App Name: Cloak & Style

# Desktop Utility (Windows)

Cloak & Style is a data-scrubbing tool for content professionals. It redacts names, emails, phone numbers, and custom keywords from structured content files (CSV, XLSX, PDF)—with clarity, elegance, and control. Think: editorial quality meets data privacy.

## Use Cases:

* Privacy-safe draft sharing
* Prepping training datasets or samples
* Scrubbing internal decks before executive review
* Cleaning legacy docs before knowledge base import

## Value to the Organization:

* Saves legal review cycles
* Promotes internal AI use without PII risk
* Empowers content, design, and UX teams to move faster with compliance in mind
* Aligns with AI-readiness and content infrastructure modernization

**Purpose:** A user-friendly data scrubber tool that uses Python to identify, mask, or redact personally identifiable information (PII) and brand-sensitive data from structured data files (e.g. CSV, Excel, and PDF). The tool supports data privacy compliance, simplifies audit preparation, and streamlines content review workflows.

# 1) Product Overview

**Purpose:** a self‑contained Windows desktop tool that scans and masks PII across common business file types (DOCX, PPTX, PDF, XLSX, CSV, TXT). It emphasizes accuracy over speed, optional human review for edge cases, and safe-by-default operations that never persist PII mappings beyond a single run.

**Primary users:** privacy/compliance analysts, program managers, engineers, and vendors working with sensitive artifacts prior to external sharing.

**Key principles**:

* **Accurate by default** (deterministic patterns + NLP NER).
* **No PII persistence** (mappings live in-memory per run only).
* **Reversible not required** (destructive/transformative masking).
* **Safe outputs** (explicit out-dir, versioned file naming).
* **Transparent** (rich summary report & audit log).

A screenshot of a computer

AI-generated content may be incorrect.

# 2) Requirements

PII Data Scrubber Tool (Python-Based)

* Tool must:
  + Parse tabular data from .csv and .xlsx
  + Extract **structured tables and readable text blocks** from .pdf files
  + Preview uploaded content in the UI
* Scrubbing modes:
  + **Masking:** Partial reveal (e.g., A\*\*\*)
  + **Redacting:** Replace with [REDACTED]
  + **Removal:** Delete from text/cells
* User options:
  + Select columns or regions to scrub
  + Upload brand keyword list
  + Choose scrubbing method per field
  + Configure case-sensitivity and pattern matching preferences
* **Usability Requirements**
* Drag-and-drop file upload
* Tooltips for scrubbing options
* Inline preview for spreadsheets and PDFs
* Flag un-processable pages in PDFs

## 2.1 Supported platforms

* Windows 10/11, 64‑bit.
* Self‑contained EXE packaged by PyInstaller + install.bat.
* Runs without admin rights (per‑user install).
* Memory profile: 8–16 GB RAM (default tuning for 8 GB).

## 2.2 File types & surfaces

* **DOCX**: body, tables, headers/footers, footnotes/endnotes, comments, tracked changes (insertions & deletions), text in shapes, alt text.
* **PPTX**: body text in shapes, tables, charts’ data labels/titles (where textual), speaker notes, slide master placeholders, alt text.
* **PDF (text-based)**: page text and metadata (no OCR at v1).
  + **Image‑only PDFs**: fail fast with a per-file error; add warning to the run report.
* **XLSX**: cell values, comments/notes; mask literals embedded inside formulas (leave functions intact where feasible).
* **CSV/TXT**: full-text scan (configurable structured vs free‑text handling).
* **Embedded/linked objects**: attempt textual extraction when feasible; otherwise add a warning to the report.

## 2.3 Languages

* v1: English only.
* Architecture prepared for FR/ES/DE models and regex/pattern packs.

## 2.4 PII scope (entities)

* Names (person, organization where sensitive), postal addresses, emails, phone numbers, IPs, credit cards, bank accounts/ABA/IBAN (if present), national IDs (SSN, SIN, etc.), dates of birth, usernames/handles, geolocation coordinates if identifiable.
* Extensible via pluggable recognizers.

## 2.5 Masking policy

The tool must explicitly support three scrubbing modes: **Mask**, **Redact**, **Remove**. **Partial-reveal Mask** is available only for free-text entities (PERSON, BRAND) only; **structured identifiers** (EMAIL, CREDIT\_CARD, PHONE, IP, SSN/NationalID) are **never partially revealed** and must be fully masked or redacted. Per-entity defaults are configurable; TOKEN remains the default mask format.

Partial-reveal masking is allowed for PERSON/BRAND only; structured identifiers are always fully masked or redacted.

* **Consistency:** same entity string → same replacement within **the active run scope** (single file or batch).
* **No cross-run persistence:** mapping dictionary is in‑memory only; destroyed on exit.
* **No partial retention:** do **not** retain domains, last‑4, area codes, etc.
* **Replacement format:** tokenized placeholders for traceability, e.g.
  + [PERSON\_001], [EMAIL\_003], [CARD\_002], [ADDR\_004]  
    (IDs are per-run counters; entity type visible for review/reporting.)
* For fixed‑width contexts (e.g., table cell alignment), allow “same‑length masking” (\*\*\*\*\*\*\*\*) as an optional format, but default to tokens.

## 2.6 Accuracy & “questionables”

* Pipeline = deterministic rules (regex/validators) **+** NLP NER (Presidio + DeBERTa‑based model plus PII Masker implementation).
* Ambiguous detections flagged as **“Questionable”** with confidence scores and shown in:
  + Optional **Review Queue** (GUI) for Accept/Reject.
  + Summary report (JSON + human‑readable HTML).

### Backends:

* **PII Masker (DeBERTa)** as the primary NER.
* **Deterministic rules** (regex/validators) for structured entities (email, phone, CC/Luhn, IP, SSN, etc.).
* **NLTK** for tokenization/normalization (no spaCy).

## 2.7 Performance guards (defaults)

* **XLSX/CSV structured data:** up to **100,000 rows** per file (cap hard‑stops with warning).
* **PDF:** up to **100 pages** **or** **10 MB**, whichever is hit first.
* Exceeding caps → skip file, add error to report (non‑zero CLI exit).
* Overrides: **backlog** item—allow advanced users to raise limits (and declare available RAM).
* **Backlog:** allow overrides / “declare RAM”.

## 2.8 Batch & headless

* GUI supports folder selection with recursive processing.
* CLI supports folder recursion, file globs, JSON/CSV reports, and non‑zero exit codes on policy violations (e.g., image‑only PDFs, caps exceeded).

## 2.9 Review & Dry‑Run

Review Queue (optional): Accept/Reject findings; filters and bulk actions.

Dry‑run: generate side‑by‑side diffs and findings without touching originals; allow exporting masked artifacts from the queue.

## 2.10 Output policy

* **Required** explicit out‑dir.
* Mirror input tree under out‑dir.
* If re‑processing to the same out‑dir: **version** files (file (2).ext, file (3).ext, …).

## 2.11 Reports & audit

* Per‑run **summary report** (HTML + JSON) capturing:
  + Inputs, timestamp, tool version, config snapshot.
  + File‑level outcomes (masked, skipped, errored).
  + Entity counts by type, confidence histograms.
  + Questionables & user decisions (if review queue enabled).
  + Warnings (e.g., embedded objects not processed, image‑only PDFs).
* Optional CSV “findings” table for downstream review.
* Contents: tool version, config snapshot, file statuses, entity counts, confidence histograms, questionables & decisions, warnings (image‑only PDFs, embedded/unchanged content), and caps violations.
* Reports never include raw PII—only masked tokens and metadata.

# 3) Architecture

## 3.1 Tech stack

* **Language:** Python 3.11
* **Packaging:** PyInstaller (one‑file EXE)
* **GUI:** Qt for Python (PySide6) — modern Windows UI with table and document viewers. Supports drag‑and‑drop (DnD), split views, progress, and a web view for HTML previews
* **PII/NLP:** PII Masker (DeBERTa) + NLTK, and Microsoft Presidio (Analyzer + Anonymizer) with:
  + Regex recognizers (email, phone, SSN, CC with Luhn, IP v4/v6, etc.).
  + ML NER model: “PII Masker” and Presidio.

PII Masker for NER. Deterministic regex/validator rules run alongside it automatically. We’ll keep an internal config knob for dev (e.g., switch to Presidio‑DeBERTa for benchmarking), but **no user choice** in the GUI/CLI.

* **File I/O:**
  + DOCX/PPTX: python-docx, python-pptx, plus lxml for tracked changes/notes/alt text.
  + PDF: PyMuPDF (faster, better layout); no OCR in v1.
  + XLSX: openpyxl (values, notes); formula literal parsing where feasible.
  + CSV/TXT: standard libs. csv/streaming readers.
* **Config & Profiles:** local JSON config; optional named JSON “profiles” (nice‑to‑have).

## 3.2 Detection pipeline

1. **Text extraction layer** (per file type) → normalized text segments with location metadata (range, shape ID, cell addr, page/para runs).
2. **Entity detection**
   * Rule‑first pass (high‑precision regex/validators with black/whitelists).
   * NER pass (ML) to capture names/addresses/etc., with thresholds.
   * Merge & de‑duplicate overlaps; compute confidence; mark **Questionables** under threshold band.
3. **Mapping & masking**
   * Per‑run in‑memory map {raw\_value → token} scoped to file/batch.
   * Apply masking to underlying OOXML runs/cell values/page spans.
4. **Validation**
   * Re‑scan masked output to ensure no residual PII above min confidence.
   * If residuals exist → raise **Warn/Fail** per policy (configurable).

## 3.3 Data model (high level)

* RunContext: id, start/end time, config snapshot, caps.
* FileJob: path\_in, path\_out, type, size, pages/rows, status, errors/warnings.
* Finding: file\_id, entity\_type, raw\_span, normalized\_value, confidence, status (auto‑masked/accepted/rejected/ignored), location metadata.
* Mapping: run‑scoped dict with token counters per entity type.
* RunContext (id, time, tool version, config snapshot).
* FileJob (in/out paths, type, size, pages/rows, status, errors/warnings).
* Finding (file id, entity type, span/locator, confidence, status, location metadata).
* Mapping (per‑run {raw → token} with per‑type counters).

## 3.4 Processing Flow

1. Extract text with location metadata per file type.
2. Detect entities (rules → NER; merge/resolve overlaps; score; mark questionables).
3. Build per‑run mapping and apply mask to underlying structures.
4. Validate (re‑scan outputs for residual PII above min‑confidence).
5. Emit artifacts (files + reports), handle versioning and errors.

## 3.5 Security & privacy

* No network calls; offline only. Clearly message: “No data is sent to the cloud or stored after session ends” in UI and responses from CLI
* No telemetry or analytics.
* No PII persistence: mapping and findings live in memory; reports include **only masked tokens** and detection metadata (not raw values). Reports never include raw PII—only tokens and detection metadata.
* Temp files cleaned on exit; overwrite-on-close for temp content where possible.
* Reports contain only masked tokens and context metadata.

# 4) User Experience (GUI)

**Home**: drag‑and‑drop or browse files/folders (recursive toggle).

**Options**: Review Queue on/off; Dry‑run on/off; caps shown (read‑only); choose out‑dir (required); optional brand keyword list; masking format.

* **Spreadsheets**: column picker with per-column scrubbing method (Mask/Redact/Remove).
* **PDFs**: entity-type toggles; free-draw region selection deferred to backlog.
* **Brand keywords**: case-sensitivity toggle.

**Run**: progress with per‑file status; cancel button (graceful stop).

**Review Queue**: table of findings; filters (entity, confidence, file); diff/preview pane (DOCX/PPTX/XLSX/CSV/TXT textual diffs; PDF rendered as masked HTML). Bulk Accept/Reject. Export masked outputs from dry‑run.

**Results**: links to out‑dir, HTML/JSON report, CSV findings.

**Tooltips**: Add tooltips for scrubbing modes, caps, residual-policy, etc. Add these to all relevant UI controls.

**Progress Display**: Provide progress via progress bars or hourglass function to allow users to understand how tasks are proceeding

## 4.1 Main flow

1. **Select input**: files or folder (with recursion toggle). Via Explorer browse function or via drag-and-drop.
2. **Select out‑dir** *(required)*.
3. **Options**:
   * Enable **Review Queue** (on/off).
   * **Dry‑run** (generate diff preview & reports only).
   * Caps shown (read‑only); note “raise limits” is a future enhancement.
   * Profile (when implemented).
4. When possible based on input file format, allow user to
   * (**XLSX/CSV**) select column with per-column method dropdown (Mask/Redact/Remove) and a “select all PII columns” quick action.
   * **(PDF)** allow user to select **entity-type toggles** (PERSON/EMAIL/etc.)
5. Select a **case-sensitivity** toggle scoped to **brand keywords** (regex flags), not NER, to avoid recall loss.
6. **Run** → progress panel (file count, successes, warnings, errors).

## 4.2 Review Queue (optional for user to select)

* Grid of findings (sortable/filterable by entity type, confidence, file).
* Preview pane:
  + **Textual diff** (pre/post) for DOCX/TXT/CSV/XLSX cells.
  + **PDF preview** with highlight overlays for text spans.
  + **PPTX slide preview** (textual surrogate if rendering is heavy).
* Actions: **Accept**, **Reject**, **Accept all of type in this file**, **Accept all exact matches**.
* Export from queue (for dry‑run) to produce masked artifacts after approval.

## 4.3 Results

* Completion banner with counts and a link to:
  + HTML Summary report
  + JSON Report
  + CSV Findings (optional)
* Per‑file status list with quick links to open in out‑dir.

# 5) CLI

piimasker run   
--in "C:\input" --out "D:\out" --recursive --include "\*.docx;\*.pptx;\*.pdf;\*.xlsx;\*.csv;\*.txt" ^

--dry-run --review-queue off --report json,html,csv   
--exit-on image-only-pdf,caps-exceeded

Key flags:

* --in,
* --out (required out)
* --recursive, --include, --exclude
* --dry-run
* --review-queue on|off (default off in CLI)
* --report json|html|csv (multi)
* --caps pdf-pages=100 pdf-mb=10 rows=100000 *(read-only in v1; overrides are backlog)*
* --concurrency N *(fixed low default; overrides are backlog)*
* Exit codes:
  + 0 success with/without warnings
  + 2 policy violations (image‑only, caps exceeded, residual PII on validate)
  + 3 fatal errors

# 6) Implementation Details (by file type)

## 6.1 DOCX

* Parse all w:p/w:r text; headers/footers via sectPr; footnotes/endnotes via related parts.
* **Tracked changes:** inspect w:ins and w:del runs; scrub both.
* **Comments:** comments.xml; **Alt text:** docPr/@descr on drawing.
* Replace text **in-place** while preserving run formatting; split runs as needed.
* Edge cases: fields (e.g., MERGEFIELD), hyperlinks text vs URL (mask link text and, if URLs contain PII, replace with placeholder and add warning).

## 6.2 PPTX

* Iterate shapes on all slides & master layouts; text in tables; speaker notes via NotesSlidePart.
* Charts: read textual labels/titles when exposed via drawing XML; otherwise warn.
* Alt text via shape properties.

## 6.3 PDF

Detect text per page. For text-based pages, extract reading-order text and write <name>\_masked.html and <name>\_masked.txt with **page markers**. For image-only pages, **flag page numbers** in the report and continue processing other pages. Inline PDF redaction and metadata stripping are scheduled for v1+ backlog due to fidelity and corruption risk.

* Use PyMuPDF to extract text spans with coordinates; map back to page text AND page ID for highlight previews; do not modify PDFs visually— **exporting masked content as HTML (and .txt)** that mirrors the text flow (with page breaks and page IDs). True PDF rewriting risks layout corruption, embedded fonts issues, and text reflow edge cases.
* Detect image‑only PDFs (no extractable text) → error + report.

## 6.4 XLSX

* Iterate worksheets row‑by‑row up to 100k rows.
* Mask values in cells (string and rich text runs).
* Comments/Notes masked; formula literals parsed and masked (leave function intact).
* Beware shared strings table: replace centrally to keep file size stable.

## 6.5 CSV/TXT

* CSV: stream rows; per‑cell scan and mask.
* TXT: full‑text stream with chunked processing to keep memory flat.

# 7) Configuration

config.json (read at start; optionally saved as profile)

json

{

"language": "en",

"entities": ["PERSON","ADDRESS","EMAIL","PHONE","IP","CREDIT\_CARD","BANK\_ACCT","NATIONAL\_ID","DOB","USERNAME","GEO"],

"mask\_format": "TOKEN",

"review\_queue": false,

"dry\_run": false,

"caps": {"pdf\_pages": 100, "pdf\_mb": 10, "rows": 100000},

"reports": ["html","json"],

"validation": {"residual\_action": "warn", "min\_confidence": 0.35, "questionable\_band": [0.35,0.65]},

"backend": "pii-masker"

}

# 8) Error Handling & Messages

* **Image‑only PDF**: “No extractable text detected—consider OCR. File skipped.”
* **Caps exceeded**: “Row/page/size limit exceeded for configured cap; file skipped.”
* **Embedded object not processed**: “Embedded binary content not text‑parsable; review manually.”
* **Write conflict**: handled via versioned naming automatically.

# 9) Packaging & Distribution

* Artifacts:
  + PIIMasker-Setup\install.bat
  + piimasker.exe (PyInstaller one‑file)
  + README.md (install + usage)
  + licenses/ (third‑party notices)
  + models/ (bundled NER weights; sized for 8–16 GB)
* No system prereqs; no registry changes beyond per‑user shortcuts.

# 10) QA & Test Plan

* Unit tests for:
  + PDF extraction
  + PII regex matchers
* Sample PDFs with test PII for validation
* Manual tests: English and UTF-8 PDFs

## 10.1 Test corpus

* **Golden files** for each type (DOCX/PPTX/PDF/XLSX/CSV/TXT) with seeded PII:
  + Names (varied capitalization & punctuation), addresses, CCs (valid/invalid with Luhn), phones (E.164 and local), emails (edge cases), IPs, SSN formats, DOBs.
  + Docs with headers/footers, comments, tracked changes, alt text, speaker notes.
  + PDFs: text‑based and image‑only.
  + XLSX: 100k‑row sheets; formulas with literals; comments.
  + CSV/JSON/XML mixed structured/unstructured.

## 10.2 Acceptance criteria

* 0 residual PII above min confidence in masked outputs (warn/abort policy exercised).
* 100% masking consistency within a run.
* Correct handling of caps (skip + report).
* Review queue Accept/Reject behavior reflected in outputs and logs.
* Output naming/versioning exact.
* PDF runs produce a masked HTML output with zero residual PII above threshold.

## QA Checklist

**A. Platform, Packaging, Privacy**

* **Windows-only, self-contained:** App installs via install.bat and runs a single PyInstaller EXE on a clean Windows 10/11 x64 machine with 8–16 GB RAM (no extra deps, no admin rights).
* **Offline-only:** No network calls during detection or masking.
* **No PII persistence:** No raw PII written to disk; mapping dictionaries exist in memory only and are destroyed on exit.
* **Licenses & README:** Distribution includes README (usage, caps, error messages) and third-party licenses.

**B. File Types & Surfaces (In Scope)**

* **CSV/TXT:** Full-text scan; encoding-safe (UTF-8/UTF-16/BOM).
* **XLSX:** Masks values and comments/notes; handles sharedStrings; preserves formulas (only literal substrings masked).
* **DOCX:** Masks body, headers/footers, footnotes/endnotes, comments, tracked changes (insertions/deletions), shapes text, alt text; preserves visual formatting.
* **PPTX:** Masks shapes/tables text, speaker notes, master placeholders, alt text; best-effort on chart label text; non-text objects reported.
* **PDF (text-based):** Produces \*\_masked.html and \*\_masked.txt with page markers (one pair per input PDF).
* **PDF (image-only pages):** Page numbers with no extractable text are **flagged per-page** in the report; file still processes other pages.
* **Out of scope enforced:** JSON/XML inputs rejected with a clear error.

**C. Scrubbing Modes & Policy**

* **Three modes available:** **Mask**, **Redact** ([REDACTED]), **Remove** (delete).
* **Per-column method (spreadsheets):** User can choose Mask/Redact/Remove per column; “Apply to all detected PII columns” quick action exists.
* **PDF controls:** Entity-type toggles (PERSON/EMAIL/PHONE/etc.) available; free-draw region selection deferred (not present).
* **Partial reveal constraints:**
  + Allowed only for **free-text** entities (PERSON, BRAND keywords) when user selects Mask.
  + **Never** allowed for **structured identifiers** (EMAIL, CREDIT\_CARD, PHONE, IP, SSN/NationalID); these must be fully masked/redacted/removed.
* **Default mask format:** [TYPE\_###] tokens; optional “asterisks same length” mode available.
* **Consistency scope:** Same raw value is masked consistently across the **active run** (single file or batch), and **not** reused across separate runs.

**D. Detection & Quality Controls**

* **Backends:** NER uses **PII Masker (DeBERTa)**; deterministic regex/validators handle structured entities; **NLTK** present; **no spaCy** in dependencies.
* **Fusion logic:** Overlaps resolved deterministically (structured rules take precedence); duplicates merged.
* **Questionables:** Findings within the configured confidence band are marked **Questionable**.
* **Residual validation:** Outputs are re-scanned; residuals above threshold are listed in the report.
  + GUI default: **Warn** (continue).
  + CLI: --exit-on residuals causes non-zero exit.

**E. Review Queue & Dry-Run**

* **Optional Review Queue:** Can be enabled/disabled; defaults **off** in GUI and CLI.
* **Queue functions:** Filter/sort by file, entity, confidence; **Accept/Reject** single or bulk (by entity type / by file / all exact matches).
* **Previews:**
  + Textual side-by-side for TXT/CSV/XLSX/DOCX/PPTX.
  + Embedded web view for **PDF masked HTML**.
* **Dry-run:** Generates findings + diffs **without** touching originals; user can **Export masked outputs** from the queue.

**F. UX: Usability & Controls**

* **Drag-and-drop:** Files/folders can be dropped onto the main window; recursion toggle available.
* **Required out-dir:** User must pick an output directory to start; app mirrors source tree there.
* **Versioning:** Re-processing to same out-dir creates name (2).ext, name (3).ext, etc. (no silent overwrite).
* **Tooltips:** Present for scrubbing modes, caps, residual policy, and entity toggles.
* **Inline previews:**
  + Spreadsheets: grid preview with column selectors.
  + PDFs: preview via masked HTML (page markers visible).
* **Brand keywords:** User can upload a keyword list; **case-sensitivity toggle** applies to brand matching (regex flags), not NER.

**G. Reporting & Logging**

* **Per-run HTML + JSON report** generated; optional CSV findings export.
* **Report contents:** Inputs, start/end time, tool version, config snapshot, file statuses (masked/skipped/errors), entity counts by type, confidence histogram, list of **Questionables** and user decisions, warnings (embedded/bitmap content, image-only PDF pages), and caps violations.
* **No raw PII in reports:** Only masked tokens and metadata.
* **Open from app:** Completion screen links open out-dir and report(s).

**H. Performance, Caps, and Stability**

* **Performance target (typical files):**
  + ≤ 25-page PDFs or ≤ 50k-row spreadsheets finish in **< 30 seconds** on an 8 GB system, with a responsive progress UI.
* **Caps enforced:**
  + CSV/XLSX: **≤ 100,000 rows/file**.
  + PDF: **≤ 100 pages** or **≤ 10 MB** (whichever triggers first).
  + Exceeding caps → file skipped with a clear error and report entry.
* **Memory bound:** Under default caps, peak memory stays within 8–16 GB.
* **CLI exit codes:** 0 success, 2 policy violation (image-only pages only, caps exceeded, residuals if requested), 3 fatal.

**I. CLI Parity**

* **Batch execution:** --in, --out (required), --recursive, --include/--exclude globs.
* **Modes & reporting:** --dry-run, --review-queue on|off, --report html,json[,csv], --exit-on image-only-pdf,caps-exceeded,residuals.
* **Mirrors GUI rules:** Same caps, masking policy, outputs, and versioning.

**J. Errors & Messages (Examples must display)**

* **Image-only PDF pages:** “Pages X, Y, Z contain no extractable text and were skipped.”
* **Caps exceeded:** “Limit exceeded (pages/size/rows); file skipped.”
* **Unsupported type:** “File type not supported: JSON/XML.”
* **Write conflict handled:** “Output already exists; writing name (2).ext.”

**Per-Entity Mode Matrix (spot-checks)**

| **Entity Type** | **Mask (Partial Reveal allowed?)** | **Redact** | **Remove** |
| --- | --- | --- | --- |
| PERSON | **Yes (allowed)** | Yes | Yes |
| BRAND (keywords) | **Yes (allowed)** | Yes | Yes |
| EMAIL | **No (full only)** | Yes | Yes |
| CREDIT\_CARD | **No (full only)** | Yes | Yes |
| PHONE | **No (full only)** | Yes | Yes |
| IP (v4/v6) | **No (full only)** | Yes | Yes |
| SSN/National ID | **No (full only)** | Yes | Yes |
| ADDRESS | Prefer full; partial reveal optional **only** if configured as free-text | Yes | Yes |

**Done when:** matrix is enforced by UI defaults and back-end policy; attempts to partially reveal a structured identifier are blocked and explained.

## 10.3 Performance checks

**Target** <30s per file for typical workloads (≤25 PDF pages or ≤50k rows). For cap-max workloads (100 pages or 100k rows), maintain responsive progress and stay under memory caps.

* Memory below 8 GB peak on default caps.
* Throughput targets are secondary; ensure UI stays responsive and progress is accurate.

# 11. Delivery Plan & Backlog

**Phases**

* **P0:** repo scaffold, CI, PyInstaller baseline, config loader, logging.
* **P1:** detection core (PII Masker + rules + NLTK), TXT/CSV handlers, JSON/HTML report.
* **P2:** DOCX/PPTX/XLSX handlers; tracked changes/comments; report enhancements.
* **P3:** PDF text extraction; masked HTML/TXT output; image‑only detection.
* **P4:** PySide6 GUI with DnD, dry‑run, Review Queue, previews, progress.
* **P5:** CLI, exit codes, caps enforcement, versioned outputs.
* **P6:** Hardening, perf checks (≤8 GB), golden tests, README/user guide.

## Backlog (ranked)

1. Caps overrides.
2. FR/ES/DE language packs.
3. OCR for image‑only PDFs (Tesseract) with language packs.
4. True PDF rebuild from text spans (high‑fidelity) with stripped metadata.
5. Support ‘Free-Draw’ regions in PDF files for redaction
6. Named profiles (config presets).
7. Concurrency tuning; “declare RAM” & optional GPU model variants.

# Epics → Stories (with acceptance criteria)

## EPIC A — Core detection & masking engine

**Goal:** Local, accurate PII detection; per‑run consistent masking; no PII persistence.

* **A1. Entity rules & validators (deterministic)**
  + Build regex+validators for: EMAIL, PHONE (E.164 + common US), CREDIT\_CARD (Luhn), IP(v4/v6), SSN/NationalID (US at v1), DOB date patterns.
  + ✅ Accept: Unit tests show ≥99% precision on structured entities across golden set; Luhn enforced; false‑positive suite for order numbers, GUIDs, etc.
* **A2. NER backend integration (PII Masker + NLTK)**
  + Vendor‑lock to PII Masker (DeBERTa) as the only NER; NLTK for tokenization/normalization.
  + Confidence scores exposed; “questionable band” configurable.
  + ✅ Accept: NER callable returns typed entities with start/end offsets and confidence; model loads offline; no spaCy in requirements.
* **A3. Fusion & overlap resolution**
  + Merge deterministic + NER detections; tie‑break rules (prefer deterministic for structured entities).
  + ✅ Accept: Overlaps resolved deterministically; 100% mapping stability within a run.
* **A4. Masking mapper**
  + In‑memory {raw → [TYPE\_xxx]} with per‑type counters; no disk persistence.
  + Two output modes: TOKEN (default), ASTERISKS\_SAME\_LENGTH.
  + ✅ Accept: Same raw value masked identically within a file/batch run; no cross‑run reuse.
* **A5. Residual validation**
  + Re‑scan outputs; on residual PII above threshold: **warn** (GUI default) and optional **fail** in CLI.
  + ✅ Accept: Toggle honored; residuals listed in summary.

## EPIC B — File handlers

**Goal:** High‑fidelity text extraction + safe write for each format.

* **B1. TXT/CSV streaming**
  + Chunked streams to keep memory flat; per‑cell masking for CSV.
  + ✅ Accept: Handles arbitrarily large TXT and CSV within caps; BOM & UTF‑8/UTF‑16 safe.
* **B2. XLSX (openpyxl)**
  + Values + comments/notes; sharedStrings table write‑back; mask literals inside simple formulas.
  + ✅ Accept: 100k‑row cap enforced; schema preserved; formulas remain valid.
* **B3. DOCX (python‑docx + lxml)**
  + Body, headers/footers, foot/endnotes, comments, tracked changes (w:ins/w:del), shapes text, alt text.
  + ✅ Accept: Formatting preserved; all surfaces covered; hyperlinks’ visible text masked, URLs containing PII replaced with placeholder and warning.
* **B4. PPTX (python‑pptx)**
  + Slide & master shapes, tables, speaker notes, alt text; best‑effort chart labels/titles.
  + ✅ Accept: Text surfaces masked; non‑text objects reported as “not processed”.
* **B5. PDF (read → masked HTML/TXT)**
  + Detect text‑based vs image‑only; use PyMuPDF for reading order; output <name>\_masked.html + \_masked.txt with page‑break markers.
  + ✅ Accept: Image‑only PDFs error out cleanly; masked HTML contains zero residuals above threshold.

## EPIC C — Caps, policies, & reporting

* **C1. Caps enforcement**
  + CSV/XLSX ≤100,000 rows; PDF ≤100 pages or ≤10 MB.
  + ✅ Accept: Exceed → skip + error entry; CLI returns non‑zero exit code when configured.
* **C2. Output policy & versioning**
  + --out required; mirror tree; automatic versioning name (2).ext.
  + ✅ Accept: No in‑place writes unless explicitly added later; collisions always version.
* **C3. Summary reports (HTML + JSON; optional CSV findings)**
  + Per‑run: inputs, config snapshot, results, entity counts, confidence histograms, questionables/decisions, warnings (image‑only, embedded).
  + ✅ Accept: Reports never include raw PII; only tokens + metadata.

## EPIC D — GUI (PySide6)

* **D1. Shell & DnD**
  + Drag‑and‑drop files/folders; recursive toggle; required Out‑dir.
  + ✅ Accept: DnD on main panel; validates caps and types pre‑run.
* **D2. Options panel**
  + Review Queue toggle (default off), Dry‑run toggle (default off), masking format, brand keyword list file.
  + ✅ Accept: Options persisted per session (not disk) for v1.
* **D3. Run & progress**
  + Per‑file status, totals (masked/skipped/errors), cancel (graceful).
  + ✅ Accept: UI responsive; logs accessible.
* **D4. Review Queue**
  + Grid with filters; diff/preview: text diffs for TXT/CSV/XLSX/DOCX/PPTX, embedded webview for PDF masked HTML.
  + Actions: Accept/Reject, bulk by type/file.
  + ✅ Accept: Dry‑run path allows “Export masked outputs” from queue.
* **D5. Results screen**
  + Links to out‑dir, HTML/JSON report, CSV findings.
  + ✅ Accept: Clicking opens location; errors summarized.

## EPIC E — CLI

* **E1. Batch runner**
  + --in, --out (required), --recursive, --include/--exclude.
  + ✅ Accept: Mirrors GUI logic; same caps and outputs.
* **E2. Modes & exits**
  + --dry-run, --review-queue on|off (default off), --report html,json[,csv], --exit-on image-only-pdf,caps-exceeded,residuals.
  + ✅ Accept: Exit codes {0 ok, 2 policy violation, 3 fatal}.

## EPIC F — Packaging, docs, and samples

* **F1. PyInstaller build + install.bat**
  + Per‑user install; README; licenses; bundled models.
  + ✅ Accept: Fresh Windows host can run with no extra deps.
* **F2. Sample corpus**
  + Seeded DOCX/PPTX/PDF/XLSX/CSV/TXT for demos & tests.
  + ✅ Accept: Covers each entity type and edge cases.
* **F3. User docs**
  + README + short “Getting Started” and “FAQ (caps, errors, image‑only PDFs)”.
  + ✅ Accept: Screenshots of main flows; clear “local‑only” privacy callout.

# Architecture notes (dev‑facing)

* **Module layout**

pgsql

CopyEdit

app/

core/

detect\_rules.py

detect\_ner.py # PII Masker + NLTK wrapper

fuse.py

mask\_map.py

validate.py

io/

txt\_csv.py

xlsx.py

docx.py

pptx.py

pdf\_read.py # to HTML/TXT writer

report/

html\_report.py

json\_report.py

findings\_csv.py

ui/

main\_window.py

options\_panel.py

review\_queue.py

preview\_widgets.py

cli/

main.py

config/

defaults.json

utils/

caps.py

paths.py

versioning.py

logging.py

* **Interfaces (selected)**
  + detect\_ner.analyze(text: str, lang="en") -> List[Entity]
  + fuse.merge(dets\_rules, dets\_ner) -> List[Finding]
  + mask\_map.apply(text\_or\_runs, findings, mode="TOKEN") -> masked
  + validate.scan(masked) -> Residuals
* **Config defaults (hardcoded + load from file)**
  + language: "en"
  + entities: ["PERSON","ADDRESS","EMAIL","PHONE","IP","CREDIT\_CARD","BANK\_ACCT","NATIONAL\_ID","DOB","USERNAME","GEO"]
  + mask\_format: "TOKEN"
  + caps: {rows: 100000, pdf\_pages: 100, pdf\_mb: 10}
  + validation: {min\_confidence: 0.35, questionable\_band: [0.35,0.65], residual\_action: "warn"}
  + backend: "pii-masker"