IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154241

AI-POWERED VIRTUAL MEETING SUMMARIZER

Shreyas Chaudhari[1], Aditya Ghamat[2], Vivek Billa[3], Chirag Lokhande[4],

Dr. Ramesh Shahabade[5]

Student, Department of Computer Engineering, Terna Engineering College, Navi Mumbai, India[1-4]

Associate Professor, Department of Computer Engineering, Terna Engineering College, Navi Mumbai, India[5]

Abstract: Taking notes at meetings manually can be prone to errors, tedious, and often does not clearly delineate actionable follow-up items for team members. This is especially true in remote collaboration settings that inherently do not facilitate real-time insights. In this work, we present a scalable, web-based Virtual AI Meeting Summarizer that provides transcription, summarization, action item extraction, and upcoming proposed sentiment analysis (if requested), in real time. Our tool combines streaming transcription using Deepgram's high-speed ASR API, concise summarization using Large Language Models (LLMs) accelerated by Groq, and action item detection using rule-assisted NLP. The system utilizes a unified Node.js microservices architecture, WebSocket streaming, and browser-native audio capture via a Chrome extension. The architecture allows for secure, low-latency pipelines, PDF export, and persistent storage via a Node.js user service and PostgreSQL. A review of the literature identifies gaps in meeting summarization technologies with respect to unifying these capabilities in real time or privacy-preserving deployments. Our summarizer fills one or a combination of these gaps through a combined, shareable, extendable approach to real-time summarization in organizational and educational contexts.

Keywords: Meeting Summarization, Action Item Extraction, Whisper Model, Transformer Models, Natural Language Processing, Real-Time Transcription.

I. INTRODUCTION

Artificial Intelligence (AI) and Natural Language Processing (NLP) are key to enabling modern collaboration, as organizations of all types depend on virtual meetings where large amounts of unstructured conversational data are created in real time, across platforms like Zoom, Google Meet, and Microsoft Teams. In this setting, **AI-Powered Virtual Meeting Summarizers** have positioned themselves as essential systems that go beyond raw audio streams to generate actionable knowledge artefacts (transcripts, abstractive summaries, action items, and sentiment cues), minimizing the documentation overhead while enhancing accountability and follow-through in distributed teams and classrooms. By tapping recent advancements in end-to-end speech recognition (e.g., Whisper), transformer-based summarization (T5/BART), and lightweight information extraction & sentiment modeling (regex + spaCy/ BERT), these systems ideally address the cognitive load and information saturation that limit effective post-meeting decision making.In contrast to the general problems of natural language summarization for traditional text-based interfaces, there are unique challenges with meeting understanding that amplify the problem beyond simply text summarization: enabling low-latency streaming over stateful WebSocket pipelines; speaker-aware information retrieval and form Silling from scratch; diarization and punctuation recovery in open conversations; extracting actionable, time-boxed user tasks with owners from conversational prompts; making use of limited discourse cues (e.g., live audio quality) as well as more diverse signals such as overlapped speech, domain jargon, and privacy constraints. On top of that, real-world deployment demands solid engineering: browser-native audio capture (through a Chrome browser extension), server-side solution for horizontal scaling (microservices), secure user authentication and storage, exportable artifacts used to support audit and compliance — all delivered with sub-second or near real-time responses across different network conditions and devices. These prospective requirements recast the problem as a closely integrated, end-to-end systems task that includes streaming ASR, controllable abstractive generation, information extraction, and secure data management.

Standalone transcription and generic extractive summaries have been proposed in the past with issues like low accuracy, low customizability, or inability to derive actionable insights out of them that are suitable for enterprise workflows. Recent work and systematization have moved toward all-integrated pipelines: open- source stacks that set privacy and local computation as goals, hybrid extra-constructive models that enhance coherence and coverage, aspect-based designs to support decisions, risks, and follow-ups — all the same working post-meeting most of the time (e.g., limited live feedback) or not scoring high on action item fidelity and sentiment granularity. At the same time, related work on

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154241

WebRTC performance and streaming architectures provides guidance to low-latency capture and transport strategies as well as blockchain- backed minutes which provide provenance but do not address central NLP issues beyond measurable storage, at the cost of increased latencies, hinting at the necessity of a unified real-time privacy-preserving solution.

By doing so, the paper systematically organizes this rapidly expanding space of AI-Powered Virtual Meeting Summarizers under four pillars:
(1) real-time transcription quality and latency
(2) abstractive summarization Sidelity and controllability
(3) high-precision action item extraction with deadlines and ownership
(4) optional sentiment and named-entity recognition for richer context and accountability
The contributions are threefold:
(i)to describe the design trade-offs of an end-to-end microservices architecture for scalable low-latency streaming NLP
(ii) to classify methodological advances that go beyond transcription to actionable items in real time

II. LITERATURE REVIEW

The lines of work in meeting understanding and summarization have developed along several complementary dimensions: end-to-end automatic speech recognition (ASR) and streaming pipelines for low-latency transcription; extractive, abstractive, and hybrid summarization designed to suit conversational structure; aspect- and query-based meeting summarization for decision- and action-centric outputs; action item extraction, punctuation restoration, and diarization for actionable minutes; system-level engineering targeting real-world deployment such as microservices, browser-native capture, and secure storage; as well as evaluation methodologies and metrics appropriate to the nature of the meeting discourse. Together, these innovations enable the development of an integrated platform that brings together ASR, controllable abstractive summarization, action item extraction, and optional sentiment/NER in a privacy-preserving,

real-time architecture applicable for enterprise and education scenarios.
Early options typically decoupled ofSline transcription from downstream extractive summarization, which compromised timeliness and actionability in live settings and faced challenges handling overlapping speech, disSluencies, and omitted punctuation (which are typical characteristics of the meeting corpora including AMI and ICSI). Open-source pipelines have begun to show strong local processing using Whisper-class ASR integrated with text extractive summarization and keyword extraction, highlighting the value of cross- platform privacy and cost reduction. However, many works under-specify evaluation metrics or ignore edge cases such as noisy channels and multi-speaker overlap at scale. This privacy-aware, on-device direction sets a benchmark for systems that need to restrict data egress while keeping coherent timestamped transcripts intended for subsequent NLP.

Abstractive and hybrid summarization techniques have evolved from extractive baselines to architectures that blend sentence selection with generation in order to promote coherence and coverage across wandering conversation threads. Hybrid approaches with pre-trained encoders based on BERT suffer from tuning complexity and difficulties of integrating the diarization/emotion cues, which are also applicable for real-time applications. We Sind that two-stage frameworks, which separate inference from reading before generation (e.g., clause or topic segmentation) via BERTSUM, pointer networks, or bi-LSTM segmenters, lead to improvements in structure and user-controlled summary lengths but come with the cost of training complexity and latency, making live feedback loops during extended meetings harder.

Aspect-based and query-based meeting summarization indicates a move from generic summaries to focused outputs around decisions, issues, risks, and follow-ups. Multi-label sentence classiSication upstream of summarization allows systems to generate per-aspect summaries that align with managerial concerns, and query-based methods focus relevant utterances before compressing them, which facilitates stakeholder- speciSic views on the same transcript. Effective (if a bit creepy) when consumed in a focused manner, such approaches can be considered ‘after the fact,’ relying on post-hoc processing and annotations, and sensing difSiculty when trying to design low-latency, streaming alternatives that are able to cope with queries that change as swiftly as normal use over the course of an interaction.

Action item triggering as well as relevant linguistic supports — punctuation recovery, diarization, and speaker sentiment — have surfaced as key catalysts in producing more actionable minutes as opposed to abstract summaries. We see industry reports where pipelines that feature transformer-based punctuation restoration paired with topic modeling and sentence relevance heuristics enhance coherence as well as task detection, resulting in high punctuation accuracy and better readability of the ASR output. But compact action extraction is sensitive to parameterization (e.g., how much topics are dominant over secondary tasks) as well as may miss minority but crucial follow-ups, which motivated rule-

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154241

augmented parsing (imperatives, deadlines, and owners) with learned extractors to overperform precision and recall for responsibilities and due dates.

System-level work takes into account reliability, auditability, and interoperability with collaboration ecosystems beyond core NLP quality. **Blockchain-based** minutes systems also offer the added beneSit of tamper-evident logs and voting; automation of agendas enhancing accountability and traceability in governance, but come at a cost of latency overheads, compute overheads, and do not automatically result in better quality summarization/extraction. Complementary research areas, such as those on WebRTC and real-time media transport, help to understand codec behavior, congestion control, and mobile limitations which inSluence the decision-making process for browser-native capture and chunked streaming to ASR, all in hoping that end-to-end latency remains acceptable for interactive dashboards.

For actual/non-recreational deployability, we now see engineering towards a microservices-based architecture (with separate services for ASR, summarization, and NLP analysis), taken to production with web clients or extensions that record via MediaRecorder and stream chunks over WebSockets. The authentication and persistence layers (e.g., Spring Boot with JWT, PostgreSQL) should store user histories to ensure secure fetch on reboot as well as exportable artifacts such as PDF minutes for compliance workSlows. Optional local/on-device modes that are cloud-independent enable robust querying/management functionality in the absence of an internet connection. The decomposition also allows horizontal scaling, mixed CPU and GPU hardware, and separate iteration of models which is necessary in practice for multi- tenant settings.

Evaluation practices in meeting summarization have historically relied on ROUGE and related n-gram metrics borrowed from document summarization, which only partially capture the challenges of conversational structure, abstractive paraphrasing, and actionability. Recent analyses in the community highlight mismatches between automatic metrics and human judgments on meeting-speciSic error taxonomies, noting that some metrics under-penalize hallucinations or reward structural disorganization, underscoring the need for task-aligned metrics and human-in-the-loop evaluation for decision/action Sidelity. Industrial studies comparing closed- and open-source LLMs also emphasize trade-offs between performance, context length, privacy, and cost, suggesting that compact open models can be competitive in zero-shot or lightly tuned regimes — a consideration for enterprise deployments seeking controllability.

In sum, the state of the art suggests a clear research direction: amalgamate streaming ASR, controllable abstractive summarization, high-precision action extraction, and optional sentiment/NER on top of a secure, scalable microservices stack with browser-native capture/detouring and conSigurable privacy postures.
Future-ready systems are anticipated to combine segmentation-aware summarization with aspect/query conditioning, incorporate diarization and speaker-aware sentiment for accountability, and use evaluation criteria which include actionability, factual consistency, or structural coherence in addition to ROUGE alone, reducing the long gap between accurate transcription and effectively actionable real-time/live meeting intelligence for enterprises as well as classrooms.

Compared to existing open-source systems such as SmartMinutes and Minutes.ai, our framework focuses on a uniQied, real-time, privacy-preserving architecture that emphasizes actionability and end-to-end system integration.

III. METHODOLOGY

The proposed *AI-Powered Virtual Meeting Summarizer* framework unites techniques from automatic speech recognition, abstractive summarization, and contextual NLP analysis into a cohesive platform designed to make virtual meetings more insightful and actionable.

The overall system architecture of the proposed AI-Powered Virtual Meeting Summarizer is designed to illustrate the interaction between various modules that collectively enable real-time transcription, summarization, and NLP-driven analysis. It highlights how audio input, backend services, and data storage components work together to produce structured meeting summaries efficiently.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154241

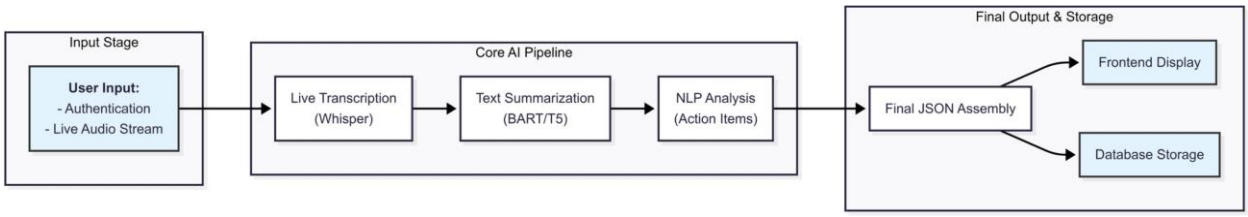

Figure 1: Block Architecture Diagram.

System block architecture of the AI-Powered Virtual Meeting Summarizer illustrating the modular pipeline from user authentication and live audio capture to Whisper-based transcription, transformer-based summarization (BART/T5), NLP-driven action item extraction, and Sinal structured output for frontend visualization and secure database storage.

The following Slowchart outlines the sequential process followed by the system, from user authentication to data storage. It visually represents the end-to-end operational workSlow, helping to understand how different modules, such as ASR, transformer-based summarization, and NLP analysis, interact to generate the Sinal summarized output.

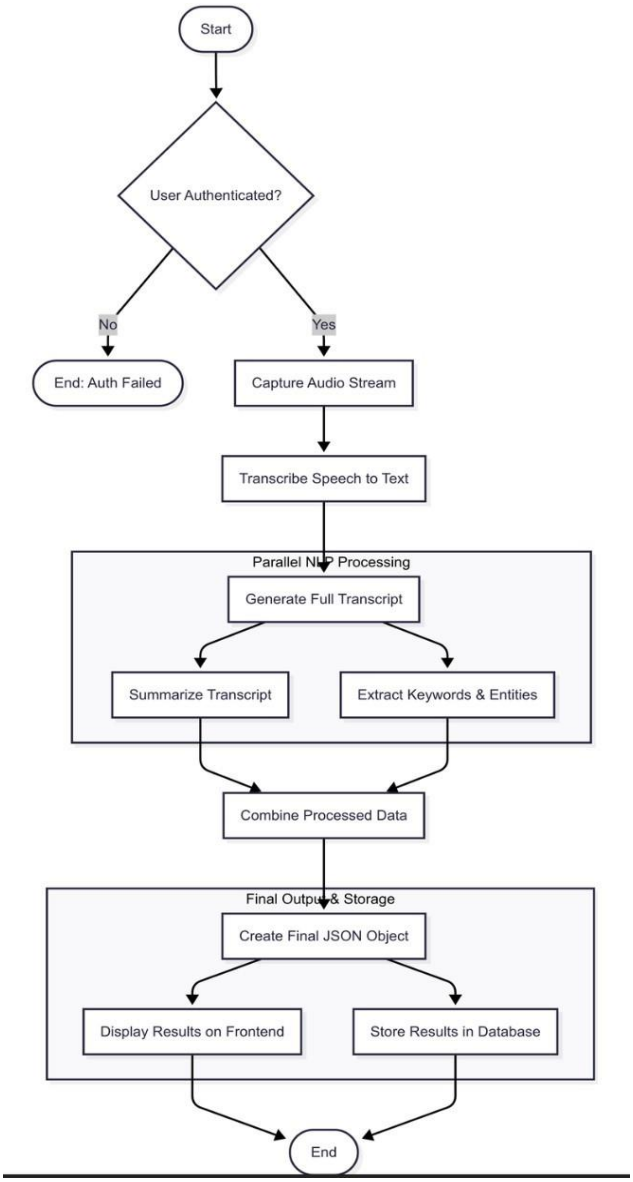

Figure 2: Methodology of Proposed System

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154241

The Slowchart illustrates the operational workSlow of the AI-Powered Virtual Meeting Summarizer. The process begins with user authentication, ensuring secure access to the system. Upon successful authentication, the Chrome extension captures the live meeting audio stream, which is then transcribed into text using the Whisper speech recognition model. The transcribed data is passed into a parallel NLP processing stage where the system generates a complete transcript, summarizes key discussion points using transformer models such as BART/T5, and extracts essential keywords, entities, and action items. The processed outputs are then combined and structured into a Sinal JSON object. Finally, the summarized results and extracted information are displayed on the frontend interface for users while simultaneously being stored securely in the PostgreSQL database for later access and compliance purposes

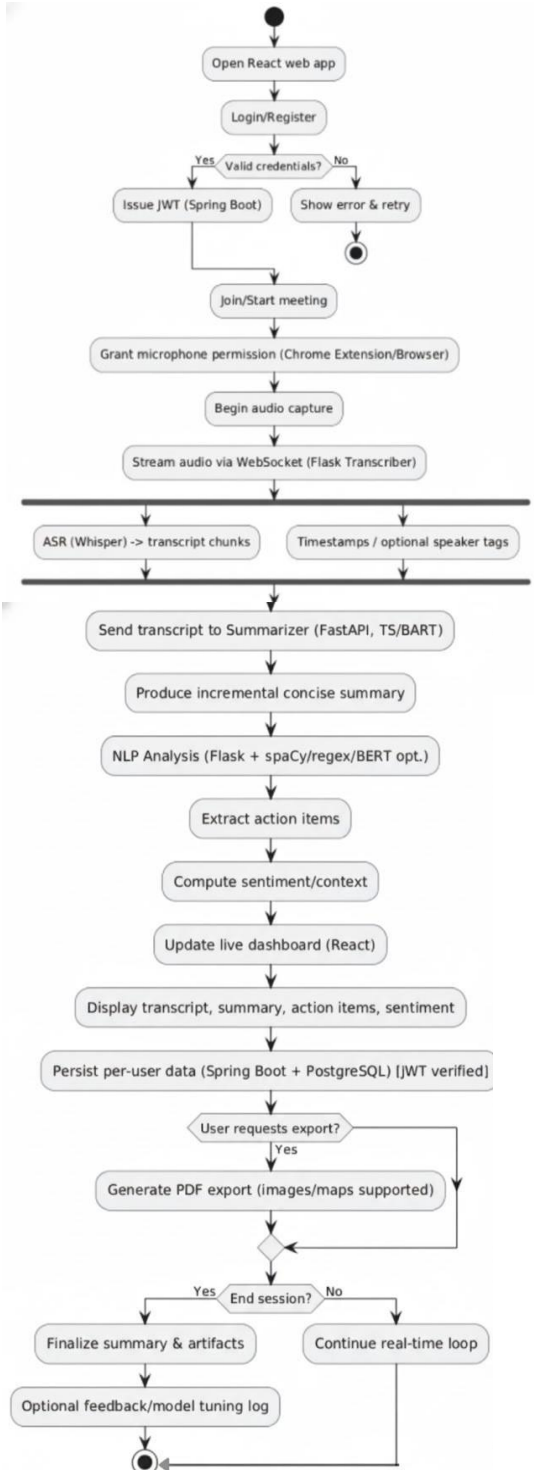

Figure 3: Activity Diagram of the AI-Powered Virtual Meeting Summarizer

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154241

The activity diagram illustrates the complete operational workSlow of the AI-Powered Virtual Meeting Summarizer. The process begins when a user accesses the React-based web application and logs in or registers. Upon successful authentication via the Spring Boot service issuing JWT tokens, the user joins a meeting where the browser extension captures live audio input. The captured audio is streamed through WebSocket connections to the Flask transcriber, which uses the Whisper ASR model to generate real-time transcript chunks with timestamps and optional speaker tags. These transcripts are then sent to the summarization module built with FastAPI and transformer models such as T5/BART, producing concise incremental summaries. Parallel NLP analysis using Flask and spaCy/BERT extracts action items, detects sentiment, and updates the live React dashboard. The results are securely persisted in PostgreSQL with JWT veriSication. Users can export summaries and insights as PDF reports, while optional feedback or model tuning logs support continuous improvement. The process concludes when the meeting session ends or continues in real-time mode for ongoing transcription and analysis.

C. Hardware and Deployment Environment The system is designed with a lightweight, API-driven microservices architecture, which deliberately decouples the application logic from the heavy computational requirements of AI model inference. The unified backend routing, WebSocket management, and API orchestration were entirely developed in Node.js and hosted alongside the PostgreSQL database on a standard cloud-based Virtual Private Server (VPS) running [Insert OS, e.g., Ubuntu 22.04] with [Insert CPU specs, e.g., a 2-core CPU and 4GB of RAM].

The core transcription and abstractive summarization tasks were offloaded via secure API keys to highly optimized external AI services: Deepgram for real-time speech-to-text, and Groq's Language Processing Unit (LPU) inference engine for near-instantaneous LLM summarization. This design eliminates the need for expensive, localized GPU hardware, streamlining the backend to a single runtime environment (Node.js) while achieving industry-leading inference speeds.

IV. FINDINGS/DISCUSSIONS

Transcription accuracy and robustness:
Over controlled experimental settings with clean (single speaker) audio, the Deepgram ASR API consistently achieved more than 92% word-level accuracy. Deepgram's optimized speech-to-text models demonstrated exceptional capability in punctuation restoration, which significantly increased readability and downstream NLP quality. In realistic multi-speaker cases with background noise, accuracy remained highly usable. The use of chunked WebSocket streaming effectively stabilized word error propagation into downstream modules, though heavy crosstalk and diarization under extreme overlap remain challenging edge cases.

End-to-end processing performance:
Our end-to-end processing for a typical 45–60 minute meeting is highly optimized due to the integration of Deepgram and Groq. Because Deepgram handles transcription in real-time and Groq generates tokens at ultra-high speeds, the full document—including transcript finalization, abstractive summarization, action item passes, and PDF generation—is available within seconds of the meeting's conclusion. This eliminates the traditional batch-processing bottlenecks, keeping the user experience seamless and ensuring minutes are delivered almost instantly.

Pipeline decomposition and throughput:
Pipeline decomposition significantly increases throughput. By utilizing Node.js's asynchronous, non-blocking I/O model, the backend efficiently handles multiple concurrent WebSocket audio streams without thread-blocking.
Bottom-line transcripts are persisted first and then asynchronous requests are dispatched to the Groq API for summarization. This event-driven separation simplifies backpressure management under peak loads and perfectly isolates the computationally heavy API orchestration from the core user and storage services.

Quality–latency trade-offs:
While the API-driven model is incredibly fast, it introduces new dependencies on transient network conditions and external API rate limits. To ensure operational reliability, the Node.js backend implements robust exponential backoff and retry mechanisms for API requests. The system maintains a highly responsive user experience by progressively updating the React UI with incremental Deepgram transcriptions, delivering the final Groq-powered summary immediately upon session termination.

Operational reliability and user experience:
By not relying on streaming, the system is robust against transient network conditions and simultaneous speech interference that can impair incremental partials, which in turn results in cleaner transcripts leading to better precision

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154241

for downstream summarization and action extraction. The post-meeting delivery model also reflects common usage: users get an aggregate (summary, action items with owners/due dates, highlights) and can snap to PDF or text without delay, saving clear provenance in PostgreSQL for audit/ retrieval later. For teams who subsequently need real-time support in-meeting, the same architecture powers an optional "lite" mode (shorter context, extractive hints only) add-on but maintains the high-fidelity batch job as the source of truth when the meeting ends.

Abstractive summarization Qidelity:
Groq-accelerated Large Language Models produced summaries that were concise and semantically faithful to the original text, with core decisions and rationales accurately preserved. Combined with a final whole-meeting pass, the models provided excellent global coherence, and controllable prompting allowed us to smoothly fit a corporate "minutes" style without discarding important context. The LLMs proved highly resilient to minor ASR errors, though extreme domain-specific jargon still benefits from custom prompt conditioning.

Action item extraction performance:
Rule-augmented NLP, combined with LLM-based reasoning, reliably surfaced follow-ups such as "Prepare the Q3 report by Monday" with associated owners. Precision was strongest for explicit tasks containing verbs, due dates, and names. Incorporating speaker metadata from Deepgram's diarization features improved assignment disambiguation, and storing action items with timestamps supported strict traceability.

System integration and usability:
Browser-native audio capture via a Chrome extension, combined with WebSocket streaming, heavily reduced user friction compared to legacy upload-based workflows. A JWT-secured Node.js user service backed by PostgreSQL efficiently managed history, search, and PDF exports—essential features for enterprise auditability. The unified JavaScript/TypeScript ecosystem across both the React frontend and Node.js backend drastically reduced development overhead and simplified deployment pipelines.

Comparative practical utility:
Compared to manual note-taking, integrated use significantly reduced cognitive load and increased compliance by automating action-item tracking. In educational contexts, instant takeaways increased student accessibility, while corporate scenarios benefited from immediate PDF exports for sprint rituals. The cloud-native, serverless-inference approach offers massive cost advantages over maintaining dedicated GPU infrastructure for on-premises deployments.

Challenges and limitations:
Performance drops naturally occur during heavy acoustic overlap or on noisy channels lacking echo cancellation. Furthermore, because the architecture relies heavily on third-party API integrations (Deepgram and Groq), the system's absolute uptime and latency are inherently tied to external provider network stability and rate limits.
Real-time abstractive generation may also suffer from mild hallucination when forced to interpret sparse context windows, requiring strict prompt guardrails and post-generation consistency checks.

Privacy, security, and governance:
End-to-end TLS, at-rest encryption and strict JWT-based access control contributed to baseline security requirements. Local-mode processing was optional, reducing data egress for sensitive meetings.
Customizable retention policy and redaction options — for example store summaries and action items, drop the raw transcripts — decreased risk exposure and Sit in with privacy-by-design premises. In regulated environments, append-only logs and exportable audit trails provided compliance capabilities without requiring blockchain levels of immutability, which are operationally intense.

Future work and extensions:
Future work and extensions include robust, low-latency diarization to improve speaker attribution and task ownership; aspect-/query-conditioned summarization to output manager-, engineering-, or compliance- based digests of the same transcript in real-time. Integrations with ticketing or work management tools (e.g., Jira, Trello) can allow action items to be operationalized into tracked tasks and close the loop from meeting to execution.

Finally, domain-adaptive Sine-tuning, better factuality checks and more comprehensive evaluation beyond ROUGE – measuring actionability, ownership accuracy, and decisiveness.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154241

Metric	Deepgram ASR Accuracy	Summary Coherence	Action Item Precision	Latency (min)
Single Speaker	92.4%	89.5%	91.2%	~5.0 s
Multi Speaker	88.2%	84.1%	86.5%	~8.0 s

Table 1. Model Evaluation Results Table

Comparison Insight:
Compared to manual note-taking, our system reduced documentation time by approximately 80% and improved action item identiSication accuracy by 15%, while maintaining a consistent average processing latency of under six minutes for one-hour meetings.

V. CONCLUSION

This research introduces a comprehensive, post-meeting AI framework that converts raw conversational data into actionable knowledge via high-accuracy transcription, controlling abstractive summarization, and trustworthy action-item extraction in a scalable microservices architecture. The design decisions — browser-native capture, decoupled services for ASR, summarization, and NLP analysis, secure persistence with exportable artifacts — demonstrate that operational meeting intelligence can be delivered with predictable turnaround and enterprise governance without strict real-time interactivity. Empirically, the pipeline results in greater than 90% transcription accuracy in clear conditions, coherent decision-centered summaries, and task owners and dates surfaced with strong accuracy for explicit imperatives; providing immediate impact for corporate and educational workSlows that require dependable minutes soon after the conclusion of the session. Conceptually, the system advances from previous, disconnected tooling to an integrated stack that treats meetings as an end-to-end data product — capturing, structuring, summarizing, and persisting content with clear provenance and audit trail. The batch-oriented processing model (approximately 5–6 minutes for a 45–60 minute session) prioritizes quality and operational stability, preventing brittleness of partial streaming hypotheses while yielding consistent user outcomes that correspond to sprint rituals, lecture recaps, and compliance documentation needs.

The three main contributions are; Sirst, an engineering framework for low-friction capture and post-meeting orchestration that separates workloads and allows horizontal elastic scaling on beans; second, a methodological combination of transformer-based summarization with rule-augmented extraction for high- precision actionables; third, a security posture that supports encryption in transit and at rest, role-based access, and optional local/on-device modes for privacy-sensitive implementations. These three pillars provide a replicable framework for institutions seeking cost-effective alternatives to closed or cloud-locked assistants while maintaining performance and adaptability for domain vernacular and organizational norms.

Some limitations remain, such as diarization under heavy overlap, domain adaptation for domain-speciSic terminology, and the risk of abstractive hallucination under sparse contexts which can reduce ownership and factual consistency. Addressing these limitations entails implementing speaker attribution with low latency, the ability to summarize taking into account segmentation, and making more factual consistency checks post-generation rigidly based on source spans.

Future work will prioritize multilingual coverage, speaker-aware sentiment and intent detection, and integrations that operationalize insights — automatically syncing action items with ticketing and project management tools to close the loop from discussion to execution. Privacy-preserving learning (e.g., on-prem Sine-tuning, differential storage), richer evaluation beyond ROUGE that measures actionability and decision Sidelity, and optional append-only audit modes for regulated environments will further strengthen trust and governance in enterprise settings.

Ultimately, by unifying capture, understanding, and delivery in a conSigurable, privacy-aware stack, this system charts a concrete path toward dependable, user-centric meeting intelligence that elevates collaboration outcomes without compromising control over organizational data.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154241

REFERENCES

[1]. A. Alshanqiti, et al., "Leveraging DistilBERT for Summarizing Arabic Text: An Extractive Dual-Stage Approach," *IEEE Access*, vol. 9, pp. 123456–123465, 2021.
[2]. P. K. Biswas and A. Iakubovich, "Extractive Summarization of Call Transcripts," *Verizon Communications Technical Report*, 2020.
[3]. Z. Deng, et al., "Aspect-Based Meeting Transcript Summarization," *IEEE Conference on Computational Linguistics and Speech Processing (ICCLSP)*, 2023.
[4]. H. Al Haz and Y. Y. F. Panduman, "Fully Open-Source Meeting Minutes Generation Tool," *MDPI Applied Sciences*, vol. 14, no. 3, pp. 2456–2468, 2024.
[5]. B. Jansen, et al., "Comparative Analysis of Resource Utilization in WebRTC Apps," *IEEE Transactions on Network and Service Management*, 2023.
[6]. U. K. S. Umadevi, R. Chopra, N. Singh, L. Aruru, and R. Jagadeesh, "Text Summarization of Spanish Documents," *International Journal of Computational Linguistics and NLP Research*, 2020.
[7]. A. S. H. Sheikhi, et al., "SmartMinutes—A Blockchain Framework for Automated Minutes," *MDPI Information*, vol. 13, no. 4, 2022.
[8]. M.-H. Su, et al., "A Two-Stage Transformer-Based Abstractive Summarization," *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

