Artifact

Eduardo Oliveira

March 3, 2025

1 Open Science Platform

1.1 Overview

Traditional centralized systems often exhibit data silos, limited verifiability, and susceptibility to manipulation, impeding the openness and reliability of scientific practices. The decentralized model introduced in this work is designed to mitigate these challenges by enabling efficient data sharing, fostering collaboration, and enhancing the validation of research outputs, thereby strengthening reproducibility and transparency.

This chapter details the design and implementation of the Open Science Platform, a decentralized system that integrates blockchain, IPFS, and smart contracts to improve research reproducibility. By leveraging immutable records and decentralized storage, the platform ensures transparent and verifiable research artifact management. Additionally, off-chain services are incorporated to facilitate file indexing, metadata extraction, and search functionality. The proposed platform aligns with Open Science principles by providing verifiable and persistently traceable access to research artifacts.

1.2 Technology Stack

The Open Science Platform is developed using a hybrid architecture that combines decentralized and off-chain technologies to ensure secure, traceable, and efficient data management.

1.3 Decentralized Components

- **Jupyter Notebooks (Python):** Serves as the front-end interface, enabling the automation and visualization of execution steps. Blockchain interactions are facilitated through the Iroha v1 Python library, while communication with the IPFS network is managed via the HTTPS client library.
- **InterPlanetary File System (IPFS):** Provides decentralized, tamper-proof storage for research artifacts and metadata, ensuring persistent and verifiable access to shared data.

1.4 Off-Chain Components

- **Jupyter Notebooks (Python):** Powers the front-end interface, facilitating the automation and display of the execution steps. Blockchain interactions are managed via the Iroha v1 Python library, while communication with the IPFS network is handled through the HTTPS client library.
- **Apache Tika:** Extracts metadata from uploaded files, enhancing artifact organization and searchability.
- **Whoosh:** Facilitates efficient indexing and keyword-based search for stored artifacts.

1.5 Platform Operations

The platform supports a set of core operations that regulate user interactions with projects and data management.

1.6 User Self-Enrollment

Users can register on the platform by submitting a private key that complies with the ED25519 and SHA-3 standards. During registration, they provide identity details, including full name, institution, email, ORCID, and role. This information is structured as a JSON object and uploaded to IPFS, with the corresponding Content Identifier (CID) recorded on the blockchain.

1.7 Project Registration

Users can register a project by specifying a descriptive name, an abstract, relevant keywords, start and end dates, funding agency, and location. Upon creation, a blockchain account is automatically assigned to the project.

1.8 Linking User and Project Accounts

Once both user and project accounts are created, the system updates their attributes to establish a bidirectional association. This ensures that querying a user account reveals linked project accounts, and vice versa, facilitating traceability and efficient project management.

1.9 Artifact Management

1.9.1 File Upload

Users can upload research artifacts, including papers, datasets, and images. Each file is stored on IPFS, generating a unique Content Identifier (CID) that ensures traceability and integrity. The CID is then recorded on the blockchain attributes of the project , establishing a verifiable reference to the artifact.

1.9.2 Metadata Extraction and Storage

During the upload process, Apache Tika automatically extracts metadata from the file, including information such as title, author, creation date, and file format. The extracted metadata is structured in JSON format, stored on IPFS, and its CID is linked to the corresponding project account on the blockchain, ensuring metadata provenance and accessibility.

1.9.3 Indexing and Search

To facilitate efficient retrieval, the system indexes both files and metadata. Users can perform keyword-based searches to locate relevant research artifacts, with search results displaying metadata details, including file descriptions, authorship, and retrieval information.

1.10 Verification and Access

1.10.1 File Validation

To ensure data integrity and authenticity, the platform verifies whether the CID of a file stored on IPFS matches the CID recorded on the blockchain. A discrepancy between these identifiers signals potential tampering or corruption, prompting an integrity check.

1.10.2 File Download

Users can retrieve and download validated files from IPFS to their local systems. The platform ensures that only authenticated users can access the files, preserving controlled dissemination while maintaining Open Science principles.

2 Security and Integrity Considerations

This section examines the security mechanisms implemented to preserve the authenticity and integrity of stored research artifacts. It discusses cryptographic hashing for ensuring data integrity, blockchain immutability to prevent unauthorized modifications, and IPFS redundancy for enhanced availability and fault tolerance. Additionally, it explores potential attack vectors, including unauthorized access, metadata manipulation, and denial-of-service threats, alongside mitigation strategies to safeguard the platform's reliability.

3 Conclusion

This chapter has detailed the technological underpinnings and operational workflows of the Open Science Platform. By leveraging blockchain, IPFS, and smart contracts alongside off-chain indexing and metadata extraction, the platform

enhances research reproducibility through immutable data storage, verifiable metadata, and automated validation mechanisms.

3.1 Technology Stack

The Open Science platform is built upon a robust technical foundation, comprising:

- Hyperledger Iroha v1 Blockchain: The core infrastructure for account management and transaction recording and business rules enforcement through Smart Contracts ensuring secure and transparent data exchange.
- IPFS (InterPlanetary File System): The decentralized storage for project artifacts and metadata, guaranteeing tamper-proof and persistent access to shared information.

Aside from the decentralized technologies above, the platform also relies on the following off-chain, centralized components:

- The platform's front-end interface utilizes Jupyter Notebooks in Python to automate and present the execution steps of its activities. Blockchain interactions are facilitated through the Iroha v1 Python library, while communication with the IPFS network is handled via an HTTPS client library.
- Apache Tika: Utilized for extracting file metadata, enhancing the platform's ability to manage and describe artifact content.
- Woosh: For efficient indexing and search capabilities for artifacts stored on the platform.

3.2 Operations

The Open Science platform is comprised of the following operations:

- User self-enrollment: Any user can self-enroll in the platform, with only a set of public/private keys conformant with standard ED25519 and SHA-3 as a requirement. The user must provide identity information such as full name, institution, email, ORCID, and role (e.g., author, publisher, reviewer). A JSON formatted representation of the user metadata is stored on IPFS, and the generated CID (Content Identifier) is stored in the blockchain of the user account.
- Project registering: Once enrolled, a user can register a project by providing a descriptive name for the project, an abstract summarizing the scope and goals of the project, keywords related to the project, start and end dates, funding agency, and location. The system automatically assigns an account in the blockchain for the project and links the user as the project

owner bi-directionally. A JSON formatted representation of the project metadata is stored on IPFS, and the generated CID (Content Identifier) is stored in the blockchain of the project account.

- File upload: Users can upload artifacts such as papers, reports, images, datasets, etc., from their local machine to the platform. These files are stored securely on IPFS. A unique identifier (CID - Content Identifier) is generated for each artifact uploaded, which is used to track data provenance.
- Metadata extraction: In tandem with the upload process, metadata information from each uploaded file is extracted.
- Metadata upload: A JSON formatted representation of the metadata extracted from a file is uploaded on IPFS. The generated CID (Content Identifier) is stored in the blockchain of the project account.
- File indexing: The system indexes the files and their corresponding metadata, enabling efficient search functionality for users.
- Keyword search: Any user can perform searches based on keywords. Positive occurrences are displayed along with metadata information from the files.
- File validation: The file validation is performed. A file is considered valid if the CID in IPFS and the CID stored on the blockchain match exactly.
- File download: The valid file is downloaded to the local file system.

3.3 Data Model

The data model that supports the platform is comprised of two main classes User and Project. The User class contains attributes for user identity information, while the Project class contains attributes for project metadata. A many-to-many relationship exists between Users and Projects where, a single user can be associated with multiple projects.

To describe the attributes of each entity in the data model, three main ontologies were considered: FOAF (Friend of a Friend), Dublin Core and Schema.org. These standard vocabularies provide a common language for describing metadata information and can potentially easy the integration with other systems adopting W3C standards for semantic Web, like knowledge graphs, for instance.

3.4 Entity-relationship model

blokchains, smart contracts are verifiable piece of code

3.5 Ontologies

To describe the attributes of each entity in the data model, three main ontologies were considered: FOAF (Friend of a Friend), Dublin Core and Schema.org. These standard vocabularies provide a common language for describing metadata information and can potentially easy the integration with other systems adopting W3C standards for semantic Web, like knowledge graphs, for instance.

3.6 Entity-relationship model

blokchains, smart contracts are verifiable piece of code

3.7 Metadata

Metadata in the context of the platform has a two fold approach. The first is related to the identity of an user account, holding key value pairs of attributes related to the user such as full name, email, institution, ORCID.

```
{
    "@context": {
        "schema": "http://schema.org/",
        "foaf": "http://xmlns.com/foaf/0.1/"
    },
    "@graph": [
        {
            "@type": "foaf:Person",
            "foaf:name": "Jolly Noether",
            "foaf:mbox": "jolly_noether@email.com",
            "foaf:organization": {
                "@type": "foaf:Organization",
                "foaf:name": "Morris College"
            },
            "schema:identifier": {
                "@type": "PropertyValue",
                "propertyID": "ORCID",
                "value": "9833-6461-2701-X"
            },
            "foaf:holdsAccount": {
                "schema:identifier": "jolly_noether@test",
                "schema:roleName": "author",
                "schema:publicKey": <publick ey value>},
            "schema:linked_project": "10278@test"
        }
    ]
}
```

3.8 Smart Contract

The platform deploys standard Ethereum EVM contracts in Solidity for account creation and detail setting. These contracts are deployed through the Iroha v1 Python Library.

3.9 Benefits

The Open Science platform offers numerous benefits for researchers and members of the scientific community, including:

- Secure data sharing: By utilizing blockchain technology and IPFS, the platform ensures tamper-proof data exchange.
- Transparent data management: The use of smart contracts and decentralized storage guarantees transparency in data access and modification history.
- Collaborative research environment: The platform enables researchers to collaborate on projects, share artifacts and results, and track progress.

3.10 Challenges

The Open Science platform faces several challenges, including:

- Scalability: As the number of users increases, the platform needs to be able to handle a growing amount of data and transactions efficiently.
- Interoperability: Ensuring seamless integration with existing research platforms and tools is crucial for widespread adoption.
- User Adoption: Educating researchers about the benefits of decentralized technologies and the Open Science platform can be an uphill battle.

3.11 Future Work

The Open Science platform has several areas for future development, including:

- Integration with existing research platforms: Collaborations with established research platforms to expand the platform's reach and user base.
- Enhanced security measures: Implementing additional security protocols to protect against potential threats and maintain the integrity of shared information.
- User interface improvements: Enhancing the web interface to make it more user-friendly and accessible for researchers from diverse backgrounds.

4 Conclusion

The Open Science platform is a comprehensive solution for secure, transparent, traceable, and tamper-proof data sharing and collaboration. By leveraging decentralized technologies, the platform empowers researchers to share project artifacts and data in a reliable and trustworthy manner.