# Technologien

## Backend-Entwicklung und API-Integration

* **Flask**:
  + **Begründung**: Flask ist ein leichtgewichtiges Web-Framework, das Entwicklern Flexibilität bietet und nicht mit unnötigen Komponenten belastet.
  + **Einsatz**: Als zentrale Webanwendung, die die Hauptlogik, Endpunkte und die Kommunikation mit anderen Diensten steuert.
* **Requests**:
  + **Begründung**: Einfaches und effizientes Python-Modul für HTTP-Anfragen.
  + **Einsatz**: Integration mit der OpenAI API und eventuellen Drittanbieter-Diensten für zusätzliche Funktionalitäten.

## Datenbank

* **MSSQL**:
  + **Begründung**: Enterprise-fähige relationale Datenbank, bekannt für Performance und Sicherheit.
  + **Einsatz**: Speicherung von Benutzerprofilen, Transkriptdaten und generierten Keynotes.
* **pyodbc**:
  + **Begründung**: Etablierte Bibliothek zur Verbindung von Python mit SQL-basierten Datenbanken.
  + **Einsatz**: Abfrage, Einfügung und Aktualisierung von Daten in der MSSQL-Datenbank.

## Transkription und Audioverarbeitung

* **SpeechRecognition**:
  + **Begründung**: Umfassendes Toolkit für Sprach-zu-Text-Konvertierungen.
  + **Einsatz**: Konvertieren von Audioaufnahmen von Meetings in lesbare Transkripte.
* **Pydub**:
  + **Begründung**: Bietet eine benutzerfreundliche Schnittstelle für die Bearbeitung von Audiodateien.
  + **Einsatz**: Formatkonvertierung und mögliche Audiooptimierung vor der Transkription.

## Sprecheridentifikation

* **Kommerzielle Lösungen und APIs** (z.B. Google Cloud, IBM Watson):
  + **Begründung**: Spezialisierte Dienste, die in der Sprechererkennung trainiert sind.
  + **Einsatz**: Unterscheidung und Kennzeichnung von Sprechern innerhalb von Meeting-Transkripten.

## GUI (Benutzeroberfläche)

* **KivyMD**:
  + **Begründung**: Bietet eine reaktive Benutzeroberfläche mit modernem Design für verschiedene Plattformen.
  + **Einsatz**: Hauptbenutzeroberfläche für Benutzer, um Audio hochzuladen, Transkripte zu überprüfen und Keynotes zu generieren.

## Integration mit Microsoft Teams

* **Microsoft Graph Python SDK**:
  + **Begründung**: Offizielles SDK, das den Zugriff auf Microsoft-Dienste vereinfacht.
  + **Einsatz**: Integration in Microsoft Teams für direkte Transkription und Keynote-Erstellung innerhalb der Plattform.

## Textverarbeitung, Keynote-Erstellung und Protokollierung

* **spaCy**:
  + **Begründung**: State-of-the-Art NLP-Tool mit hervorragender Performance und Genauigkeit.
  + **Einsatz**: Extraktion von Schlüsselinformationen aus Transkripten, Identifikation von Hauptpunkten und Strukturierung von Text für Keynotes.
* **Gensim**:
  + **Begründung**: Spezialisiert auf Textzusammenfassung und -modellierung.
  + **Einsatz**: Ergänzung von spaCy bei Bedarf für spezifische Textzusammenfassungsaufgaben.
* **OpenAI API**:
  + **Begründung**: Bietet fortschrittliche Textgenerierungs- und Analysefunktionen.
  + **Einsatz**: Ergänzen und Verfeinern von Keynotes und Protokollen, um sie kohärent und prägnant zu gestalten.

## Authentifizierung

* **OAuthLib** und **Flask-OAuthlib**:
  + **Begründung**: Standardisierte und sichere Authentifizierungsmethoden.
  + **Einsatz**: Benutzerauthentifizierung, um sicherzustellen, dass nur berechtigte Benutzer auf den Dienst zugreifen können.

## Datenverwaltung und Speicherung

* **Boto3** (optional):
  + **Begründung**: Standardbibliothek für den Zugriff auf AWS-Dienste.
  + **Einsatz**: Speichern und Abrufen von Audio-, Transkript- und Keynote-Dateien aus AWS S3, wenn Cloud-Speicherung gewünscht ist.
* **Python os** und **shutil** Module:
  + **Begründung**: Kernbibliotheken für das Dateimanagement in Python.
  + **Einsatz**: Verwaltung von Dateien, insbesondere für temporäre Speicherung und Datenzugriff.

## Fehlerprotokollierung und Monitoring

* **Sentry**:
  + **Begründung**: Professionelles Tool für Fehlerüberwachung mit detaillierten Berichten und Benachrichtigungen.
  + **Einsatz**: Echtzeit-Überwachung der Anwendung, Identifizierung und Behebung von Fehlern, um die Servicequalität sicherzustellen.

## Tests

* **Pytest**:
  + **Begründung**: Flexibles und mächtiges Testing-Framework für Python.
  + **Einsatz**: Automatisierung von Tests, um die Funktionalität und Zuverlässigkeit des Systems während der Entwicklung und nach Updates sicherzustellen.