JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 11, pp. 101-115, November 2026
https://doi.org/10.9708/jksci.2026.31.11.101

Design and PoC Validation of a Secure On-Premises Meeting Intelligence Pipeline for Construction Enterprises

Jaehwan Kim*, Jaeseong Yun**, Jongseok Lee***, Kyungmin Kim****

*Doctoral Researcher, Department of AI Convergence, aSSIST, Seoul, Korea
**AI Engineer, Research and Development Department, XAIKOREA Inc., Seongnam, Korea
***Team Leader, Infrastructure Team, RAD CNS Co., Ltd., Seoul, Korea
****Principal Researcher, Information Security Team, Security Platform, Seoul, Korea

[Abstract]

This paper develops and evaluates artifact validation and checkpoint recovery within an on-premises meeting-intelligence pipeline for construction enterprises. Using the same ASR model and decoding configuration, we compared full restart, syntax-based reuse, and reuse with combined integrity and execution-condition checks. The first 120 seconds of 16 AMI test recordings were processed in independent 30-second chunks. All 912 runs terminated: 48 no-fault controls and 864 fault-injection runs. Full restart and combined validation preserved the uninterrupted reference transcript in every fault-injection run, whereas syntax-based reuse returned incorrect transcripts in 96 text-corruption runs. After interruption following the third saved chunk, the mean of recording-level median recovery latencies was 26.19 seconds for both reuse policies and 83.52 seconds for full restart; the recording-level paired reduction for combined validation was approximately 57.3 seconds. These results show the role of content validation and valid-chunk reuse under the tested conditions. They do not establish whole-system security or accuracy on Korean construction meetings. The additional benefit of execution-condition checks requires a checksum-only comparison.

▸ **Key words**: On-Premises AI, Meeting Intelligence, Artifact Integrity, Checkpoint Recovery, Fault Injection

[요 약]

본 논문은 건설기업을 위한 온프레미스 회의 인텔리전스 파이프라인의 산출물 검증과 체크포인트 복구를 구체화하고 평가한다. 동일한 자동음성인식(ASR) 모델·설정에서 전체 재시작, 구문 검사 재사용, 해시·실행조건 결합 검증을 비교하였다. AMI test의 16개 녹음에서 처음 120초를 독립적인 30초 구간으로 처리했으며, 무장애 48회와 장애 주입 864회, 총 912회가 종료되었다. 전체 재시작과 결합 검증은 모든 장애 실행에서 기준 전사를 보존했지만 구문 검사군은 내용 변경 96회에서 다른 전사를 반환하였다. 세 번째 구간 저장 후 단순 중단에서 장애 후 지연의 녹음별 중앙값 평균은 두 재사용 정책 모두 26.19초, 전체 재시작은 83.52초였으며, 결합 검증의 녹음별 쌍체 지연 감소는 약 57.3초였다. 결과는 지정한 결함의 내용 검증과 정상 구간 재사용 효과이며, 전체 시스템 보안성이나 한국어 건설 회의의 인식 정확도를 입증하지 않는다. 실행조건 검사의 추가 효과는 해시 전용 비교가 필요하다.

▸ **주제어**: 온프레미스 AI, 회의 인텔리전스, 산출물 무결성, 체크포인트 복구, 장애 주입

• First Author: Jaehwan Kim, Corresponding Author: Jaehwan Kim
*Jaehwan Kim (kjh861213@naver.com), Department of AI Convergence, aSSIST
**Jaeseong Yun (yun777567@gmail.com), Research and Development Department, XAIKOREA Inc.
***Jongseok Lee (lodossw@gmail.com), Infrastructure Team, RAD CNS Co., Ltd.
****Kyungmin Kim (xelose1004@gmail.com), Information Security Team, Security Platform
• Received: 2026. 08. 03, Revised: 2026. 10. 13, Accepted: 2026. 10. 31.

Copyright © 2026 The Korea Society of Computer and Information
http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945
www.kci.go.kr

I. Introduction

조직 내부에서 음성을 전사하는 시스템은 모델의 인식 성능뿐 아니라 장시간 작업의 중단과 재실행을 다루어야 한다. 회의록 작성 과정에서 일부 구간의 결과가 남아 있으면 이를 재사용하여 계산을 줄일 수 있다. 그러나 부분 파일, 읽을 수 있지만 내용이 바뀐 파일, 다른 입력이나 설정에서 생성된 파일을 정상 체크포인트로 취급하면 작업은 종료되어도 전사 결과는 보존되지 않는다. 따라서 처리 완료 여부와 전사 보존 여부를 함께 확인해야 한다.

Whisper와 WhisperX는 음성인식 및 시간 정렬의 기반을 제공한다[1,2]. 이와 별개로 복구 정책은 중단 전 산출물 중 무엇을 신뢰하고 어떤 구간을 다시 계산할지 결정한다. 본 연구는 같은 모델로 수행한 무장애 기준 실행의 전사를 지정된 장애 후에도 보존하면서 재계산 범위를 줄일 수 있는지 살펴본다. 모델 오인식이 그대로 보존되는 경우도 있으므로 복구 보존과 공식 정답 대비 인식 정확도를 분리한다.

기존 건설 회의 처리 개념검증(Proof of Concept, PoC)은 음성인식, 화자분리, 요약, 저장 및 키 관리 기능을 통합하고 데이터·키 경계, 산출물 완전성, 체크포인트 복구와 증거 수준 분리를 설계 원칙으로 두었다. 본 연구는 이 파이프라인의 산출물 완전성과 복구 원칙을 전사 구간의 재사용 판정 및 복구 비용으로 구체화하여 심화 평가한다. 새로운 음성 모델을 제안하거나 전체 파이프라인의 보안성을 입증하는 것이 아니라, 기존 설계 중 측정 가능한 복구 동작을 검증한다. 실험에는 공개 영어 회의 자료를 사용하며 한국어 건설 현장에 대한 적용 가능성은 후속 검증 과제로 둔다.

연구질문(Research Question, RQ)은 두 가지다. RQ1은 지정한 파일 손상에서 재사용 정책별로 전사 보존이 어떻게 달라지는가이다. RQ2는 전사를 보존한 실행에서 장애 종류와 중단 위치에 따라 재계산량과 시간이 어떻게 달라지는가이다. 두 질문은 완료된 E1으로 분석한다. 해시 검사에 실행조건 검사를 추가했을 때의 효과와 기존 워크플로 대비 차이는 별도 검증 질문으로 구분한다.

본 연구는 독립 구간으로 처리한 음성 전사의 복구 정책을 비교하는 경험적 시스템 연구이다. 동일한 음성·모델·장애 조건을 사용하여 정책에 따른 전사 보존과 계산 비용을 함께 분석한다. 그 결과를 통해 파일을 읽을 수 있다는 사실만으로 재사용을 결정할 때의 문제와, 정상 구간을 유지할 때의 비용 변화를 설명한다. 입력·모델·코드의 식별자와 실패를 포함한 실행 기록은 이 비교의 근거를 추적하는 데 사용한다.

실험의 해석은 검증 정보와 비용을 분리하는 데 초점을 둔다. 출력 내용의 변경은 저장된 해시와의 비교로 확인할 수 있고, 다른 작업의 정상 산출물은 현재 입력과 설정의 적합성을 따로 살펴야 한다. E1에서는 해시와 실행조건 검사를 결합하였으므로 관측된 개선은 결합 정책의 결과로 보고한다. 각 검사의 추가 효과와 기존 구현 대비 차이를 밝히는 비교는 후속 검증으로 이어진다.

II. Related Work and Research Scope

1. Speech Processing and Transcript Preservation

Whisper는 음성과 전사의 대응 관계를 대규모 약지도 자료에서 학습하여 음성을 문장으로 변환하는 자동음성인식(Automatic Speech Recognition, ASR) 모델이다[1]. WhisperX는 전사에 시간 정렬을 결합하여 단어가 발화된 시점을 찾고[2], pyannote.audio는 발화 구간을 화자별로 묶는 화자분리를 다룬다[3]. 음성 처리 연구에서는 무엇을 말했는지, 언제 말했는지, 누가 말했는지를 각각 평가한다. 본 연구의 관심은 이 가운데 생성된 전사가 작업 중단과 재시작 후에도 유지되는가이다. 이를 확인하기 위해 음성 인식 모델과 처리 구간을 고정하고 복구 정책만 비교한다.

2. Checkpoint Recovery and Fault Injection

체크포인트는 계산을 처음부터 반복하는 대신 저장된 중간 상태에서 재개하기 위한 수단이다. Elnozahy 등은 체크포인트와 로그를 이용한 복구 기법을 정리하면서, 복구에 사용할 상태의 일관성과 되풀이할 계산의 범위를 주요 문제로 다루었다[4]. 본 연구에서는 한 음성 구간의 전사가 끝날 때 그 결과를 응용 수준의 체크포인트로 저장한다. 네 구간 중 세 구간이 정상적으로 남아 있다면 마지막 구간만 다시 처리할 수 있다. 다만 남은 결과가 손상되었거나 현재 입력과 다른 조건에서 생성되었다면 재계산이 필요하다. 따라서 저장 여부와 함께 재사용 적합성을 판단한다.

장애 주입(fault injection)은 중단이나 손상을 의도적으로 일으켜 시스템의 탐지·복구 동작을 관찰하는 방법이다. Hsueh 등은 주입 대상과 방식, 관측 수단을 함께 설계하는 실험 접근을 설명하였다[5]. 본 연구는 자연 발생 장애를 기다리는 대신 동일한 음성에 세 종류의 장애를 적용한다. 단순 중단은 정상 결과의 재사용을, JSON 잘림은 읽기 어려운 파일의 처리를, 내용 변경은 읽을 수 있는 파일의

www.kci.go.kr

내용 검증을 살펴본다. 같은 조건에서 정책을 비교하므로 결과 차이를 검사 방식과 연결할 수 있다. 조건별 동일 실행 횟수는 비교를 위한 배분이며, 실제 서비스에서 각 장애가 발생하는 빈도는 별도 조사 대상이다.

3. Execution-Aware Caching and Crash Consistency

캐시는 이미 계산한 결과를 보관하여 같은 작업에서 다시 사용하는 기법이다. 여기서 핵심은 저장된 결과가 지금 요청한 작업에 대응하는지 판단하는 것이다. Bazel의 원격 캐시는 입력과 실행 명세에 연결된 결과 메타데이터와 출력 내용을 저장하는 공간을 구별한다[6]. 예를 들어 파일 내용이 그대로여도 입력 음성이 바뀌었다면 이전 전사를 그대로 쓸 수 없다. 본 연구는 이러한 기존 원리를 전사 구간에 적용하여, 내용 유지 여부와 생성 조건의 적합성을 나누어 확인한다.

Di Tommaso 등이 제시한 Nextflow는 실행 환경과 워크플로를 관리하여 계산 재현성을 지원한다[7]. 캐시 재사용은 입력·스크립트·컨테이너 등을 반영한 작업 해시와 저장된 출력을 기준으로 판단하며[8], DVC도 선언된 의존성과 매개변수를 이용해 작업 재실행과 출력 복원을 관리한다[9]. Nextflow 25.04에서 도입된 실험적 실행 이력(data lineage) 기능은 출력의 체크섬과 생성 작업의 관계를 기록한다[10]. Snakemake의 워크플로 간 캐시도 입력, 매개변수 및 소프트웨어 환경을 식별에 사용한다[11]. 이력의 기록, 재사용 판정 및 출력 내용 검증은 구현과 설정에 따라 역할이 다르므로, 기능 목록과 실제 장애 대응을 각각 대조해야 한다.

저장 중단을 다루는 연구에서는 쓰기를 요청한 상태와 중단 후 복구 가능한 상태의 관계가 중요하다. Pillai 등은 응용 프로그램의 갱신 절차가 파일시스템의 지속성 특성에 의존하며, 그 차이가 장애 후 일관성에 영향을 준다는 점을 보였다[12]. 이 관점에서 전사 파일과 검증 기록의 갱신 사이에 중단이 발생하면 두 파일이 서로 다른 시점의 상태로 남을 수 있다. 본 연구는 구간 저장 응답 뒤의 중단을 E1에서, 세부 저장 단계 사이의 프로세스 중단을 별도 파일 실험에서 관찰한다. 두 실험을 연결하여 재사용 판정과 저장 순서가 맡는 역할을 해석한다.

4. Implications of On-Premises Deployment

온프레미스 처리는 모델과 데이터를 조직 내부에서 관리할 수 있는 배치 조건이다. NIST의 제로트러스트 지침은 물리적 또는 네트워크 위치와 별개로 접근 주체와 자원의 신뢰를 검증할 것을 강조한다[13]. 본 연구는 입력·모델·실행 코드와 제어기를 신뢰하는 조건에서 복구를 평가한다. 인증, 암호화, 키 분리, 관리자 권한 및 로그 관리는 운영 시스템에서 별도로 다루어야 할 보안 통제이다.

이들 연구의 공통 관심은 저장된 결과의 재사용 조건과 중단 뒤 반복할 계산의 범위이다. 체크포인트 연구는 복구 상태의 일관성을, 워크플로 연구는 입력·설정과 결과의 연결을, 장애 일관성 연구는 저장 과정에 남는 상태를 다룬다. 본 연구는 이 관점을 음성 전사에 적용하여 동일한 입력과 모델에서 정책별 전사 보존과 비용을 함께 측정한다. 현재의 기여는 지정한 결함에 대한 비교 결과와 재현 가능한 평가 기록에 있다. 실행조건 검사의 독립 효과 및 기존 도구 대비 차이를 판단하기 위한 비교 범위는 표 1에 정리하였다.

Table 1. Comparison scope of related work and this study

Study or system	Main evaluation focus	Relation to this study
Whisper and WhisperX [1,2]	Recognition and alignment	Whisper is the ASR backbone in E1
pyannote.audio [3]	Speaker diarization	Diarization component of the existing application
Bazel remote caching [6]	Reuse based on task specifications and content	Prior concept separating execution compatibility from output integrity
Nextflow · DVC · Snakemake [8,9,10,11]	Reuse based on inputs, configuration, and environment	No direct comparison yet; matched conditions required
Application crash consistency research [12]	Storage updates and durability conditions	Distinguishes evidence for process interruption from power loss
This study	Transcript preservation and recomputation cost	Policy comparison under injected faults with the same worker; failures retained in the analysis

III. Selective Recovery with Execution-Condition Validation

1. Tasks and Trust Boundaries

본 연구에서 산출물은 음성 구간을 처리하여 얻은 전사와 관련 정보를 저장한 파일을 뜻한다. 하나의 녹음을 네

www.kci.go.kr

구간으로 나누고, 구간 i의 산출물 A_i에 전사 문자열(text), 시간 구간 등의 목록(segments), 생성에 사용한 실행조건(binding)을 기록한다. 파일 형식은 JavaScript Object Notation(JSON)이다. 별도의 검증 기록 L_i에는 파일의 SHA-256(Secure Hash Algorithm 256-bit) 해시와 실행조건을 보관하며, 이하 검증 기록으로 부른다. 해시는 파일 바이트가 달라졌는지 확인하기 위한 요약값으로 사용한다. 검증 기록의 작성자를 인증하는 전자서명과는 역할이 다르다.

실행조건 B_i는 구간 식별자, 음성 입력 바이트의 해시, 모델 가중치의 해시, 추론 설정을 포함한 규약 파일의 해시로 구성한다. 제어기는 동결한 입력 명세를 기준으로 현재 작업의 기대 조건 B_i*를 계산한다. 파일과 검증 기록에 저장된 조건을 이 기대값과 대조하여 다른 입력·모델·설정의 결과를 구별한다. 실행 코드와 컨테이너 이미지는 별도의 동결 명세로 식별한다. 따라서 E1의 재사용 판정에 사용하는 정보와 실험 환경을 재현하기 위한 정보는 서로 연결되지만 역할이 다르다.

본 연구는 제어기·입력 명세·모델·실행 코드·검증 기록이 지정된 상태로 유지된다고 가정한다. E1에서는 검증 기록을 유지한 채 출력 파일에 손상을 적용하여, 저장된 검증 정보로 산출물의 변화를 탐지하는지 확인한다.

Fig. 1은 음성 구간과 실행조건에서 생성된 산출물이 재사용 판정을 거쳐 복구에 이용되는 관계를 보여준다. 입체 표현은 논리적 구성요소를 구분하기 위한 것이다.

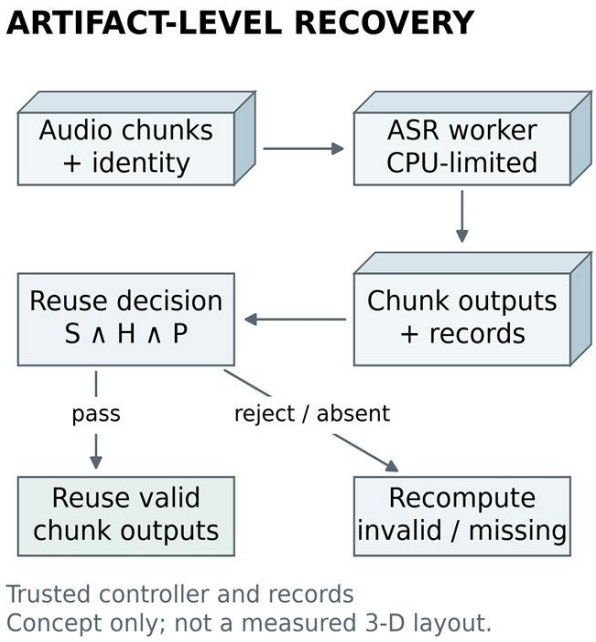


Fig. 1. Three-dimensional conceptual diagram of transcript artifact generation, validation, and reuse.

2. Reuse Decisions

재사용 검사는 세 질문으로 나뉜다. 첫째, 파일을 읽고 필요한 항목을 해석할 수 있는가이다. 구문 조건 S(A_i)는 JSON 해석 가능 여부, text의 문자열 여부와 segments의 목록 여부를 확인하는 최소 형식 검사다. 무음 구간의 빈 전사도 정상적인 출력으로 허용한다. 둘째, 저장 이후 파일이 바뀌었는가이다. 무결성 조건 H(A_i,L_i)는 현재 바이트의 해시와 검증 기록에 저장한 값을 비교한다. 셋째, 이 파일이 지금 수행하는 작업의 결과인가이다. 실행조건 일치 조건 P(A_i,L_i,B_i*)는 파일과 검증 기록의 생성 조건을 현재 작업이 요구하는 조건과 비교한다.

예를 들어 현재 작업은 모델 B를 요구하지만 모델 A가 만든 전사와 검증 기록이 함께 남아 있다고 하자. JSON을 읽을 수 있고 파일과 기록의 해시도 일치하면 구문 검사 S와 내용 검사 H를 통과한다. 생성 모델과 현재 요구 모델을 비교하는 P는 이 차이를 식별한다. 이는 검사 역할을 설명하는 사례이며, E1의 내용 변경 결함과 구별된다. 결합 정책은 세 조건을 모두 만족하는 구간을 재사용하고, 나머지 구간을 다시 추론한다.

재사용 규칙은 $R_i = S(A_i) \wedge H(A_i,L_i) \wedge P(A_i,L_i,B_i*)$이다. A_i는 전사 파일, L_i는 저장 당시의 검증 기록, B_i*는 현재 작업이 요구하는 실행조건이며, ∧는 모든 조건을 만족함을 뜻한다. 정상 산출물은 세 검사를 통과하여 재사용된다. JSON이 잘리면 S에서, 전사만 변경되면 H에서 재계산 대상으로 판정된다. 다른 모델로 생성된 파일과 그 검증 기록이 함께 남은 경우에는 S와 H를 통과할 수 있으나 P에서 현재 요구 모델과의 차이가 드러난다. 이 예시는 판정 규칙의 설명이며, 마지막 조건의 효과는 E1과 구별하여 검증한다.

판정은 현재 구간의 기대 조건을 읽는 단계에서 시작한다. 이어 정책별 검사를 수행하여 재사용할 구간을 선택하고, 나머지 구간을 추론한 뒤 결과와 검증 기록을 저장한다. 모든 구간이 준비되면 작업을 종료한다. 평가기는 이 결과를 무장애 기준 전사와 대조하여 전사 보존 여부를 집계한다. 실행 완료 상태와 평가 결과는 각각 기록한다.

전체 재시작은 모든 구간을 다시 추론하고, 구문 검사 정책은 S를 만족하는 결과를 재사용한다. E1의 결합 검증군은 S·H·P를 모두 적용한다. 후속 보완 실험의 출력 해시 전용 정책은 S와 H를 적용하며, 결합 검증군과 비교하여 P의 추가 효과를 살펴보도록 구성하였다.

Fig. 2는 세 검사의 질문과 실제 비교 정책을 대응시킨다. 파일이 읽힌다는 사실, 저장 내용이 유지됐다는 사실, 현재 실행에 적합하다는 사실은 서로 구별된다.

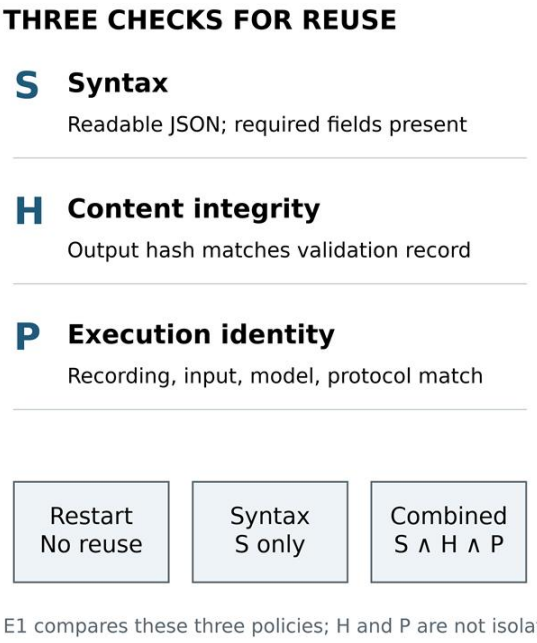


Fig. 2. Roles of syntax (S), content-hash (H), and execution-condition (P) checks and the policies compared in E1.

구문 검사군은 최소한의 파일 검사에 따른 동작을 관찰하는 비교군이며, 전체 재시작군은 전체 구간을 다시 계산하는 비용 기준이다. E1의 내용 변경 조건에서는 출력 해시 검사로 손상을 탐지할 수 있다. 실행조건 검사 P의 효과를 구별하려면 출력 해시 전용 정책과의 비교가 필요하다.

본 논문에서 결과의 확정은 복구 작업이 결과 파일을 남기고 종료한 상태를 뜻한다. 그 결과가 기준 전사와 다르면 정확성 실패로 집계하고, 원시 기록의 incorrect_commit 항목에 대응시킨다.

3. Storage and Result Commitment

저장 도중 중단될 때의 상태를 이해하려면 결과 파일과 검증 기록의 갱신 순서를 구분해야 한다. E1 작업자는 먼저 임시 파일에 전사를 쓴 뒤 결과 파일을 교체하고, 이어서 검증 기록도 같은 방식으로 갱신한다. 각각의 파일 교체와 두 파일을 함께 확정하는 것은 다른 문제다. 두 갱신 사이에서 중단되면 서로 다른 시점의 파일과 검증 기록이 남을 수 있다. E1은 구간 처리가 끝났다는 응답 이후 작업자를 중단하며, 저장 경계 자체의 관측은 별도 파일 실험에서 다룬다.

재시작한 작업자는 구간별로 검사를 수행해 유효한 결과를 재사용하고 나머지를 추론한다. 모든 구간을 수집한 뒤 작업 종료 상태를 기록하고, 장애 없이 전체 구간을 처리한 기준 결과의 구간별 전사 문자열 목록과 비교한다. 기본 실험의 정확성 지표는 이 문자열 목록의 완전 일치이다. 보완 실험은 문자열 일치와 전체 구조화 결과 일치를 별도로 기록하도록 확장하였다.

4. Interpretation and Dependency Scope

평가에서는 재사용 적합성, 기준 전사 보존, 정답 대비 인식 품질을 구분한다. 재사용 적합성은 현재 작업에 쓸 수 있는 결과인지를, 전사 보존은 장애 없이 얻은 결과와 같은지를 묻는다. 정답 대비 품질은 사람이 작성한 정답을 얼마나 정확히 인식했는지를 다룬다. 예를 들어 모델이 처음부터 틀리게 인식한 문장을 복구 후 그대로 반환하면 전사는 보존되지만 인식 오류는 남는다. 또 문장이 같아도 시간 정보가 달라질 수 있다. 이에 E1은 구간별 문자열의 완전 일치를 평가하고, 보완 실험은 구조화 결과와 실행조건 준수 여부도 각각 기록한다.

현재 구현은 각 30초 구간을 별도의 문맥으로 해독한다. 따라서 한 구간을 다시 처리할 때 다른 구간의 전사를 입력으로 사용할 필요가 없다. 앞선 전사를 문맥으로 이용하는 장시간 전사에서는 상황이 달라진다. 앞 구간의 변경이 뒤 구간의 해독에도 영향을 줄 수 있으므로 사용한 문맥과 생성 조건을 함께 기록하고, 재계산 범위가 후속 구간으로 확장되는지 검증해야 한다. 본 연구의 구간별 복구 결과는 이 독립 처리 조건에서 해석한다.

5. Operational Controls

모델·코드·입력을 읽기 전용으로 제공한 격리 컨테이너에서 하나의 실험 작업만 수행한다. 중앙처리장치(Central Processing Unit, CPU) 사용량은 2개, 메모리와 메모리·스왑 합계 제한은 각각 8GiB이며 그래픽처리장치(Graphics Processing Unit, GPU)와 외부 네트워크 접근을 비활성화한다. 기존 서비스의 상태와 가용 메모리를 감시하고, 사전에 정상인 서비스의 상태 악화나 자원 기준 위반이 관측되면 실험 작업만 중단한다. 서비스 감시는 실행 상태를 기준으로 수행한다.

IV. Experimental Design and Analysis

1. Data and Fixed Conditions

Augmented Multi-party Interaction(AMI) 자료는 공개 다자 회의 녹음을 제공하며[14], 본 연구는 고정 revision의 분할 명세[15]에 해당하는 16개 test 녹음을 사용하였다. 각 녹음의 처음 120.000초를 무음 포함 그대로 선택하여 네 개의 연속된 30초 구간으로 나누었다. 입력은 16kHz 단일채널 16비트 펄스부호변조(Pulse Code

Modulation, PCM) 음성이며, 원본 파형과 고정 구간을 그대로 사용하였다. 고유 입력량은 1,920초이다.

16개 녹음은 EN2002, ES2004, IS1009, TS3003의 네 시리즈에 속한다. 같은 녹음의 반복 실행은 처리 시간의 변동을 관찰하기 위한 기술적 반복이다. Hurlbert의 의사 반복(pseudoreplication) 논의는 측정 횟수와 독립 표본 수를 구별할 필요성을 설명한다[16]. 본 연구는 녹음과 시리즈의 의존성을 고려하여 기술통계로 결과를 요약하였다. 또한 test 자료의 기존 결과를 열람한 뒤 후속 실험을 설계한 이력이 있다.

추론 모델은 Whisper large-v3-turbo이며 CPU에서 두 스레드로 실행하였다. 언어는 영어, temperature는 0, beam size는 1, fp16과 이전 구간 text 조건화는 비활성화하였다. 모델 가중치, 실행 이미지, 프로토콜, 입력 명세 및 일정의 SHA-256을 동결하였다. 이 설정과 식별자를 재현을 위한 고정 조건으로 사용하였다.

E1 작업자는 Python의 OpenAI Whisper 구현에서 load_model과 transcribe를 직접 호출한다. 실험 작업자는 음성인식 단계로 구성하고, 입력 채널은 AMI Mix-Headset으로 고정하였다. 공유 호스트는 aarch64 구조의 Cortex-X925 10코어와 Cortex-A725 10코어를 갖추었으며, 실험 컨테이너에는 전체 호스트 성능과 구별되는 2 CPU 상당의 사용량 제한을 적용하였다. 실행 환경은 동결한 컨테이너 이미지로 식별하고, 모델·실행 코드·입력 명세의 해시를 함께 보관한다. 이 식별자는 부록 C에 제시하였다.

2. Fault Matrix and Execution Order

기본 실험 E1은 세 정책, 세 장애 유형, 두 중단 위치와 세 번의 기술적 반복을 결합하였다. 장애 실행은 16×3×3×2×3=864회이며 무장애 대조군 16×3=48회를 더해 총 912회이다. 각 녹음의 장애 없이 전체 구간을 처리한 기준 결과를 기준으로 두고 무장애 대조군의 일치를 확인하였다. 나머지 실행은 고정 seed 20261026에 따른 동결 일정으로 수행하였다.

첫 번째 또는 세 번째 구간이 저장된 뒤 준비 응답을 보내면 제어기가 실험 작업자를 SIGKILL로 종료한다. 단순 중단은 추가 파일 변경 없이 재시작한다. 잘림은 첫 구간 JSON 바이트를 절반 길이로 자르고, 내용 변경은 JSON 구조를 유지한 채 첫 구간 text를 CORRUPTED OUTPUT으로 바꾼다. 두 손상 조건에서 검증 기록은 그대로 둔다. 의도된 SIGKILL은 장애 주입 조건이며, 작업자 비정상 종료나 감시 실패 같은 예상하지 못한 실행 실패와 구별한다.

실험은 다음 순서로 진행한다. 먼저 장애 없이 네 구간을 처리해 비교 기준 전사를 만든다. 각 장애 실행에서는 처음부터 첫 번째 또는 세 번째 구간까지 실제로 추론하고, 결과 파일과 검증 기록 저장 뒤 준비 응답을 기다린다. 제어기는 그 시점의 작업자를 종료한 후 정해진 손상을 적용한다. 이어 새 작업자를 같은 정책으로 실행하여 구간별 재사용 여부를 판정하고, 재사용할 수 없는 구간만 추론한다. 마지막으로 네 구간의 전사 문자열을 기준 전사와 비교한다. 따라서 중단 전 추론, 의도된 중단, 손상 적용, 복구 및 결과 판정은 구분되는 단계다.

Table 2. Design of the baseline experiment E1

Item	Condition
Input	First 120 s of each of 16 recordings; 4 series
Processing unit	4 chunks of 30 s; no cross-chunk transcript conditioning
Policies	Full restart; syntax-based reuse; combined validation (S+H+P)
Fault	Interruption only; JSON truncation; syntax-preserving text corruption
Interruption position	After saving chunk 1 or 3
Repetitions and runs	3 technical repetitions per fault condition; 864 fault runs; 48 no-fault controls
Correctness	Exact chunk-level string equality with the clean full-restart transcript of the same recording
Limits	600 s per run; unexpected failures and partial results retained

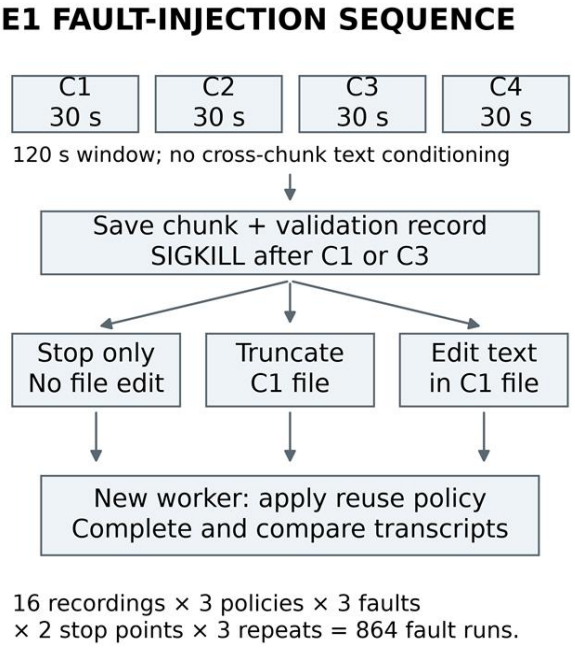


Fig. 3. Storage, interruption, corruption, and recovery sequence in E1. C1–C4 are 30-second audio chunks; execution is interrupted after the save acknowledgment for C1 or C3.

www.kci.go.kr

Fig. 3은 E1의 장애 주입 순서를 요약한다. 중단 후 변조는 첫 번째 구간 산출물에 적용하며, 복구가 끝난 뒤 무장애 기준 전사와 비교한다.

무장애 대조는 녹음 16개와 정책 3개의 조합으로 48회이며 각 조합을 한 번 실행하였다. 장애 실행은 16녹음×3장애×2중단 위치×3정책×3반복으로 864회이다. 장애 복구 결과의 분모는 864회이다.

3. Evaluation Metrics and Conditional Timing Analysis

복구의 정확성은 동일 입력과 모델을 장애 없이 실행했을 때의 전사를 기준으로 판단한다. 이 기준 전사와 구간별 문자열 목록을 비교하여 복구 과정의 결과 변화를 확인한다. 정상 종료한 실행이라도 문자열 목록이 다르면 잘못된 확정으로 집계한다. 실행 실패와 시간 초과, 무장애 대조는 각각 별도 항목으로 기록한다.

재계산 음성량 C는 장애 후 실제 추론한 구간 길이의 합이다. 장애 후 지연 L은 SIGKILL 직전부터 복구 작업자가 종료되고 결과와 기준 전사를 읽은 직후까지 측정한다. 손상 적용, 작업자 재시작, 모델 로딩, 검사, 추론 및 저장이 포함된다. 전체 경과시간 T는 최초 작업자 실행 직전부터 같은 종료 시점까지로, 중단 전 추론까지 포함한다. 최종 result.json 저장, 컨테이너 준비와 배치 간 대기는 별도의 운영 시간이다. 따라서 L은 실험 제어와 결과 읽기를 포함한 경과시간이며, 실제 서비스의 사용자 체감 복구시간으로 적용하려면 같은 측정 경계를 확인해야 한다. 본문에서는 L을 주 지표로 사용하고, 보관된 T의 사후 분석을 표 5와 부록 E에 제시한다.

녹음마다 발화량과 처리 난도가 다르므로 같은 녹음을 처리한 정책끼리 짝지어 비교한다. 전체 재시작 대비 정책 a의 차이는 ΔC=C_a-C_restart, ΔL=L_a-L_restart이며 음수는 비용 감소를 뜻한다. 같은 녹음·장애·중단 위치·반복에 해당하는 두 실행의 차이를 먼저 구하고, 녹음별 세 차이의 중앙값을 계산한다. 마지막으로 녹음별 값에 동일한 가중치를 부여하여 평균한다. 중앙값은 한 번의 큰 시간 변동이 요약값을 좌우하는 영향을 줄인다. 세 반복 모두에서 두 정책이 기준 전사를 보존한 녹음을 분석 대상으로 하였고, 표 4의 각 조건에는 16개 녹음이 포함되었다. 이 순서를 사용하므로 정책별 중앙값을 각각 계산한 뒤 뺀 값과는 차이가 생길 수 있다.

복구 정책을 평가할 때는 정확성과 속도를 함께 살펴야 한다. 틀린 전사를 빠르게 반환한 실행까지 지연 평균에 섞으면 복구 성능을 잘못 해석할 수 있다. 따라서 모든 정확성 실패는 표 3에 집계하고, 시간은 올바르게 복구된 조건을 대상으로 비교한다. 이때 분석에서 빠지는 조건과 분모도 함께 제시한다. 30초 구간 k개를 재사용하면 재계산 음성량은 30k초 감소한다. 반면 실제 시간에는 모델 로딩, 검사, 저장 및 호스트 상태도 영향을 준다. 음성량과 시간을 각각 보고하는 이유는 계산량의 감소가 실제 대기시간의 감소로 얼마나 이어지는지 확인하기 위해서다.

4. Role of Supplementary Validation

실행조건 검사 P의 효과를 구별하기 위해 E1 중간 결과를 검토한 뒤 출력 해시 전용 비교군과 오래된 정상 산출물·검증 기록 쌍 조건을 추가한 보완 실험을 설계하였다. 보완 설계와 결과는 별도 프로토콜 및 경로로 관리하며, 기존 동결 결과와 실패 사례를 함께 보존한다.

보완 검증은 확인하려는 수준에 따라 나누었다. 파일 저장 경계 실험 E3a에서는 실제 프로세스 중단 14사례와 무중단 대조 2사례를 통해, 중단 시점에 남은 파일과 검증 기록을 복구 정책이 어떻게 선택하는지 확인하였다. 개발 통합 검증에서는 다른 모델·설정으로 실제 생성한 대체 산출물(donor)을 사용하여 ASR 중단부터 복구까지 두 작업이 이어지는지 확인하였다. 전자는 파일 선택 규칙을, 후자는 구현과 실행 비용을 점검하며 E1과 별도로 보고한다. 별도 보완 본실험은 무장애 64회와 장애 896회의 총 960회 및 대체 산출물 생성 32회로 동결하여 진행 중이다. 보완 본실험의 정책 간 효과는 전체 결과를 확인한 뒤 별도로 평가할 예정이다.

V. Results

1. Run Completion and Transcript Preservation

총 912회는 모두 정상 종료했고 예상하지 못한 실행 실패와 시간 초과는 없었다. 기준 전사와 일치한 816회는 무장애 대조 48회와 장애 실행 768회로 구성된다. 장애 실행 864회 가운데 나머지 96회에서는 잘못된 전사가 확정되었다.

표 3의 각 장애·정책 셀은 16개 녹음, 두 위치 및 세 반복을 포함한 96회이다. 내용 변경에서 구문 검사군은 96회 모두 잘못된 전사를 확정하였다. 중단 후 text만 변경하고 JSON 형식과 자료형을 유지했으므로 재사용 조건을 통과하였다. 이는 정상 종료한 작업이 변경된 전사를 반환한 정확성 실패이다. 단순 중단과 잘림에서는 세 정책 모두 셀별 96회 전사를 보존하였다. 검증 정책과 전체 재시작은 내용 변경에서도 각각 96회 전사를 보존하였다.

www.kci.go.kr

Table 3. E1 results by fault type and policy

Fault	Policy	Runs	Matches	Incorrect commits
Stop only	Full restart	96	96	0
Stop only	Syntax check	96	96	0
Stop only	Combined validation	96	96	0
Truncated JSON	Full restart	96	96	0
Truncated JSON	Syntax check	96	96	0
Truncated JSON	Combined validation	96	96	0
Changed text	Full restart	96	96	0
Changed text	Syntax check	96	0	96
Changed text	Combined validation	96	96	0
Total	Fault runs only	864	768	96

96회 오류는 동일한 손상·정책 조합을 반복 관측한 결과이다. 핵심은 정상 종료 상태가 같은 실행들 사이에도 전사 보존 여부가 달랐다는 점이다.

Fig. 4에서 실행 종료와 전사 보존의 차이를 확인할 수 있다. 내용 변경 조건의 구문 검사군은 96회 모두 종료했지만 기준 전사를 보존하지 못했다.

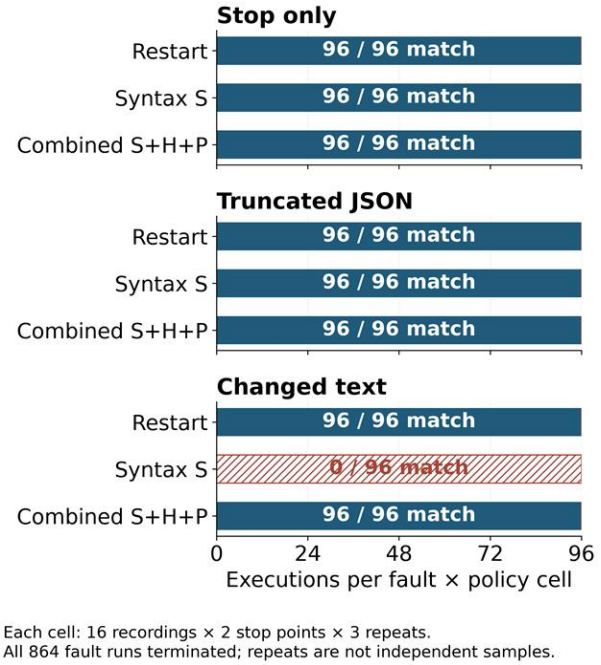


Fig. 4. Transcript preservation by policy in 864 E1 fault-injection runs. Each bar represents 16 recordings x 2 interruption positions x 3 repetitions = 96 runs. Hatching denotes incorrect transcript commitments; the 48 no-fault controls are excluded. Technical repetitions of the same recordings are included.

2. Recomputation and Latency in Correct Recovery

표 4는 결합 검증군과 전체 재시작군을 비교한다. 모든 행에서 16개 녹음의 세 쌍체 반복이 전사 보존 조건을 만족하였다. 첫 번째 구간 저장 후 단순 중단하면 검증군은 나머지 90초만 계산하여 30초를 절감하였다. 세 번째 구간 저장 후에는 마지막 30초만 계산하여 90초를 절감하였다. 대응하는 지연 차이는 각각 -16.061초와 -57.312초였다.

첫 번째 구간에서 잘림 또는 내용 변경이 발생하면 그 구간도 다시 계산해야 하므로 네 구간 모두, 즉 120초를 재계산하였다. 재계산 절감은 0이며 지연 차이도 각각 -0.001초와 +0.023초였다. 두 조건의 지연 차이는 0에 가까웠다. 세 번째 구간까지 저장한 뒤 첫 구간이 손상되면 두 번째와 세 번째 구간을 유지할 수 있어 60초를 절감하였다.

Table 4. Conditional differences between combined validation and full restart (n: valid recordings)

Fault	Saved chunks	Recomp. audio (s)	ΔC (s)	ΔL (s)	Valid n
Stop only	1	90	-30	-16.061	16
Stop only	3	30	-90	-57.312	16
JSON trunc.	1	120	0	-0.001	16
JSON trunc.	3	60	-60	-41.366	16
Text edit	1	120	0	+0.023	16
Text edit	3	60	-60	-41.439	16

ΔL은 녹음별 세 쌍체 차이의 중앙값을 동일 가중 평균한 값이다. E1의 전체 재시작은 모든 조건에서 장애 후 120초를 재계산한다. 구문 검사군도 단순 중단과 잘림에서는 비슷한 재계산 절감을 보였으나, 내용 변경에서는 두 위치 모두 16개 녹음에 정확성 실패가 있어 성공 조건부 지연 대비를 정의하지 않았다.

Fig. 5는 표 4의 결합 검증군과 같은 집계 규칙으로 구문 검사군도 함께 나타낸다. 올바르게 복구한 조건에서는 두 재사용 정책의 비용 절감이 유사하며, 결합 검증의 의미는 절감량 자체뿐 아니라 재사용을 허용하는 조건에 있다.

www.kci.go.kr

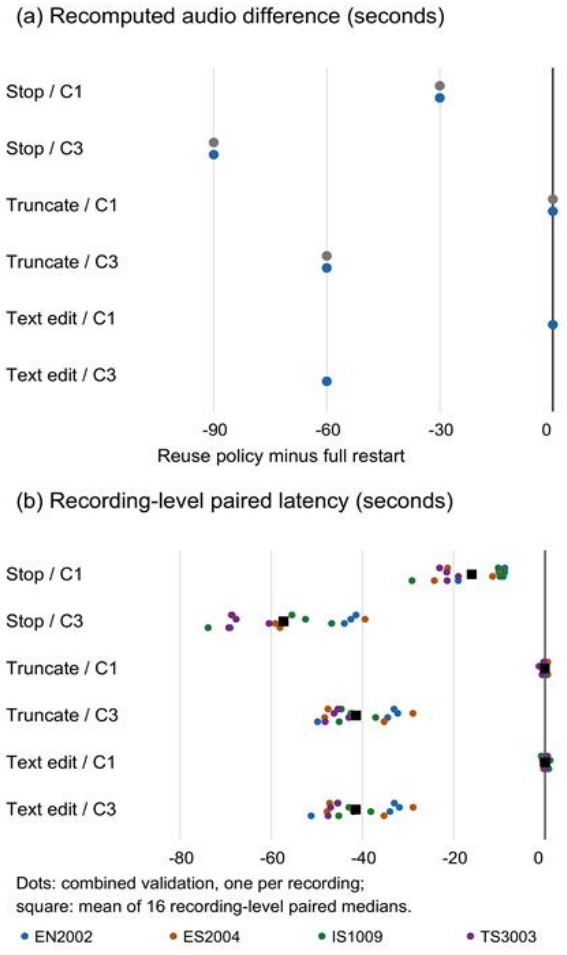


Fig. 5. Recomputed audio duration (a) and recording-level paired latency differences (b), relative to full restart. In (a), gray denotes syntax-based reuse and blue denotes combined validation; conditions with correctness failures are excluded. In (b), dots show each recording's median of 3 paired differences for combined validation, and squares show the equally weighted mean across 16 recordings. Colors distinguish the 4 meeting series.

기존 E1 기록의 사후 분석으로 중단 전 비용까지 포함한 전체 경과시간 T를 표 5에 제시한다. 단순 중단 위치 3의 장애 후 지연 L은 전체 재시작 83.52초, 구문 검사 26.19초, 결합 검증 26.19초였다. T는 각각 145.20초, 87.80초, 87.93초로 나타났다. 두 재사용 정책에 공통으로 나타난 절감은 정상 산출물의 재사용과 연결되며, 추가 검증의 가치는 정확성을 유지하는 조건과 추가 비용을 함께 살펴야 한다.

첫 구간이 손상된 위치 1 조건에서는 결합 검증군의 전체 경과시간 차이가 잘림 +0.066초, 내용 변경 +0.009초였다. 이 조건에서도 전체 경과시간 차이는 0에 가까웠다. 부록 E는 성공 조건부 절대 시간과 전체 기록 비용을 구분하여 제시한다.

Table 5. Post hoc analysis of total elapsed time T for correct recovery

Condition	Pos.	Full restart	Syntax check	Combined validation	Runs per policy
Stop only	1	104.01	87.73	87.97	48
Stop only	3	145.20	87.80	87.93	48
Truncated JSON	1	103.82	103.90	103.80	48
Truncated JSON	3	145.36	103.68	103.95	48
Changed text	1	103.77	N/A	103.81	48
Changed text	3	145.24	N/A	103.76	48

표 5의 시간 단위는 초이며, 각 값은 16개 녹음의 반복 중앙값에 동일 가중치를 부여한 값이다. 내용 변경의 구문 검사군은 위치별 48회 모두 정확성 실패여서 성공 조건부 시간은 해당 없음으로 표시하였다. 이 96회의 누적 전체 경과시간 8,417.51초는 전체 비용에 포함하였다. 표 4의 쌍체 차이는 계산 순서가 달라 표 5의 두 열을 뺀 값과 구별된다.

3. Storage Boundaries and Development Integration

E3a의 16개 파일 사례에서는 기대한 저장 상태와 정책 결정이 모두 관측되었다. 오래된 정상 산출물과 그에 대응하는 해시 검증 기록이 함께 남는 세 사례에서 구문 검사군과 출력 해시군이 각각 세 번 잘못된 재사용을 허용하였다. 전체 재시작과 결합 검증군의 잘못된 재사용은 이 사례 집합에서 0이었다. 이 관측은 파일 선택 단계에서 출력 해시가 일치하는 다른 작업의 산출물까지 재사용될 수 있음을 보여준다. 저장 경계별 파일 상태와 사례 식별자는 부록 C에 정리하였다. E3a의 파일 내용과 실행조건 표지는 합성 자료이며, 실제로 중단한 것은 파일 작성 프로세스이다. 실제 음성 추론을 중단한 개발 통합 두 작업과 구별한다.

별도 공개 개발 회의에서 다른 모델과 다른 추론 설정의 donor를 실제 생성한 뒤 두 ASR 중단·복구 통합 작업을 실행하였다. 첫 작업은 한 구간 추론 후 실제 SIGKILL, 다른 모델 산출물 적용 및 검증 정책 복구를 수행했고, 두 번째는 세 구간 추론 후 SIGKILL, 다른 설정 산출물 적용 및 전체 재시작을 수행하였다. 두 작업 모두 전사와 구조화 결과가 기준과 일치하였다. 두 작업은 서로 다른 조건에서 중단과 복구의 연결을 확인한 개발 사례이다.

www.kci.go.kr

표 C2는 두 실행의 구현 및 자원계획 증거다. 각각 120초를 다시 추론했고 같은 개발 회의에 의존한다. 통합 실행의 예상외 실패와 시간 초과는 없었으며 76개 서비스 감시 표본에서 사전에 정상인 서비스의 상태 악화를 관측하지 않았다. 비용은 중단 전·후 추론시간과 호스트 경과시간으로 구분하여 기록하였다.

VI. Discussion and Threats to Validity

1. Interpretation of the Research Questions

RQ1의 결과는 파일을 해석할 수 있다는 사실과 전사 내용이 보존되었다는 사실이 다름을 보여준다. 내용이 바뀌어도 JSON 형식이 유지되면 구문 검사만으로 이를 가려내지 못했다. RQ2에서는 유효한 구간을 재사용할 때 재계산량과 장애 후 지연이 줄었지만, 첫 구간부터 다시 계산해야 하는 조건에서는 이득이 거의 없었다. 두 결과를 함께 보면 복구 정책은 종료 여부뿐 아니라 결과 보존 여부로 평가하고, 비용은 실제로 재사용할 수 있는 구간의 범위와 연결해 해석해야 한다.

구문 검사군은 최소 기능 대조군이며, 내용 변경에서의 실패는 검사 범위로부터 예상되는 결과다. 특히 text만 바꾼 결함은 필드 간 일관성 검사로도 탐지될 가능성이 있다. E1은 기본 검사의 한계와 복구 비용을 설명하는 근거로 해석한다. E1에서 함께 적용한 해시와 실행조건 검사의 효과는 해시 전용군을 포함한 동결 보완 실험에서 구별하고, 기존 방법과의 차이는 동일 조건의 별도 비교로 확인할 과제다.

2. Metric Scope and Reorganization of Prior Results

전사 보존은 같은 모델의 무장애 출력을 기준으로 평가하므로 원래 모델의 오인식도 유지될 수 있다. 반면 단어오류율(Word Error Rate, WER)과 문자오류율(Character Error Rate, CER)은 정답 전사를 기준으로 인식 품질을 평가한다. 화자분리오류율(Diarization Error Rate, DER)은 화자·시간 참조와 평가 구간·겹침 처리 조건을 요구한다. 기존 원고의 한국어 인식·화자분리·요약 점수는 이러한 채점 절차의 재검증이 필요한 항목으로 구분하였다.

기존 개념검증에서 자동 화자 수 일치는 0/8이고 구간 간 화자 연결 DER은 74.05%였다. 이 결과는 원래 조건의 진단 기록으로 보존하며 E1의 복구 평가와 구분한다. 실제 인물 식별과 담당자 자동 귀속의 적용 가능성을 판단하려면 화자 연결 성능에 대한 추가 검증이 필요하다.

3. Generalization and Statistical Interpretation

평가 자료는 영어 공개 회의 네 시리즈의 16개 녹음에서 추출한 앞 120초 구간이다. 따라서 장시간 회의의 중·후반 발화, 구간 간 문맥 의존성, 한국어 건설 회의에 대한 일반화에는 한계가 있다. 녹음과 시리즈의 의존성 및 조건별 세 번의 기술적 반복을 고려하여 결과를 기술통계로 제시하였다. 모집단 신뢰구간·유의성 검정·꼬리 지연 추정을 위해서는 독립 자료와 추가 측정이 필요하다.

사전에 열람한 test 자료와 E1 중간 결과가 보완 설계에 영향을 주었다는 점도 한계다. 후속 독립 코퍼스 검증에서는 평가 전에 모델과 조건을 확정하고, 자료 및 정답 참조를 열람한 이력을 함께 기록해야 한다.

주입한 CORRUPTED OUTPUT과 파일 절반 잘림은 재현 가능한 인위적 결함이다. 결과는 지정한 결함에 대한 정책의 반응을 설명하며, 실제 운영의 장애 분포와 신뢰도를 추정하려면 현장 발생 빈도 자료가 추가로 필요하다. 보완 실험도 같은 16개 녹음을 사용하므로 독립 코퍼스 검증은 별도 과제로 남는다.

4. Operational and Security Limitations

CPU 사용량을 제한한 공유 호스트에서 측정한 절대 지연은 장치·부하·캐시 상태와 처리 순서의 영향을 받는다. 서비스 감시는 사전에 기록한 상태와 비교하는 운영 통제로 수행하였다. 요청 지연과 가용성에 대한 영향은 별도 측정이 필요한 항목이다.

E1과 E3a의 관측 범위는 신뢰된 검증 기록과 제어기 아래에서 발생한 프로세스 중단이다. 검증 기록과 파일의 동시 위조, 호스트 침해, 전원 상실 및 저장장치 손상에 대한 방어와 내구성은 추가 검증이 필요하다. 실제 운영에서는 입력·모델 배포, 권한 분리, 로그 보존, 키 관리 및 사고 대응 절차를 함께 마련해야 한다.

5. Reproducibility and Research Ethics

실행별 결과·오류·부분 파일과 동결 명세는 실패 사례까지 포함하여 원래 상태로 보존한다. 기본 프로토콜은 KSCI-ARTIFACT-RECOVERY-MAIN-20261026으로 식별하며, 해시와 입력·코드·일정의 대응은 별도 실행 명세에 제시한다. 재현 자료의 배포 시에는 데이터·코드·모델의 이용조건을 각각 확인하고 내부 접속 정보와 자격증명을 제거해야 한다.

실행 증거는 현재 로컬 재현 자료로 보관하고 있으며,

www.kci.go.kr

공개 접근 위치와 외부 재현 실행은 후속 확인 항목이다. 이전 학술대회 개념검증[17]에서 응용 배경과 시스템 구성을 계승하고, 이번 연구에서 장애 프로토콜·실험·분석을 추가하였다.

6. Total-Cost Interpretation and Comparison Conditions

운영 총비용을 비교하려면 정상 실행 비용, 중단 전 추론, 장애 후 재추론·모델 로딩·검사·저장·제어 비용을 같은 측정 경계로 합산해야 한다. 본문은 E1의 장애 후 지연을 주 지표로 사용하고, 중단 전 비용을 포함한 전체 경과시간의 사후 분석을 부록 E에 제시하였다. 실제 운영의 평균 절감률을 추정하려면 장애 발생 빈도도 함께 측정해야 한다.

기존 워크플로와의 비교에서는 동일한 음성·모델·설정·구간 의존성과 자원 제한을 제공해야 한다. 먼저 출력 해시 전용 정책과 결합 검증을 비교하여 실행조건 검사의 추가 효과를 분리한다. 다음으로 기존 도구를 같은 정보와 재사용 단위로 구성하여 정상 파일의 재사용, 내용 손상 탐지, 다른 작업의 정상 산출물 처리 및 복구 비용을 대조한다. 정확성이 같으면 정상 실행의 추가 비용과 재계산 범위를 분석한다. 이 비교와 독립 장시간 자료 검증은 현재 결론의 적용 범위를 확장하기 위한 후속 과제이다.

VII. Conclusion

본 연구는 건설기업을 위한 온프레미스 회의 파이프라인의 산출물 완전성과 체크포인트 복구를 구체화하였다. 전사 재사용 문제를 산출물 구문, 바이트 무결성 및 실행조건 일치로 분해하고, 같은 모델·입력·저장 형식의 장애 주입 실험으로 평가하였다. 총 912회가 종료되었지만 내용 변경과 구문 검사 정책의 조합에서는 96회 모두 잘못된 전사가 확정되었다. 이 결과는 실행 종료와 전사 보존을 함께 평가할 필요성을 보여준다. 검증 기반 재사용은 이 제한된 행렬에서 전사를 보존했으며 유효한 체크포인트가 남는 조건에서만 재계산 음성과 장애 후 지연을 줄였다.

지정한 장애 조건에서 내용 검증은 형식을 유지한 손상을 탐지하는 데 기여하였고, 비용 절감의 크기는 재사용 가능한 정상 구간의 범위에 따라 달라졌다. 따라서 운영자는 파일 검사 수준과 복구 비용을 함께 고려하여 정책을 선택할 수 있다. 다음 검증은 출력 해시 전용 정책 및 동일 조건으로 구성한 기존 워크플로와의 비교를 중심으로 수행하며, 독립 장시간 자료에서 적용 범위를 확인한다.

REFERENCES

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, Robust Speech Recognition via Large-Scale Weak Supervision, Proceedings of the 40th International Conference on Machine Learning, PMLR, Vol. 202, pp. 28492–28518, 2023. *https://proceedings.mlr.press/v202/radford23a.html*

[2] M. Bain, J. Huh, T. Han, and A. Zisserman, WhisperX: Time-Accurate Speech Transcription of Long-Form Audio, INTERSPEECH 2023, pp. 4489-4493, 2023. *https://doi.org/10.21437/Interspeech.2023-78*

[3] H. Bredin, pyannote.audio 2.1 Speaker Diarization Pipeline: Principle, Benchmark, and Recipe, INTERSPEECH 2023, 2023. *https://doi.org/10.21437/Interspeech.2023-105*

[4] E. N. Elnozahy, L. Alvisi, Y.-M. Wang, and D. B. Johnson, A Survey of Rollback-Recovery Protocols in Message-Passing Systems, ACM Computing Surveys, Vol. 34, No. 3, pp. 375-408, 2002. *https://doi.org/10.1145/568522.568525*

[5] M.-C. Hsueh, T. K. Tsai, and R. K. Iyer, Fault Injection Techniques and Tools, Computer, Vol. 30, No. 4, pp. 75-82, 1997. *https://doi.org/10.1109/2.585157*

[6] Bazel Contributors, Remote Caching, Bazel Documentation. *https://bazel.build/remote/caching* (Accessed October 29, 2026)

[7] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, Nextflow Enables Reproducible Computational Workflows, Nature Biotechnology, Vol. 35, pp. 316-319, 2017. *https://doi.org/10.1038/nbt.3820*

[8] Nextflow Contributors, Caching and Resuming, Seqera Documentation. *https://docs.seqera.io/nextflow/cache-and-resume* (Accessed October 29, 2026)

[9] Iterative, Running Pipelines, DVC Documentation. *https://doc.dvc.org/user-guide/pipelines/running-pipelines* (Accessed October 29, 2026)

[10] Seqera, Getting Started with Data Lineage, Nextflow Documentation, experimental feature introduced in version 25.04. *https://docs.seqera.io/nextflow/tutorials/data-lineage* (Accessed October 29, 2026)

[11] Snakemake Contributors, Between Workflow Caching, Snakemake 8.20.4 Documentation. *https://snakemake.readthedocs.io/en/v8.20.4/executing/caching.html* (Accessed October 29, 2026)

[12] T. S. Pillai, V. Chidambaram, R. Alagappan, S. Al-Kiswany, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, All File Systems Are Not Created Equal: On the Complexity of Crafting Crash-Consistent Applications, OSDI 14, pp. 433-448, 2014. *https://www.usenix.org/conference/osdi14/technical-sessions/presentation/pillai*

[13] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero Trust Architecture*, NIST SP 800-207, 2020. *https://doi.org/10.6028/NIST.SP.800-207*

www.kci.go.kr

[14] AMI Consortium, AMI Corpus. *https://groups.inf.ed.ac.uk/ami/corpus/* (Accessed October 29, 2026)

[15] Pyannote Contributors, AMI Diarization Setup, a fork of the BUT Speech@FIT setup, revision 67c2d539286e89f68952d5dcf83912bd9f01dfae. *https://github.com/pyannote/AMI-diarization-setup/tree/67c2d539286e89f68952d5dcf83912bd9f01dfae*

[16] S. H. Hurlbert, Pseudoreplication and the Design of Ecological Field Experiments, Ecological Monographs, Vol. 54, No. 2, pp. 187-211, 1984. *https://doi.org/10.2307/1942661*

[17] J. Kim, J. Yun, J. Lee, and K. Kim, A Secure On-Premises Meeting Intelligence Pipeline Using Edge AI for Construction Enterprises, Proceedings of the 74th Summer Conference of the Korea Society of Computer and Information, Jeju, Republic of Korea, July 2026.

Appendix A. Evaluation Scope and Further Validation

본문 표 3과 표 4는 완료된 E1에서 계산한 값이고, 표 C2는 별도 공개 개발 회의에서 완료된 두 통합 작업의 값이다. 각 프로토콜의 결과는 해당 실험의 분모로 별도 집계하였다. 본 논문의 결과 범위는 2026년 10월 29일 원고 확정 시점에 완료하고 검토한 이 증거로 한정하였다.

출력 해시 전용 비교군, 다른 입력·모델·설정에서 생성한 산출물, 독립 코퍼스 검증은 후속 과제다. 진행 중인 보완 본실험의 결과는 완료된 E1 및 개발 사례와 구분하여 관리한다.

공식 정답 대비 품질 평가는 별도 과제다. 기존 자체 정규화 점수와 원 배포 채점 절차의 동등성을 확인하고, 화자 귀속 평가는 화자·시간 참조를 갖춘 조건에서 수행해야 한다. 본 논문의 평가는 동일 입력과 모델의 무장애 기준 실행 대비 전사 보존을 기준으로 한다.

집계 순서는 반복별 쌍체 차이를 계산한 뒤 녹음별 중앙값을 구하고, 마지막으로 16개 중앙값을 평균하는 것이다. 예를 들어 한 녹음의 세 반복 차이가 −10, −12, −30초라면 해당 녹음의 대표 차이는 −12초이다(설명용 가상 수치). 정책별 중앙값을 먼저 구해 빼면 같은 값이 되지 않을 수 있다. 부록 E의 절대 시간 평균과 쌍체 차이는 각각의 집계 순서에 따라 계산한다.

Appendix B. Variation across Recordings

표 B1은 기존 표 4의 추정량과 포함 규칙을 유지하면서 녹음별 쌍체 차이를 요약한 사후 기술 분석이다. 범위는 관측된 최소·최대값이며, 시리즈 평균은 각 시리즈의 네 녹음에 동일 가중치를 부여하였다.

Table B1. Latency-difference distributions and series means for combined validation

Condition	Min-max (s)	EN2002	ES2004	IS1009	TS3003
Stop only, position 1	-29.131 ~ -8.801	-11.757	-16.722	-14.513	-21.251
Stop only, position 3	-73.841 ~ -39.440	-49.238	-56.353	-57.140	-66.518
Truncated JSON, position 1	-1.287 ~ 0.613	0.037	0.416	0.030	-0.487
Truncated JSON, position 3	-49.834 ~ -28.943	-37.416	-40.007	-42.350	-45.693
Changed text, position 1	-0.798 ~ 1.117	0.158	-0.465	0.394	0.007
Changed text, position 3	-51.253 ~ -28.901	-37.542	-39.787	-42.928	-45.498

작은 양수와 음수가 함께 나타나는 조건은 전체 재시작과의 차이가 0 부근에서 변동했음을 보여준다. 세 번의 기술적 반복은 해당 실행 조건의 변동을 살펴보기 위한 것이다.

Appendix C. Execution Specifications and Storage-Boundary Observations

E1은 AMI Mix-Headset의 16 kHz 모노 PCM 입력, 30초 구간, OpenAI Whisper large-v3-turbo, CPU 두 스레드, 영어, temperature 0, beam size 1, fp16 비활성화 및 이전 구간 전사 조건화 비활성화를 사용한다. 컨테이너에는 2 CPU 상당과 메모리 8 GiB를 할당하고 GPU 및 외부 네트워크 접근을 비활성화하였다. 이미지·모델·코드·입력·일정의 전체 식별자는 함께 제공한 execution_manifest.json에 보관한다. Python·PyTorch·Whisper 패키지 버전과 동결 이미지 내부의 대조 및 외부 재현 실행은 후속 확인 항목이다.

E3a에서는 빈 초기 상태와 다른 입력 조건으로 생성된 정상 파일·검증 기록 쌍이 남은 초기 상태에 대해 같은 여덟 위치를 확인하였다. 앞의 일곱 위치에서는 파일 작성 프로세스를 실제로 종료하고 마지막은 무중단 대조로 두어 총 16사례를 구성하였다. 임시 파일은 재사용 대상에서 제외하며, 표 C1의 파일과 검증 기록은 최종 경로에 남은

www.kci.go.kr

상태를 뜻한다. 새 파일은 현재 기대 실행조건, 옛 파일은 다른 입력 조건을 가진 합성 자료이다.

Table C1. Expected states and observations at storage boundaries

Case ID	Interruption or control point	Artifact / validation record (empty state)	Artifact / validation record (old pair)
00 / 08	Before opening the temporary artifact file	Absent / Absent	Old / Old
01 / 09	After a partial write to the temporary artifact file	Absent / Absent	Old / Old
02 / 10	After synchronizing the temporary artifact file, before replacement	Absent / Absent	Old / Old
03 / 11	After artifact replacement, before writing the validation record	New / Absent	New / Old
04 / 12	After a partial write to the temporary validation record file	New / Absent	New / Old
05 / 13	After synchronizing the temporary validation record file, before replacement	New / Absent	New / Old
06 / 14	After validation record replacement, before the completion marker	New / New	New / New
07 / 15	Uninterrupted control	New / New	New / New

사례 번호는 보관 기록 e3a-plan-00부터 e3a-plan-15에 대응한다. 16사례 모두 표 C1의 기대 상태와 관측이 일치하였다. 08·09·10에서는 옛 파일과 그 해시 검증 기록이 함께 남아 구문 검사와 출력 해시 검사만으로는 잘못된 재사용을 막지 못했다. 결합 검증은 현재 기대 입력 조건과 다름을 확인하여 이를 거부하였다. 이 결과는 지정한 저장 경계에서 파일 선택 규칙의 동작을 보여준다.

개발 ASR 통합 두 작업은 같은 공개 개발 회의를 사용했다. 다른 모델 조건의 대체 산출물은 tiny.en으로 생성하고 복구 목표는 large-v3-turbo로 두었다. 다른 설정 조건은 large-v3-turbo의 beam size 2로 생성한 산출물을 beam size 1인 목표 작업에 적용하였다. 각각 첫 구간과 세 번째 구간 추론 후 중단하였으며 정책은 결합 검증과 전체 재시작으로 서로 달랐다. 표 C2는 이 두 개발 사례에서 실제 중단·복구 연결과 실행 비용을 기록한 결과이다.

Table C2. Cost breakdown of development ASR integration runs

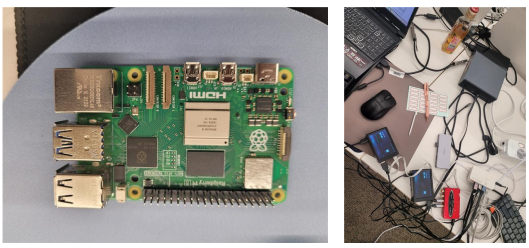
Fixed case	Pre-stop inference (s)	Recovery inference (s)	Total inference (s)	Host time (s)
Different model; stop 1; validated reuse	25.811	116.741	142.552	155.433
Different config.; stop 3; full restart	85.552	117.222	202.775	215.426

Appendix D. PoC Materials and Their Relation to the Experiment

기업 회의 전사 시스템의 응용 배경을 설명하기 위해 PoC 장치 구성을 Fig. 6에 제시한다. Raspberry Pi 5 보드와 기가바이트 DGX Spark, 오디오 인터페이스 및 수집 장치가 연결된 개발 환경을 함께 보여준다.

프로젝트 중간보고서에는 보안 환경용 음성 수집, 전사·발화자 구분 및 회의록 요약을 개발 대상으로 명시하고, 회의 기록 목록과 참석자별 전사문·요약 화면을 확인한 사실을 기록하였다. 이는 전사 산출물이 후속 열람과 요약에 사용되는 응용 맥락을 보여준다. 이 과정에서 잘못된 중간 산출물을 재사용하면 오류가 후속 처리로 이어질 수 있으므로, 본 연구는 전사 파일의 재사용 판정과 복구 비용에 평가 범위를 한정하였다.

본문의 정량 평가는 공개 AMI 녹음을 사용하였다. PoC 사진은 음성 수집 장치와 연산 장비가 연결되는 응용 환경을 설명한다.



(a) (b)

Fig. 6. PoC hardware setup: (a) Raspberry Pi 5 board; (b) GIGABYTE DGX Spark connected to peripheral devices.

자료 출처: 저자 제공 PoC 장비 사진 및 「협업과제 중간보고서」(2026년 8월 17일).

www.kci.go.kr

Appendix E. Post Hoc Analysis of Total Elapsed Time

이 사후 분석은 E1 종료 후 보관된 912개 result.json을 다시 읽어 수행하였다. 각 조건의 48회는 16녹음의 3회 반복이다. 아래 수치는 올바른 결과를 반환한 실행에서 녹음별 반복 중앙값을 구한 뒤 16녹음을 동일 가중 평균한 초 단위 값이다. 정책 순서는 전체 재시작, 구문 검사, 결합 검증이다. 괄호 앞은 장애 후 지연 L, 괄호 안은 중단 전 추론을 포함한 전체 경과시간 T이다.

단순 중단 위치 1: 83.63 (104.01), 67.39 (87.73), 67.64 (87.97).

단순 중단 위치 3: 83.52 (145.20), 26.19 (87.80), 26.19 (87.93).

JSON 잘림 위치 1: 83.55 (103.82), 83.59 (103.90), 83.49 (103.80).

JSON 잘림 위치 3: 83.49 (145.36), 42.06 (103.68), 42.12 (103.95).

내용 변경 위치 1: 83.48 (103.77), 해당 없음, 83.49 (103.81).

내용 변경 위치 3: 83.53 (145.24), 해당 없음, 42.09 (103.76).

내용 변경의 구문 검사군은 각 위치에서 48회 모두 잘못된 전사를 반환하여 성공 조건부 시간은 해당 없음으로 표시하였다. 이 실행들도 전체 결과와 비용 집계에 포함하였다. 이 96회의 전체 경과시간 합은 8417.51초이다.

무장애 48회와 장애 864회를 모두 포함한 실행별 T의 합은 95043.46초(26.40시간)이다. 이는 중단 전 추론과 복구 과정을 포함한 실행 구간의 누적 경과시간이다. 원자료별 SHA-256, 조건별 집계, 녹음별 집계 및 분석 코드를 별도 재현 자료에 보관하였다.

E1에서는 공통 계측을 위해 모든 정책이 검증 기록을 작성한다. 검증 기록 작성과 검증 기능 전체의 추가 비용을 분리하려면, 동일한 저장 절차에 검증 기능만 달리한 무장애 비교 실험이 필요하다.

Appendix F. Transcription and Summarization Service Interface

Fig. 7은 강의 음성을 이용한 PoC 서비스의 개발 화면이다. 사용자는 왼쪽에서 시간 순서에 따른 전사문을 확인하고, 오른쪽에서 요약을 열람한다. 저장된 전사문이 후속 요약의 입력으로 사용되는 응용 흐름을 보여준다.

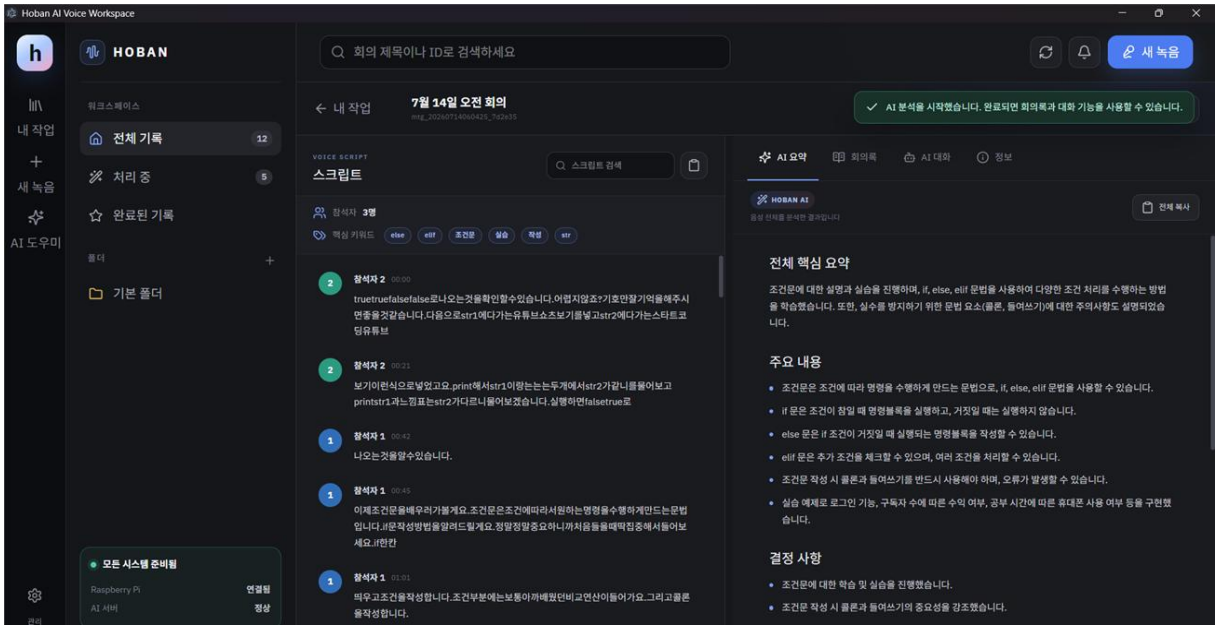


Fig. 7. Development interface of the PoC transcription and summarization service using lecture audio.
자료 출처: 저자 제공 서비스 화면(2026년 7월 개발 자료).

www.kci.go.kr

Authors



Jaehwan Kim is a doctoral researcher in the Department of AI Convergence at aSSIST (Seoul School of Integrated Sciences and Technologies), Seoul, Republic of Korea. His work in this study covers the overall system design, pipeline implementation, empirical measurements, and manuscript preparation. His research interests include on-premises meeting intelligence and edge AI.



Jaeseong Yun is an AI Engineer in the Research and Development Department at XAIKOREA Inc., Seongnam, Republic of Korea. His work in this study covers speech-AI processing, vLLM serving, and concurrency for meeting summarization.



Jongseok Lee is the Infrastructure Team Leader at RAD CNS Co., Ltd., Seoul, Republic of Korea. His work in this study covers the two-node GPU infrastructure and service orchestration.



Kyungmin Kim is a Principal Researcher in the Information Security Team at Security Platform, Seoul, Republic of Korea. His work in this study covers the security architecture, AES-256-GCM protection, and Vault KEK integration.

www.kci.go.kr