MetaStrip: A Web Extension for File Metadata Inspection and Removal with FastAPI and ExifTool

Aisha Nicole Dones   
*College of Computer and Information Science  
Mapúa Malayan Colleges Mindanao*Davao City, Philippines anDones@mcm.edu.phJames Andrei Nadela  
*College of Computer and Information Science  
Mapúa Malayan Colleges Mindanao*Davao City, Philippines  
jaNadela@mcm.edu.phKen Gyrro P. Acquiat  
*College of Computer and Information Science  
Mapúa Malayan Colleges Mindanao*  
Davao City, Philippines  
kgAcquiat@mcm.edu.ph

***Abstract*—In modern digital workflows, embedded file metadata poses a persistent privacy risk, particularly in document sharing, media publishing, and collaborative environments. While tools like ExifTool offer robust command-line functionality for metadata inspection and removal, they remain inaccessible to non-technical users and lack integration with standard browser workflows. This paper presents MetaStrip, a hybrid privacy-preserving system that combines a web extension frontend with a FastAPI-powered local backend to automate metadata detection and removal. MetaStrip supports common file types (PDF, DOCX, JPEG, PNG) and offers two operational modes: manual pre-upload sanitation and automatic post-export cleaning, triggered by browser events. It also allows users to perform either full or user-specified metadata removal, supporting granular control over which fields are retained or stripped. All file processing occurs locally, ensuring full data ownership and offline privacy assurance. By integrating open-source tools with browser automation, MetaStrip provides a lightweight, user-centric approach to digital file sanitization. Functional and performance evaluations confirm the system’s effectiveness in metadata removal, processing speed, and usability, while literature comparisons situate it within current privacy-preserving technologies.**

***Keywords—metadata removal, digital privacy, Chrome Extensions API, file sanitization, local processing, privacy-by-design, user-centric security***

# INTRODUCTION

1. *Background of the Study*

In modern digital workflows, the exchange of documents and media files is routine. However, these files frequently carry embedded metadata—hidden information such as author names, GPS coordinates, timestamps, device identifiers, and software versions. While metadata serves functional roles in indexing and interoperability, it also poses serious privacy risks when shared unintentionally. This concern is particularly relevant in journalism, academia, and collaborative platforms, where file-based communication is prevalent. Although tools such as ExifTool offer powerful command-line capabilities for metadata inspection and removal, they remain inaccessible to the average user due to their technical complexity. As digital privacy becomes increasingly important, the need for usable, transparent, and effective metadata removal tools has grown significantly [1].

1. *Problem Statement*

Despite the privacy implications of metadata, many users are unaware of its existence, and few tools provide seamless, automated means of removing it as part of standard file-handling routines. Existing solutions are often fragmented—limited to specific file types, reliant on manual workflows, or dependent on cloud services that themselves may introduce privacy concerns. Platforms such as Google Docs, for example, lack built-in options for stripping metadata upon export. Consequently, users remain exposed to privacy leakage through common actions such as uploading a document or exporting a file for sharing. The absence of a locally operated, automation-aware, user-accessible solution for metadata removal constitutes a key gap in current privacy tools.

1. *Project Objectives*

This study aims to develop *MetaStrip*, a metadata removal tool that integrates a browser extension with a locally hosted FastAPI server and ExifTool backend. The project will support common file types such as PDF, DOCX, JPEG, and PNG, enabling both user-initiated (pre-upload) and automated (post-export) metadata cleaning workflows. The system will operate entirely within the user’s local environment, with an emphasis on privacy, usability, and efficient file handling.

1. *Significance*

This project contributes to the field of privacy-preserving technologies by proposing a hybrid, locally executable tool for metadata removal that integrates browser automation with server-side file sanitization. Unlike many cloud-dependent solutions, *MetaStrip* prioritizes local processing, ensuring that no files leave the user’s device. By offering both manual and automatic workflows, the system lowers the barrier to entry for non-technical users while addressing real-world privacy risks in digital communication. In doing so, it advances practical tools for safeguarding metadata privacy across file sharing and publishing scenarios [2, 3].

1. *Scope and Limitations*

*MetaStrip* supports four widely used file types: PDF, DOCX, JPEG, and PNG. The initial version focuses on compatibility with Chromium-based browsers, using the Chrome Extensions API for event monitoring and interaction. While the system offers both pre-upload and post-export metadata cleaning, it does not cover mobile platforms or cloud-native document formats such as Google Docs in their native states. Moreover, while ExifTool provides robust metadata coverage, some proprietary or deeply embedded metadata fields may not be fully removed. The project is scoped to prioritize local execution, extensibility, and modular design to accommodate future enhancements.

# Review of Related Literature

Existing literature highlights the growing importance of metadata sanitation in safeguarding user privacy during digital communication and file sharing. While metadata serves functional purposes such as search optimization, version control, and geotagging, it has been increasingly recognized as a vector for privacy breaches, especially in journalism, healthcare, and legal contexts.

ExifTool by Phil Harvey is the de facto standard for metadata inspection and removal across various file formats, including images, videos, and documents [1]. Despite its versatility and reliability, ExifTool's command-line interface and complex output structures make it less accessible for non-technical users. This usability gap motivates the integration of its capabilities into more intuitive, automated systems like MetaStrip.

Khalid and Zimanyi [2] proposed a human-in-the-loop framework for cleaning raw metadata files, introducing a semi-automated workflow for enhancing metadata quality while preserving semantic consistency. Their study emphasized the potential for user-guided metadata editing, laying the groundwork for systems that combine automation with user feedback—an avenue considered for future iterations of MetaStrip.

Privacy risks associated with social media metadata were examined by Tayeb et al. [3], who developed a prototype for selective metadata stripping to mitigate unintentional information disclosure. Their work reinforced the necessity of user-controlled metadata management, especially in environments where automated tools risk oversanitizing or undersanitizing critical fields. Unlike social-media-targeted solutions, MetaStrip focuses on general-purpose file formats and operates entirely offline, reducing dependency on external cloud APIs.

The development of browser extensions for metadata removal was explored by Varona et al. [4], who created a plug-in prototype for Firefox that could sanitize image metadata prior to upload. Their work demonstrated the feasibility of integrating file sanitization directly into the browser interface, an approach further extended by MetaStrip’s dual-mode (pre-upload and post-export) capabilities using the Chrome Extensions API.

Finally, Westbrook et al. [5] conducted a digital library audit aimed at identifying and purging metadata artifacts across institutional repositories. Their findings revealed the persistence of sensitive metadata in supposedly anonymized academic documents, underlining the limitations of manual review processes and the need for systemic tools like MetaStrip that enforce metadata hygiene at the point of file interaction.

Taken together, these studies provide foundational support for MetaStrip's architecture: a hybrid local system that fuses ExifTool’s robustness with FastAPI’s responsiveness and browser extension automation to deliver a seamless and secure user experience in metadata privacy management.

# Methodology

The development of *MetaStrip* follows an Agile methodology to allow for iterative prototyping, testing, and integration of features. This approach supports flexibility in responding to user feedback and aligning development with evolving privacy and usability goals. The system comprises two main components: a browser extension for user interaction and event detection, and a locally hosted backend server responsible for metadata operations.

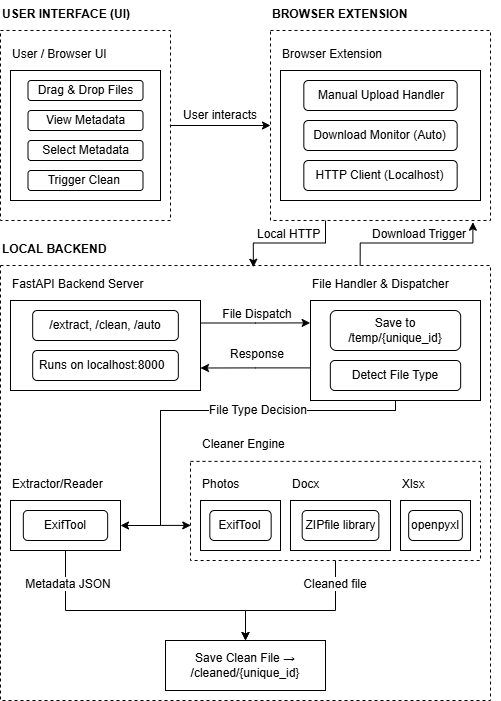
The backend server is implemented using FastAPI, a Python-based web framework optimized for high-performance asynchronous applications. FastAPI is chosen for its lightweight design, compatibility with Python tooling, and ability to expose endpoints for local use without external dependencies. For metadata parsing and removal, the system integrates ExifTool, an open-source utility known for its broad file format support and proven reliability in metadata manipulation [1]. The extension and server communicate through HTTP calls restricted to the localhost domain, maintaining strict data confinement.

A core principle of *MetaStrip* is “privacy by design.” All file processing is performed locally on the user’s device. STRIDE threat modeling was conducted during the planning phase to identify security vulnerabilities and inform architectural decisions. This included mitigation strategies for spoofing, information disclosure, denial of service, and privilege escalation. No external logging or remote storage is employed. This approach aligns with prior research that highlights the importance of local, user-controlled metadata management systems in minimizing risk while enhancing usability [2, 4, 5].

# System Implementation

*MetaStrip* is implemented as a dual-component system integrating a web extension frontend with a FastAPI-powered backend. The system supports two operational workflows: pre-upload metadata cleaning initiated manually by the user, and post-export sanitization triggered automatically upon file download events. In the pre-upload workflow, MetaStrip introduces a “Select Fields” option in the UI that enables users to choose which metadata fields to remove from the file. After viewing extracted metadata, users can mark specific fields—such as GPS location, author name, or software version—for deletion. These user selections are passed to the FastAPI backend as part of the HTTP request, enabling dynamic, field-specific metadata removal instead of blanket sanitization. Figure 1 illustrates the high-level system architecture of MetaStrip, showing the interaction between the browser extension, local server, and ExifTool module.

Fig. 1. System Architecture of MetaStrip.



In the manual workflow, users can drag and drop supported files—such as PDF, DOCX, JPEG, and PNG—into the browser extension interface. The extension forwards the file to the local backend server via HTTP, where ExifTool extracts its metadata. This metadata is displayed to the user, who may then choose to proceed with sanitization. Upon confirmation, ExifTool removes the embedded metadata and the server returns a cleaned version of the file, ready for download.

The Cleaner Engine adapts its removal logic based on the selected metadata fields received from the frontend. For files processed by ExifTool, commands are constructed to target only the user-specified tags. Similarly, for DOCX and XLSX files, internal cleaning routines are conditionally applied based on the requested fields. This allows for a customizable, minimally destructive sanitation process aligned with user intent.

The automated workflow activates when the user exports or downloads a supported file type from a browser-based platform, such as a Google Doc exported as a PDF. The extension listens for download events using the Chrome Downloads API. When such an event is detected, the extension sends the file to the local backend, which processes it immediately using ExifTool. The sanitized file is then saved to the user’s system, optionally renamed to reflect its cleaned state. This mode ensures metadata hygiene without requiring additional user interaction.

All processes are executed on the local device, and the extension and server are isolated from internet-based services. This architecture guarantees that sensitive files are never transmitted externally and remain under the full control of the user throughout their lifecycle. The use of asynchronous request handling in FastAPI ensures low-latency responses, even during concurrent operations. Together, these design elements enable a privacy-focused, responsive, and user-accessible system.

# Testing and Evaluation

To evaluate the comprehensiveness of metadata removal, the table below outlines common metadata fields grouped by file type. This mapping guided the functional test design for MetaStrip.

Table 1. Common metadata fields across file types

| **Metadata Field** | **File Type Presence** | | | |
| --- | --- | --- | --- | --- |
| *PDF* | *DOCX* | *JPEG* | *PNG* |
| Title | ✓ | ✓ | X | ✓ |
| Author | ✓ | ✓ | ✓ | ✓ |
| Subject | ✓ | ✓ | X | X |
| Keywords | ✓ | ✓ | ✓ | ✓ |
| Creator | ✓ | X | X | X |
| Producer | ✓ | X | X | X |
| Creation Date | ✓ | ✓ | ✓ | ✓ |
| Modification Date | ✓ | ✓ | X | X |
| Last Modified By | X | ✓ | X | X |
| Trapped | ✓ | X | X | X |
| Page Count | ✓ | X | X | X |
| File Size | ✓ | ✓ | ✓ | ✓ |
| Encryption | ✓ | X | X | X |
| Revision Number | X | ✓ | X | X |
| Total Editing Time | X | ✓ | X | X |
| Language | X | ✓ | X | X |
| Template | X | ✓ | X | X |
| Make (Camera) | X | X | ✓ | X |
| Model (Camera) | X | X | ✓ | X |
| Orientation | X | X | ✓ | X |
| Exposure Time | X | X | ✓ | X |
| FNumber | X | X | ✓ | X |
| ISO Speed Ratings | X | X | ✓ | X |
| Focal Length | X | X | ✓ | X |
| GPS Coordinates | X | X | ✓ | X |
| Software | ✓ | ✓ | ✓ | ✓ |
| Description | X | ✓ | X | ✓ |
| Category | X | ✓ | X | X |
| Content Status | X | ✓ | X | X |
| DPI / Resolution | X | X | ✓ | ✓ |
| Gamma | X | X | X | ✓ |
| Comments | X | X | X | ✓ |
| Copyright | X | ✓ | ✓ | ✓ |

Metadata field associations were compiled based on documentation from ExifTool [1], Adobe PDF Specification [6], Microsoft Office Open XML [7], and the PNG RFC standard [8].

The system will be evaluated across both functional and performance dimensions. Functional testing will assess the ability of MetaStrip to detect and remove metadata from a variety of file types. Selective cleaning functionality was tested by simulating user interactions with the “Select Fields” feature. For each file type, tests verified that only the chosen metadata fields were removed while untouched fields remained intact post-cleaning. This validated that the field selection mechanism and backend filtering logic functioned correctly across PDF, DOCX, JPEG, and PNG files. Sample files will be prepared with known metadata fields, including author name, creation and modification timestamps, GPS data, device model identifiers, and software version tags. ExifTool and independent metadata viewers will be used to verify the presence of metadata before processing and its successful removal afterward.

Performance benchmarks will focus on three core metrics: average processing time per file, CPU usage during concurrent operations, and change in file size post-sanitization. These metrics will be collected under controlled conditions and compared across file types and workflows. Additionally, usability testing will be conducted by simulating typical user scenarios, such as batch uploads, drag-and-drop cleaning, and Google Docs exports. The system will be deployed and tested primarily on Chromium-based browsers, including Google Chrome and Microsoft Edge.

To assess robustness, the system will be exposed to malformed and unsupported file types to observe failover behavior and error handling. These test cases will validate whether the application gracefully handles edge conditions without compromising data integrity. Future development may explore the integration of user feedback mechanisms for guided metadata cleaning, inspired by recent work in metadata enhancement and human-in-the-loop systems [2, 4, 5].

##### References

1. P. Harvey, “ExifTool.” Apr. 2025. [Online]. Available: https://exiftool.org/
2. H. Khalid and E. Zimanyi, “Cleaning Raw Metadata Files with Human-in-The-Loop,” Data & Knowledge Engineering, Jan. 2023, doi: 10.2139/ssrn.4581057.
3. S. Tayeb et al., “Toward metadata removal to preserve privacy of social media users,” 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pp. 287–293, Jan. 2018, doi: 10.1109/ccwc.2018.8301741.
4. D. A. Varona, G. G. Mansilla, and R. O. Z. Sánchez, “A browser plug-in for metadata removal,” Universidad Complutense de Madrid, 2020. [Online]. Available: https://eprints.ucm.es/id/eprint/61630/
5. R. N. Westbrook, D. Johnson, K. Carter, and A. Lockwood, “Metadata Clean Sweep: a Digital Library Audit project,” D-Lib Magazine, vol. 18, no. 5/6, May 2012, doi: 10.1045/may2012-westbrook.
6. Adobe Systems Incorporated, PDF Reference, Sixth Edition: Adobe Portable Document Format Version 1.7, Nov. 2006. [Online]. Available: https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000\_2008.pdf
7. ECMA International, Office Open XML File Formats – 2nd Edition (ECMA-376), Dec. 2011. [Online]. Available: https://www.ecma-international.org/publications-and-standards/standards/ecma-376/
8. T. Boutell and the PNG Development Group, PNG (Portable Network Graphics) Specification Version 1.0, RFC 2083, Mar. 1997. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2083