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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023

DOI: 10.17148/IJARCCE.2023.125258

# Integrating Public Reported Evidence Collection, Public Court Records Archive And Realizing Secure And Decentralized Case Document Management Using IPFS And Hyperledger Fabric Blockchain: An Implementation Study

**Karthik Banjan[1], Jishnu Pillai Anilkumar[2], Harshit Singh[3], Kumar Sunny[4], Ruhin Kouser[5]**

Bachelor of Technology (CSE), School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka[1]

Bachelor of Technology (CSE), School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka[2]

Bachelor of Technology (CSE), School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka[3]

Bachelor of Technology (CSE), School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka[4]

Assistant Professor, School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, [5]

**Abstract**: Cloud and Blockchain-enabled case document management system that combines InterPlanetary File System (IPFS) and Hyperledger Fabric blockchain. The system aims to provide secure, transparent, and decentralized storage and management of case documents, while ensuring the privacy and confidentiality of the documents. The system further implements evidence collection from the public and providing public access to the court records.

**Keywords:** Hyperledger Fabric, InterPlanetary File System, Microsoft Azure, Google Firebase, Docker, React, NodeJS, Spring Boot

## I. INTRODUCTION

The flow of information has been completely transformed by the digital environment, impacting a wide range of businesses, including the legal and law enforcement sectors. Collaboration between law enforcement, judicial institutions, and the public is essential in today's rapidly growing digital age to uphold justice and preserve societal order. Demand for creative solutions that make use of emerging technologies is rising as a result of the growing volume of data and the requirement for reliable and efficient document management.

This research is highly significant when seen in the perspective of the Indian judicial system. Being one of the biggest democracies in the world brings India certain difficulties in managing case documents and gathering evidence. This study seeks to contribute to the existing initiatives to streamline and improve the Indian legal system by implementing new technologies to bridge the gap between law enforcement, the judiciary, and the public.

The InterPlanetary File System (IPFS) and Hyperledger Fabric blockchains are combined in this study to create a blockchain-enabled case document management system. Transparency, accountability, and cooperation between the public, law enforcement organisations, and judicial systems are greatly enhanced by the combination of public evidence collection and the disclosure of public or redacted court documents. The proposed approach intends to encourage community involvement and participation in the pursuit of justice by giving the public the ability to actively provide evidence and access court documents that are fit for public consumption.

  

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023
DOI: 10.17148/IJARCCE.2023.125258

With the help of this implementation study, we hope to shed light on the possibility of combining IPFS, the Hyperledger Fabric blockchain, and Cloud storage to handle case documents, gather evidence from the public, and enable safe access to court approved records. The results and conclusions of this study can act as a basis for future developments in the use of technology to improve the Indian judicial system and promote cooperation among key stakeholders in the search for justice.

The design and implementation of the blockchain-enabled case document management system will be thoroughly examined in this study's technical aspects. Additionally, it will provide insights on how its performance and security were assessed, highlighting how crucial it is to store case files in a secure, impenetrable, and transparent manner. The system's viability and efficiency will be shown through the use of open-source technologies, exhibiting its capacity for deployment in practical situations.

## II. LITERATURE SURVEY

### *A. Literature Review*

#### *1) File sharing with IPFS and blockchain*

As a decentralized system for file distribution and storage, the InterPlanetary File System (IPFS) provides a safe and dependable file storage solution, IPFS makes use of a content-addressable network, where files are recognized by their content rather than by their location. The study [1] proposes a secure file sharing system by building on the merits of IPFS. Traditional centralized file sharing systems still have problems with single points of failure, privacy, and security flaws. Decentralized and tamper-proof file storage is provided by IPFS, while an extra layer of security, immutability, and transparency is provided by blockchain technology. The suggested approach [1] makes the most of both technologies' advantages to offer improved data integrity, user control, and file sharing security.

However, this system does have a few disadvantages – scalability, regulatory compliance, general usability without technical expertise, constant requirement of strong network reception, high storage requirements as it keeps multiple copies of dispersed files indefinitely, and the metadata of files may reveal sensitive information.

The proposed approach [1] shows how file sharing behaviours may be changed, and it acts as a starting point for further study into secure distributed systems. A promising path toward safe file sharing systems is provided by the combination of IPFS with blockchain technology, opening the door for improved data security and public confidence in digital ecosystems.

#### *2) Decentralized Multi-Party Consent Management For Secure Patient Health Records Transmission And Interchange*

In order to provide secure and regulated exchange of patient health information, this research [2] offers a completely decentralized multi-party permission management architecture that makes use of blockchain technology. This blockchain based system provides fine-grained access restrictions through the use of smart contracts, guaranteeing that data sharing adheres to patient-defined norms. The content of shared records is protected by encryption techniques, and digital signatures guarantee the validity of the data. The blockchain keeps track of all consent requests, pseudonymity, approvals, and revocations, enabling auditing and enforcing accountability.

However, this system does have a few disadvantages – scalability, complex user experience and data access control, governance/standardization, privacy, access control, regulatory compliance, and legacy system integration.

The proposed research work's [2] importance rests in its potential to transform consent management, secure patient health record exchange, and promote improvements in healthcare systems all across the world.

#### *3) Forensic Evidence Management With Blockchain Technology*

An implementation of a blockchain-based system for managing forensic evidence is presented in the paper [3]. The system makes use of the immutability, transparency, and decentralised consensus that are built into blockchain technology to guarantee the integrity and traceability of forensic evidence throughout its entire existence. The integrity and validity of the evidence are guaranteed by the use of cryptographic primitives and the forensic evidence management procedure is more efficient and transparent because to the automated documentation and audit trail. It offers a transparent and democratic approach to decision-making, guaranteeing that all stakeholders have a vote in the management and access control of forensic evidence by employing blockchain's consensus processes and smart contracts.

DOI: 10.17148/IJARCCE.2023.125258

However, this system does have a few disadvantages – scalability, privacy, regulatory compliance, interoperability/integration and governance and consensus mechanism

This paper's [3] investigation of blockchain technology in forensic evidence management has the power to completely change the discipline, enhancing the effectiveness and reliability of the criminal justice system.

*4) Antitampering Scheme Of Information Transfer In Judicial Systems With Blockchain*

In order to safeguard information about evidence transfer in the legal system, the study [4] suggests an antitampering technique that makes use of blockchain technology. The application of smart contracts in the process of transferring evidence eliminates the need for mediators and increases transparency. It maintains a visible and immutable audit trail of all the transactions and establishes hash functions, digital signatures and consensus (proof-of-work or proof-of-stake) to reduce the possibility of illegal alterations, collusion or manipulation.

However, this system does have a few disadvantages – scalability, standardization, adoption barriers, privacy/confidentiality, integration/interoperability, regulatory compliance, and governance and consensus mechanism.
The paper [4] makes a significant contribution by outlining a blockchain-based antitampering mechanism for the transfer of evidence information in the legal system. The recommended solution has a significant deal of potential to revolutionize the security framework of legal systems and improve the reliability of the evidence used in court cases.

*5) Hyperledger Fabric*

The research [5] highlights Hyperledger Fabric's scalable modular architecture and channel-based privacy, which enables flexibility in building and executing blockchain networks customized to distinct business requirement using chaincode. The paper highlights the responsibilities played by peer gossips, ordering services, and membership services in preserving the integrity and consensus of the blockchain network as it analyzes the major architectural elements.

However, this system does have a few disadvantages – interoperability/integration, consensus mechanism, privacy, regulatory compliance, adoption/network governance, and standardization.

As a distributed operating system for permissioned blockchains, the paper [5] provides a thorough description of Hyperledger Fabric. Hyperledger Fabric is a desirable option for business blockchain applications, due to its modular design, privacy features, scalability mechanisms, and support for smart contracts.

*B. Existing Methods*

*1) Traditional Paper Based Systems*

Court case documents have always been stored and retrieved using traditional paper-based systems as it is tangible, familiar, secure and easy to use. However, its limited accessibility and search capabilities, space and storage requirements, vulnerability to loss/damage, difficulty of version control and updates, time consuming retrieval and collaborative challenges forced us to look for alternatives.

*2) Electronic Document Management Systems (EDMS)*

EDMS offers a centralized, digital repository for the storage of several legal document types, including pleadings, motions, exhibits, court orders, and transcripts, in the context of managing court case records. The typical capabilities offered by these systems include document version management, indexing, search functionality, access control, accessibility and collaboration tools. Its disadvantages are – initial implementation and transition cost, privacy, data security, workflow integration and technological dependencies.

*3) Cloud Based Systems*

In recent years, cloud-based solutions for document retrieval and storage for legal proceedings have attracted a lot of attention and acceptance. These systems offer scalability, data security, flexibility, cost efficiency, accessibility, collaborative ability and data backup. There are concerns over its data privacy, legal compliance, vendor resilience, service level agreements, data transfer/migration and dependence on internet connectivity.

*4) Hybrid Systems*

Hybrid systems integrate both physical and electronic storage technologies, and is one strategy that has attracted a lot of interest recently. It is known for its accessibility, convenience, cost efficiency, searchability, indexing, collaborative ability

DOI: 10.17148/IJARCCE.2023.125258

and preservation of original documents. However, there are challenges associated with it – integration, security, data migration and conversion, legal compliance, legacy document archival, and technological dependencies.

*5) Distributed Ledger Technology (DLT) - Based Systems*

DLT, also known as blockchain, is a decentralized and unchangeable digital ledger that makes it possible to maintain records in a safe, immutable, tamper-proof, secure and transparent environment. It however has issues with scalability, privacy, confidentiality, regulatory compliance, and in migration of legacy systems.

## III. PROPOSED METHOD

*A. Evidence Archives - Evidence Collection Website*

- Develop a user-friendly evidence collection website accessible to both the public and law enforcement (police) officials.
- Enable the public to securely upload evidence through a comprehensive form.
- Implement a search functionality for law enforcement (police) officials to retrieve evidence based on specific criteria.
- Utilize Microsoft Azure Blob for secure evidence storage and seamless retrieval.
- Utilize Google Firebase for police credential authentication.

Evidence Archives, developed with React, Express and NodeJS is an advanced web application that aims to give a trustworthy platform for the public to upload and contribute crucial information regarding accidents and crimes that are being perpetrated. In cases when law enforcement agencies are unable to quickly obtain real-time crime information, this innovative website offers a substantial advantage. Evidence Archives fills the gap between the general public and law enforcement organizations by utilizing the capabilities of contemporary technology, making it easier to gather important evidence quickly. The user-friendly design of Evidence Archives is one of its key benefits as it makes it simple for users to submit different types of evidence, such as documents, photos, audio recordings, and videos. This guarantees that a wide variety of data may be gathered and made accessible to law enforcement officers for additional analysis and investigation.

Evidence Archives makes use of Microsoft Azure Blob to guarantee the safe storage and quick access of the uploaded evidence. This cloud-based storage option has a number of benefits, including scalability, robustness, and high availability. The use of Azure Blob guarantees that the public's evidence is securely saved and is unaffected even in the event of unanticipated hardware problems or system outages. This reliability is essential for protecting the integrity of the evidence and enabling quick data retrieval when necessary. Exclusive access to the evidence is offered to law enforcement agents via pre-configured login credentials using Google Firebase. With this restricted access, sensitive data is kept only in the hands of authorised individuals, increasing security and preventing misuse or manipulation.

The benefits of evidence archives go beyond the short-term gains from gathering and preserving evidence. The website encourages a feeling of community participation and collaboration in ensuring public safety by allowing the general public to actively participate in the reporting and documenting of crimes. Additionally, the platform's real-time capabilities enable law enforcement organisations to quickly obtain vital information, allowing them to effectively respond to ongoing events and deploy resources.

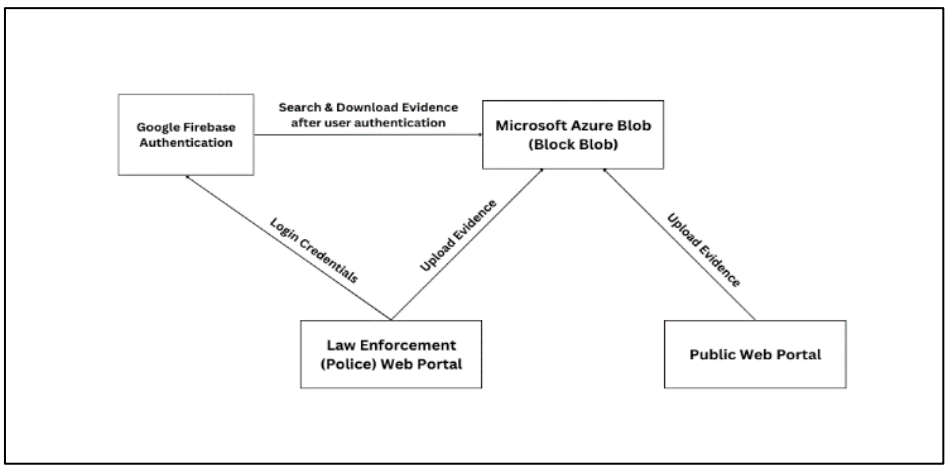


Figure 1: Architecture of Evidence Archives

  

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023

DOI: 10.17148/IJARCCE.2023.125258

*B. Court Vault – Court Records Website*

- Develop a secure login system for court officials to ensure data privacy and authorized access.
- Create a user-friendly interface for court officials to upload court documents, incorporating a comprehensive document upload form.
- Implement a search functionality for the public to find and download court documents based on keywords or by selecting a specific state.
- Integrate Microsoft Azure Blob for secure and scalable document storage, ensuring data reliability and availability.
- Integrate Google Firebase for police credential authentication.

The main objective of the Court Vault website developed with React, Express and NodeJS is to give the public a user-friendly portal for browsing and downloading court documents and judgments. This technology provides important benefits by making it simple to access essential legal data, encouraging openness, and expediting the distribution of court-related papers.

One of Court Vault's main advantages is its simple user-centric search tool, which enables users to quickly find and download relevant court records and judgments. Court staff are offered login credentials that have already been set up, allowing them to submit a variety of assets, including pictures, audio files, videos, and other pertinent document types which are eventually stored in Microsoft Azure Blob. The public can view case-related materials when the court staff uploads them to the website, enabling open access to court records and decisions. It incorporates Google Firebase to provide safe user authentication and maintain the platform's integrity.

Court Vault democratizes legal information and improves public involvement with the judicial system by offering a consolidated platform for accessing court-related resources.

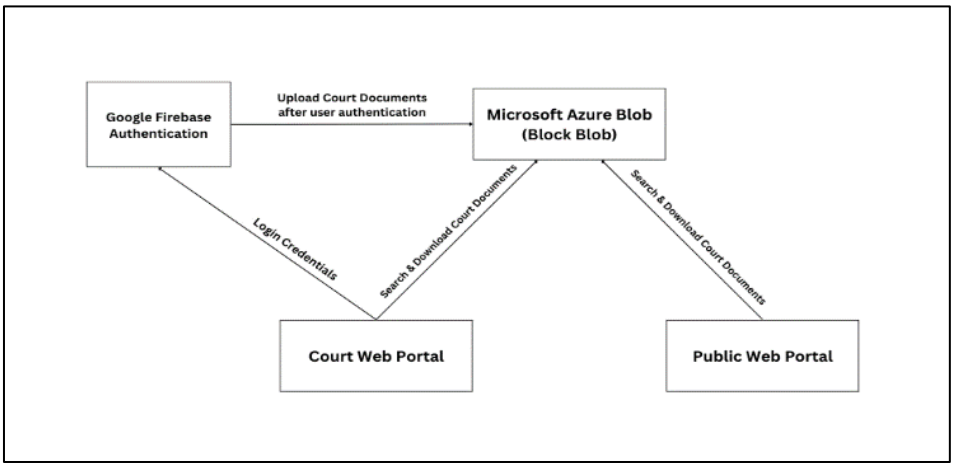


Figure 2: Architecture of Court Vault

*C. Case Cloud Blockchain –Blockchain based Evidence & Investigative Report Storage API Server*

- To develop a secure and decentralized case management system using Hyperledger Fabric, IPFS, and Spring Boot.
- To establish a transparent and auditable blockchain network for case management, involving the police and court organizations.
- To leverage the power of Fabric's modular architecture and Docker containerization to ensure efficient resource allocation and enhanced security.
- To enable essential case management functions, including case creation, query, update of IPFS hash, and deletion, through a Go-based Chaincode deployed on a dedicated channel.
- To facilitate seamless integration and interaction with the blockchain network through a Java-based application acting as a Rest API server, powered by the robustness of Spring Boot.

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023

DOI: 10.17148/IJARCCE.2023.125258

Case Cloud Blockchain is a sophisticated system that leverages cutting-edge technologies such as Hyperledger Fabric, IPFS, and Spring Boot to provide a robust and secure platform for case management. At its core, this system utilizes the power of Fabric, a powerful blockchain framework, to establish a decentralized network comprising two key organizations: the police and the court. Both organizations and their respective users actively participate in this transformative ecosystem.

Hyperledger Fabric, known for its resilience and scalability, forms the backbone of the blockchain aspect of Case Cloud. Leveraging the power of Fabric's modular architecture, the system ensures that each organization's components operate within dedicated Docker containers. This containerization approach not only promotes efficient resource allocation but also enhances security by isolating each organization's sensitive data and functionalities.

The heart of the Case Cloud Blockchain lies in its Chaincode, which is written in Go. This Go-based Chaincode is deployed to a dedicated channel established between the police and court organizations. By employing a shared ledger maintained by the blockchain network, the system ensures transparency, immutability, and auditability of case-related information. Through the Chaincode, essential functions such as creating a case, querying case details, updating case IPFS hash, and even deleting a case can be seamlessly executed.

To facilitate seamless integration and interaction with the blockchain network, Case Cloud employs a Java-based application that connects to the network through the Fabric Gateway. This application serves a dual purpose by acting as a Rest API server, allowing authorized users.

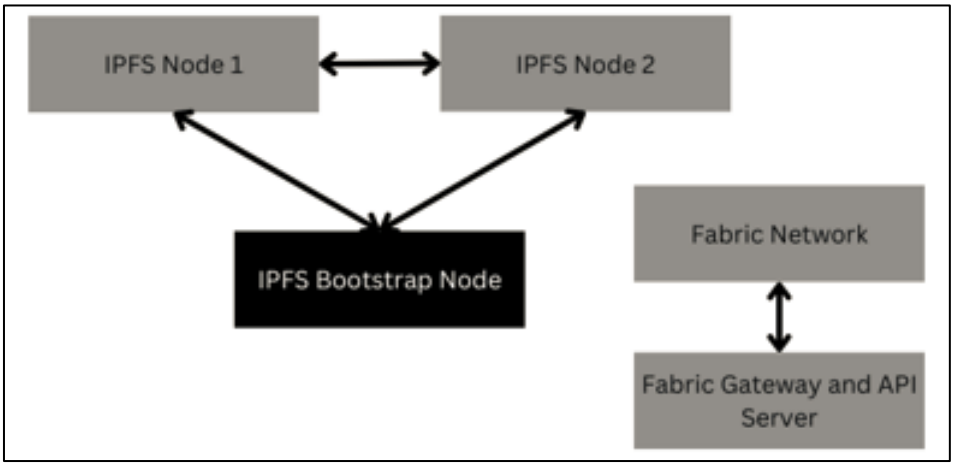


Figure 3: Architecture of Case Cloud Blockchain

## IV. METHODOLOGY

### *A. Evidence Archives – Evidence Collection Website*

A comprehensive and professional strategy is used in the development of this evidence collection website. The front-end implementation puts a focus on responsive design and makes use of Bootstrap's features to guarantee smooth adaptability to various screen sizes and devices. React libraries are used to provide form validation, assuring data integrity and good user experience. Users are guided by integrated error handling and feedback methods that deliver the proper information based on whether an activity was successful or unsuccessful. CSS classes and theming options in Bootstrap are used for front-end customizations. React Router makes it easy to navigate a website by building many pages or routes for various features.

The frontend interacts with backend APIs, utilizing React's fetch API or other HTTP client libraries to send requests and update the application state based on the received data. Testing and debugging techniques, including React Developer Tools, browser developer consoles, and unit testing frameworks like Jest, ensure optimal functionality and performance.

The application's backend depends on Microsoft Azure Blob, Google Firebase, NodeJS, ExpressJS, BodyParser, Axios, Cors, and Multer. Microsoft Azure Blob offers safe cloud-based storage for massive amounts of unstructured data, while Google Firebase delivers crucial tools and services for developing, testing, and deploying web apps. BodyParser makes

DOI: 10.17148/IJARCCE.2023.125258

it easier to handle and parse incoming HTTP requests, while NodeJS and ExpressJS enable server-side development. Multer makes file handling and data processing from multipart forms easier while Cors enables secure communication between web browsers and servers. Axios handles HTTP requests and responses.
In order to guarantee the application's best functionality and performance, testing and debugging techniques—both manual and automated—are used.

*B. Court Vault – Court Records Website*

The Court Vault application was developed using an approach that emphasizes professional implementation. The frontend uses a component-based modular architecture to encourage code reuse and maintainability. It makes use of Bootstrap's responsive design features to offer seamless device and screen size adaptation. Enhanced user experience and data integrity are ensured via form validation utilizing React frameworks. Users are guided by integrated error handling and feedback methods that provide success messages. To improve aesthetic appeal, we make use of Bootstrap's robust style and theming features. Seamless website navigation is made possible by React Router, and data input and retrieval are easily managed by connectivity with backend APIs.

For building, testing, and deploying web applications as well as for managing and storing massive amounts of data securely, the backend makes use of Google Firebase and Microsoft Azure Blob. For effective server-side development, libraries like Node.js, Express.js, and others used in Evidence Archives (*Section IV - A*) are used.

In order to guarantee the application's best functionality and performance, testing and debugging techniques—both manual and automated—are used.

*C. Case Cloud Blockchain –Blockchain based Evidence & Investigative Report Storage API Server*

A methodology that places a focus on professional implementation was used to develop the Case Cloud Blockchain. The Create Case, Query Case, Delete Case, and Update Case methods make up the core functionality of the APIs. The creation, retrieval, deletion, and modification of case records inside the ledger are all supported by these APIs. To handle the necessary data, each API uses path variables and JSON request bodies.

The smart contract that controls the system's business logic is called chaincode and was developed using the Go programming language. It receives API requests from the server and responds to them by running appropriate routines. The application is then given the results. The Hyperledger Fabric blockchain network relies on the fabric gateway, which makes it easier for client apps to submit transactions and request information about the ledger's current state. It gives clients connecting to the fabric network a single endpoint.

The creation of standalone web apps is made easier by the auto-configuration feature of the Java-based Spring Boot platform. The InterPlanetary File System (IPFS) permits distributed file storage and transfer by leveraging unique hashes for content addressing. Data pertaining to cases will hence be stored safely and independently.
In order to guarantee the best functionality and performance, testing and debugging techniques—both manual and automated—are used.

## V. IMPLEMENTATION

*(Blue = user-defined functions and Red = built-in functions)*

*A. Evidence Archives – Evidence Collection Website*

*1) Evidence Storage Algorithm*

```
// Use the function getContainerClient(containerName) to get the containerClient value

FUNCTION uploadEvidence (file, timeRange, place, description, name, address, contactNumber, email):

  // Get file paths
  fileExt = path.extname(file.originalname)

  // Give the blob being created a name
```

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023
DOI: 10.17148/IJARCCE.2023.125258

```
  blobName = "Your Blob Name" + fileExt

  // Setup an Azure Blob
  blockBlobClient = containerClient.getBlockBlobClient(blobName)

  // If optional fields are not filled, replace it with NA
  IF name is false THEN
    name = "NA"
  END IF
  IF address is false THEN
    address = "NA"
  END IF
  IF contactNumber is false THEN
    contactNumber = "NA"
  END IF
  IF email is false THEN
    email = "NA"
  END IF

  // Tag the user inputs as the Blob's metadata
  metadata = {
    timeRange: timeRange, place: place,
    description: description, name: name,
    address: address, contactNumber: contactNumber,
    email: email
  }

  // Upload the file and metadata into our Azure Blob
  blockBlobClient.upload(file.buffer, file.size, metadata)

END FUNCTION
```

*2) Law Enforcement Login Algorithm*

```
DEFINE handleLogin event handler
  prevent default form submission
  check if password and email fields are not empty

    IF not empty:

  // Call the built-in signInWithEmailAndPassword and pass the firebase object with your credentials, email ID and password
    Call signInWithEmailAndPassword(firebaseDetails, email, password):
        IF successful:
          redirect to "/search" page
        ELSE:
          show "Incorrect credentials" alert
```

*3) Evidence Search Algorithm*

```
// Use the function getContainerClient(containerName) to get the containerClient value

FUNCTION searchEvidence(containerClient, searchText):
  DECLARE an empty array called blobs

  // Iterate through all the available blobs
  FOR EACH blob IN containerClient.listBlobsFlat():
```

DOI: 10.17148/IJARCCE.2023.125258

```
// Assign the name of blobs to blobName
    SET blobName to blob.name

// Get the blob with the same name as in blobName
    SET blobClient to
      containerClient.getBlockBlobClient(
        blobName
      )

    SET blobUrl to blobClient.url

// Get the Blob's properties
    SET blobProperties to
      await blobClient.getProperties()

// Get the metadata from Blob's properties
    SET metadata to blobProperties.metadata

// Check if the search text is present in the Blob's name and if not look in the Blob's metadata for a match
    IF blobName includes searchText:
      ADD { name: blobName, url: blobUrl,
          metadata: metadata
        } to blobs
    ELSE:
      FOR EACH key, value IN metadata:
        IF value.toLowerCase() includes
          searchText.toLowerCase():
          ADD { name: blobName, url:
              blobUrl, metadata: metadata
            } to blobs
          BREAK the loop

  RETURN blobs
```

*B. Court Vault – Court Records Website*

*1) Court Documents Storage Algorithm*

```
// Use the function getContainerClient(containerName) to get the containerClient value

FUNCTION uploadCourtDocs(file, caseNum, station, FIR, act, section, caseType, petitioner, respondent, advocate,
court, state, dateFiled, dateListed, status):

        // Get file path
  fileExt = path.extname(file.originalname)

        // Give the blob being created a name
  blobName = "Your Blob Name" + fileExt

        // Setup an Azure Blob
  blockBlobClient = containerClient.getBlockBlobClient(blobName)

        // Tag the user inputs as the Blob's metadata
  metadata = {
    caseNum, station, FIR, act, section, caseType,
    petitioner, respondent, advocate, court, state,
    dateFiled, dateListed, status
  }
```

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023

DOI: 10.17148/IJARCCE.2023.125258

```
            // Upload the file and metadata into our Azure Blob
   blockBlobClient.upload(file.buffer, file.size, metadata)

END FUNCTION
```

*2) Court Officials' Login Algorithm*

```
DEFINE handleLogin event handler
   prevent default form submission
   check if password and email fields are not empty

     IF not empty:

     // Call the built-in signInWithEmailAndPassword and pass the firebase object with your credentials, email ID and password
     Call signInWithEmailAndPassword(firebaseDetails, email, password):
          IF successful:
             redirect to "/documentUpload " page

          ELSE:
             show "Incorrect credentials" alert
```

*3) Court Documents Search Algorithm*

```
// Use the function getContainerClient(containerName) to get the containerClient value

FUNCTION searchCourtDocs(containerClient, searchText):
   DECLARE an empty array called blobs

   // Iterate through all the available blobs
   FOR EACH blob IN containerClient.listBlobsFlat():

     // Assign the name of blobs to blobName
     SET blobName to blob.name

     // Get the blob with the same name as in blobName
     SET blobClient to
        containerClient.getBlockBlobClient(
          blobName
        )

     SET blobUrl to blobClient.url

     // Get the Blob's properties
     SET blobProperties to
        await blobClient.getProperties()

     // Get the metadata from Blob's properties
     SET metadata to blobProperties.metadata

     // Check if the search text is present in the Blob's name and if not look in the Blob's metadata for a match
     IF blobName includes searchText:
        ADD { name: blobName, url: blobUrl,
            metadata: metadata
          } to blobs

     ELSE:
        FOR EACH key, value IN metadata:
```

DOI: 10.17148/IJARCCE.2023.125258

```
            IF value.toLowerCase() includes
              searchText.toLowerCase():
               ADD { name: blobName, url:
                    blobUrl, metadata: metadata
                  } to blobs
               BREAK the loop

  RETURN blobs
END FUNCTION
```

*C. Case Cloud Blockchain – Blockchain based Evidence & Investigative Report Storage API Server*

*1) Chaincode (Smart Contract)*

*a. Init Ledger*

```
FOR EACH case IN the array of cases:
    MARSHAL the case into JSON format
    PUT the JSON data into the world state using the case ID as the key
RETURN nil
```

*b. Case Exists*

```
GET the case data from the world state using the given case ID
IF there's an error while getting the data:
  RETURN false and the error

IF the case data is null:
   RETURN false and no error

ELSE:
  RETURN true and no error
```

*c. Create Case*

```
IF the case already exists using the CaseExists function:
  IF it exists:
    RETURN an error saying the case already exists

ELSE:
   CREATE a new case with the given details

MARSHAL the new case into JSON format
PUT the JSON data into the world state using the case ID as the key

RETURN nil
```

*d. Query Case*

```
GET the case data from the world state using the given case ID
IF there's an error while getting the data:
  RETURN nil and the error

IF the case data is null:
  RETURN nil and an error saying the case doesn't exist

ELSE:
  UNMARSHAL the case JSON data into a Case struct
  RETURN it with no error
```

*e. Delete Case*

```
GET the case data from the world state using the given case ID
IF there's an error while getting the data:
   RETURN the error
```

  

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023
DOI: 10.17148/IJARCCE.2023.125258

```
IF the case data is null:
    RETURN an error saying the case doesn't exist

ELSE:
    DELETE the case data from the world state

RETURN nil
```

*f. Update Case IPFS Hash*

```
GET the case data from the world state using the given case ID
IF there's an error while getting the data:
    RETURN the error

IF the case data is null:
    RETURN an error saying the case doesn't exist

ELSE:
    UNMARSHAL the case JSON data into a Case struct

UPDATE the CaseIPFSHash field of the Case struct with the given IPFS hash
MARSHAL the updated Case struct into JSON format
PUT the JSON data into the world state using the case ID as the key

RETURN nil
```

*2) Spring Boot Application*

*a. Case Service Constructor*

```
FUNCTION CaseService():
   CREATE a new gRPC connection
   CREATE a new Gateway with identity, signer, and connection
   SET gRPC call options and deadlines
   CONNECT to the gateway
   GET the network and contract
   CALL initLedger()
```

*b. Functions Interacting with Ledger*

The functions in Case Service that interact with the ledger either submit or evaluate a transaction.

*c. Rest API Controller*

The Rest API Controller maps the different paths in the URL and specifies how and what type of data each path accepts. Then it calls the necessary Case Service functions.

*d. IPFS Private Cluster*

A three-node private IPFS cluster is created. The initial node acts as a bootstrap node and the other two as peers in the network. All three nodes are in a private network and the data cannot be accessed from outside the network. The nodes also run a cluster, meaning data uploaded at one node is also replicated at the other two nodes.

## VI. RESULTS AND DISCUSSION

*A. Evidence Archives – Evidence Collection Website*

The implementation of Evidence Archives demonstrates the successful completion of the research's objectives. The website provides a user-friendly platform for the public to upload evidence securely. The search functionality enables law enforcement officials to retrieve evidence based on specific criteria, facilitating efficient investigations. Integration with Microsoft Azure Blob and Google Firebase ensures secure evidence storage, reliable retrieval, and user authentication.

  

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023

DOI: 10.17148/IJARCCE.2023.125258

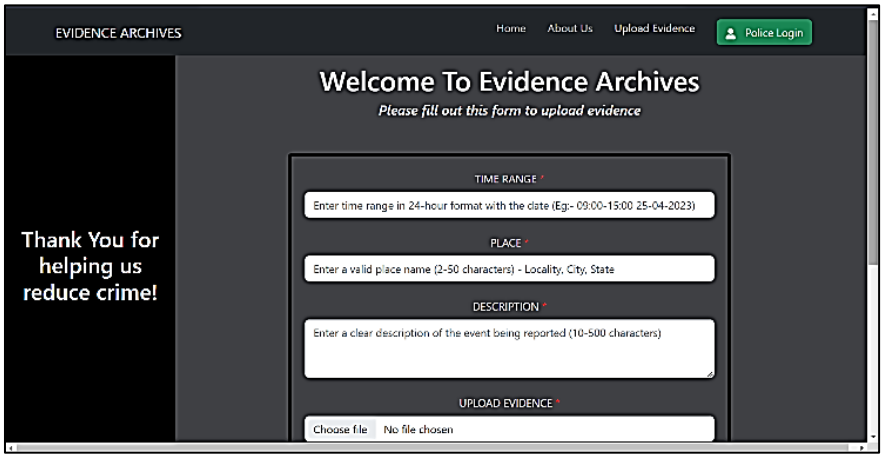


Figure 4: Evidence upload functionality of Evidence Archives

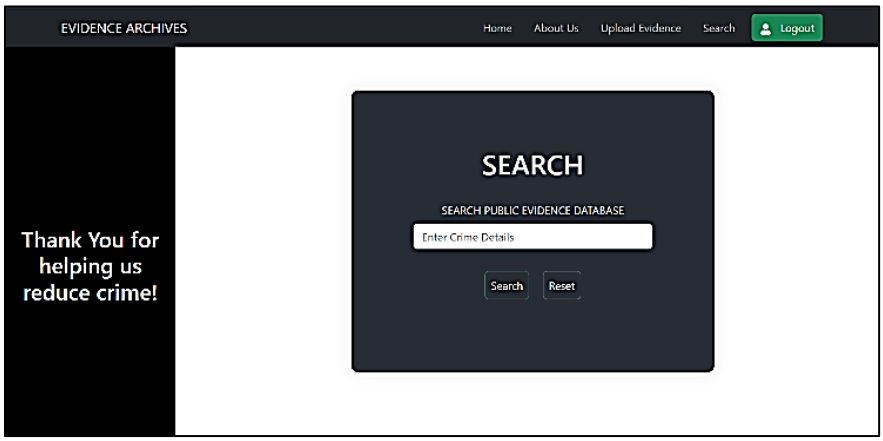


Figure 5: Evidence search functionality of Evidence Archives

*B. Court Vault – Court Records Website*

The implementation of Court Vault demonstrates the successful achievement of the research's objectives. The system provides a secure and user-friendly platform for court officials to upload and manage court documents, significantly reducing the administrative burden. The search and download functionalities enable the public to access relevant court documents conveniently. The integration of Google Firebase and Microsoft Azure Blob ensures robust user authentication, document storage and retrieval capabilities.

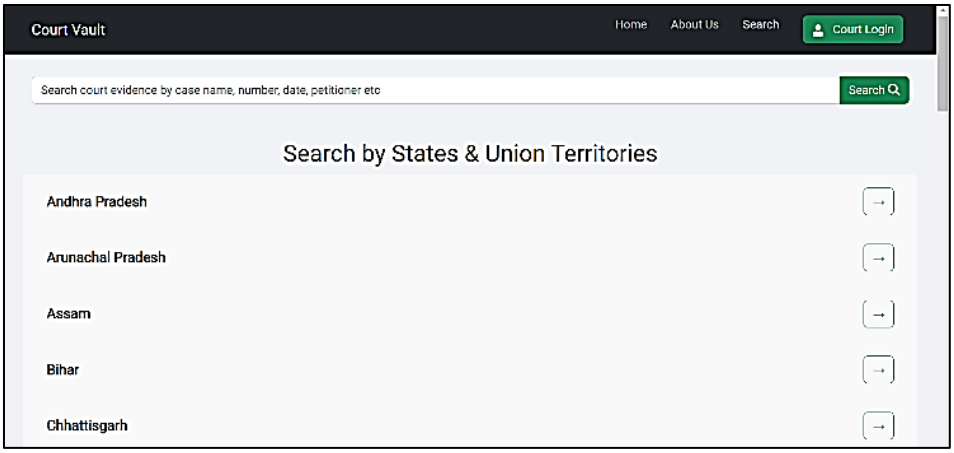


Figure 6: Keyword and State-based search functionality of Court Vault

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023

DOI: 10.17148/IJARCCE.2023.125258

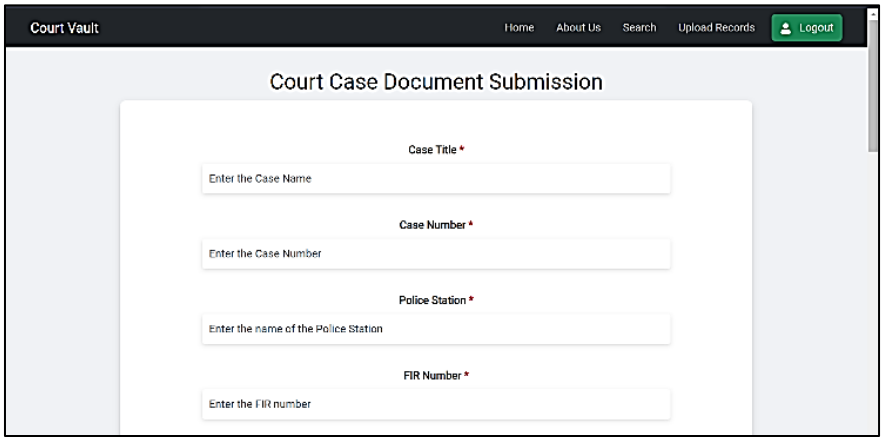


Figure 7: Court Documents and Verdicts upload functionality of Court Vault

*C. Case Cloud Blockchain –Blockchain based Evidence & Investigative Report Storage API Server*

Implementing the Case Cloud Blockchain with IPFS and Hyperledger Fabric blockchains has shown to be a major improvement in the secure storage of case evidence. The backends' integration with a Spring Boot application enabled an efficient and smooth user-system communication. The solution ensured that the evidence gathered by law enforcement authorities was stored securely and decentralized by utilizing IPFS and the Hyperledger Fabric blockchain. Using IPFS made it possible to store evidence records in a dispersed manner while yet maintaining redundancy and accessibility in the event of network outages. Another degree of security, immutability, and transparency was provided to the stored evidence through the connection with the Hyperledger Fabric blockchain. The system was effective in achieving its goal of giving court and law enforcement institutions access to the tamper-free evidence.


Figure 8: Hyperledger Fabric Network of Case Cloud Blockchain

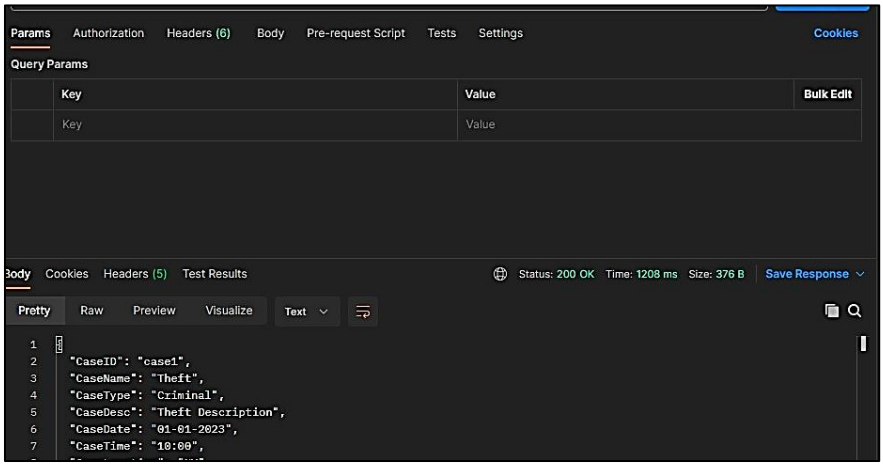


Figure 9: Interaction with Case Cloud Blockchain's API Server

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023
DOI: 10.17148/IJARCCE.2023.125258

Two websites—Evidence Archives and Court Vault, and an API server—Case Cloud Blockchain—were successfully developed and implemented as part of the research. These solutions addressed various facets of gathering evidence, accessing court documents, and safely storing evidence, respectively.

The results indicate that the websites and the server successfully achieved their goals by offering reliable access methods, secure storage architecture, and user-friendly interfaces. The research's success was largely due to the integration of technologies including React, NodeJS, Microsoft Azure Blob, IPFS, Hyperledger Fabric blockchain, Spring Boot, Google Firebase, and numerous more modules.

## VII. FUTURE WORKS

Future developments and enhancements are highly likely to result from the integration of public reported evidence collection and secure decentralized case document management via IPFS and Hyperledger Fabric Blockchain.

*Enhanced Data Security:* Constant research and development efforts should concentrate on enhancing the security safeguards of blockchain based storage, leveraging cutting-edge cryptographic methods and consensus mechanisms to guarantee immutable and tamper-proof storage of investigation reports and evidence documents.

*Scalability and Performance Optimization:* It is crucial to look at scalability solutions that can effectively manage increasing data storage and retrieval expectations as the user base of Evidence Archives, Court Vault, and Case Cloud Blockchain grows. It is important to continue performance optimization of the system in order to improve user experience and make it possible for users to easily obtain court rulings and case-related information.

*Interoperability and Integration:* Future work should focus on facilitating seamless interoperability between Case Cloud Blockchain, Court Vault, and Evidence Archives, allowing for easy data sharing and cooperation among diverse stakeholders in the legal ecosystem. To facilitate information exchange and economize on administrative tasks, integration with other pertinent platforms and systems should also be investigated.

*Usability and Accessibility Enhancements:* To make the platforms more accessible and usable, continued user research and feedback should be undertaken. To offer the best user experience for both law enforcement agencies, judicial systems and the general public, user-friendly interfaces, clear navigation, and thorough documentation should be developed.

*Continued Collaboration and Partnerships:* It is important to develop collaboration with industry stakeholders, technical specialists, and legal authorities in order to gather insightful information and confront new challenges. Together, the system can adapt to meet changing legal requirements and take use of new technologies to spur further innovation in areas such as blockchain-based solutions, evidence management, and court document management.

We must seek to actively refine the system and enhance its capabilities to ensure its effectiveness in serving law enforcement, legislature, judiciary and the general public.

## VIII. CONCLUSION

This research successfully created and deployed two interrelated websites and an API server — Evidence Archives, Court Vault, and Case Cloud Blockchain. As a result, the pursuit of justice is more transparent, approachable, and effective owing to these websites' facilitation of cooperation between law enforcement, legal institutions, and the general public.

The research has given robust and reliable solutions for evidence collection, court document management, and secure storage through the seamless integration of contemporary technologies, including React, NodeJS, Microsoft Azure Blob, Google Firebase, IPFS, and Hyperledger Fabric blockchain. The research's results enable the general people to actively participate in crime prevention and sleuthing while also guaranteeing that vital legal information is easily available to all parties engaged in the judicial process.

This research represents a significant advancement in leveraging technology for the improvement of law enforcement, judicial collaboration, and public participation in the administration of justice by combining user-friendly interfaces, secure authentication mechanisms, and advanced storage technologies.

  

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 5, May 2023
DOI: 10.17148/IJARCCE.2023.125258

## REFERENCES

[1] Huang, H. S., Chang, T. S., & Wu, J. Y. (2020, July). A secure file sharing system based on IPFS and blockchain. In Proceedings of the 2nd International Electronics Communication Conference (pp. 96-100).

[2] Madine, M. M., Salah, K., Jayaraman, R., Yaqoob, I., Al-Hammadi, Y., Ellahham, S., & Calyam, P. (2020). Fully decentralized multi-party consent management for secure sharing of patient health records. IEEE Access, 8, 225777-225791.

[3] Sathyaprakasan, R., Govindan, P., Alvi, S., Sadath, L., Philip, S., & Singh, N. (2021, March). An implementation of blockchain technology in forensic evidence management. In 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) (pp. 208-212). IEEE.

[4] Guo, J., Wei, X., Zhang, Y., Ma, J., Gao, H., Wang, L., & Liu, Z. (2022). Antitampering scheme of evidence transfer information in judicial system based on blockchain. Security and Communication Networks, 2022.

[5] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Yellick, J. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys conference (pp. 1-15).

  