IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

SyncSphere: A Secure Cross-Device Intelligent Personal Data Sharing and Clipboard Synchronization System

H. V. Pavan Kumar[1], Saranya Elango[2], Dr. K. Akila[3]

Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India[1,2]

Guide, Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India[3]

Abstract: Personal data is frequently copied between laptops, desktops, smartphones, and tablets. Existing workflows often require users to transfer clipboard content manually through messaging applications, email, or other intermediary services. These extra steps interrupt workflow and may expose copied information to unnecessary services. This paper presents SyncSphere, a proposed secure cross-device clipboard synchronization system designed to make copied content available across a user's authorized personal devices. The system combines device connection and authentication, device pairing and management, clipboard monitoring, content processing and classification, cross-device synchronization, encryption, clipboard history, smart content recognition, and user notifications. The planned system supports multiple content formats—including text, links, images, and files—and aims to recognize URLs, email addresses, and phone numbers for quick actions. The project presentation identifies Python for backend logic, MongoDB for data storage, and Flutter for the cross-platform interface, with devices connected to the same network. This paper describes the motivation, related-work themes, proposed architecture, module design, implementation requirements, and an evaluation plan. Performance will be assessed using synchronization time, transfer success rate, content accuracy, time and effort saved, resource usage, and security checks. The supplied project review does not report completed experiments or measured results; therefore, no quantitative performance claims are made.

Keywords: cross-device synchronization; clipboard sharing; device pairing; data security; encryption; clipboard history; smart content recognition; personal computing

1. INTRODUCTION

Copy-and-paste is a basic interaction used in everyday computing. People commonly copy text, web links, contact details, images, and files on one device and then need the same content on another. When the devices do not share a clipboard, users may resort to sending a message to themselves, composing an email, using a cloud drive, or transferring a file manually. Such workarounds add steps and can scatter personal information across services that are not primarily intended for clipboard transfer.

A cross-device clipboard system can reduce this friction by transferring copied content between devices that belong to the same user. However, clipboard data may contain sensitive material such as personal messages, account details, work content, links, or contact information. Convenience therefore needs to be balanced with authentication, controlled device pairing, and protection of data during transfer and storage.

SyncSphere is proposed as a secure and intelligent personal data-sharing system that synchronizes clipboard content across authorized devices. Its design includes multi-format sharing, clipboard history, smart recognition of common content types, and a user interface that reports connection and synchronization status. The project aims to combine these capabilities in one workflow rather than requiring users to switch between separate applications.

1.1 Objectives

The project objectives are to develop a secure cross-device synchronization system for sharing clipboard content across multiple personal devices; provide real-time and reliable data transfer; support multiple content formats; protect access through authentication and device pairing; and make previously copied content available through clipboard history.

 This work is licensed under a Creative Commons Attribution 4.0 International License

1.2 Contributions

This paper documents the proposed SyncSphere design, its nine functional modules, the technologies identified in the project review, and a set of performance and security measures for future testing. It also describes the intended benefits and clarifies the distinction between proposed functionality and experimentally verified results.

2. LITERATURE SURVEY

The project presentation identifies ten base papers or research themes related to clipboard monitoring, AI-assisted copy-and-paste, cross-device agents, hybrid user interfaces, synchronization, authentication, and secure device-to-device communication. The review presents these works as relevant background, while noting that their primary focus varies across security analysis, agent-based interaction, healthcare or biomedical synchronization, and IoT communication.

No.	Research theme / work named in the presentation	Method or focus reported	Relevance / limitation noted
1	Clipboard Monitoring	Static analysis; 1.2M+ apps	Security focus; base paper
2	MagicCopy	AI copy-paste; 16 participants	Cross-application focus; base paper
3	DevicesWorld	Cross-device AI; 6,140 tasks	Agent focus; base paper
4	Unified Agent	Stateful AI agent; cross-device tasks	Agent focus; base paper
5	Hybrid User Interfaces	Survey / taxonomy	Mixed-reality focus; base paper
6	Multi-Device Synchronization	Data synchronization; 31 participants	Healthcare focus; base paper
7	Multimodal Data Synchronization	Timestamp synchronization	Biomedical focus; base paper
8	Device Authentication	Key agreement; protocol evaluation	IoT focus; base paper
9	Secure D2D Communication	Blockchain and key exchange	Complex security method; base paper
10	BEdX25519 Secure Key Exchange	ECC authentication; security evaluation	IoT focus; base paper

Taken together, these topics motivate a system that treats synchronization, security, and ease of use as connected design concerns. The supplied slides provide brief titles and summaries rather than complete bibliographic records. Accordingly, the entries above preserve the descriptions in the presentation; full author names, exact paper titles, publication details, and identifiers should be checked against the original papers before journal submission.

2.1 Research Gap

Based on the focus labels in the project review, several related approaches concentrate on one area, such as clipboard security, AI-assisted interaction, multi-device synchronization, or device authentication. SyncSphere proposes to combine these concerns in a personal clipboard workflow: automatic synchronization among paired devices, support for text, links, images, and files, encrypted transfer and storage, clipboard history, and recognition of common content such as URLs, email addresses, and phone numbers. This is a design-level gap statement derived from the supplied review, not evidence that no existing system offers these capabilities.

3. PROPOSED SYSTEM

SyncSphere is designed to synchronize newly copied content between authorized personal devices. Devices connect through Wi-Fi and are expected to be on the same network, as specified in the project presentation. When a user copies content, the system monitors the clipboard, identifies the content type, and prepares it for transfer to paired devices. The receiving device should make the content available for pasting and allow the user to access earlier items through clipboard history.

3.1 Key Features

- Cross-device synchronization: copied content is intended to become available on paired personal devices.
- Multi-format sharing: support is planned for text, links, images, and files.
- Secure device pairing: authentication associates devices with a trusted personal group.

- Encrypted data transfer and storage: clipboard information is intended to be protected in transit and at rest.
- Clipboard history: previously copied items can be viewed and reused.
- Smart content recognition: URLs, email addresses, and phone numbers can be identified to suggest quick actions.
- User interface and notifications: the interface displays connected devices, synchronization status, and updates.

3.2 System Architecture

The proposed architecture can be understood as a sequence of connected components. First, the user connects and authenticates, then pairs the devices that are permitted to exchange content. A clipboard-monitoring component detects new copied material. The content-processing component classifies the item and prepares the appropriate representation for transfer. The synchronization component sends the content to authorized paired devices, while the security component applies the planned access controls and encryption. A history component stores clipboard entries for later retrieval, and smart recognition identifies selected content types. Finally, the user interface and notification component communicates connection and transfer status.

Stage	Module	Main responsibility
1	User Connection & Authentication	Establish a user session and connect over the supported network.
2	Device Pairing & Management	Pair, authorize, and manage personal devices.
3	Clipboard Monitoring	Detect newly copied clipboard content.
4	Content Processing & Classification	Identify text, links, images, and files.
5	Cross-Device Synchronization	Transfer content between paired devices.
6	Security & Encryption	Protect clipboard content during transfer and storage.
7	Clipboard History	Store and retrieve earlier clipboard items.
8	Smart Content Recognition	Recognize URLs, email addresses, and phone numbers.
9	User Interface & Notifications	Show devices, synchronization state, and updates.

Figure 1. Proposed SyncSphere workflow: authentication and pairing → clipboard monitoring → content classification → secure synchronization → clipboard history and smart actions → status notification.

3.3 Processing Workflow

The intended processing workflow begins when the user connects a device and completes the required authentication and pairing steps. The clipboard monitor detects a new item and passes it to the content-processing component. The item is classified by format, and the system checks that the destination is an authorized paired device. The content is then transmitted using the planned security controls. On receipt, the target device makes the content available for reuse and updates the clipboard history. Smart recognition may provide quick actions for links, email addresses, and phone numbers. The interface reports the transfer state so that the user can identify successful or failed synchronization.

3.4 Functional Modules

3.4.1 User Connection and Authentication

This module establishes the user's connection to SyncSphere and provides the authentication boundary for access to the system. The presentation specifies Wi-Fi connectivity. The precise authentication protocol and account-management flow need to be detailed during implementation.

3.4.2 Device Pairing and Management

The pairing module registers trusted personal devices and maintains their association with the user. Synchronization should be restricted to devices that have been paired and verified, and users should be able to review and manage their connected devices.

3.4.3 Clipboard Monitoring

The clipboard-monitoring module detects newly copied content and initiates processing. The implementation should account for operating-system permissions and platform-specific clipboard restrictions.

 This work is licensed under a Creative Commons Attribution 4.0 International License

3.4.4 Content Processing and Classification

This module identifies whether the copied item is text, a link, an image, or a file. Classification allows the system to apply an appropriate representation and transfer procedure for the content type.

3.4.5 Cross-Device Synchronization

The synchronization module transfers clipboard content between paired devices. Its implementation should handle network interruptions, duplicate events, transfer failures, and the ordering of updates. The project review describes real-time and reliable transfer as an objective; measured latency and reliability remain to be established.

3.4.6 Security and Encryption

This module is intended to protect clipboard content during transfer and storage and to enforce the rule that only authenticated, paired devices may access synchronized information. The final implementation should document its encryption mechanisms, key handling, authentication flow, and treatment of sensitive clipboard entries.

3.4.7 Clipboard History

The history module stores clipboard entries so users can view and reuse previously copied content. The implementation should define retention limits, deletion controls, and safeguards for sensitive items.

3.4.8 Smart Content Recognition

This module detects selected content types, including URLs, email addresses, and phone numbers, and suggests relevant quick actions. Recognition should be designed to avoid changing the original copied content.

3.4.9 User Interface and Notifications

The interface presents connected devices, synchronization status, and relevant updates. Notifications should communicate success or failure clearly without unnecessarily displaying sensitive clipboard contents.

4. IMPLEMENTATION REQUIREMENTS

The project review identifies a cross-platform application with Python for backend logic and processing, MongoDB for clipboard and user records, and Flutter for the front-end interface. The listed hardware includes laptops, desktops, smartphones, tablets, and other personal computing devices capable of running the application. The presentation states that all devices must be connected to the same network.

Component	Technology / requirement	Purpose stated in the project review
Backend	Python	Core backend logic and processing.
Data storage	MongoDB	Storage for clipboard content and user records.
Cross-platform interface	Flutter	Front-end interface across supported devices.
Hardware	Laptops, desktops, smartphones, tablets	Devices capable of running the application.
Network	Wi-Fi / same network	Connection condition described in the presentation.

The presentation does not specify a complete network protocol, encryption algorithm, key-management design, synchronization conflict policy, or detailed deployment configuration. These should be selected, implemented, and documented before a reproducible technical evaluation can be performed.

5. PERFORMANCE METRICS AND EVALUATION PLAN

The project presentation proposes six measures for evaluating SyncSphere. The following plan translates those measures into observable tests. No measured results were included in the supplied slides, so this section describes planned evaluation rather than completed experiments.

Metric	Definition	Suggested measurement procedure
Synchronization time	Time between copying content on the source device and its availability on the target device.	Record timestamps across repeated transfers under stated network conditions; report median and spread.
Transfer success rate	Share of attempted transfers that complete successfully.	Divide successful, verified transfers by total attempts and report by content type.
Content accuracy	Whether received content matches the original without loss or corruption.	Compare source and destination content; use byte-level or hash comparison where applicable.
Time and effort saved	Difference in steps and time compared with traditional manual sharing.	Run matched tasks using manual sharing and SyncSphere; record task time and actions.
Resource usage	CPU, memory, battery, and network consumption.	Measure idle and active use over a defined duration on representative devices.
Security	Whether only authenticated paired devices can access synchronized data and whether transfer data is protected.	Test authorized and unauthorized pairing, access attempts, and transfer/storage protection; document threat assumptions.

5.1 Experimental Protocol

A future evaluation should define a test matrix covering each supported content type, multiple device combinations, and representative network conditions. Each transfer should be repeated sufficiently to reduce the influence of transient network variation. Tests should include successful transfers, disconnections, device removal, unauthorized access attempts, duplicate clipboard events, and recovery after reconnecting. The study should report the number and type of trials, device specifications, operating systems, network conditions, and any excluded observations. A comparison against manual sharing should use the same tasks and participants where feasible.

5.2 Reporting Results

Results should include synchronization-time distributions, transfer success rates, content mismatch counts, task completion times, resource consumption, and security-test outcomes. If user studies are conducted, participant count, task design, consent process, and rating scales should be reported. Performance claims should be limited to the devices, platforms, content formats, and network conditions actually tested.

6. ADVANTAGES, NOVELTY, AND LIMITATIONS

6.1 Expected Advantages

The proposed system is intended to reduce manual transfers, provide seamless access to copied content across personal devices, support multiple formats, and reduce the steps needed to reuse information. Clipboard history can help users recover earlier items, while smart recognition may make common content actionable. Authentication, pairing, and encryption are intended to provide security controls for personal data.

6.2 Intended Novelty

The project frames its novelty as the combination of real-time cross-device clipboard synchronization, smart recognition, secure device authentication and encrypted transfer, and reusable clipboard history in a single system. This is a statement of the project's design objective; novelty relative to existing products or published research requires a more detailed and verified comparison.

6.3 Limitations and Risks

The supplied review does not establish measured performance or a completed implementation status for each module. The same-network requirement may limit use when devices are on different networks. Clipboard APIs and background execution rules differ across operating systems, potentially affecting monitoring and synchronization. Sensitive clipboard contents create privacy risks, especially if history is retained indefinitely or if a device is lost. Network interruptions, conflicting clipboard updates, large files, and incompatible formats may also affect reliability. These issues require explicit design decisions and empirical testing.

7. CONCLUSION AND FUTURE WORK

This paper presented SyncSphere, a proposed secure cross-device personal data-sharing and clipboard synchronization system. The design aims to synchronize text, links, images, and files across authenticated paired devices while providing clipboard history, smart content recognition, encryption, and status notifications. The project review identifies Python,

MongoDB, and Flutter as the principal software technologies and lists nine functional modules that support the intended workflow.

The next stage should complete the implementation details for authentication, pairing, encryption, clipboard monitoring, and transfer recovery, then test the integrated system across supported devices. Evaluation should measure synchronization time, transfer success rate, content accuracy, time and effort saved, resource usage, and security behavior. Since the supplied presentation contains no experimental results or full bibliographic references, quantitative claims and final literature citations should be added only after testing and verification.

ACKNOWLEDGMENT

The authors acknowledge Dr. K. Akila for guidance on the SyncSphere project and thank the department and institution for supporting the project work.

REFERENCES

[1]. E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 9846, Internet Engineering Task Force, July 2026. https://www.rfc-editor.org/rfc/rfc9846.html

[2]. D. Temoshok et al., "Digital Identity Guidelines: Authentication and Authenticator Management," NIST Special Publication 800-63B-4, National Institute of Standards and Technology, 2025. https://doi.org/10.6028/NIST.SP.800-63B-4

[3]. OWASP Foundation, "OWASP Top 10:2025," 2025. https://owasp.org/www-project-top-ten/

[4]. M. Souppaya, K. Scarfone, and D. Dodson, "Secure Software Development Framework (SSDF) Version 1.2: Recommendations for Mitigating the Risk of Software Vulnerabilities," NIST SP 800-218 Rev. 1, Initial Public Draft, 2025. https://csrc.nist.gov/projects/ssdf/publications

[5]. P. Saint-Andre and R. Salz, "Service Identity in TLS," RFC 9525, Internet Engineering Task Force, November 2023. https://www.rfc-editor.org/rfc/rfc9525.html

[6]. E. Barker, "Recommendation for Key Management: Part 1 - General," NIST SP 800-57 Part 1 Rev. 5, May 2020. https://doi.org/10.6028/NIST.SP.800-57pt1r5

[7]. National Institute of Standards and Technology, "Recommendation for Key Derivation Using Pseudorandom Functions," NIST SP 800-108 Rev. 1, February 2024. https://csrc.nist.gov/pubs/sp/800/108/r1/final

[8]. OWASP Foundation, "Cryptographic Storage Cheat Sheet," OWASP Cheat Sheet Series, accessed October 24, 2026. https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html

[9]. OWASP Foundation, "Authentication Cheat Sheet," OWASP Cheat Sheet Series, accessed October 24, 2026. https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

[10]. OWASP Foundation, "Logging Cheat Sheet," OWASP Cheat Sheet Series, accessed October 24, 2026. https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html

[11]. OWASP Foundation, "OWASP API Security Top 10 - 2023," 2023. https://owasp.org/API-Security/

[12]. MongoDB, Inc., "Security Checklist for Self-Managed Deployments," MongoDB Manual, accessed October 24, 2026. https://www.mongodb.com/docs/manual/administration/security-checklist/

[13]. MongoDB, Inc., "Encryption at Rest," MongoDB Manual, accessed October 24, 2026. https://www.mongodb.com/docs/manual/core/security-encryption-at-rest/

[14]. MongoDB, Inc., "Encryption," MongoDB Manual, accessed October 24, 2026. https://www.mongodb.com/docs/manual/core/security-data-encryption/

[15]. Android Developers, "Copy and paste," Android Developers documentation, accessed October 24, 2026. https://developer.android.com/develop/ui/views/touch-and-input/copy-paste

[16]. Flutter Team, "State management," Flutter documentation, accessed October 24, 2026. https://docs.flutter.dev/data-and-backend/state-mgmt

[17]. Flutter API documentation, "Clipboard class," Flutter services library, accessed October 24, 2026. https://api.flutter.dev/flutter/services/Clipboard-class.html

[18]. Python Software Foundation, "asyncio - Asynchronous I/O," Python 3 documentation, accessed October 24, 2026. https://docs.python.org/3/library/asyncio.html

[19]. Python Software Foundation, "secrets - Generate secure random numbers for managing secrets," Python 3 documentation, accessed October 24, 2026. https://docs.python.org/3/library/secrets.html

[20]. Python Software Foundation, "ssl - TLS/SSL wrapper for socket objects," Python 3 documentation, accessed October 24, 2026. https://docs.python.org/3/library/ssl.html

[21]. Python Software Foundation, "hashlib - Secure hashes and message digests," Python 3 documentation, accessed October 24, 2026. https://docs.python.org/3/library/hashlib.html

[22]. A. Langley, M. Hamburg, and S. Turner, "Elliptic Curves for Security," RFC 7748, Internet Engineering Task Force, January 2016. https://www.rfc-editor.org/rfc/rfc7748.html

 This work is licensed under a Creative Commons Attribution 4.0 International License

[23]. S. Josefsson and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)," RFC 8032, Internet Research Task Force, January 2017. https://www.rfc-editor.org/rfc/rfc8032.html

[24]. Apple Inc., "UIPasteboard," Apple Developer Documentation, accessed October 24, 2026. https://developer.apple.com/documentation/uikit/uipasteboard

[25]. "flutter_secure_storage," pub.dev package documentation, version 11.2.0, accessed October 24, 2026. https://pub.dev/packages/flutter_secure_storage