Files

main

Go to file t

aiwhisperer
tests
.gitignore
CLAUDE.md
LICENSE
README.md
article_aiwhisperer.txt
download_models.py
requirements.txt
setup.py
start.py

voelspriet Add CLAUDE.md with project context for future sessions

f84a37d · 5 minutes ago History

aiwhisperer / CLAUDE.md

↑ Top

Preview Code Blame 63 lines (53 loc) · 2.45 KB

Raw

AIWhisperer - Project Context

Project Info

- **Name:** AIWhisperer (renamed from DocSanitizer)
- **GitHub:** https://github.com/voelspriet/aiwhisperer
- **Author:** Henk van Ess
- **License:** CC0 (Public Domain)
- **Current version:** 0.3.0

What It Does

Tool to analyze confidential documents with cloud AI while reducing risk of exposing sensitive data.

Pipeline: PDF → convert → split → encode (sanitize) → upload to AI → download → decode → real names restored

Two main selling points:

1. Files too big to upload to cloud AI (170MB+ PDFs)
2. Local AI too slow, but uploading unredacted confidential data is risky

Style Preferences (Henk van Ess / Digital Digging)

- No emoji in code or docs
- No cocky/absolute claims ("without exposing" → "with reduced risk")
- Honest about limitations - tool reduces risk, doesn't eliminate it
- Short sentences, direct, practical
- No marketing fluff

Key Language Choices

- "Whisper your documents to AI—with reduced risk of exposing sensitive data"
- "This reduces—but does not eliminate—the risk"
- "helps minimize the risk of leaks"
- NOT: "The AI never sees the real data" (too absolute)

Package Structure

```
aiwhisperer/
├── __init__.py      # Main exports: encode, decode, Mapping
├── cli.py           # CLI commands: convert, encode, decode, check, analyze
├── converter.py     # PDF to text (marker-pdf or tesseract)
├── encoder.py       # Sanitization logic
├── decoder.py       # Restore real names
├── mapper.py        # Mapping file handling
├── detectors/       # NER and pattern detection
│   ├── hybrid.py    # spaCy + patterns (default)
│   ├── patterns.py  # Regex patterns only
│   └── ...
└── strategies.py    # Anonymization strategies (replace, redact, mask, hash)
```

CLI Commands

```
aiwhisperer convert document.pdf --split --max-pages 500
aiwhisperer encode document.txt --legend
aiwhisperer decode ai_output.txt -m mapping.json
aiwhisperer check    # Show installed dependencies
aiwhisperer analyze  # Preview what would be detected
```

Related Files

- `article_aiwhisperer.txt` - Draft article for Digital Digging blog
- `test_output/` - Test files from 4713-page PDF conversion

Original Story

Based on real 170MB cocaine investigation case file (4,713 pages). Article: https://www.digitaldigging.org/p/speed-reading-a-massive-criminal