18 Digital Rights Management & Sovereignty (DRM)

18.1 Digital Rights Management & Sovereignty (DRM)

Owner(s): ATC
DOA Task: T3.3
Tier: 3 (Main), Also 2
Nature: Optional/System
Result: K3.3

This task will address the issue of data rights management via a blockchain-based scheme for DRM, which will provide trusted and high-level credible content protection (this can also be used to track who has access/ed the data). This will be based on ATC's blockchain-based technical approach and will be extended to include a hybrid on-chain and off-chain system to store and manage the rights of digital assets. The result will be enhanced DRM for stakeholders with non-repudiable audit trails.

18.1.1 Introduction

Purpose: To enhance the management and security of digital asset transactions through a robust blockchain-based Data Rights Management (DRM) system. It is designed to perform critical functions, including the notarization, tracking, and validation of all data rights transactions both within individual Dataspaces and across multiple participating Dataspaces

Description: The Blockchain-based Data Rights Management module (based on the IDSA clearing house module specifications) will be a two-fold solution that will provide both inter (between Dataspaces) and in (within Dataspaces) connectivity for data-sharing between Data providers and Data consumers of the participating Dataspaces. Leveraging blockchain technology, the initiative will incorporate a hybrid model of on-chain and off-chain mechanisms, enabling robust and scalable management of digital content rights. This approach ensures trusted, high-level content protection and facilitates precise tracking of access to data. The result will be a strengthened DRM framework that provides stakeholders with clear, non-repudiable audit trails, enhancing trust and compliance across digital transactions.

18.1.1.1 Big Picture

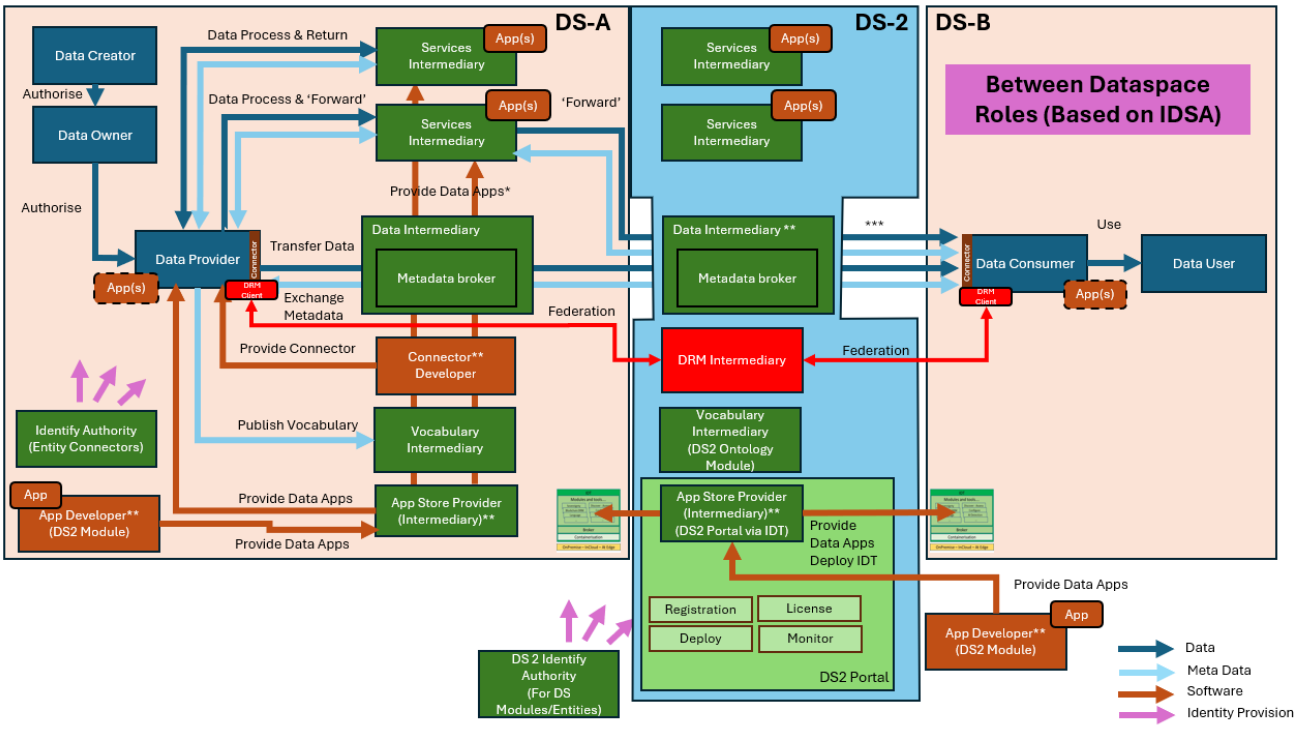
DRM can be used by a participant and between participants who themselves could be within the same Dataspaces or different ones:

- **Inter-Dataspace:** The module will function as a blockchain network within the DS2 SW platform, acting as a trusted intermediary between Dataspaces. This setup will enable DRM clients' blockchain networks to connect and participate in a federated environment for the **inter-connection** of participants from two Dataspaces (Data provider- Data consumer) as shown in the diagram below: DSA Provider IDT will connect with DSB Consumer IDT utilizing the optional Blockchain-based DRM. The DRM Module will interface locally through the "DRM Client" modules which is an optional part of IDT. These will provide all the trustworthy and reliable information that

both the producer and the consumer need on their respective ends. This includes providing immutable, verifiable logs that are more secure against tampering and fraud compared to traditional logging systems. Blockchain's decentralized nature ensures that no single entity can alter past transactions, thus increasing trust among participants. Therefore, the implementation of the DRM Module ensures a robust, transparent, and secure environment for data transactions across different Dataspaces, enabling both providers and consumers to engage with confidence in the system's ability to protect and monitor their data and transactions effectively.

- **In-Dataspace** (within): The DRM Client can also be used by Dataspace participants that are based within the same Dataspace, say DSA IDT (of the diagram above). In this case the connector will be fully operational when both parties (DSA1 and DSA2) have installed/deployed the Blockchain-based DRM. In this case, the connector will be able to log, monitor etc all transactions/ actions made between the two participants. This functionality will be achieved using the autonomous blockchain network provided by the DRM Client, eliminating the need for a core module to manage these processes. This is because, both parties have already accepted the policies/guidelines/ rules of the DS they are hosted and there is no need to have the assistance of the DRM module as in the case of the inter-connection between two DSs hosted by different DS owners/ hosts.

The big picture follows with the fit to the IDSA architecture shown in the subsequent diagram:

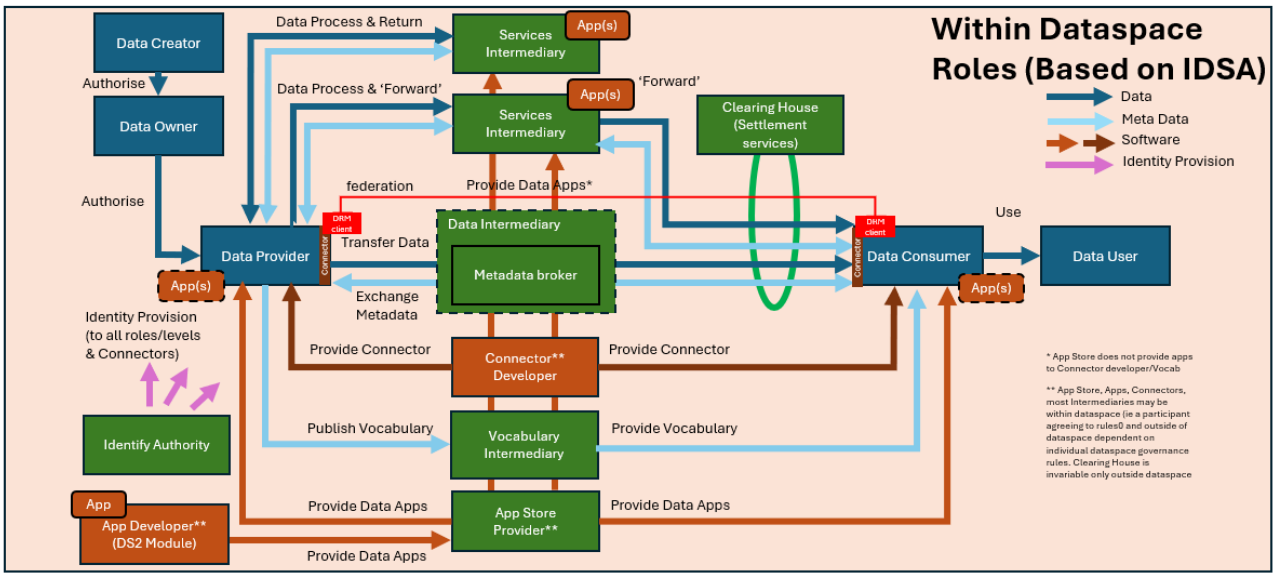


Where	Status
Within a single Dataspace	DS2 DRM can be deployed by participants within a single Dataspace to enable high-level DRM protection using blockchain technology. This is achieved by deploying the DRM Client and connecting through the DRM Core as an intermediary. DRM client can act also as a standalone blockchain network, they can synergy, and enable DRM protection without intermediary as well.

Deployed and used by a single participant to enable the participant in either an In-Data space or Inter- Data space scenario Within a single Dataspace	The DRM Client can function as an autonomous blockchain network to enable this functionality. Single participant can use the DRM client to log all operation into blockchain
Across Dataspaces without Service Intermediary	N/A
Across Dataspace with Intermediary	The DRM Intermediary serves as a central intermediary, facilitating the connection of DRM clients' blockchain networks in a federated environment enabling logs of all transactions across dataspaces. Hosted on DS2, a trusted environment, the DRM Intermediary eliminates the need for supplementary agreements between dataspaces.
Other Comments	N/A

18.1.1.2 Within a single Dataspace (where applicable)

DS2 DRM leverages blockchain technology to provide high-level DRM protection. Transaction logs, along with DRM and policy records, are securely stored on the blockchain, ensuring they are immutable. This not only enhances security but also increases transparency and reliability between the dataspace and its consumers. This can be achieved by deploying DRM client consumer and on provider side and by configuring federation between those clients. The DS2 DRM app will log transaction actions between consumer and provider, data rights, and record policy events. This information will be accessible to both parties through a user-friendly interface.



18.1.1.3 Deployed and used by a single participant

DS2 DRM can be utilized by individual participants. The DRM client functions as an autonomous blockchain network, logging all transactions, DRM, and policy records directly onto the blockchain, ensuring they are immutable. Participants can access and view this information through a user interface provided by the DS2 DRM. This can be achieved by deploying DRM client on participant side. Across Dataspaces without an Intermediary (where applicable)

18.1.1.4 Across Dataspace with Intermediary (where applicable)

N/A

18.1.1.5 Across Dataspace with Intermediary (where applicable)

The DRM Intermediary serves as a central intermediary, facilitating the connection of DRM clients' blockchain networks in a federated environment. Across Dataspaces, the DRM Intermediary offers a comprehensive suite of functionalities that bolster security, transparency, and reliability. As a central intermediary on the DS2 S/W Platform, it enables DRM clients to connect their blockchain networks in a federated environment, facilitating seamless interactions between participants. Key features include the provision of immutable, verifiable logs that significantly enhance security against tampering and fraud, surpassing traditional logging systems. The DS2 DRM app will log transaction actions between consumer and provider, data rights, and record policy events. This information will be accessible to both parties through a user-friendly interface.

18.1.2 Component Definition

The figure below represents the internal structure, sub-components, DS2 module interfaces, and primary other interfaces of the DRM module.

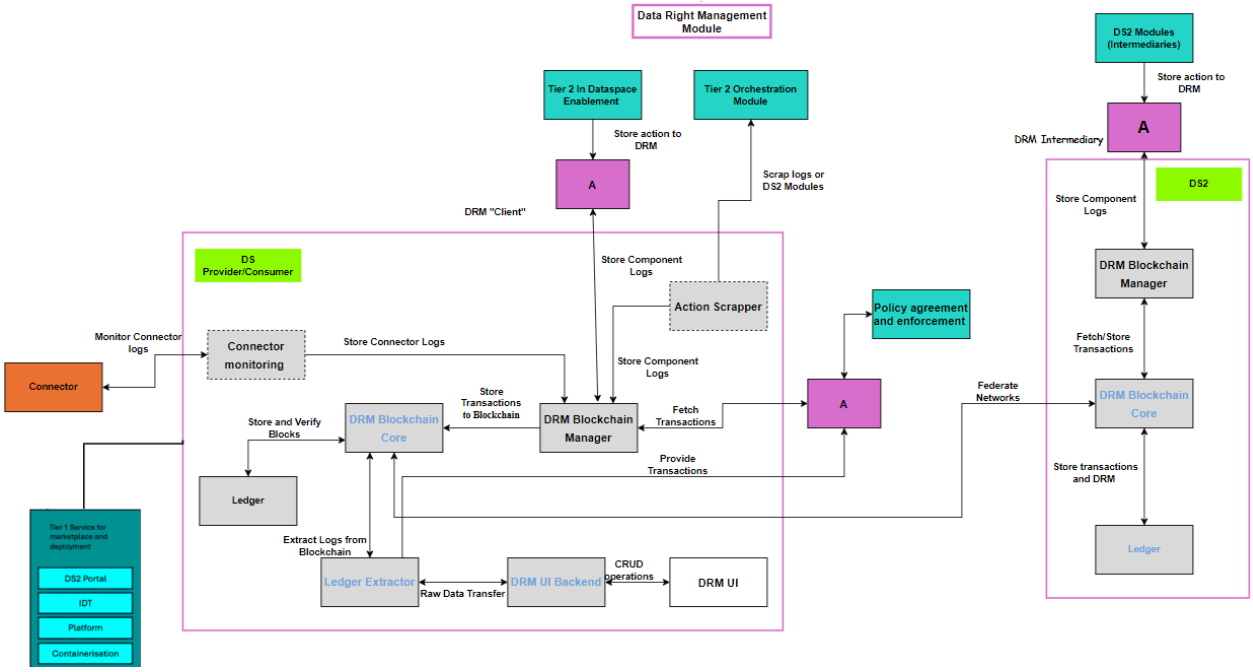


Figure 1: Schema for the Module

These modules have the following subcomponent and other functions:

- **DRM UI:** This subcomponent is a dashboard that allows users to view logs and transactions occurring between or across dataspaces. Users can easily access detailed records of their interactions with other partners. This dashboard interacts with the DRM UI Backend subcomponent, which performs the necessary transformations of data extracted from the ledger extractor to be appropriately displayed on the dashboard and supports all necessary CRUD operations.
- **DRM UI Backend:** This subcomponent retrieves log information extracted from the ledger and performs additional transformations to make the data viewable on the dashboard. It interacts with the Ledger Extractor to access the requested data and communicates with the DRM UI to display the transformed data to users.
- **Ledger Extractor:** This subcomponent securely extracts data from the ledger, converts it into a readable format, and forwards it to other services that require access to ledger information. Additionally, it records these actions on the blockchain to ensure transparency and traceability. It securely interacts with the DRM Blockchain Core to retrieve the necessary logs that are stored to the Blockchain's ledger, DRM UI Backend to provide information that need to be viewed in UI and with Policy Agreement and Enforcement through API to provide information that is needed for the app.
- **Connector monitoring:** This subcomponent is designed to retrieve logs from the connector and forward them to the blockchain for secure storage on the ledger. It interacts with the DRM Blockchain Manager to ensure that these logs are effectively recorded in the ledger. It is compatible with EDC connector and IDSA connector. This component is optional, allowing participants to use it solely for DRM to log transactions through supported APIs or restrict its use to DS2 tools if preferred.
- **DRM Blockchain Core:** This subcomponent is the core functionality of the blockchain incorporating essential components to initiate the blockchain network. It provides the functionality of storing immutable, verify information and encrypt information to the ledger. It interacts directly with the ledger to store information securely and works in conjunction with the Blockchain Node Manager to store additional logs provided by other DS2 modules.
- **Ledger:** The ledger is the central subcomponent of blockchain networks, securely storing all information in an immutable format. It interacts with the Blockchain Node subcomponent to ensure seamless integration and data management.
- **DRM Blockchain Manager:** The subcomponent is responsible for fetching, transforming, and storing information to and from the blockchain from modules which do not have a connector. Additionally, it is responsible for transforming incoming data logs into the correct schema so they can be stored on the blockchain, and ensuring they are placed correctly within the blockchain's ledger. Furthermore, it interacts with the DS2 Policy Agreement and Enforcement Module via API to access and provide logs essential for its operations. Additionally, it interacts with other DS2 modules, that interact with data, via API to store logs about data process
- **Action Scrapper Subcomponent:** The subcomponent is responsible for collecting all logs exported by the DS2 modules, utilizing a (Prometheus-like) exporter module that is deployed within the orchestration system. In this setup, the DS2 services report to the exporter module instead of the DRM API. This method offers a more efficient and streamlined approach for log collection.
- **Services and APIs:**
  - **Other External (non DS2) Modules/Services:**
    - **DS Connector:** Connector will provide logs to the connector monitoring subcomponent.

- **Tier 3 DS Modules/Service**
  - DS2 Policy Agreement and Enforcement Module.
- **Tier 2 DS Modules/Services**
  - Any DS2 Module that performs data manipulation eg Transformation module should log their action
- **Tier 1 Service Stack for Marketplace and deployment** and **API:** The DRM Client will be implemented as generically described elsewhere in this document. Exceptions: The Platform will only be needed for inter-participant service orchestrations were used
- **DRM APIs:**
  - Tier 2 DS2 Modules API: API will be used to store logs and transactions from other DS2 tools.
  - DS2 Intermediates Modules API: API will be used to store logs and transactions from other DS2 intermediates services.
  - Policy Agreement and Enforcement API: API will be used by DS2 Policy Agreement and Enforcement Module to extract the necessary information for enabling policy enforcement and receiving updates on policy enforcement status, indicating whether policies are respected or rejected..

18.1.3 Technical Foundations and Background

ATC will apply its blockchain expertise to enhance Digital Rights Management, ensuring data transparency and immutability which secures records against alterations and unauthorized tampering. The system allows only authorized parties to manage and access digital assets through a permissioned network, enhancing overall security. It also supports fine-grained access control for detailed data rights management and robust audit trails for non-repudiable records of all transactions, improving compliance and accountability. Additionally, the platform utilizes a hybrid system that integrates both on-chain and off-chain data management, optimizing storage and scalability to meet diverse stakeholder needs in rights management.

Subcomponent/Component	Owner	License
Hyperledger Fabric framework	The Linux Foundation	Apache 2.0

18.1.4 Interaction of the Component

The following table specifies the primary input/output controls/data to blocks which are not part of the module

With Module/Feature	Receives From/Gives To	What
Connector	Receives From	Receive logs of Dataspace's transactions
Tier 2	Receives From	Receive logs of other DS2 tools
Policy Enforcement Module	Gives To	Logs and transactions that are necessary for policy enforcement
Policy Enforcement Module	Receives From	Policy enforcement status

DS2 Modules	Receives From	Logs and transactions from other DS2 tools

18.1.5 Technical Risks

Risk	Description	Contingency Plan
Complex Setup	DRM module architecture is complex, involving multiple components such as peers and certificate authorities. This complexity can lead to challenges in setup	Carefully plan the network architecture, remove unnecessary components, maintain a detailed documentation and make use of automated deployment tools.
Scalability Concerns	Performance can be impacted as the network scales in terms of transaction volume and number of nodes. This could lead to bottlenecks, especially in networks that require high throughput and low latency.	Carefully plan the network architecture, including the appropriate number of nodes and channels, to optimize performance and scalability
Blockchain federation	Node communication between separate networks is challenging.	Need further investigation and the developing of additional components to support the federation
Integration with dataspace connectors	The DRM module needs to be integrated with the Connector to manage logs and DRM information effectively. Each connector has its own technical requirements, and we must ensure that all integrations comply with applicable regulations.	To ensure the successful integration of the DRM module with each Connector, a detailed investigation of the technical aspects of each connector is necessary

18.1.6 Security

Security Issue	Description	Need
Inter-participant trust	Unknown actor join a DRM Intermediary blockchain network and have access on data	strong identity management, strict access controls, and enforce robust governance and monitoring practices across all network participants
In-Dataspace	There is no specific DS security risk	N/A

18.1.7 Data Governance

Data Governance Issue	Description	Need
Handling sensitive information	Logs, DRM content and logs of transactions represent confidential information	Secure data transfer of the logs, DRM and transactions logs. Encrypt sensitive information.
Handling of personal data	This component is not set up to deal with the monitoring of personal data.	N/A

18.1.8 Requirements and Functionality

This module will be used in the following usecases:

- City Scape ✓
- Green Deal ✓
- Agriculture ✓
- Inter-Sector TBD

Their requirements and functions/extensions to achieve them relative to this module, specifically extracted from the use case are as per the table below noting that in many cases further discussion might need to take place between pilot partner and module partner to determine if a fit or the scope of the precise fit.

In respect to all use cases, the DS2 DRM module is classified as an Optional Module but also a System one. It is “System” in that it is not part of a user scenario in that it can simply be optionally used to maintain notarised logs from other modules which are used within any use case. As such it does not appear explicitly in any user scenarios which are connected with datasharing vs configuring for datasharing.

WHERE	WHAT	WHY	Run/Design Time	Priority
	Use Case 1: City Scape			
N/A			R&D	S
	Use Case 2: Green Deal			
N/A			R&D	M
	Use Case 3: Precision Agriculture			
N/A			R&D	C

18.1.9 Workflows

The following sub-sections describe the sequence diagrams of the DRM module.

18.1.9.1 Record Connector’s Transaction and logs

This feature allows to record all transaction and logs connector provides and store them into blockchain network.

The main steps/functionalities are as follows:

- Fetch Transactions, logs and ownership information
- Store information to the blockchain

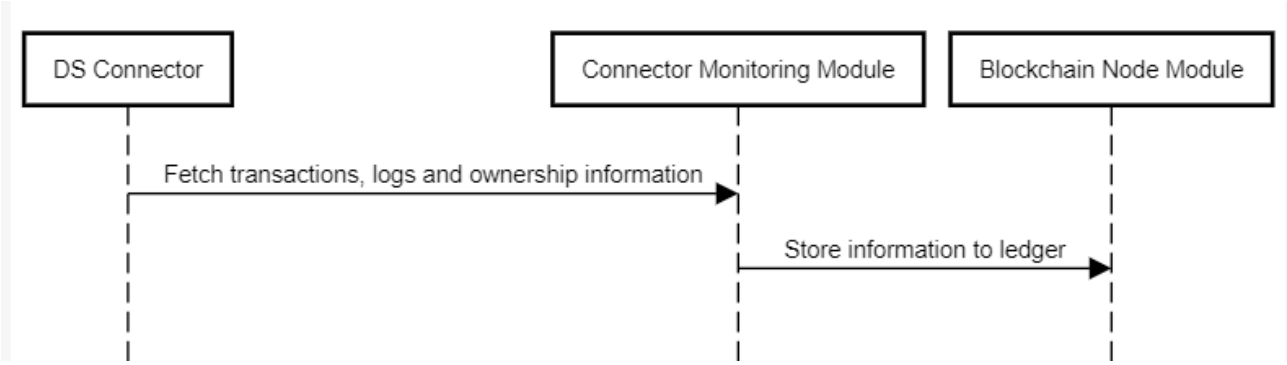


Figure 2: Record Connector's Transaction and logs

18.1.9.2 Policy enforcement monitoring

This feature records details about policy enforcement activities, documenting which policies were enforced and which were denied.

The main steps/functionalities are as follows:

- Send/Fetch information from policy enforcement module
- Forward information to operation service module
- Store information to blockchain network

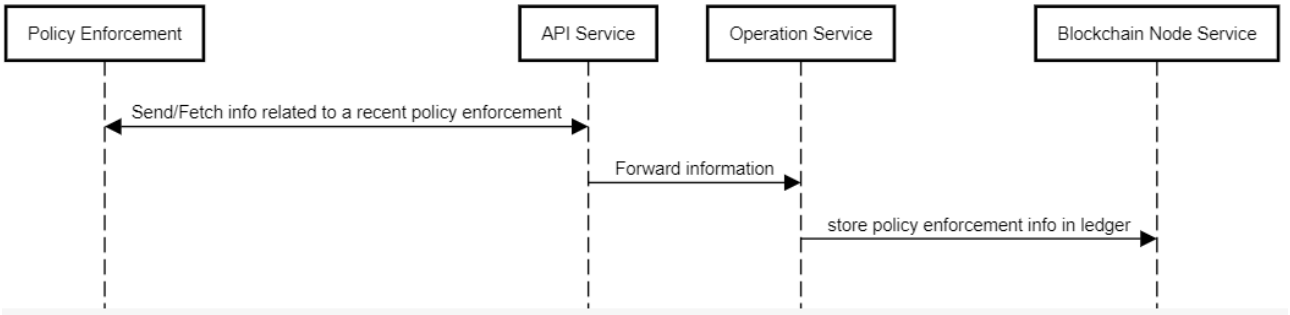


Figure 3: Policy enforcement monitoring

18.1.9.3 DS2 Modules Monitoring

This feature allows DS2 module to record logs and transaction about data into the blockchain network.

The main steps/functionalities are as follows:

- Send/Fetch information from DS2 Modules
- Forward information to operation service module
- Store information to blockchain network

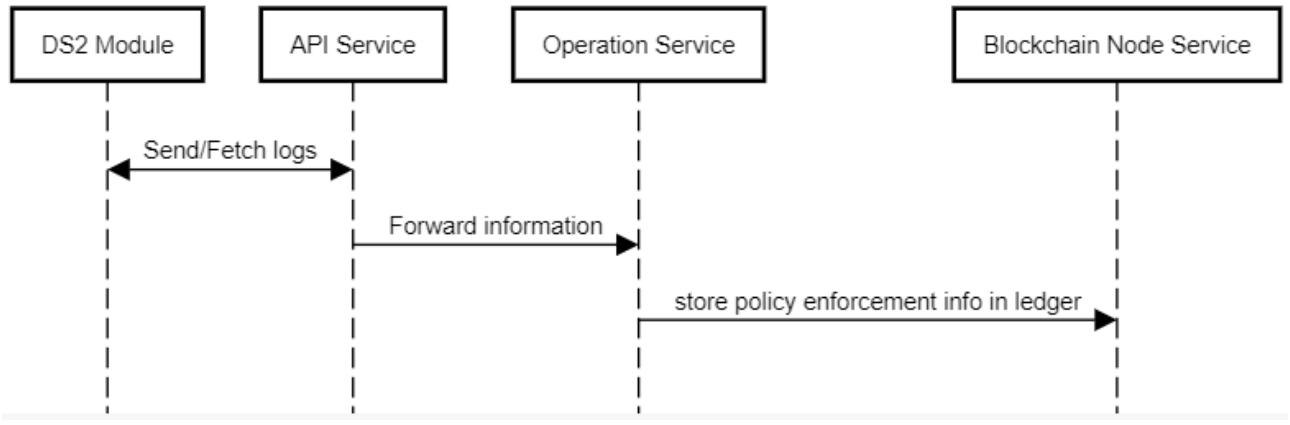


Figure 4: DS2 Modules Monitoring

18.1.9.4 User Interface

This feature allows DS2 participants to view all logs, transactions and DRM information into a friendly UI Dashboard.

The main steps/functionalities are as follows:

- Send/Fetch information from DS2 Modules
- Forward information to operation service module
- Store information to blockchain network

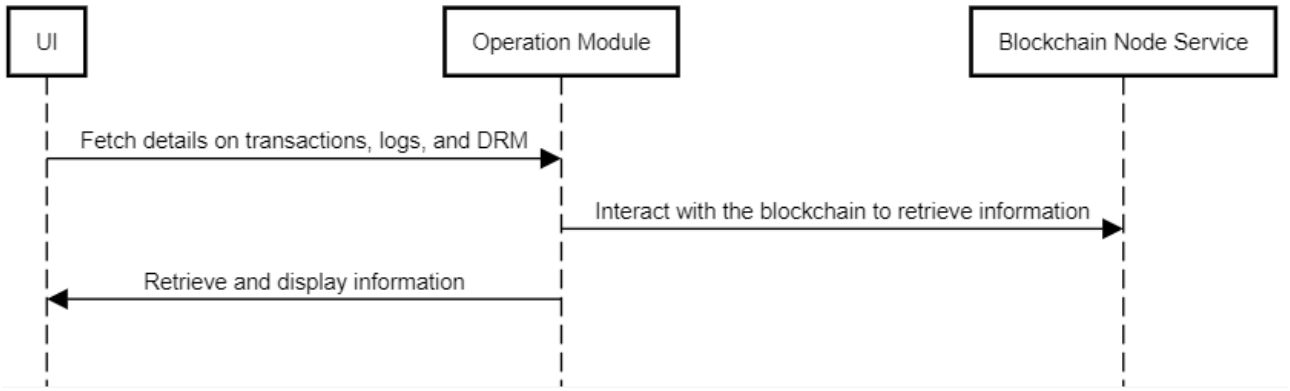


Figure 5: User Interface

18.1.10 Role, Resourcing, and Milestones

Sub-component	Main Activity	M18	M24	M30	M36
DRM Blockchain Core	The development and configuration of the blockchain network				
DRM Blockchain Core	Development of smart Contracts and the business logic for the ledger				
DRM Blockchain Manager	Application that interacts with the deployed blockchain network (DRM Blockchain Core), using the Fabric Gateway Service to invoke a smart contract, which queries and updates the ledger.				
Connector Monitoring	Develop an app to integrate with the connector and retrieve the necessary logs.				
Connector Monitoring	Process and store logs into blockchain's ledger				
Ledger Extractor	This sub component will extract and transform information from the blockchain's ledger.				
User Interface	The development of the User Interface from where the user can see the content of the ledger they have.				
DRM User Interface Backend	This subcomponent will support all CRUD operations that User Interface need				
Table Total/DOA Task Total/Resilience	**Comments:**				

18.1.11 Open Issues

The following table summarise open issues/uncertainties that need to be resolved during the next stages or implementation.

Issue	Description	Next Steps	Lead or Related Component
Risk Analysis - Risk Knowledge base	Risk assessment use of DRM tool and environment	Need to define better, schedule call with UOS	UOS
Policy enforcement	The Policy Enforcement Tool requires data from the DRM tool to effectively enforce policies. Similarly, the DRM tool needs access to these policies to accurately record and log their application.	Need to establish clear protocols for how the tools will communicate and share information	INDRA
Logs and Transaction of other DS2 Modules	Any component in the control plane that makes a decision regarding data should be able to log their action into blockchain	Need to establish clear protocol for how these logs will be stored	Various