Definition of done

Whisper : Mawen

Définition

Whisper est un **modèle open-source de reconnaissance vocale automatique (Speech-to-Text, STT)** développé par OpenAI. Contrairement aux solutions propriétaires comme **Google Speech-to-Text**, **Azure Cognitive Services** ou **Amazon Transcribe**, Whisper repose sur une approche **end-to-end** utilisant des réseaux de neurones de type **Transformer**. Ce modèle a été entraîné sur un corpus massif et multilingue d'audio-text aligné, ce qui lui confère une **robustesse élevée** face aux accents, aux environnements bruyants et aux contextes spécialisés (ex. médical).

Concept clés

- **STT (Speech-to-Text)** : Conversion de la parole en texte via modèles neuronaux.
- **Multilinguisme** : Whisper couvre plus de 90 langues, avec des capacités de traduction intégrées.
- **Robustesse au bruit** : entraînement sur données réelles incluant bruit, accents et variations.
- **Taille de modèle** : différentes variantes (Base, Small, Medium, Large) permettant un compromis entre **précision, latence et coût**.
- **Open Source** : à la différence des API cloud fermées, Whisper peut être déployé **on-premise**, garantissant **RGPD-compliance** et contrôle des coûts.

Pourquoi Whisper et non une autre solution ?

- **Contrôle des données** : contrairement aux services cloud, Whisper peut tourner **en local** → indispensable en contexte médical (confidentialité, RGPD).
- **Open Source & gratuit** : pas de coûts variables à l'usage, contrairement aux facturations à la minute des API SaaS.
- **Robustesse multilingue** : adapté aux environnements hospitaliers multilingues ou multiculturels.
- **Flexibilité de déploiement** : sur serveur local, cloud privé, ou intégré dans un pipeline existant.

Modèles Adaptés au Médical

- **Base (74M paramètres)** → compromis performance / coût, utile pour prototypage.

- **Small (244M paramètres)** → meilleure précision, adapté à une utilisation régulière.
- **Medium (769M paramètres)** → très haute précision, adapté aux environnements cliniques exigeants.
- **Large (1550M paramètres)** → qualité quasi-professionnelle, utile pour **comptes-rendus médicaux** ou applications critiques.

Enjeux Techniques Majeurs

- **Latence & performance** : gestion en temps réel (asynchrone, batching GPU).
- **Conformité & sécurité** : respect du **RGPD**, traitement local des données sensibles.
- **Scalabilité** : capacité à gérer plusieurs flux audio simultanés dans un hôpital.
- **Qualité de transcription** : réduire les erreurs sur le **lexique médical** (potentiellement via fine-tuning ou correction post-traitement).
- **Monitoring & maintenance** : suivi continu des performances, intégration dans un pipeline médical fiable.

Temps de Développement & Complexité

L'implémentation d'une solution STT robuste basée sur Whisper suit une trajectoire progressive :

1. **Phase 1 – Recherche et Choix (1-2 semaines)**
    a. Benchmark entre solutions (Google, Azure, AWS, Whisper)
    b. Tests comparatifs : précision, latence, coût d'inférence
    c. Analyse des contraintes : conformité **RGPD**, coûts d'exploitation, complexité d'intégration
2. **Phase 2 – Implémentation de Base (2-3 semaines)**
    a. Installation de Whisper et configuration des environnements GPU/CPU
    b. Développement d'un **wrapper API** (WhisperSTT) avec gestion d'erreurs
    c. Écriture de **tests unitaires** pour valider la transcription sur divers cas
3. **Phase 3 – Optimisation & Production (1-2 semaines)**
    a. Optimisations de performance : **asynchrone, batching, cache de modèles**
    b. Renforcement de la robustesse : gestion des fichiers corrompus, formats audio variés
    c. Monitoring et observabilité : **logs, métriques de précision, latence**
4. **Phase 4 – Intégration Complète (1 semaine)**
    a. Intégration au pipeline médical : **Audio → Texte → Analyse IA**
    b. Tests end-to-end avec l'application finale
    c. Documentation (API, guides d'utilisation, best practices)

⏱️ **Total estimé : 5-8 semaines** pour une implémentation robuste.

Maquette Figma - UX UI - Design : Thibaud (4 jours)

J'ai travaillé sur l'ensemble de l'UX et de l'UI de l'application. Pour cela, j'ai réalisé une maquette Figma, retravaillée à de nombreuses reprises afin de trouver le bon équilibre entre une application complète et une expérience simple et accessible.

Pourquoi la simplicité est-elle primordiale ? Parce que l'application a pour vocation d'offrir un premier diagnostic santé à l'utilisateur : rassurer, donner de bons conseils, et l'orienter vers le service le plus adapté, qu'il s'agisse d'un simple problème bénin ou d'une situation plus sérieuse.

L'accès à la fonctionnalité principale – échanger avec l'IA – devait être immédiat. Dès la connexion, l'utilisateur doit voir et comprendre instantanément comment lancer une conversation. L'application a été pensée pour être intuitive et facile à utiliser, même pour les personnes peu à l'aise avec la technologie. Nous partons du principe que notre public cible est très large, allant des adolescents jusqu'aux personnes âgées.

Un autre point essentiel était l'intégration de services externes comme Google Maps ou Doctolib, afin de faciliter la prise en charge après le premier échange. Pour cela, nous avons ajouté un dashboard clair et facilement accessible, regroupant toutes les fonctionnalités principales en une ou deux actions seulement. Ce tableau de bord permet notamment :

- de consulter l'historique des conversations,
- de suivre son état de santé,
- d'accéder rapidement au bon service de santé.

Enfin, une attention particulière a été portée au design. L'objectif était de proposer une interface rassurante, épurée et moderne, qui puisse convenir à tous les âges. Nous avons privilégié une esthétique proche de celle des applications de santé classiques (comme Doctolib), afin de refléter sérieux et crédibilité. Le choix des couleurs et du dégradé de fond sur la page d'accueil vise à transmettre un sentiment de confiance, tout en rappelant que l'application est un outil d'accompagnement et ne remplace pas les services de santé officiels.

Tests et Qualité du Code – Backend – Frontend – CI/CD : Thibaud (2 jours)

Un point essentiel du projet a été la mise en place d'une stratégie complète de tests et de contrôle qualité. L'objectif était de garantir un code propre, fiable et maintenable, tout en facilitant le travail collaboratif.

Côté backend, nous avons utilisé différents outils : Pytest pour lancer les tests et vérifier la couverture du code, Black pour uniformiser le style, Flake8 pour détecter les erreurs de logique et MyPy pour le typage statique. Ces outils permettent de corriger très tôt les incohérences et d'imposer une rigueur dans le développement.

Côté frontend, nous avons adopté une approche similaire avec ESLint pour détecter les erreurs, Prettier pour assurer un formatage homogène et le système de vérification TypeScript pour sécuriser les types. Là encore, l'objectif était d'éviter les erreurs courantes et de garder un projet clair et robuste.

Un autre élément clé a été la mise en place d'un pipeline CI/CD complet avec GitHub Actions. Chaque fois qu'un développeur pousse du code, plusieurs vérifications automatiques sont lancées :

- le respect du style de code,
- l'exécution des tests et le suivi de la couverture,
- une analyse de sécurité pour prévenir les failles,
- la validation des commits selon le standard *Conventional Commits* afin de garder un historique lisible.

Nous avons également ajouté des hooks pre-commit sur les postes de développement. Ces vérifications locales permettent d'éviter d'envoyer du code incorrect ou non formaté dès le départ. Cela concerne aussi bien le formatage automatique que la détection de secrets sensibles ou la validation des messages de commit.

Enfin, cette architecture nous apporte plusieurs bénéfices concrets :

- un code plus fiable et sécurisé,
- une homogénéité de style sur tout le projet,
- une meilleure traçabilité grâce à la gestion stricte des versions et des commits,
- une réduction des erreurs humaines grâce à l'automatisation.

Cette mise en place témoigne de notre volonté d'avoir une base technique solide et professionnelle, comparable aux standards utilisés dans des environnements de production réels.