

生成 AI 活用のための学内クラウドストレージを介したファイル自動処理マイクロサービス

白井 伸宙^{*1)} [<https://orcid.org/0000-0001-7641-6190>], 池田 真樹²⁾ [<https://orcid.org/0009-0004-9875-7446>], 田ノ上 飛翔¹⁾ [<https://orcid.org/0009-0009-6784-4324>]

1) 三重大学情報基盤センター, 2) 三重大学財務部

* 責任著者: shirai@cc.mie-u.ac.jp

Automated File Processing Microservice via On-Campus Cloud Storage for Utilizing Generative AI

Nobu C. Shirai¹⁾, Masaki Ikeda²⁾, Tsubasa Tanoue¹⁾

1) Center for Information Technologies and Networks, Mie University, 2) Finance Department, Mie University

概要: 高等教育機関の多様なデータは適切な活用で教育・研究・業務の質の向上に有用である一方、機密情報を含む場合があるため学外クラウド処理時には各ファイルの機密性評価が必要である。本研究では、学内データの整理や検索、機密性の判断の効率化を目的として、セキュアな学内クラウドストレージを介してマルチメディアファイルを自動でテキスト化し、ローカルな生成 AI と連携して活用できるファイル自動処理マイクロサービスのシステムを設計・実装し、試験運用を行った。本システムでは、ユーザが対象ファイルを含むフォルダをサービス提供用のアカウントに共有するだけで処理が始まり、出力ファイルが同じ共有フォルダに保存される仕組みにより、ユーザ側の学習コストを抑えつつ、既存の学内クラウドストレージのフロントエンドを活用した管理側の運用負担軽減を実現している。実証実験の結果、機密データを扱う学内環境においても心理的ハードルなく活用できる安全性と、会議録作成等の日常業務における顕著な効率化効果が確認され、本システムが学内でのセキュアかつ実用的な生成 AI 活用基盤として教職員の業務支援に有効であることが明らかになった。

キーワード: 学内クラウドストレージ, ローカル生成 AI, 音声認識, 光学文字認識, 定型プロンプト処理

ランニングタイトル: 生成 AI 活用のための学内ファイル自動処理マイクロサービス

1 はじめに

高等教育機関では、授業・研究会の映像、会議の音声記録、講義メモや黒板写真など、多様な形式のデータが日々生成・蓄積されている。これらのデータは教育・研究・管理業務における貴重な資源であり、適切に活用することで学習効果の向上、研究品質の改善、業務効率化などの多くの利点をもたらす可能性を秘めている。しかしながら、これらのデータには学生の個人情報や未公開研究内容など機密性の高い情報が含まれる場合があり、外部クラウドサービスでの処理においては情報漏洩のリスクが懸念される。特に近年普及している生成 AI 技術を活用する際には、データの機密性を個別に評価し、適切な

処理環境を選択する必要があるが、厳格なセキュリティポリシーに従うと十分にクラウドツールを活かすことは難しく、こうした理由から導入を控えている教職員は多い。近年、セキュリティ上の懸念を解決するため、オープンソースソフトウェアを用いてローカルな環境で生成 AI 活用に関する研究・運用がなされている [1, 2] が、このようなアプローチでは通常チャットインターフェースが提供され、新規ユーザがプロンプト作成技術を習得する必要があるため、日常的な業務利用までの障壁となっている。

本研究では、先行研究の技術的知見を基礎としつつ、組織全体での生成 AI 活用を促進するため、学内のセキュアなクラウドストレージ [3] を基盤とした自動ファイル

処理マイクロサービスのシステムを構築・運用し、その有効性を検証する。このシステムは、各種メディアファイルのテキスト化によりローカルおよびパブリッククラウド生成 AI のプロンプト作成を支援すると同時に、音声認識や OCR 機能自体も独立した用途で活用できるよう設計されている。また、既存の学内クラウドストレージをフロントエンドとして活用することで維持コストを最小化し、他機関での同様のシステム [4, 5] への展開時の導入障壁を低減している点も特徴である。

本システムの特徴は、ユーザ側の操作が単純なフォルダ共有のみであり学習コストが低いこと、既存の学内クラウドストレージインフラをそのまま活用できるため管理負担が軽減されること、そして、学内サービスとして提供されることで、外部サービス利用時の機密性に関する懸念が解消され、教職員がより積極的かつ安心して日常業務に活用できる点である。また、定型プロンプト処理の仕組みにより、プロンプトエンジニアリングの専門知識がなくても様々な用途に合わせた AI 活用が可能となり、ユーザは実際の利用過程で生成 AI の機能や性能の理解を深められるため、組織全体での生成 AI 技術の普及を促進し、生成 AI 技術の日常的な業務・教育活動への浸透を加速させると考えられる。

さらに、本研究は生成 AI を活用した開発アプローチの有効性も示している。学内の比較的セキュアな環境下での試行サービス提供は生成 AI のアシストを受けながらプログラムをアジャイルに作成する方法と相性がよく、プログラミングの詳細な知識がなくても仕様書が書ける情報センターの教職員が学内向けのサービスを構築するのに極めて効果的である。これにより、従来よりも迅速かつ柔軟なサービス開発が可能となり、ユーザの声を反映したシステム構築を実現できる。

本稿では、システムの全体設計、システムの実装、および実証実験結果を報告する。

2 システムの全体設計

本システムは「ユーザによる簡単な操作」と「バックグラウンドでの高度な自動処理」を両立させることを設計思想としている。図 1 に示すように、学内クラウドストレージを中心としたファイル同期とファイル生成イベント検知による処理フローを構築している。

2.1 ユーザ操作の流れ

システムの利用手順は極めて単純である：

1. 処理対象ファイルを含むフォルダをクラウドストレージを用いてサービス提供用アカウントと共有

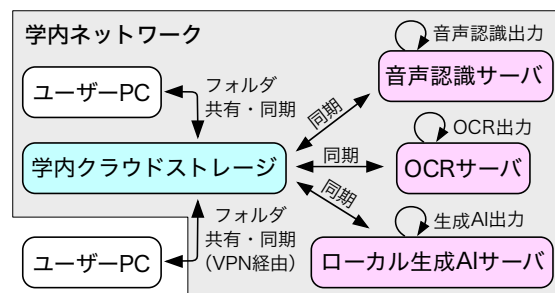


図 1 学内クラウドストレージを介したファイル自動処理マイクロサービスのシステム構成図

する。

2. ファイル処理および同期待ち。
3. 共有フォルダ経由で生成ファイルにアクセス可能となる。

この単純な操作により、技術的知識に関わらず全ての学内ユーザがサービスを活用できる。

2.2 システム処理の流れ

バックグラウンドでは以下の処理が自動的に実行される：

1. **ファイル同期**：クラウドストレージのクライアントによって、共有されたファイルが処理サーバに同期される。
2. **処理スケジューリング**：ファイル生成イベント検知により、共有フォルダ内に新規ファイルが追加または変更された際に自動的に処理を開始する。
3. **メディア処理**：
 - i. 音声・動画ファイル：文字起こし処理を実行し、出力されたテキストファイルを共有フォルダに保存。
 - ii. 画像・PDF ファイル：光学文字認識 (OCR) 処理を実行し、出力されたテキストファイルを共有フォルダに保存。
4. **生成 AI 処理**：
 - i. テキスト化されたファイルおよびユーザが直接共有したテキストファイルに対して、ローカル生成 AI による要約処理を実行。
 - ii. 要約を行うデフォルトプロンプトまたはユーザ定義プロンプト（フォルダ内に配置する設定ファイル）による柔軟な処理を実現。
5. **処理結果の配置**：
 - i. 処理結果は元の共有フォルダに保存され、自動的にユーザと同期される。
 - ii. ログ情報も記録され、処理状況の把握や問

題診断に活用できる。

すべての処理は学内ネットワーク内で完結し、データセキュリティを確保しつつ、VPN 経由でのリモートアクセスも可能である。この設計により、プロンプトエンジニアリングの専門知識がなくても、様々な用途に合わせた生成 AI 出力を簡単に得ることができる。

3 システムの実装

本章では、第 2 章で述べたシステムの全体設計に基づき、学内クラウドストレージを基盤としたファイル自動処理マイクロサービスについて、三重大学情報基盤センターが提供している「三重大学自動テキスト化サービス」で使用しているシステムの実装を例に説明する。

3.1 学内クラウドストレージ

本システムは、三重大学情報基盤センターが提供する「三重大学 ownCloud サービス」[3] を学内クラウドストレージの基盤として採用している。このサービスは、学内ネットワークからの Web アクセスに加え、VPN を利用することで学外からも安全にアクセスでき、セキュアなファイル共有を実現している。ユーザは ownCloud のデスクトップクライアントを利用することで、ローカル環境とクラウド間でファイルを自動的に同期できる。これにより、ユーザは特別な操作を意識することなく、日常的にファイルを管理できる。また、学内クラウドストレージを活用することで、処理サーバ側での独自バックアップが不要となり、データ管理をクラウドストレージに集約できるため、運用コストの削減にもつながっている。

処理サーバ側でも通常の ownCloud クライアントを使用している。ownCloud クライアントは共有されたフォルダ名の競合を自動で解決するため、運営側は共有するフォルダ名に規則を設ける必要はなく、ユーザは任意のフォルダ名を使用して共有することができる。これにより、ユーザは業務内容や目的に応じた直感的なフォルダ名を自由に設定でき、管理者側も複雑な命名規則の策定・周知・監視といった運用の手間を省くことができる。

本研究は機密性を考慮して学内クラウドストレージを基盤としたが、API を直接利用しない設計により、ファイル処理や結果ファイルの配置の仕組みは ownCloud 以外の外部クラウドストレージサービスにも同様に適用できる。本研究のプロトタイプ開発時には Dropbox を用いて同様の処理も検証している。

3.2 フォルダの共有設定と組織アカウント

フォルダを共有する先のアカウントは一般ユーザと区別しやすい名前とアイコンを持つ組織アカウント¹⁾を作成し、ユーザはこのアカウントに対してフォルダを共有することで処理を開始できる。図 2 にユーザが ownCloud の Web インタフェースからフォルダを共有する画面を示す。



図 2 三重大学 ownCloud サービスの Web インタフェースからのフォルダ共有画面

三重大学自動テキスト化サービスでは、現時点では一つの組織アカウントのみを共有先としているが、サービスごとに複数の組織アカウントを設定することで、負荷分散やサーバ側ストレージ容量の軽減も実現可能である。また、組織アカウントの運用方法を工夫することで、特定部局向けの専用アカウントを設けて優先的な処理を行うなど、学部や研究科ごとに最適化したサービス提供も可能である。本方式の利点は、ユーザがどのサービスを利用する場合でも同じ窓口から操作でき、サービスごとに共有するアカウントを選択するだけでよいことである。これにより、サービスごとに異なる Web ページやアカウントを準備する手間が不要となり、学内クラウドストレージの既存インタフェースをそのまま活用できる。

3.3 処理サーバの構成

本システムでは、システムの有効性検証とサービス提供のために 2 台の処理サーバを運用した。まず、プロトタイプの開発・検証を目的とした試用用サーバを 2024 年 11 月 11 日から 2025 年 3 月 25 日まで運用し、システムの基本機能と処理性能の評価を実施した。その後、2025 年 3 月 26 日からは本格的なサービス提供を目的としたサービス提供用サーバに移行し、現在に至る。

試行用サーバの構成は以下の通りである。

- OS: Ubuntu Desktop 24.04 LTS (Noble Numbat)
- CPU: Intel(R) Core(TM) i9-14900K (24 コア / 32 スレッド, 最大 5.09GHz)
- メインメモリ (RAM): 64GB (実効 62.6GB)
- GPU: NVIDIA GeForce RTX 4060 Ti (VRAM 16GB)

サービス提供用サーバの構成は以下の通りである。

- OS: Ubuntu Desktop 24.04 LTS (Noble Numbat)
- CPU: Intel(R) Core(TM) i7-14700F (20 コア / 28 スレッド, 最大 4.84GHz)
- メインメモリ (RAM): 32GB (実効 31.2GB)
- GPU: NVIDIA GeForce RTX 4070 (VRAM 12GB)

三重大学自動テキスト化サービスでは、試行期間中は試行用サーバで、2025 年 3 月 26 日以降はサービス提供用サーバでマイクロサービスの処理を行っている²⁾。音声文字起こしの Whisper [6] (3.4 節) を GPU 実行する場合、PyTorch の CUDA 対応を考慮すると NVIDIA 社設計の GPU を使用が推奨され、さらに Whisper の高精度モデル (large) を利用する場合に GPU の VRAM が 10GB 以上である必要があるため、両サーバとも上記の条件を満たす GPU を選定した³⁾。

3.4 自動テキスト化・生成 AI 処理

本システムの中核となる自動テキスト化・生成 AI 処理機能は、マルチメディアファイルからのテキスト抽出と、抽出されたテキストの生成 AI による分析・要約を組み合わせることで、学内データの安全かつ効率的な活用を実現している。

3.4.1 ファイル命名規則と進捗管理

ユーザビリティ向上のため、処理状況をリアルタイムで把握できる仕組みを実装している。例えば、音声ファイル audio.mp3 が処理された場合、以下の命名規則で各種ファイルが生成される。

- audio_mp3_info.txt: 処理待ちキュー状況ファイル (未処理ファイルがある場合のみ生成)
- audio_mp3_dump.txt: 処理進捗状況ファイル (リアルタイム更新)
- audio_mp3.txt: 文字起こし結果の本体
- audio_mp3_summary.txt: 要約結果 (ローカル生

成 AI 処理; 文字起こし結果から生成される)

処理待ちキュー状況ファイル (.info.txt) は、システム全体の処理待ちキュー情報を提供する。フォルダ内に未処理ファイルが存在する場合、現在の処理待ちファイル数を表示する。処理進捗状況ファイル (.dump.txt) は処理開始時に即時生成され、推定処理時間や開始タイムスタンプを記録した後、処理の進行状況をリアルタイムで追記する。これにより、ユーザは処理がどの段階まで進んでいるかを随時確認できる。どちらもユーザからの要望に基づいて実装されたもので、処理完了までの待ち時間の目安を知りたいという声に応えている。

これらの進捗管理機能はすべて元のファイルと同じ共有フォルダ内に自動生成されるため、ユーザは追加のインターフェースなしに処理状況を確認できる。この設計は「すべての操作を学内クラウドストレージ内で完結させる」というシステムの基本思想に沿ったものであり、ユーザ体験の一貫性を高めている。

3.4.2 メディアファイルのテキスト化処理

本システムの自動テキスト化処理は、本研究のために開発した音声・動画・画像・文書ファイルを対象とした Python パッケージ textify [11] により実現されている。textify は、音声・動画ファイルに対しては Whisper [6] を、PDF や画像ファイルに対しては EasyOCR [8] を用いてテキスト抽出を行うラッパープログラムであり、コマンドラインでの引数・オプション設定を用いて、ファイル単位またはディレクトリ単位での一括処理が可能である。音声認識に Whisper を用いた学内サービスとしては京都大学の「kwhisper」[9] などの先行事例があるが、本システムでは既存の学内クラウドストレージとの連携による自動処理と、生成 AI との統合処理を特徴としている。

出力は全て元のファイル名に対応した名前のテキストファイル (3.4.1 節参照) として保存される。なお、本研究では生成 AI 活用の理念を体現するように生成 AI によって生成されたコードを活用しており、textify もその一つである。textify は GitHub 上で公開されており [11]、他の教育機関でも容易に導入・運用できるようになっている。

3.4.3 Whisper の処理時間計測

音声・動画ファイルの文字起こしにおいて、Whisper の実行時間はファイルの長さに応じて変化する。本システムでは、ユーザビリティ向上のため、以下に示す処理時間の計測結果を基に、ファイル長から処理時間を推定する機能を実装している。

図 3(a) に示すように、試行用サーバおよびサービス提

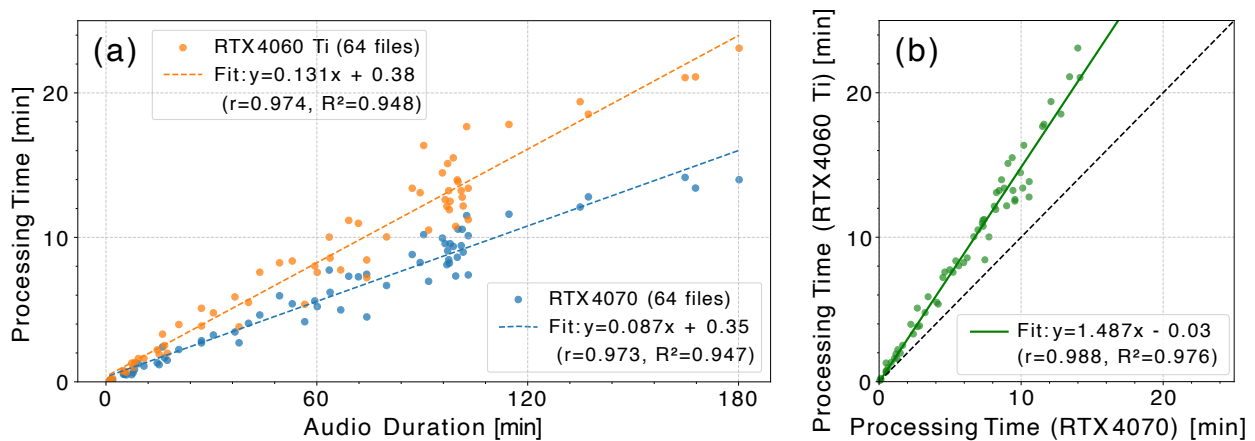


図3 Whisper の処理時間計測結果. (a) 音声ファイルの長さ (分) と文字起こし処理時間 (分) の関係. 試用用サーバ (RTX4060Ti) をオレンジ, サービス提供用サーバ (RTX4070) を青で表示. (b) 同一音声ファイルでの両サーバ間の処理時間比較. 横軸: サービス提供用サーバ, 縦軸: 試用用サーバの処理時間 (分).

供用サーバでの音声ファイル処理について, 散布図と回帰直線を示す. 試用用サーバでは 64 件の音声ファイルについて回帰直線 ($y = 0.131x + 0.38$) を得て, ピアソンの相関係数 $r = 0.974$, 決定係数 $R^2 = 0.948$ となった. サービス提供用サーバでは回帰直線 ($y = 0.087x + 0.35$) を得て, ピアソンの相関係数 $r = 0.973$, 決定係数 $R^2 = 0.947$ となった.

図 3(b) では, 同一音声ファイルを両サーバで処理した際の処理時間比較を示している. 同一ファイルでの処理では, 両サーバが元の回帰直線から同じ方向にずれる傾向が観察され, 散布図の点を線形回帰すると $y = 1.487x - 0.03$ ($r = 0.988$, $R^2 = 0.976$) という強い相関が得られた⁴⁾. この結果は, 音声ファイル固有の特性が処理時間に与える影響が, システム間で一貫していることを示している. また, 両サーバ間の処理時間差は主に GPU 性能差に起因していると考えられる.

この知見は実用的な運用において重要な示唆を与える. コンピュータの性能や OS, プログラム処理系のバージョンにより回帰直線の傾きは変化するものの, 一度様々な長さの音声ファイルで回帰直線を推定すれば, その後は実用に足る精度での処理時間推定が継続的に可能であることが確認された. この結果から, **ffprobe** を用いて音声ファイルの長さを取得することで, 両サーバとも実用に足る精度で処理時間の推定が実施できることが確認された.

ただし, 本研究で計測した処理時間は, 最も時間がかかる Whisper 実行から結果ファイル生成までの時間のみを対象としており, ownCloud クライアントによるファイル同期時間やネットワーク速度依存性は含まれていない. 実際のユーザ体験における総処理時間は, ファイルのアップロードから最終的な結果ファイルの同期完了ま

でを含むため, ネットワーク環境 (学内ネットワーク接続または VPN 接続) やファイルサイズによって追加の時間が発生する可能性がある. この要因を含めた総合的な処理時間モデルの構築は, より正確なユーザ体験予測のための今後の課題である.

3.4.4 テキスト化自動処理システムの実装

systemd サービスとして **inotifywait**⁵⁾ によるディレクトリ監視スクリプトを常時稼働させ, 新規ファイルの追加をリアルタイムで検知して自動テキスト化処理を実行している. 加えて, cron による定期実行スクリプトを導入し, 未処理ファイルの有無を一定間隔でチェックすることで, **inotifywait** のイベント取りこぼし時にも確実な処理を担保している. 多重起動を防ぐため, **flock** コマンドによる排他制御を導入している. この冗長化構成により, 学内クラウドストレージと連携したファイル自動処理サービスの高い信頼性と運用効率を両立している. 処理実行では, GPU 実行・CPU 実行の両方に対応し, GPU が占有されている場合に処理を停止する機能も備えている.

3.4.5 生成 AI によるテキスト処理と要約

メディアファイルをテキスト化したものを含むテキストファイルを対象とした生成 AI による自動処理機能を実装している. この機能はローカル環境で完結する生成 AI 推論基盤を構築することで, 学内データのセキュリティを確保しつつ, テキストの要約や分析などの用途で利用できる処理を提供している. ユーザは特別な知識なく定型処理の恩恵を受けられるよう設計されており, 処理結果は元ファイルと同じディレクトリに保存される.

この機能は, Ollama [12] をバックエンドとして利用し, Gemma 3 [13] (後に Gemma 3n [14] へ更新) をローカルで実行することで, テキストファイルの内容を

自動的に要約する。

自動要約処理のワークフローは以下の通りである：

1. cron ジョブが 10 分おきに共有フォルダをスキャンし、未処理のテキストファイルを検出
2. 検出されたテキストファイルに対して、Gemma 3n を用いた要約処理を実行
3. 命名規則に従い、生成 AI 処理結果のテキストファイルを共有フォルダに保存

当初は音声文字起こしや OCR 処理によって生成されたテキストファイルのみを処理対象としていたが、ユーザからの要望を受け、テキストファイルを直接共有した場合も処理対象とするよう機能を拡張した。これにより、会議録などの既存テキストデータに対しても、学内のセキュアな環境でローカル生成 AI を用いた要約処理が可能となった。

3.4.6 プロンプトカスタマイズ機能と活用例

デフォルトのプロンプトとして「以下を日本語で簡潔に要約してください：」を使用しているが、ユーザが自身の目的に合わせてプロンプトをカスタマイズできる機能も実装している。具体的には、以下の 2 種類のカスタムプロンプト設定方法を提供している：

- prompt.txt**：共有フォルダ内に配置することで、そのフォルダ内の全テキストファイルに対して指定のプロンプトを適用
- prompt_filename.txt**：特定のファイル（**filename.txt**）に対してのみカスタムプロンプトを適用（**prompt.txt** より優先される）

この自動要約機能により、長文のテキストデータの内容を素早く把握することが可能となり、特に会議録や報告書などの要約作成に活用されている。表 3 に示すように、会議録の要約作成を中心に複数の用途で利用されており、テストユーザからは業務効率化に貢献しているとの評価を得ている。

この自動要約機能も、本システムの基本思想に沿って、すべての処理が学内クラウドストレージ内で完結し、追加の操作や専門的な知識を必要としない設計となっている。プロンプトエンジニアリングの知識がなくても、定型の要約処理が自動的に実行される点が特徴であり、組織全体での生成 AI 活用の敷居を下げることに貢献している。

3.5 生成 AI とアジャイル開発の有効性

本研究で構築したシステムの開発プロセス自体が、生成 AI を活用した新たなソフトウェア開発アプローチの

具体的な成功事例となっている。学内の比較的セキュアな環境下での試行サービス提供は、生成 AI のアシストを受けながらプログラムをアジャイルに作成する方法と相性がよく、プログラミングの詳細な知識がなくても仕様書が書ける情報センター教職員であれば、実用的な学内向けサービスを迅速に構築できることを示している。

具体的には、本システムの開発では ChatGPT（使用モデル：o1-preview, o1, o4-mini-high, o3）や GitHub Copilot Chat（使用モデル：Claude Sonnet 3.7 Thinking, Claude Sonnet 4）などの生成 AI ツールを活用し、初期プログラムの作成からモジュール化、機能拡張までを実施した。実現したい機能をプロンプトとして記述し、生成 AI にプログラムを書かせる手法により、専門的なプログラミングスキルがなくても実用的なシステム開発が可能となった。

この開発アプローチの大きな利点は、従来型の委託開発と異なり、テストユーザからのフィードバックに基づく反復的な仕様改善が可能である点にある。生成 AI が実装の詳細を担うことで開発者はシステムの目的や機能の改善に集中できるため、当初想定していなかった問題や潜在的なニーズを開発過程で取り込むことができ、サービスの完成度と利便性を大幅に向上させることができた。機能拡張の開発効率も飛躍的に向上し、例えば自動要約機能などは短期間で実装・改良できた。

セキュリティ面については、学内限定サービスとして提供することで、機密データを扱う際の懸念を軽減しつつ、最新の生成 AI 技術を活用できる環境を整備できた。特に、セキュリティ面で最も注意を要するフロントエンドを既存の学内クラウドストレージサービスに任せることによって、生成 AI の補助を得た内製プログラムの開発・実行のハードルを大幅に下げることができた。この経験は、他の学内向けサービス開発にも応用可能であり、情報センターの限られたリソースでより多くの価値あるサービスを提供できる可能性を示している。

4 実証実験

本システムの有効性を検証するため、学内の教職員からテストユーザを募り、実証実験を実施した。テストユーザには自身の業務において実際にシステムを利用してもらい、その後アンケート調査によって評価を得た。

4.1 実証実験の概要

実証実験は以下の要領で実施した。

- 対象者**：教職員 47 名
- 回答者数**：15 名（回答率 31.9%）

●**利用期間**: 2024 年 11 月から 2025 年 7 月までの間に各ユーザが利用を開始

●**アンケート回答期間**: 2025 年 7 月 2 日～7 月 4 日

対象者 47 名は ownCloud 共有履歴から 1 回以上サービスを利用したことが確認されたユーザ数である。テストユーザは 2024 年 11 月から順次サービスの利用を開始し、最も早いユーザは約 8 ヶ月、最も遅いユーザは数日の利用期間を経てアンケートに回答した。特に 2024 年 12 月と 2025 年 1 月に多くのユーザが利用を開始している。実験期間中の 2025 年 3 月 26 日には、3.3 節で述べた通り、試行用サーバからサービス提供用サーバへの移行を実施した。また実証実験期間中はユーザからのフィードバックを随時取り入れ、継続的に機能改善・追加を行った。

4.1.1 機能拡張のプロセス

本研究ではアジャイル開発手法を採用し、短期間の開発サイクルを繰り返しながら以下のように機能を段階的に拡張した。

基本機能の確立（2024 年 11 月～12 月）クラウドストレージ連携と自動処理の基本機能を開発・検証

処理効率の向上（2025 年 1 月～3 月）GPU 使用状況の監視機能追加、多重実行防止の実装により安定性と効率性を向上

ユーザ体験の改善（2025 年 4 月～5 月） ● 処理状況の可視化: `dump.txt` によるリアルタイム進捗表示

- 処理待ち状況の通知: `_info.txt` による処理キュー情報の提供
- 自動要約機能: Gemma 3/3n によるテキスト要約機能の追加⁶⁾

システム監視の強化（2025 年 6 月）`inotifywait` によるファイル変更のリアルタイム検知、アウトプット形式の最適化（不要ファイル形式の削除）

4.2 サービスの利用状況

テストユーザの回答に基づくサービスの主な利用目的を表 1-3 に示す。最も多い利用目的は会議録の作成であり、音声文字起こし機能が会議関連業務の効率化に特に活用されていることがわかる。また、自動要約機能も会議録の要約作成に多く利用されている。OCR 機能については、紙媒体資料のデジタル化や手書きメモのデジタル化に活用されている。

表 1 音声文字起こし機能の主な利用目的（複数回答可）

利用目的	回答数
会議録の作成	11
打合せメモの作成	6
講演・セミナーの記録	3
インタビュー記録の作成	1
業務報告書の下書き作成	1

表 2 OCR 機能の主な利用目的（複数回答可）

利用目的	回答数
紙媒体資料のデジタル化	2
手書きメモのデジタル化	1

表 3 自動要約機能の主な利用目的（複数回答可）

利用目的	回答数
会議録の要約作成	6
報告書の要約作成	1
長文資料の概要把握	1

4.3 サービスの評価結果

4.3.1 全体満足度と継続意向

サービス全体に対する満足度では、「非常に満足」が 8 名 (53.3%), 「やや満足」が 4 名 (26.7%), 「どちらとも言えない」が 1 名 (6.7%), 「非常に不満」が 2 名 (13.3%) という結果であった。「非常に満足」と「やや満足」を合わせると 80% のユーザが満足していることがわかる。

また、今後のサービス利用継続意向については、15 名中 14 名 (93.3%) が「ぜひ継続したい」と回答し、1 名 (6.7%) が「条件付きで継続したい」と回答した。条件として「精度を見極めながら、NotebookLM も併用している」という回答が得られた。これらの結果から、アンケートに回答したテストユーザの大多数がサービスに満足し、継続利用を希望していることが確認された。

4.3.2 各評価項目の結果

各評価項目について 5 段階評価（1: 全く当てはまらない～5: 非常に当てはまる）で回答を求めた結果を表 4 に示す。「外部クラウドサービスと比べて心理的ハードルが低い」および「セキュリティ/機密性の面で安心して使える」の評価が最も高く（平均 4.47）、システムの設計思想である「学内データの安全な処理」という目標が達成できていることを示している。一方、「出力結果の精度に満足している」および「生成 AI による要約が業務に役立つ」の評価が相対的に低く（平均 3.8）、これらは今後の改善点として考慮すべき項目である。

表4 サービスの評価項目と結果 (n=15, 5段階評価)

評価項目	平均 (標準偏差)	中央値
外部クラウドサービスと比べて心理的ハードルが低い	4.47 (0.74)	5.0
操作が直感的で学習コストが低い	4.33 (0.90)	5.0
処理速度に満足している	4.27 (0.88)	4.0
出力結果 (文字起こし・OCR) の精度に満足している	3.80 (0.68)	4.0
セキュリティ／機密性の面で安心して使える	4.47 (0.74)	5.0
既存の業務フローに組み込みやすい	4.00 (0.93)	4.0
生成 AI による要約が業務に役立つ	3.80 (1.01)	4.0

4.4 ユーザからのフィードバック

4.4.1 サービスの良かった点

自由記述によるフィードバックから、サービスの良かった点として以下のようなコメントが得られた：

- 「使用に際して心理的ハードルが小さく、手続きが容易で、精度もどんどん上がってきていて、とても便利で大変助かっています。」
- 「長い録音時間のデータについて文字起こしするには時間がかかるが、並行して他の業務を行えるため全く気にならない。データを格納するだけで動くため誰にでも非常に扱いやすいと感じた。」
- 「現在担当している委員会では、2時間程度の議事録について正確な文字起こしが求められます。文字を打ち込む時間が大幅に短縮されるため、当該業務に要する時間を4分の1程度まで削減できたのではないかと考えています。」
- 「既存の ownCloud をインターフェースにしていたので、非常に直感的でよかったです。」
- 「機密性のあるものを取り扱えるのが、とてもありがたい。」

これらのコメントから、システムの設計思想である「ユーザによる簡単な操作」と「既存のクラウドストレージを活用した使いやすさ」、そして「学内でのセキュアな処理」という目標が達成できていることが確認された。また、議事録作成など実際の業務において大幅な時間短縮効果があったことも報告されている。

4.4.2 改善点と追加機能の要望

改善点や追加機能についての要望として、以下のようないコメントが得られた：

- 「文字起こしが完了した際にメール通知等があると

嬉しいかもです。」

- 「発言要旨の冒頭に『話者』を付した形式の議事メモにしたい場合のプロンプト例など、適切なプロンプトと出力結果のサンプルなどを、ツールの手引きとともに配布することで、心理的ハードルを下げ、使用への誘因ともなり、ツールが一層効果的に活用されることに繋がると考えられる。」
- 「難しいとは思いますが、『ええ』『あの』などのフィラーを自動除去できればいいな、と思います。」
- 「pyannote.audioなどを組み込んだ話者分離機能があると、より議事録作成がスムーズになると考えています。また、summaryの作成等急に機能が追加された気がしますので、ユーザ登録等により定期的なアナウンスをいただくと、新たなサービスがすぐに利用でき、ありがたいと思っています。」

これらの要望は、今後のシステム改善における重要な指針となる。特に話者分離機能や通知機能、フィラーの自動除去などは、議事録作成の効率をさらに向上させる可能性がある機能である。また、プロンプトのサンプル提供や機能追加の通知など、ユーザ支援の充実に関する要望も見られた。

4.5 考察

実証実験の結果から、本システムが当初の設計目標を概ね達成していることが確認された。特に、学内クラウドストレージを基盤としたセキュアな環境での処理と、シンプルな操作性が高く評価されている。また、会議録作成を中心とした実際の業務における時間短縮効果も確認された。

一方で、文字起こしの精度や生成 AI による要約機能については、他の項目と比較して評価が相対的に低く、今後の改善が望まれる。特に、話者分離機能やフィラー除去など、議事録作成をより効率化するための機能追加が有効と考えられる。

回答者の93.3%が「ぜひ継続したい」と回答していることから、本システムが学内での生成 AI 活用基盤として有効であることが確認された。また、自由記述からは、本システムが「教職員がより積極的かつ安心して日常業務に活用できる」という当初の目標を達成していることが示唆された。

今回の実証実験の結果は、組織内でのセキュアな生成 AI 活用のためのアプローチとして、本研究が提案する「学内クラウドストレージを介したファイル自動処理マイクロサービス」の有効性を示すものである。また、ユー

ザからの具体的な改善要望は、今後のシステム拡張における重要な指針となる。

5 むすび

本研究では、高等教育機関における生成 AI 技術の安全かつ効果的な活用を目的として、学内クラウドストレージを基盤としたファイル自動処理マイクロサービスを設計・実装し、その有効性を検証した。

実証実験の結果から、本システムは「セキュリティ／機密性の面で安心して使える」「外部クラウドサービスと比べて心理的ハードルが低い」といった点で高い評価を得ており、組織内でのセキュアな生成 AI 活用基盤として有効であることが確認された。特に、学内データの機密性を確保しつつ生成 AI 技術を活用できる点は、教育・研究機関特有の要件に対応したソリューションとして意義がある。

本システムの重要な貢献は、プロンプトエンジニアリングの専門知識がなくても様々な用途に合わせた AI 活用を可能にした点にある。定型プロンプト処理の仕組みにより、生成 AI 技術に不慣れなユーザでも実用的な成果を得られるようになり、組織全体での技術普及を促進する効果が期待できる。

また、本研究では生成 AI を活用した開発アプローチ自体の有効性も実証された。生成 AI のアシストを受けながらプログラムをアジャイルに作成する方法により、専門的なプログラミングスキルがなくても実用的なシステム開発が可能となり、教育機関における内製サービス開発の新たな可能性を示すことができた。

今後は、ユーザからの要望を反映した機能拡張を継続するとともに、本研究の知見を他の教育機関と共有し、高等教育分野における生成 AI 活用の好事例として展開していくことを目指す。本研究が提案するアプローチが、教育・研究機関における生成 AI 技術の適切な導入と活用の一助となることを期待する。

謝 辞

本研究は多くの方々の協力により実現している。筆者らが所属する三重大学生成 AI 活用検証イニシアティブ (GAUTI) の共同メンバーである平山亮氏 (三重大学医学部附属病院 医学・病院管理部) と江川昂明氏 (三重大学図書・情報部) には、サービスの活用促進、テストユーザのサポートに協力いただいた。システムの技術基盤となる「三重大学 ownCloud サービス」について、その構築・運用者である松原伸樹氏 (三重大学 情報基盤センター) から多くの技術的助言をいただいた。システムの評価と改善においては、テストユーザとして参加いただいた坂元

舞氏 (三重大学 情報基盤センター) から使い勝手に関する貴重なフィードバックを得ることができた。また、学内の複数部局から集まったテストユーザにも参加いただき、多様な視点からの評価と提案を受けることができた。協力いただいた全ての方々に心から感謝の意を表する。

注

- 1) 本システムでは `textify_gauti` という組織アカウントを使用している。`textify` は本研究で開発したプログラム名かつサービス名のア称であり、`gauti` は三重大学生成 AI 活用検証イニシアティブ (GAUTI) の略称である。アイコンは GAUTI の「ゴ」をモチーフにしたマスコット「ゴチ」で、文献 [10] にて CC BY 4.0 で公開している。
- 2) 現在のサービス提供用サーバでは ownCloud クライアントが稼働しており、三重大学 ownCloud サービスを提供している ownCloud サーバは文献 [3] に記載がある別サーバで運用されている。試用用サーバは研究目的での評価・検証に既存のものを使用した。その後、本格的なサービス提供のためのサーバを別途購入した。
- 3) 2025 年 3 月にサービス提供用サーバの購入の際、条件を満たす GPU が品薄であったため、GPU 搭載の構築済みパソコンを購入した。
- 4) この高い相関は、音声ファイルごとの処理時間のばらつきが、システム固有のランダムな要素よりも音声ファイル自体の特性に起因することを示している。これは音声ファイルの内容、品質、形式などの追加要素を考慮することで、さらに精度の高い処理時間推定が可能である可能性を示唆している。
- 5) `inotifywait` は、`inotify-tools` パッケージを導入することで利用できるコマンドラインツールで、ディレクトリやファイルの変更をリアルタイムに監視する用途に適している。
- 6) 本サービスでは 2025 年 4 月 19 日に Gemma 3[13] を用いて自動要約機能を実装し、その後 2025 年 6 月 28 日に Gemma 3n へと更新した。Gemma 3n は Gemma 3 の軽量化モデルでありながら、同等の性能を発揮する。

参考文献

- [1] J. Radas, B. Risse, R. Vogl: Building UniGPT: A Customizable On-Premise LLM-Solution for Universities, EPIc Series in Computing, Vol. 105, pp. 108–116, 2025.
- [2] A. J. O'Connell: How (and Why) the University of Michigan Built Its Own Closed Generative AI Tools, EDUCAUSE Review, 2024. <https://er.educause.edu/articles/2024/2/how->

and-why-the-university-of-michigan-built-its-own-closed-generative-ai-tools (2025 年 7 月 7 日参照)

- [3] 松原 伸樹: 三重大学 ownCloud サービスの運用・構築, 技術職員による技術報告集, Vol. 25, pp. 20 – 23, 2017.
- [4] 松岡 孝, 田島 尚徳, 出口 大輔, 森 健策: ownCloud を用いた全教職員向けファイル共有サービスの構築, 大学 ICT 推進協議会年次大会論文集 2014 年度年次大会, 2014.
- [5] 葛川 寛之, 葛西 真寿, 須藤 勝弘: ownCloud による学内クラウドサービス「弘大クラウド」の構築と運用, 第 18 回学術情報処理研究会発表論文集, 2014.
- [6] OpenAI: Whisper. <https://github.com/openai/whisper> (2025 年 7 月 7 日参照)
- [7] FFmpeg Developers: FFmpeg. <https://ffmpeg.org/> (2025 年 7 月 7 日参照)
- [8] Jaided AI: EasyOCR. <https://github.com/JaidedAI/EasyOCR> (2025 年 7 月 7 日参照)
- [9] 京都大学情報環境機構: 「文字起こし試行サービス kwhisper の紹介」, 京都大学情報環境機構広報誌「Info!」No. 28, p. 9, 2023.
<https://www.iimc.kyoto-u.ac.jp/sites/default/files/2025-02/info28.pdf> (2025 年 7 月 7 日参照)
- [10] Nobu C. Shirai: GAUTI のロゴとマスコットキャラクター「ゴチ」の画像公開用リポジトリ <https://github.com/nobucshirai/GAUTI-logo> (2025 年 7 月 7 日参照)
- [11] Nobu C. Shirai: textify: A Microservice for Automated File Processing via On-Campus Cloud Storage. <https://github.com/nobucshirai/textify> (2025 年 7 月 7 日参照)
- [12] Ollama: Ollama. <https://github.com/ollama/ollama> (2025 年 7 月 7 日参照)
- [13] Gemma Team, Google DeepMind: Gemma 3 Technical Report, 2025. <https://storage.googleapis.com/deepmind-media/gemma/Gemma3Report.pdf> (2025 年 7 月 7 日参照)
- [14] O. Sanseviero, I. Ballantyne: Introducing Gemma 3n: The developer guide, 2025. <https://developers.googleblog.com/en/introducing-gemma-3n-developer-guide/> (2025 年 7 月 7 日参照)