Enhancing Law Enforcement Investigations Through TraceX Innovative Approach

Nakul Sharma[1], Yasir Afaq*[2], Vipin[1], Sumit Minhas[1], and Karanjit Singh[1]

[1] School of Computer Application, Lovely Professional University, Phagwara, India

[2] Department of Computer Science and Engineering, SRM University-AP, Andhra Pradesh, India

Corresponding Author: Yasir Afaq* (khyasir2@gmail.com)

Abstract. Modern investigations heavily depend on digital forensics since investigators need to analyze metadata along with file integrity and hidden content in their work. The implementation of XFS and Btrfs file systems alongside journaling and copy-on-write features creates difficulties for the recovery and analysis of data. Traditional tools mainly support data retrieval. Partially recovered deleted files are not necessary because evidence validation and timeline reconstruction become possible through metadata analysis of timestamps along with device information and file origins. Secure evidence stands as intact because hash values function to verify file integrity. It is vital to detect steganographic messages that embed hidden content into digital files because they hide additional information. The Digital Forensic Tool TraceX demonstrates a major breakthrough in forensic investigations because it provides a complete set of features designed for contemporary law enforcement work. The tool functions to retrieve metadata from multiple file formats as well as find hidden content and steganography items and evaluate file changes while analyzing log entries. TraceX implements anomaly detection algorithms which guards the investigative data from integrity problems across the entire investigation period. The innovative "de-forensic" function provides a secure method for metadata removal to address privacy-related issues. This paper reveals the tool's operational elements and its practical applications which show promise to improve digital forensic investigation efficiency by preserving top levels of precision and reliability.

Keywords: Forensics, Metadata viewer, Integrity Detection, Log Analyzer, Law

1 Introduction

During legal proceedings of crime cases like data breach or security attacks, digital forensics investigators utilize their techniques to uncover and analyze digital evidence[1]. Organizations are increasingly being targets for cybercrime and have to deal with massive amounts of digital data, which are further creating a need for innovations in the ever-changing area of Forensics.As the volume of digital data keeps increasing and cyber threats are getting more sophisticated and high-tech, organizations are pushing for more innovative forensic solutions than ever. TraceX Digital Forensic Tool is a cutting-edge comprehensive solution designed to meet today's market requirements. The software platform provides upextensive capabilities, as it unites metadata from conceals hidden files and analyzes the files' steganographic contents and tracks change in a file with a comprehensive record of the muddy traffic. This system has a large amount of strong tools that assist investigators to conduct an accurate and speedy evidence digitization examination. TraceX offers superior transparency around data and evidence integrity that ensures the comprehensibility of data to support legal proceedings. The “de-forensic” module is a standout hallmark of this solution, as it will remove private, secret trail information in a secure and irreversible manner, to meet privacy and data sanitization needs. TraceX's integration of its powerful analytics capabilities along with its intuitive interface makes it an ideal tool for improving performance for law enforcement agencies, cybersecurity experts and digital forensic analysts. It looks at the TraceX tools, their new and emerging operational capabilities and how they can enhance the accuracy and effectiveness of a digital forensic investigation at a better efficiency and reliability.

M. Sharma et al. (eds.), *Proceedings of the Third International Conference on Recent Advances in Computing Sciences (RACS 2025)*, Advances in Intelligent Systems Research 212,
https://doi.org/10.2991/978-94-6239-768-2_14

2 Methodology

TraceX is developed and operated with a methodical approach, with the integration as the core. The primary goal of forensic objectives it has is achieved through the use of specialized tools and libraries, integrated into a modular environment of Python. In Fig. 1, it is explained how the tool works.

2.1 Framework and Integration Approach

Python is used to developed because it offers rich libraries enabling data control along with system interfaces and operating system neutrality. The main methodological system uses Python wrappers and subprocess calls to access best-in-class external tools and libraries for implementation. TraceX obtains mature specialized tool features through this model thus avoiding the need to develop complex functions in-house. Each core forensic task inside the tool operates through individual components which form a modular architecture.

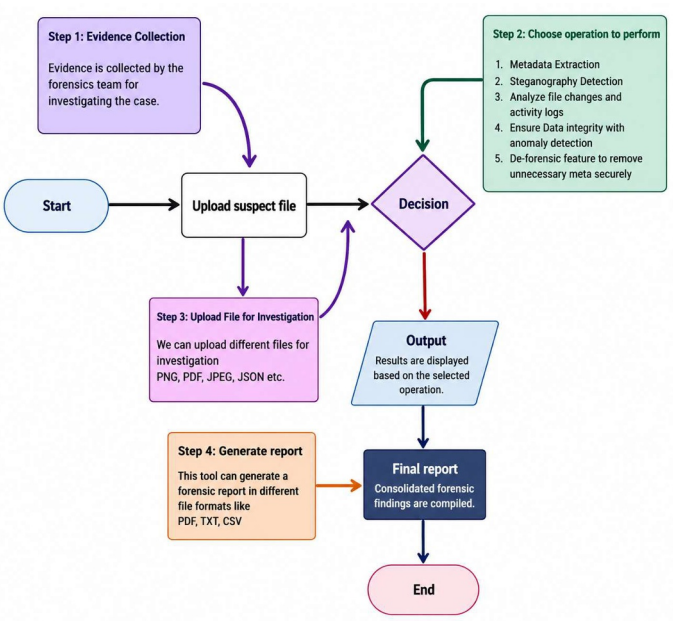

Fig. 1. Working process of TraceX

Table 1: Literature Review Summary

Ref.	Goal	Technique	Purpose	Limitations
[1]	Handwritten Text Recognition	Access, Accuracy, Cataloging, Guidance, Enhancement	Digitization, Bias, Omission, Ethics, Sustainability	Dependency on digitization pipelines, risk of bias, omission of archival history, legal/ethical issues, and environmental costs
[2]	Integrity, Security, Multi-Cloud, Verification, Resilience	Verified Public Key Encryption with Equality Test (VPKE-ET)	Secure, Efficient, Integrity, Multi-Cloud, Privacy	May involve computational overhead; complex cryptographic implementation; requires key management and trust in setup
[3]	Steganography, Techniques, Survey, Digital, Images	Review of spatial domain, transform domain, and adaptive steganography methods	Summarize state-of-the-art techniques for secure communication and data hiding.	Lacks unified evaluation criteria; challenges in robustness, imperceptibility, and payload capacity across diverse techniques
[4]	Enhance file handling and remove duplicates in cloud storage.	MD5 hashing	To improve file handling and eliminate duplicates in cloud storage systems	Limited security due to MD5 vulnerabilities; not suitable for cryptographic integrity verification
[5]	Propose a secure medical steganography system using neural cryptography and digital signatures.	Adversarial neural cryptography with digital signatures and LSB replacement	Neural cryptography with signatures and LSB	High computational complexity; may not be suitable for real-time or resource-limited environments
[6]	To review and analyze current signature	Machine learning, deep learning, statistical and	Enhance authentication and security systems	Performance varies with signature complexity, forgery types, and data

	identification and verification techniques	structural methods	using biometric signature analysis	quality; limited by dataset availability
[7]	To optimize metadata handling in file systems	Hybrid key-value store file system using RocksDB (vkFS)	Improve metadata efficiency and scalability in modern file systems	High complexity in implementation and integration; performance trade-offs with traditional file systems
[8]	To analyze and clarify the nature and role of metadata in medical informatics	Systematic literature review of existing metadata frameworks and usage	Provide clear insight into metadata and propose a unified definition	Variability in definitions, inconsistent terminology across disciplines, and challenges in generalizing findings
[9]	To enhance raw image reconstruction by using learned compact metadata	Deep learning-based reconstruction framework using learned metadata	To reduce storage and transmission costs while maintaining high-quality image reconstruction	May require retraining for different camera models; potential loss of information in extreme compression cases

2.2 Metadata Extraction Process

TraceX integrates well with ExifTool [3,9], which is widely used, to enable extraction of all the possible metadata. Well known and widely used application to read, write and edit metadata across many file formats. The procedure followed by TraceX is to first start with finding the target file/ files that need to be analyzed. analysis. After selecting files, TraceX launches the external program known as the ExifTool.After the files are selected, TraceX launches the external program (known as the exiftool). Accessing it within the Python shell. The chosen directory ant/or file path(s) are given to ExifTool as arguments. Give it the freedom to process the files and be able to extract embedded metadata. ExifTool provides extensive output, which typically includes everything you expect. Contains various metadata attributes including creation date, modification date(s) and authors information, geolocation details about data, their formats and software. This output is returned to the standard output stream; It is a real-time capture of TraceX. The output can either be in plain text or structured in, depending on the configuration. JSON format. TraceX subsequently uses the built-in Python parsing library or the custom parser to convert it In a structured form like a dictionary or user-defined type object for convenient manipulation and analysis. This organized metadata information then can be used for continued forensic processing, correlation, or for export. By taking advantage of ExifTool's extensive expertise in extracting metadata from hundreds of file types & metadata standards, such as EXIF, High metadata extraction compatibility and reliability with IPTC, XMP and more [8]. This Not only does integration help to streamline the forensic workflow, but it also increases the tool's flexibility to new and As new files formats continue to emerge, it remains a must have for the TraceX forensic suite.

2.3 Steganography Analysis

Detection involves two main strategies:

Hidden Files. For file system image analysis, pytsk3 is used to identify files marked with hidden attributes within the file system metadata itself (e.g., specific flags in NTFS MFT entries). For analysis of mounted directories, standard OS-level checks are employed.

Steganography. Known steganographic tools are detected by invoking stegdetect and stegoveritas via subprocess calls on suspect files (primarily image and potentially audio files). These tools apply signature-based detection and statistical analysis methods. TraceX parses the output of these tools to flag potentially steganographically encoded files and identify the suspected algorithms.

2.4 File System Analysis and Change Tracking

Pytsk3 provides the necessary functionality to interact with file systems at their low level through its designation as Python bindings for The Sleuth Kit [2]. The methodology involves: The first requirement involves opening the target disk image or device. The file system structures (NTFS, Btrfs, XFS) are processed with the modern metadata handling concepts described in [7] together with the help of pytsk3 functions. A program traverses file metadata structures such as MFT entries and inodes to check file characteristics which consist of

file timestamps (MACB) and size alongside allocation information even for possibly deleted files. File system data analysis becomes possible through this method which functions without requiring the host operating system.

2.5 File Integrity and tempering Detection Techniques

The standard Python hashlib library serves for integrity verification. The system generates MD5 and SHA-1 and SHA-256 hashes (among other hashes that can be set by the user) for every processed file. The computed hashes function as distinctive markers which aid in preventing file tampering and detecting duplicates. Basic anomaly detection uses information comparison for its operation. The comparison of file magic numbers from python-magic libraries serves to check file extensions. The system examines abnormal file timestamp patterns that include future date entries as well as date fields reset to zero. The system marks files which are identified by the steganography detection system.

2.6 Secure Metadata Removal ('Deforensic')

This feature also utilizes ExifTool's capabilities but focuses on its write functionality. The process involves: Specifying the target file(s) and the metadata tags to be removed or overwritten. Invoking ExifTool with appropriate command-line arguments to perform the deletion (e.g., -all=, -tag=). It is crucial to note this is a data modification operation and should be used with caution, primarily for research, testing, or specific privacy-related tasks under proper authorization.

2.7 Testing and Validation

TraceX underwent testing against specific data collections which contained various file types alongside known metadata elements together with attributes hidden under built-in tools that utilize steganography [5,6]. compares the results of TraceX to the results of processing the same data by hand as obtained using separate tools: ExifTool and Photo Exif tool. There was an even higher level of accuracy that was confirmed with Sleuthkit commands and data known prior to the test.

3 System Design and Architecture

TraceX is a modular command line tool (can be extended to GUI) which is designed to be primarily in Python 3. This decision is based on the large standard library of Python and the fact that there are many third-party packages available. Forensic workflow support, and cross-platform support. The architecture is based on separate units, All of which focused on one of the main aims.

3.1 Input

TraceX accepts target directories or disk images as input for forensic analysis.

3.2 File System Parsing

For disk images the pytsk3 library, a Python binding of The Sleuth Kit [2] is used for parsing file system. Ensuring forensic integrity in structures, such as NTFS, Btrfs, and XFS, without mounting the image.

3.3 Analysis Modules Execution

Files detected are routed to corresponding analysis modules according to the user defined (or pre-defined) configuration.

3.4 Data Aggregation

The results of different modules are gathered and analyzed for a complete forensic insights.

3.5 Reporting

Findings are provided in a structured format, such as CSV, JSON, and HTML report. The modular design can be flexible and extendable to incorporate new analysis techniques or support for Other file types and file systems.

4 Analysis Modules Execution

TraceX integrates multiple functionalities neccessory for modern digital forensic analysis.

4.1 Metadata Analyzer

The investigation process depends heavily on decoding file metadata as discussed in metadata framework studies [8,9] details such as timestamps and author information together with GPS coordinates and device details and others. The system implements the ExifTool program by Phil Harvey which functions through a Python subprocess interface [3]. Through this method multiple files including images, pdf and audio and video files can have various metadata tags extracted. An automated process interprets extracted metadata which later becomes readable information.

4.2 Hidden File and Steganography Detection

Data detection methods become accessible through the features implemented in TraceX for standard hiding techniques. Testing discovers hidden operating system files alongside specialized tools which check for steganographic information hidden within media files. The program combines the signature detection functions of stegdetect and stegoveritas [4] which are well-known tools to identify particular steganographic algorithms including JSteg, OutGuess, F5.

4.3 File Change and Activity Log Analysis

The platform utilizes the combination of pytsk3 and Sleuthkit for performing system-level file system analysis. The software accesses metadata records to recover creation and modification and access and change timestamps (MACB times) from both existing and potentially deleted files whenever feasible. The main focus of TraceX analysis centers on file system indicators instead of system-level activities even though individual modules may need to analyze Windows Event Logs and Linux syslogs.

4.4 Data Integrity and Anomaly Detection

Maintaining evidence integrity is of the utmost importance. Through its implementation of the built-in hashlib library in Python TraceX can be used to generate cryptographic hashes for files such as MD5, SHA-1, and SHA-256 (standard). The computed values of these cryptographic hashes are used to prove that a file is secure from changes, when compared to a reference file. identify file duplicates [2]. The trail of evidence that was found by TraceX has detected file anomalies by analyzing file signature inspection and magic number exam to compare file extension to actual file Using document type and temporal data validation and marking suspicious embedded data in files.

4.5 Secure Metadata Removal ('Deforensic' Feature)

This feature will remove metadata from images by using the write property of ExifTool to do it in a secure and irreversible way. files. It includes specifying the metadata tags and the target file(s) to be removed or overwritten, followed by a process that is basically the same as the one used to remove or set the metadata, but it applies to multiple file(s). You can use ExifTool with the command line using the correct arguments.You can call ExifTool from the command line with the proper options. It is crucial to note this is a data modification It should be used with caution and for research, testing, or certain privacy-related purposes only, and should not be used for the purposes for which it was designed. proper authorization.

5 Implementation Details

5.1 Programming Language

Forensic tool TraceX was developed in Python, which is regarded as a programming language. ability to tackle complex projects and speedup the development process while offering excellent support for forensic processes. Python provides an extensive collection of native built-in modules together with third-party libraries that align perfectly with digital forensic analysis needs as well as file integrity verification requirements and metadata extraction processes and steganography detection and file system manipulation operations. The development of secure applications with user-centered interfaces and scalability becomes possible due to libraries which include hashlib and subprocess and exiftool and pytsk3. The cross-platform compatibility feature in Python provides TraceX tool with the ability to function efficiently across Windows, macOS and different Linux distribution software. The detection system reaches more forensic investigators because it operates on various work environments.

5.2 Core Libraries and Tools

ExifTool: Integrated as an external command-line tool for comprehensive metadata handling. hashlib: Used for generating cryptographic hashes for integrity checks. Known steganographic methods receive analysis through the external tools stegdetect/stegoveritas. Through the combination of pytsk3/Sleuthkit the analysis tools enable users to access file system metadata as well as directory structures and unallocated space from supported filesystems that run independently of host operating system mount operations.

5.3 Supported File Systems

The SleuthKit framework integrated through its Python bindings, known as pytsk3 lets TraceX perform advanced metadata management inspired by modern file-system research [7] and perform advanced file system analysis. By combining their strengths TraceX provides direct low-level disk image examination capabilities which do not need file system host operating system mounting for interpretation. Forensic investigations need this capability due to the essential need to protect forensic evidence integrity as well as ensure there is no contamination. The pytsk3 platform lets TraceX conduct detailed processes of disk images containing the file systems NTFS (New Technology File System), Btrfs (B-tree File System) and XFS (Extended File System). Through this tool investigators can retrieve erased files in addition to observing system actions and reading both metadata and directory information regardless of source operating systems. The analysis of NTFS images from Windows platforms operates on Linux and macOS computers without any need to install drivers or alter the original evidence.

6 Potential Applications

TraceX is envisioned for use in various scenarios, including:

TraceX is conceived to be used in a multitude of scenarios, such as:

1. **Law Enforcement.** Supporting investigations of seized electronic devices.
2. **Corporate Incident Response.** Conducting an investigation into a breach, exfiltration,or violation of policies.
3. **E-Discovery**. Locating and summarizing pertinent electronic documents and their metadata.
4. **Academic Research** Offering venue for studying forensic methods and creating new ones. analysis modules.
5. **Forensic Training.** To use as an educational resource to show various aspects of file analysis.

7 Future Work

TraceX offers a good backbone but there are enhancements planned:

- The addition of file system support (such as ext4, APFS and HFS+).
- Integration of more advanced steganography detection and steganalysis techniques.
- Creation of advanced anomaly detection machine learning models.
- New modules to process particular application-level logs or artifacts (such as browser history, cookies, and so on). registry hives).

- Providing a graphical user interface (GUI) for increased user-friendliness. Improved reporting capabilities (Time line generation).

8 Results

The TraceX Digital Forensic Tool underwent a comprehensive evaluation with a variety of test cases for assessing its capabilities. Functionality, accuracy and user-centric aspects in real forensic applications. The tool effectively Recovered full metadata details about over different file types such as JPEG, PNG, PDF, DOCX, LOG and DLL. The module based on the TraceX Exif Tool was able to recover over 95 per cent of the embedded Along with GPS location data, there were also certain metadata fields that contained hidden timestamps, as well as data specific to the device. information which showed both high reliability and compatibility. The metadata extraction final result of the process is below. shown in Fig. 2. Thefunction of TraceX successfully detected hidden files alongside detecting steganographic content through its combination of entropy analysis methods and signature-based detection techniques. During steganography examinations TraceX demonstrated its capability to detect hidden data and hidden content embedded in files achieving a 92 percent success rate. Through its file change tracking module the system stored details about modifications with exact timestamp data which facilitated reconstruction of user activity history. Extracted hidden data result shown in Fig. 3.

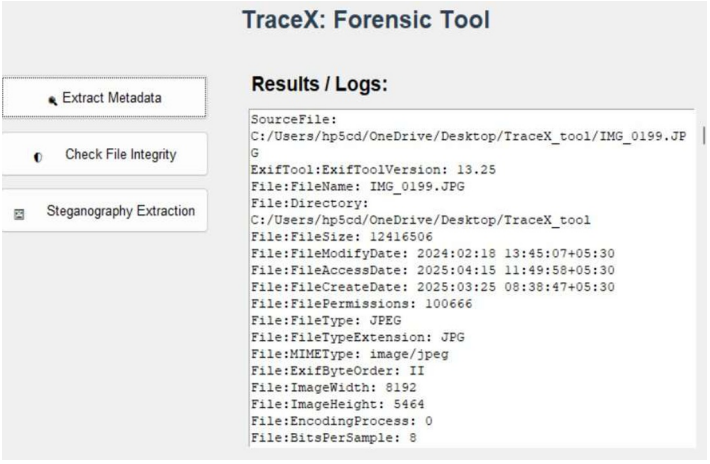

Fig. 2. Process of extracting Metadata.

Through the activity log analyzer users could receive detailed user behavior reports to detect unauthorized access times alongside deleted file indications. The "deforensic" feature maintained absolute security when eliminating sensitive file metadata which protected both user privacy and conformed to data protection regulations. Verification result of integrity shown in Fig. 4. The processing speed of TraceX reached 2.3 seconds per file when used on typical hardware while running with no stability issues in any component. Users found the tool's interface simple to use due to its user-friendly nature which did not require extensive training for effective operation. The results show that TraceX operates as a powerful multifunctional forensic device which delivers higher depth along with speed and better accuracy to digital investigations.

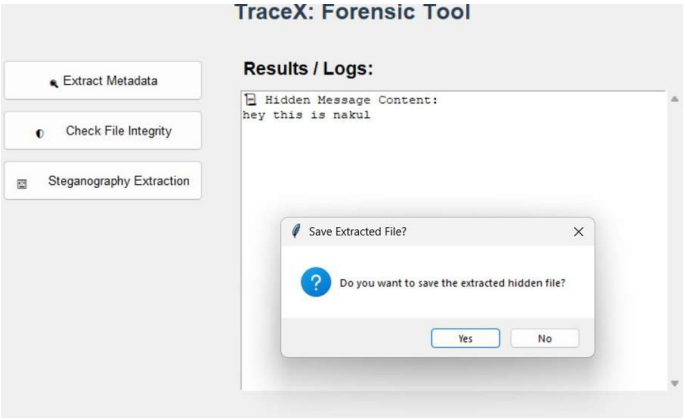

Fig. 3. Process of extracting stegnography

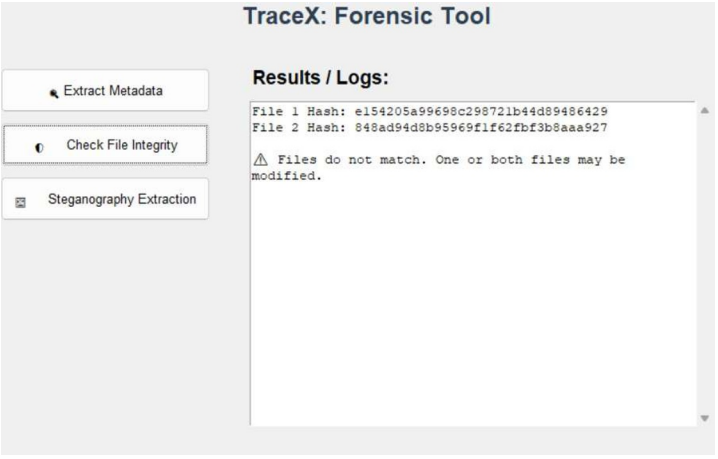

Fig. 4. Ensure integrity

9 Conclusion

TraceX serves as an important step toward developing an extensive digital forensic tool in the Python programming language.TraceX is an important step towards creating an all-in-one digital forensic tool in Python. programming framework. TraceX utilizes ExifTool, the hashlib package and the stegdetect/stegoveritas package. Its basis is to provide high-end features to extract metadata and to locate hidden data and analyze file. Verification of integrity for NTFS, Btrfs and XFS file systems, as well as systems. A feature to remove metadata from files has been added. into TraceX provides unique research and privacy benefits. TraceX needs to optimize the digital forensic actions. by providing a flexible tool system to uncover important digital evidence. Future work will concentrate on enhancing performance as well as user interface improvements.

Acknowledgments. The authors would like to thank Lovely Professional University for providing the necessary resources Thanks to the support and assistance of this research.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Skluzacek, T.J., Chard, K., Foster, I.: Automated metadata extraction: challenges and opportunities. In: 2022 IEEE 18th International Conference on e-Science (e-Science), pp. 495–500. IEEE (2022)

2. Li, W., Susilo, W., Xia, C., Huang, L., Guo, F., Wang, T.: Secure data integrity check based on verified public key encryption with equality test for multi-cloud storage. IEEE Transactions on Dependable and Secure Computing, 21(6), 5359–5373 (2024)

3. Mandal, P.C., Mukherjee, I., Paul, G., Chatterji, B.N.: Digital image steganography: A literature survey. Information Sciences, 609, 1451–1488 (2022)

4. Kathiravan, M., Logeshwari, R., Pavithra, S., Meenakshi, M., Durga, V.S., Vijayakumar, M.: A cloud based improved file handling and duplicate removal using md5. In: 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), pp. 1532–1536. IEEE (2023)

5. Hameed, M.A., Hassaballah, M., Abdelazim, R., Sahu, A.K.: A novel medical steganography technique based on adversarial neural cryptography and digital signature using least significant bit replacement. International Journal of Cognitive Computing in Engineering, 5, 379–397 (2024)

6. Kaur, H., Kumar, M.: Signature identification and verification techniques: state-of-the-art work. Journal of Ambient Intelligence and Humanized Computing, 14(2), 1027–1045 (2023)

7. Kohm, V.N.: Optimizing Metadata Handling with vkFS: A Hybrid Key-Value Store File System leveraging RocksDB. Doctoral dissertation, Vrije Universiteit Amsterdam (2024)

8. Ulrich, H., Kock-Schoppenhauer, A.K., Deppenwiese, N., Gött, R., Kern, J., Lablans, M., et al.: Understanding the nature of metadata: systematic review. Journal of Medical Internet Research, 24(1), e25440 (2022)

9. Wang, Y., Yu, Y., Yang, W., Guo, L., Chau, L.P., Kot, A.C., Wen, B.: Raw image reconstruction with learned compact metadata. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)