Audit du dˆ'pˆ·t NoXoZVorteX_prompted

1) Inventaire des fichiers (par domaine/fonction)

Backend / c¯ ur applicatif (Python)
- ‘analyse_conversations_merged.py‘
- ‘config.py‘
- ‘extractors.py‘
- ‘prompt_executor.py‘
- ‘result_formatter.py‘
- ‘utils.py‘
- ‘install.py‘
- ‘help.py‘

Frontend
- Aucun fichier front-end (pas de HTML/CSS/JS).

Data / exemples
- ‘data_example/example_claude_json.json‘
- ‘data_example/example_chatgpt_json.json‘
- ‘data_example/example_lechat_json.json‘

Prompts / templates
- ‘prompts/prompt_example_child_safety_prompt.txt‘
- ‘prompts/prompt_example_security_prompt.txt‘
- ‘prompts/prompt_example_full_security_analysis_prompt.txt‘

Scripts (CLI / outils)
- Tous les scripts sont en Python (voir section Backend).
- Aucun script ‘.sh‘.

Documentation
- ‘README.md‘
- ‘TRY_THIS_FIRST.txt‘
- ‘help_advanced.txt‘
- ‘LICENSE‘

Configuration
- ‘config.py‘
- ‘.gitignore‘

Infra / CI / dˆ'ploiement
- Aucun fichier dˆ'diˆ' (pas de Dockerfile, CI, etc.).

Tests
- Aucun dossier/fichier de tests.

2) Fichiers clˆ's par domaine et usage

Backend / c¯ ur applicatif
- ‘analyse_conversations_merged.py‘ : point dâ entrˆ'e CLI. Charge les exports JSON, dˆ'tecte les doublons, dˆ'coupe les conversations, exˆ'cute les prompts via lâ API Mistral (ou simulation), puis sauvegarde les rˆ'sultats en CSV/JSON/TXT/Markdown.
- ‘prompt_executor.py‘ : gestion des prompts (chargement, formatage), exˆ'cution des requˆ“tes API Mistral avec retry, traitement dâ une conversation pour produire un rˆ'sultat.

- ‘extractors.py‘ : dˆ'tection du format JSON (ChatGPT/LeChat/Claude) et extraction des messages.
- ‘result_formatter.py‘ : sˆ'rialisation des rˆ'sultats dans plusieurs formats (CSV/JSON/TXT/Markdown).
- ‘utils.py‘ : logging, nettoyage du texte, comptage approximatif des tokens, gˆ'nˆ'ration de noms de sorties.
- ‘config.py‘ : constantes (version, modˆ¤le, limites, dˆ'pendances, etc.) et rˆ'cupˆ'ration de la clˆ' API.
- ‘install.py‘ : crˆ'ation du venv, vˆ'rification/installation des dˆ'pendances et nettoyage de fichiers.
- ‘help.py‘ : aide CLI, aide avancˆ'e et changelog.

Data / exemples
- ‘data_example/*.json‘ : fichiers dâ exemples pour ChatGPT, LeChat et Claude.

Prompts / templates
- ‘prompts/*.txt‘ : exemples de prompts utilisˆ's par le moteur.

Documentation
- ‘TRY_THIS_FIRST.txt‘ : commande dâ exemple rapide.
- ‘help_advanced.txt‘ : aide CLI dˆ'taillˆ'e (long format).
- ‘README.md‘ : description minimale du dˆ'pˆ·t.
- ‘LICENSE‘ : licence du projet.

Configuration
- ‘config.py‘ : paramˆ¤tres du runtime (API, modˆ¤les, limites, etc.).
- ‘.gitignore‘ : exclusions Git.

3) Structure globale du projet

Points dâ entrˆ'e
- ‘analyse_conversations_merged.py‘ est le point dâ entrˆ'e principal (CLI). Il orchestre lâ ensemble du flux :
  1. Lecture des arguments et validation.
  2. Chargement des JSON et dˆ'tection de format.
  3. Dˆ'duplication et dˆ'coupage des conversations.
  4. Exˆ'cution du prompt (API Mistral ou mode simulation).
  5. Sauvegarde des rˆ'sultats (CSV/JSON/TXT/Markdown).

Modules principaux
- **Extraction & normalisation** : ‘extractors.py‘.
- **Exˆ'cution de prompts** : ‘prompt_executor.py‘.
- **Formatage des rˆ'sultats** : ‘result_formatter.py‘.
- **Utilitaires** : ‘utils.py‘.
- **Configuration** : ‘config.py‘.

Flux de donnˆ'es (rˆ'sumˆ')
- Entrˆ'e : exports JSON (ChatGPT/LeChat/Claude) -> ‘analyse_conversations_merged.py‘ -> ‘extractors.py‘.
- Transformation : assemblage du texte + dˆ'coupage par tokens -> ‘prompt_executor.py‘.
- Sortie : rˆ'sultats via ‘result_formatter.py‘.

Dˆ'pendances majeures (Python)
- ‘requests‘ (appels API Mistral), ‘tqdm‘ (progression), ‘tiktoken‘ (comptage tokens), ‘argparse‘, ‘concurrent.futures‘.

4) API
- Aucune API interne exposˆ'e (pas de FastAPI/Flask/Express). Le projet consomme lâ API Mistral via HTTP.

5) Scripts ‘.sh‘
- Aucun.

6) Scripts ‘.py‘ (usage + imports)

‘analyse_conversations_merged.py‘
- Usage : CLI principal (analyse, extraction, exˆ'cution de prompts).
- Imports notables : ‘argparse‘, ‘json‘, ‘csv‘, ‘glob‘, ‘hashlib‘, ‘concurrent.futures‘, modules locaux (‘conf
ig‘, ‘utils‘, ‘extractors‘, ‘install‘, ‘help‘, ‘prompt_executor‘, ‘result_formatter‘).

‘prompt_executor.py‘
- Usage : chargement/formatage des prompts, exˆ'cution API Mistral, traitement conversation.
- Imports notables : ‘requests‘, ‘random‘, ‘time‘, ‘pathlib‘, ‘typing‘.

‘extractors.py‘
- Usage : dˆ'tecte le format JSON (ChatGPT/LeChat/Claude) et extrait les messages.
- Imports notables : ‘typing‘, ‘utils.ecrire_log‘.

‘result_formatter.py‘
- Usage : ˆ'crit les rˆ'sultats dans plusieurs formats (CSV/JSON/TXT/Markdown).
- Imports notables : ‘csv‘, ‘json‘, ‘datetime‘, ‘pathlib‘.

‘utils.py‘
- Usage : logging, nettoyage texte, tokens, gˆ'nˆ'ration noms de sortie.
- Imports notables : ‘re‘, ‘datetime‘, ‘tiktoken‘ (optionnel).

‘config.py‘
- Usage : constantes globales + rˆ'cupˆ'ration clˆ' API.
- Imports notables : ‘os‘.

‘install.py‘
- Usage : vˆ'rification prˆ'requis, crˆ'ation venv, installation dˆ'pendances.
- Imports notables : ‘subprocess‘, ‘venv‘, ‘shutil‘.

‘help.py‘
- Usage : affichage aide standard/avancˆ'e.
- Imports notables : ‘pathlib‘, ‘config‘.

7) Duplications, conflits potentiels, obsolescence
- **Versions incohˆ'rentes** : ‘analyse_conversations_merged.py‘ annonce v3.0.1 alors que ‘config.py‘ est en v2
.7.0 (risque de confusion sur les logs et paramˆ¤tres).
- **Redondance ‘ensure_directory‘** : fonction similaire dans ‘analyse_conversations_merged.py‘ et ‘prompt_exe
cutor.py‘.
- **Dˆ'pendances divergentes** : ‘install.py‘ dˆ'clare des dˆ'pendances de fallback (‘mistletoe‘, ‘anthropic‘, ‘p
ython-dotenv‘) non listˆ'es dans ‘config.py‘.
- **Docs minimales** : ‘README.md‘ ne dˆ'crit pas lâ usage, ce qui peut ˆ“tre perˆ§u comme obsolˆ¤te/incomplet.

8) Tableau synthˆ'tique des fichiers

Chemin	Usage
‘analyse_conversations_merged.py‘	CLI principal dâ analyse et dâ exˆ'cution de prompts.
‘config.py‘	Configuration globale (versions, API, limites).
‘extractors.py‘	Dˆ'tection de format JSON + extraction de messages.
‘prompt_executor.py‘	Chargement/formatage de prompts et exˆ'cution API.
‘result_formatter.py‘	Export des rˆ'sultats (CSV/JSON/TXT/Markdown).
‘utils.py‘	Utilitaires (log, nettoyage, tokens, noms).
‘install.py‘	Installation dˆ'pendances + venv.

‘help.py‘	Aide CLI standard/avancˆ'e.
‘data_example/*.json‘	Exemples de donnˆ'es (ChatGPT/LeChat/Claude).
‘prompts/*.txt‘	Exemples de prompts.
‘README.md‘	Description minimale.
‘TRY_THIS_FIRST.txt‘	Exemple de commande rapide.
‘help_advanced.txt‘	Aide CLI dˆ'taillˆ'e.
‘.gitignore‘	Exclusions Git.
‘LICENSE‘	Licence.

9) Tableau des endpoints
- Non applicable (aucune API exposˆ'e).