Strategic Auto-Classification

The proposed system, Justice Chamber, is designed as an end-to-end intelligent legal decision support framework that systematically conceptualizes the multifaceted dynamics of judicial courts. As mapped in the structural system topology and execution routing architecture detailed in this photo, the entire runtime workflow is rigorously managed by a centralized state machine coordinated via LangGraph, which ensures deterministic state management and robust multi-turn agent coordination. When you enter an incident description, the client interface evaluates unstructured narratives to execute automatic segmentation between criminal and civil algorithmic frameworks. This automatic classification branch is strategically important for two distinct reasons: first, nonprofessional users frequently experience cognitive ambiguity when trying to distinguish between civil and criminal jurisdiction. Second, the classification explicitly dictates the structural routing of downstream lawyer persona, which maps prosecutor-to-lawyer composition for criminal proceedings and plaintiff attorney composition for civil proceedings. Furthermore, parsing and identifying case types prior to execution can lead to



significant isolation of vector search boundaries inside ChromaDB, narrowing search windows, which can quickly bring about relevant legal precedents while saving token consumption. Fig. 1 System architectural diagram.

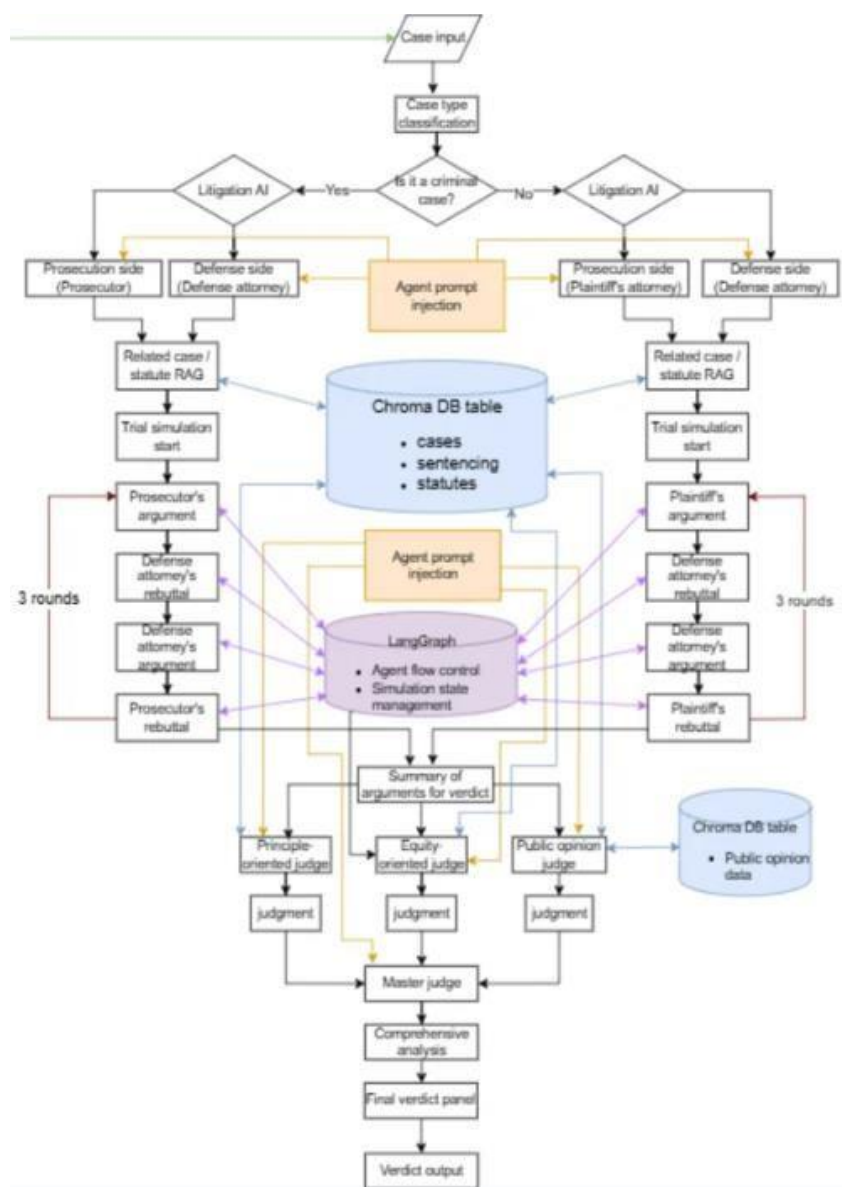


Fig. 1 System architectural diagram



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Fig.2 case input page

B. Interactive Information Supplementation and Query Loops

If the internal structural verification layer identifies an information deficiency within the user's initial baseline inputs, the backend processing pipeline initiates a sequential query generation loop. Instead of initiating the multi-agent debate with incomplete contextual parameters, the system leverages a localized scikit-learn classifier combined with prompt-driven missing feature detection to automatically analyze the case gaps. The system then generates context-aware, targeted supplementary questions regarding crucial missing variables, such as specific financial transaction records, administrative identification identifiers, or formal institutional reporting registries. By prompting the user to supply clear answers to these factual variables, the system dynamically enriches the case context model, stabilizing baseline reasoning boundaries and enhancing the overall precision of the subsequent litigation simulation.



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

"안녕하세요. 너무 억울하고 답답해서 시뮬레이션을 돌려봅니다. 지난주 토요일(5월 23일)에 유명 중고거래 사이트에서 평소 갖고 있던 태블릿 PC가 시세보다 조금 저렴하게 올라온 것을 보고 판매자에게 연락을 했습니다. 판매자는 폰인을 직장인이라고 소개하며, 오늘 당장 돈이 급해 싸게 처분하는 것이니 신임금을 해주면 바로 편의점 택배로 부치고 송장 번호를 보내주겠다고 했습니다. 판매자의 해너 온도가 높고 사기 조희 사이트에도 연락처가 뜨지 않아서 별 의심 없이 알려준 계좌번호로 45만 원을 송금했습니다. 하지만 돈을 받은 직후부터 판매자는 "지금 회의 중이다", "퇴근하고 보내주겠다"라며 핑계를 대며 시간을 끌기 시작했습니다. 급가야 다음 날부터는

03 / AI 피싱 질문

Q1. 판매자와 실명, 계좌번호 등의 정보 외에 추가적으로 신원을 확인할 수 있는 자료(예: 전화번호, 이메일 주소)가 있는지 확인이 필요합니다.

상세 답변 입력

Q2. 송금 당시의 은행 송금 기록 및 영수증이 포함되어 있는지 확인이 필요합니다.

상세 답변 입력

Q3. 현재 경찰에 피해 신고를 접수했거나, 접수할 계획이 있는지 확인이 필요합니다.

상세 답변 입력

Q4. 다른 피해자들로부터 추가적인 증거 자료나 진술을 확보할 수 있는지 여부가 확인이 필요합니다.

상세 답변 입력

Fig. 3 Additional Input plus Screen

C. Hybrid RAG Pipeline and Stateful Adversarial Simulation Loop

To enforce maximum evidentiary anchoring and eliminate domain-specific hallucinations, the architecture integrates a cross-layered, dual-track Retrieval-Augmented Generation (RAG) pipeline querying distinct collections within a centralized ChromaDB ecosystem. High-performance statutory frameworks (statutes), historical Supreme Court precedents (cases), and algorithmic sentencing templates (sentencing) are vectorized via OpenAI's text-embedding-3-small encoder to establish a dense semantic retrieval layer. In parallel, a localized, open-source Sentence-Transformers pipeline is deployed to capture the nuanced rhetorical expressions embedded within colloquial legal arguments and individual judicial opinions, which are indexed separately under the Opinion collection.

Once the enriched state is bound, LangGraph initiates an automated, stateful 3-round adversarial turn-taking loop between the assigned advocate agents as explicitly defined in the iterative loops of image_fe7b41.jpg.



During Round 1, both opposing agents independently dispatch semantic queries to the ChromaDB cluster to harvest supporting evidentiary contexts and construct their primary legal pleadings. Round 2 triggers a strict cross-examination phase where agents systematically evaluate the opponent's historical state vector to emit targeted counter-arguments. Round 3 concludes the litigation phase with the generation of closing arguments and final summations. To maximize user engagement and system transparency, the textual deltas generated during these sequential turns are pushed to the frontend user dashboard via Server-Sent Events (SSE) in realtime, rendering a streaming, live-courtroom visualization without full-page re-rendering overhead.

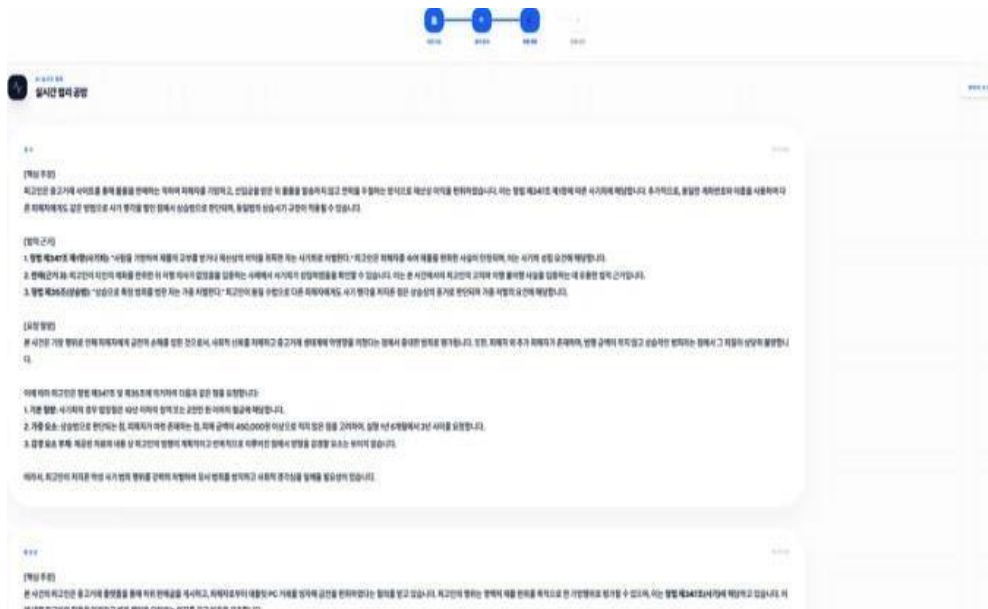


Fig. 2 SSE Streaming Screen

D. Parallel Multi-Philosophy Adjudication Layer

Following the summation of pleadings, the accumulated immutable graph state is broadcast simultaneously to three parallel judicial sub-agents as charted in the middle tier of Fig. 1 (image_fe7b41.jpg), each simulating a distinct philosophical paradigm of adjudication. The Strict Principle Judge evaluates the argument logs tightly against explicit statutory text bounds and conservative sentencing guidelines. The Equity-Centric Judge calculates mitigation metrics by parsing text parameters for remorse indicators and non-standard defendant contexts. Concurrently, the Public Opinion Judge assesses social sentiment indicators by interacting with an external ChromaDB table dedicated to opinion metrics, as visualized on the lower-right side of Fig. 1 (image_fe7b41.jpg).

This process relies on a separate, lightweight public opinion pipeline that operates externally from the LLM core to optimize inference latency. Real-time news articles harvested via a dedicated web crawler are processed using a custom string-parsing tokenization technique and scored against the hun3359/khue-bert-base-sentiment Korean Sentiment Lexicon, which is manually patched with custom weight modifiers for legal domain tuning. The



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

consolidated sentiment score \$\$\$ determines the final public opinion classification based on a strict threshold boundary, filtering the outcomes into positive, negative, or neutral categories.

E. VERDICT ACCURACY VALIDATION AGAINST REAL SUPREME COURT DECISIONS

To empirically validate the system's judgment accuracy, two real Supreme Court decisions published in 2026 were used as ground-truth references. For each case, only the factual narrative was provided as input, with the actual verdict withheld to prevent leakage into the simulation pipeline.

Case 1 — Civil Jurisdiction (Supreme Court 2025 다 211789)

The case involved an insurance death benefit dispute where the insured sustained injuries during the coverage period but died after the policy expiration. The system correctly classified the case as civil and identified the contra proferentem rule under Article 5(2) of the Act on the Regulation of Terms and Conditions as the core legal principle. The final verdict awarded the full claimed amount of KRW 35,000,000, matching the Supreme Court ruling in both judgment direction and monetary amount. All three judicial sub-agents reached consensus on full award, with only the Equity-Centric Judge applying a partial reduction, which was overridden by the Master Judge.

Case 2 — Criminal Jurisdiction (Supreme Court 2026 모 510)

The case involved a defendant who failed to submit an appellate brief within the statutory deadline, while a separately confirmed conviction created a post-appeal concurrent offense relationship under Article 37 of the Criminal Act. The system correctly classified the case as criminal and identified the ex officio review obligation under Article 361-4(1) of the Criminal Procedure Act as the dispositive issue. The judgment direction aligned with the Supreme Court's decision to quash and remand. Direct sentencing comparison was not applicable given the remand structure of the actual ruling; however, all three sub-agents consistently identified the procedural violation and recommended sentence reduction reflecting the concurrent offense relationship.

Summary

Table 1. Verdict Accuracy Comparison Against Supreme Court Ground Truth

	Case 1 (Civil)	Case 2 (Criminal)
Case Type Classification	✓ Correct	✓ Correct
Judgment Direction	✓ Match ¹	✓ Match
Core Legal Principle	✓ Match	✓ Match
Quantitative Amount	✓ Full match (KRW 35M)	N/A (remand structure)



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

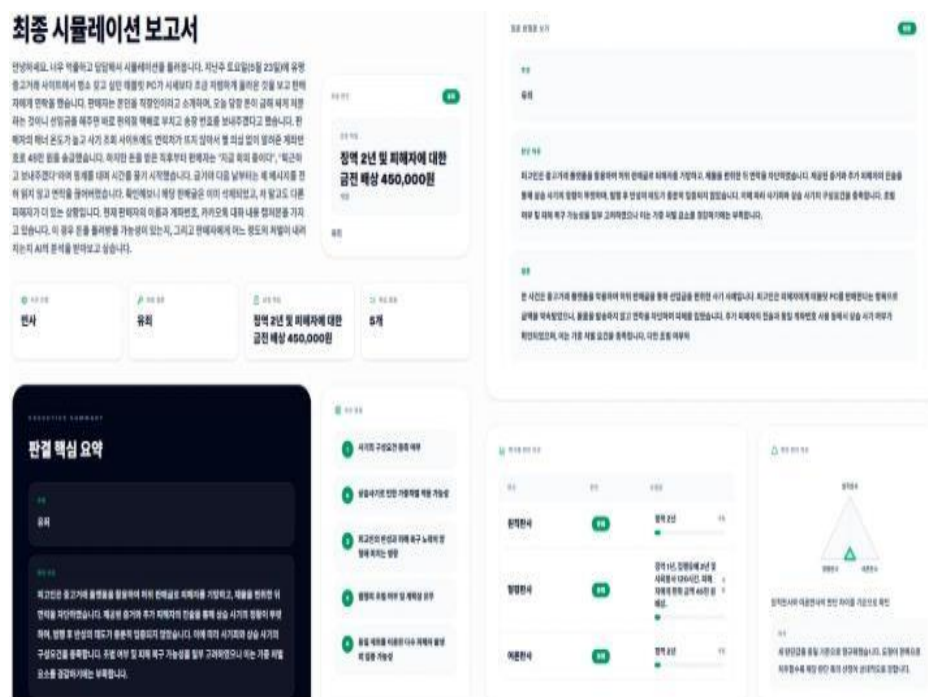
Impact Factor: 7.056

Both cases confirmed that the system accurately classified jurisdiction, identified the dispositive legal principle, and reached a judgment direction consistent with the Supreme Court. These results suggest that the RAGgrounded adversarial simulation pipeline produces legally coherent outputs aligned with actual judicial reasoning.

F. Legal Consultation Chatbot Integration

To support post-simulation workflows, the architecture deploys a context-aware Legal Consultation Chatbot that operates as a downstream consumer of the generated simulation report. Unlike general-purpose legal chatbots that rely solely on parametric LLM knowledge, the integrated chatbot parses the multi-perspective verdict report produced by the Master Judge agent, grounding its responses in the case-specific reasoning, retrieved precedents, and judicial gap analysis generated during the simulation.

The chatbot is initialized with the full simulation context upon report generation, enabling it to deliver targeted legal advice and explanatory case consultations directly relevant to the user's submitted case. This design eliminates the need for users to re-describe their case, as the chatbot inherits the complete enriched state from the preceding simulation pipeline. The chatbot interface is rendered on the result page alongside the final verdict report, providing seamless transition from simulation output to interactive legal consultation without additional navigation overhead.





International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056



Fig3.Triangular Interjudge Gap Graph and Workshop Timeline, Right Legal Advisory Chat Integration Screen

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Functional Evaluation of Case Classification and Context Ingestion

The functional validity of the automated preprocessing pipeline was evaluated using diverse scenarios representing real-world user queries. When non-professional users submitted narratives where the boundaries between civil liability and criminal offenses were ambiguous, the automated classification layer successfully routed the payloads to their respective multi-agent configurations. By explicitly confining the retrieval boundary within ChromaDB right after the classification branch, the system restricted the vector search domain to relevant document subsets, effectively avoiding cross-jurisdictional context contamination and ensuring that the advocate agents received highly targeted statutory guidelines.

B. Empirical Validation of Multi-Turn Adversarial Pleadings

The stability of the multi-turn litigation simulation was monitored across extensive execution cycles controlled by LangGraph. The implementation of localized prompt matrices for distinct agent roles proved effective in maintaining behavioral consistency over multiple turns. The Prosecutor and Plaintiff Attorney agents successfully generated objective, evidence-driven accusations anchored on the retrieved statutes, while the Defense Attorney agents maintained a defensive posture by parsing historical input vectors for mitigating circumstances. The realtime delivery of text deltas via Server-Sent Events functioned without data frame



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

dropping or front-to-back blocking states, verifying the efficiency of the asynchronous multi-agent coordination loop.

C. Architectural Assessment of Comparative System Positioning

To establish market contribution, the proposed system architecture was evaluated against existing Legaltech paradigms as delineated in image_a11e2e.png. Conventional systems are strictly limited to simple keyword precedent queries or basic document summaries, outputting a rigid, one-way information block from a single AI instance without any interactive simulation mechanics. In contrast, Justice Chamber introduces a stateful digital courtroom framework where adversarial agents run collaborative multi-turn debate loops and pass the combined results to parallel judicial entities. This design successfully delivers a comprehensive multi-philosophy risk report far superior to conventional single-threaded outputs.

D. Architectural Assessment of Comparative System Positioning

The meta-adjudication performance of the Master Judge was examined using simulated cases characterized by high social friction and variable sentencing boundaries. When the three parallel judicial sub-agents emitted highly divergent sub-verdicts—reflecting conservative statutory rigidity from the Strict Principle Judge and severe penalties from the Public Opinion Judge—the hierarchical Master Judge successfully resolved the ideological polarization. The consensus algorithm synthesized the contradictory sentencing metrics into a unified, balanced final verdict without infinite loops or deadlocks. On the client end, the geometric vertex tracking solver successfully translated these qualitative discrepancies into the SVG-based Euclidean plane, validating the interface's scannability.

V. CONCLUSION

The production instance of the Justice Chamber client ecosystem is constructed using Next.js, TypeScript, and Tailwind CSS to maximize legal document accessibility. As demonstrated in the system deployment specs, the historical repository layer—internally designated as the Cass Records module—provides users with structural dual-mode views consisting of a consolidated table interface and an intuitive card visualizer. This frontend component maps case metadata, including dynamic proceeding stamps, localized title strings, and automated data entry dates, effectively enabling high-density legal records management. When a user initiates an action, the input interface captures the unstructured narrative and utilizes the background processing pipeline to execute an automated split between criminal and civil algorithmic frameworks. If internal structural verification layers identify an information deficiency within the user's initial input, a sequential query generation loop is triggered. This sub-module generates context-aware supplementary questions to extract critical missing legal parameters—such as transaction evidence, identification variables, and institutional reporting states—prior to launching the multi-agent graph simulation.

The technical environment underpinning the AI and Backend dimensions is systematically partitioned into decoupled processing layers, as mapped out in the technical stack architecture layout in image_fdf741.png. On the artificial intelligence track shown on the left side of image_fdf741.png, Azure OpenAI serves as the enterprise-grade foundational model framework, providing secure, deterministic text generation capabilities for the advocate and judicial agents. Evidentiary context management is offloaded to ChromaDB, which handles localized vector indexing, while Sentence Transformers runs concurrently to encode raw textual expressions into dense semantic vectors. Computational workflows are handled in Python, using scikit-learn for



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Jiho Choi *et al*, Volume 15, Issue 9, September 2026, pg. 119-130

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

supplementary statistical classification tasks and Pydantic to ensure strict schema validation before data ingestion. Transient operational records and contextual state metrics are cached using Redis to minimize memory overhead.

Concurrently, the backend infrastructure detailed on the right side of image_fdf741.png is containerized using Docker to guarantee cross-environment portability and isolated network deployment. FastAPI orchestrates the high-performance asynchronous RESTful API architecture, powered by the Uvicorn ASGI server wrapper to manage high-volume transactional throughput. Persistent application metadata, user authentication files, and core case records are secured within a PostgreSQL relational database. To protect the API perimeter, pythonjose is integrated to execute secure JSON Web Token (JWT) cryptographic signatures and token decoding. System tracing, metric collection, and telemetry logging are managed via the dd module to maintain continuous operational observability across the decoupled backend topology.

ACKNOWLEDGEMENT

This research was supported by the MSIT(Ministry of Science ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) in 2026 (No. 2024-0-00023).

REFERENCES

- [1]. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, and S. Riedel, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459-9474, 2020.
- [2]. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3982-3992, Nov. 2019.
- [3]. B. Shin, "Development of KNU Korean Sentiment Lexicon for Sentiment Analysis," *Journal of Korean Language and Literature*, vol. 45, no. 2, pp. 115-138, Dec. 2018.
- [4]. LangChain Authors, "LangGraph: Stateful multi-agent orchestration for generative workflows," 2024. [Online]. Available: <https://github.com/langchain-ai/langgraph>.
- [5]. OpenAI, "New embedding models and API updates: Introducing text-embedding-3-small," OpenAI Tech Report, Jan. 2024. [Online]. Available: <https://openai.com/blog/new-embedding-models>.
- [6]. M. Sartor and G. Contissa, "Legaltech and AI in the digital courtroom: Challenges of reliability and hallucination suppression," *Artificial Intelligence and Law*, vol. 31, no. 3, pp. 412-435, Sep. 2023.
- [7]. A. Tiwari and K. L. Jones, "Adversarial multi-agent dialogue loops for multi-turn factual evaluation," in *Proceedings of the International Conference on Automated Deduction (CADE)*, pp. 204-221, Jul. 2022.
- [8]. X. Li, Y. Zhang, and L. Wang, "AgentsBench: A Multi-Agent LLM Simulation Framework for Legal Judgment Prediction," *MDPI Systems*, vol. 13, no. 8, p. 641, Aug. 2025.
- [9]. T. R. Davidson, D. MacLeod, and H. Govind, "LegalSim: Multi-Agent Simulation of Legal Systems for Discovering Procedural Exploits," *arXiv preprint arXiv:2510.03405*, Oct. 2025.
- [10]. L. Zhang, J. Chen, and H. Wang, "Legal-DC: Benchmarking Retrieval-Augmented Generation for Legal Documents," *arXiv preprint arXiv:2603.11772*, Mar. 2026.