

MEDIFIT WP1 - RBAC Component in DDSIBB

Sven Böckelmann, benelog GmbH & Co. KG

benelog 



Funded By:



Abstract

In the MEDIFIT project, a pragmatic approach to access management within the Mediterranean food supply chain has been adopted through the integration of Keycloak, an open-source Identity Access Management System. This choice is motivated by Keycloak's extensive feature set and its ability to offer federated authentication, ensuring efficient and secure management of user data. Central to the project's strategy is the application of a hybrid Role-Based Access Control (RBAC) model, which combines Core RBAC with Attributed RBAC (A-RBAC). This model provides a straightforward yet effective solution for managing user permissions in line with the specific requirements of food supply chains. By focusing on a practical and manageable RBAC approach, the MEDIFIT project addresses the need for secure, adaptable, and user-friendly access control within the EPCIS 2.0 events framework, contributing to the overall integrity and safety of the Mediterranean food industry.

Table of Contents

I. Introduction	1
II. Role-Based Access Control (RBAC): Established Techniques	1
A. Diverse Facets of Role-Based Access Control (RBAC) Models	1
1. Standard RBAC (Core RBAC)	2
2. Hierarchical RBAC (H-RBAC)	2
3. Constraint RBAC (C-RBAC)	2
4. Contextual RBAC (C-RBAC)	3
5. Attributed RBAC (A-RBAC)	3
6. Temporal RBAC (T-RBAC)	3
7. Rule-Based RBAC (RB-RBAC)	3
8. Organizational RBAC (O-RBAC)	3
B. Hybrid Role-Based Access Control Model in the MEDIFIT Project	4
III. Implementation of DDSIBB RBAC for EPCIS 2.0 Events using Keycloak	4
A. Role and Role Mapping	4
1. OpenEPCIS Core Roles	4
2. MEDIFIT Event Access Roles	5
3. Restricted Access	5
B. Streamlining Security with Keycloak in the DDSIBB Framework	7
1. Realm Roles in Keycloak	7
2. Managing Groups and User Attributes in Keycloak for DDSIBB	9
3. Integrating LDAP User Federation in Keycloak for MEDIFIT	9
4. Enhancing EPCIS 2.0 Events with RBAC in DDSIBB: A Focus on Metadata and Flexibility	13
5. Extending the EPCIS REST API with Roles-Allowed HTTP Request Header	15
IV. Demonstrator of the RBAC component in DDSIBB	16
V. Conclusion: Access Control in the MEDIFIT Project with Keycloak and EPCIS	17
A. Keycloak and EPCIS 2.0 Integration in the MEDIFIT Project	17
B. Advancing Dynamic Access Management in Supply Chains	17
C. Verifiable Credentials: Enhancing Supply Chain Verification in Food Supply Chains	18
Leveraging Verifiable Credentials for Enhanced Verification	18
Credential Issuance for Non-GS1 Identifiers	19
EPCIS 2.0: A Platform for Harmonizing Data with Verifiable Credentials	19
The Path Forward: Building a Trusted Ecosystem	19

I. Introduction

There are several Identity Access Management Systems available on the market. We have opted for Keycloak due to its Open Source nature and extensive range of features. This system not only provides the ability to manage users, organize corresponding user groups and assign roles to them but also offers Federated Authentication. This feature allows us to integrate other providers, with a significant advantage being that the actual personal data of users remain within their respective systems. A trust relationship is merely established between Medifit's Keycloak system and the other Authentication Provider.

One important deliverable for Work Package 1 involves enabling role-based access control (RBAC) for EPCIS events to limit access. Although RBAC alone may not be sufficient for supply chains in the long run, within the scope of the MEDIFIT project, it has been demonstrated that the role-based access control engine meets all necessary user requirements and was thus implemented accordingly.

A comprehensive overview of Keycloak functionalities and our analytical approach to evaluate its suitability is provided in a separate document¹.

II. Role-Based Access Control (RBAC): Established Techniques

Role-Based Access Control (RBAC) is a widely adopted model for managing users' permissions within a system based on their roles within an organization. RBAC reduces the complexity and potential for error in assigning permissions to users by enabling administrators to associate rights with roles rather than with individual users. Herein, we delineate several entrenched techniques within the domain of RBAC.

A. Diverse Facets of Role-Based Access Control (RBAC) Models

The landscape of Role-Based Access Control (RBAC) has evolved to accommodate a myriad of operational requirements and security policies in complex organizational structures. As the digital infrastructure of businesses becomes more intricate, the demand for varied and specialized RBAC models has led to the creation and adoption of several extended RBAC frameworks. These models blend core RBAC principles with additional rules, attributes, and constraints to provide tailored access control solutions.

¹MEDIFIT Project. "Role Based Access Control With Keycloak in MEDIFIT EPCIS 2.0", 2023. Online Available at: <https://github.com/MEDIFIT-PRIMA/documentation-resources/blob/main/Workpackage%201/Role%20Based%20Access%20Control%20With%20Keycloak%20in%20MEDIFIT%20EPCIS%202.0.pdf>

In practice, the lines between these RBAC extensions can often blur as implementations tend to merge and interweave various concepts to align with specific organizational needs. For instance, Constraint RBAC (C-RBAC) and Contextual RBAC (C-RBAC), despite sharing an acronym, each introduce distinct dimensions to the basic RBAC model—yet both enhance the decision-making process by considering additional variables. This overlap illustrates the complexity and the challenges in distinctly categorizing each RBAC facet, especially when they inherently embody characteristics of one another.

Such convergence within RBAC implementations demonstrates the fluidity of access control requirements and the need for a comprehensive strategy that can adapt to various scenarios. Therefore the RBAC approach in MEDIFIT also resembles a combination of these RBAC models, to build a robust and efficient access control system with clear and easily traceable effects.

1. Standard RBAC (Core RBAC)

Core RBAC² encapsulates the foundational elements of the model, providing a framework where access permissions are based on user roles. Roles correspond to job functions within an organization, and users are assigned to roles based on their responsibilities and qualifications. Permissions to perform certain operations are then tied to these roles. The core RBAC model's simplicity belies its potency; it serves as the cornerstone of access control in various environments, from corporate to cloud-based infrastructures.

2. Hierarchical RBAC (H-RBAC)

Hierarchical RBAC extends the standard model by introducing a hierarchy of roles, where higher-level roles inherit permissions from subordinate ones. This structure mirrors the organizational hierarchy, allowing for a streamlined approach to permission inheritance and a reduction in administrative overhead. It enables more granular control over access rights and can reflect complex organizational structures with multiple layers of roles.

3. Constraint RBAC (C-RBAC)

Constraint RBAC adds constraints or conditions to role assignments and permissions. These constraints could include separation of duties (SoD), which ensures that conflicting roles are not assigned to the same user, thereby preventing potential conflicts of interest or fraud. Additionally, C-RBAC can enforce context-based constraints, like time or location restrictions, further refining the access control framework.

²David Ferraiolo (NIST), Richard Kuhn (NIST). "Role-Based Access Controls", 1992. Online Available at: <https://csrc.nist.gov/files/pubs/conference/1992/10/13/rolebased-access-controls/final/docs/ferraiolo-kuhn-92.pdf>

4. Contextual RBAC (C-RBAC)

Contextual RBAC integrates context-aware policies into the traditional RBAC system, enhancing its capabilities to make decisions based on dynamic attributes. This contextual information can include factors such as the time of access, the location of the user, the current state of the system, or the particular transaction being performed. For instance, a user may only be granted certain privileges during business hours or within secure network zones. By incorporating these variable elements, Contextual RBAC allows for a more nuanced and agile approach to access control, catering to the multifaceted nature of modern enterprise environments. It ensures that permissions are not only tied to the roles but are also sensitive to the operational context, enhancing security and operational efficiency.

5. Attributed RBAC (A-RBAC)

Attributed RBAC integrates dynamic attributes into role definitions and user assignments. Attributes may contain user-specific details. A-RBAC systems can evaluate these attributes to make access decisions, thereby introducing flexibility and adaptability to changing circumstances.

6. Temporal RBAC (T-RBAC)

Temporal RBAC introduces time-based controls into the RBAC model, allowing permissions to be automatically granted or revoked based on time or duration. This is particularly useful for granting temporary access rights, for instance, to contractors or for access that should only be available during specific business hours or stages of a project.

7. Rule-Based RBAC (RB-RBAC)

Rule-Based RBAC incorporates business rules into the access control mechanism. These rules can be defined by the organization to enforce specific policies, ensuring that the access rights are governed by the organization's regulations and policies. RB-RBAC systems use these rules as a basis for dynamically adjusting user permissions in response to changing business needs or operational contexts.

8. Organizational RBAC (O-RBAC)

Organizational RBAC tailors access control mechanisms to the specific structures of an organization. It involves defining roles not just based on job functions but also on the organization's units, such as

departments or teams. O-RBAC takes into account the interrelations between different organizational entities and the specific access needs that arise from an individual's place within the overall structure.

B. Hybrid Role-Based Access Control Model in the MEDIFIT Project

The MEDIFIT project implements a hybrid access control model, combining Core RBAC and A-RBAC, to handle the complex security requirements of food integrity related supply chain systems. Core RBAC establishes a baseline by associating user permissions with specific roles within the system, adhering to a structured approach that mirrors organizational roles and responsibilities. This is augmented with A-RBAC, which introduces a layer of attributes, providing the means to tailor access controls dynamically.

The use of an attribute such as "epcis-capture-default-roles-allowed" exemplifies the integration of A-RBAC within MEDIFIT. This attribute specifies which roles are permitted to query data entries, ensuring a controlled yet flexible access mechanism that can adapt to various scenarios inherent in food traceability and safety applications. The hybrid model secures access rights that are not only pre-defined but also adaptable, ensuring the system is user-friendly and its validations are straightforward to verify.

III. Implementation of DDSIBB RBAC for EPCIS 2.0 Events using Keycloak

A. Role and Role Mapping

For quick identification of relevant roles for event-access within Keycloak, all roles were prefixed with event-access.

1. OpenEPCIS Core Roles

The main DDSIBB REST API framework also requires some basic roles for capture and query and admin of course. These roles control access to the RESTful capture or data retrieval services. For any capture document the default value for the role name allowed to access and EPCIS Document would be query, if not overwritten by the accessing application of course.

Role Name	Description
capture	Capture Interface Access
query	Query Interface Access

Role Name	Description
admin	Administrator

2. MEDIFIT Event Access Roles

To control access for different players or have another special interest, the following roles have been defined for a fitting, but also easy to validate use case in MEDIFIT.

Role Name	Description
event-access-medifit-supplier	Supplier Event Access Role
event-access-medifit-manufacturer	Manufacturer Event Access Role
event-access-medifit-distributor	Distributor Event Access Role
event-access-medifit-surveillance	Surveillance Authority Event Access Role
event-access-medifit-lab	Analytical Lab Event Access Role

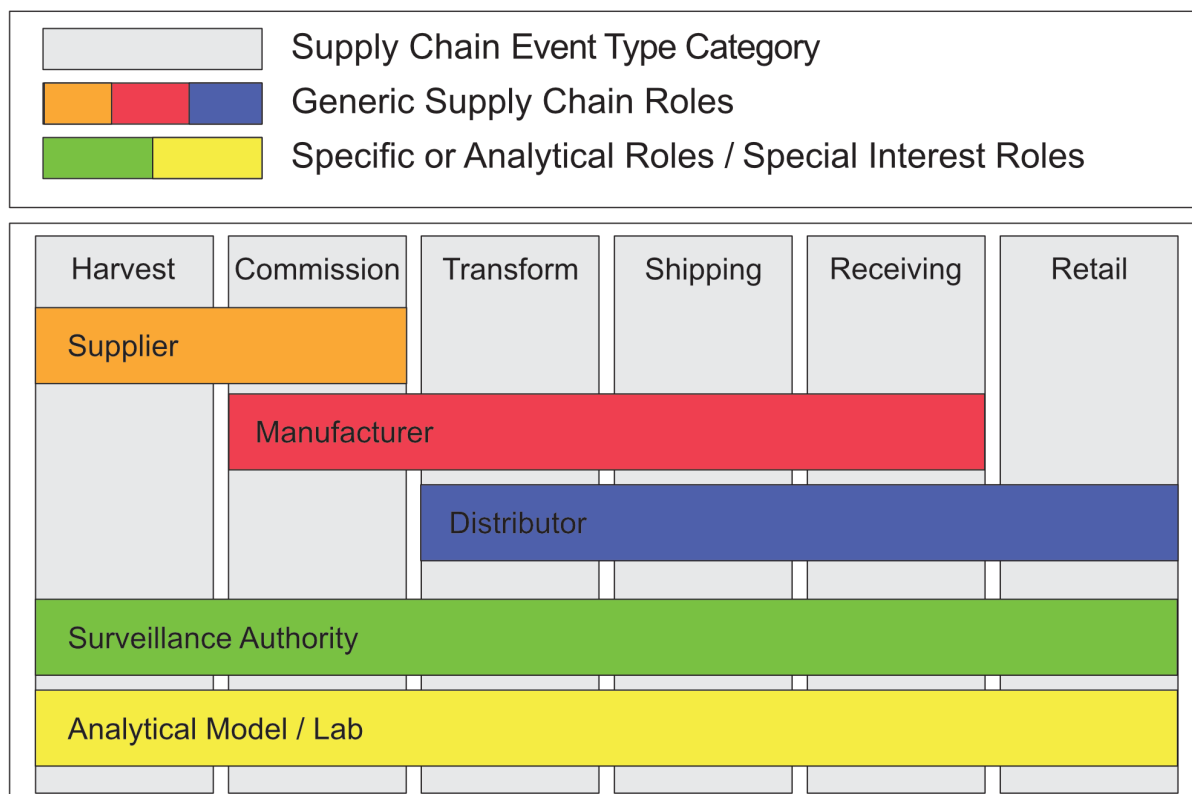
3. Restricted Access

A user has to be associated with the role `capture` - whenever an EPCIS document is being captured, a list of role names that are allowed access will be stored as additional metadata to the document within the database. Access to the document will be only allowed if the accessing user is associated with at least one of roles that were stored together with the EPCIS Document's metadata.

For MEDIFIT, we started with a simple supply chain event type category to user role mapping. In real-life this mapping will surely be more complex, but it's a good overview and illustrated some of the most common supply chain event categories mapped to very generic roles user's may be associated with in a supply chain.

Supplier, **Manufacturer** and **Distributor** may only have access to certain parts of the supply chain for a product. Surveillance Authorities may be granted access upon request or to very specific events, that may be required for regulatory purposes. **Analytical Models**, or **Lab Users** may use very specific events for triggering pipelines or reporting analytical insights, that other users may not even be allowed access to at all.

For a comprehensive illustration see **Figure 1: Generic Supply Chain Role Mapping**.

**Figure 1:** Generic Supply Chain Role Mapping

B. Streamlining Security with Keycloak in the DDSIBB Framework

In the MEDIFIT project, the integration of Keycloak, an open-source tool for managing access, plays a crucial role in securing our Distributed Data and Service Integration BackBone (DDSIBB). To understand our choice of Keycloak and how it fits with our project's needs, we have a detailed paper that explains our decision-making process and the benefits of using Keycloak.

We've set up a specific Keycloak instance for DDSIBB at <https://keycloak.medifit-prima.net>. This instance is vital for enabling Single Sign-On (SSO) across various services within the DDSIBB. SSO makes it easier and more secure for users to access different parts of our system, as they only need to log in once to use all the services.

Configuring realms in Keycloak is a key part of this setup. A realm in Keycloak is like an independent area where we control user access, roles, and settings. Each realm acts as a separate security zone, often linked to different parts of the DDSIBB. This setup helps us keep user access well-organized and secure, ensuring that permissions in one realm don't accidentally affect another.

In each realm, we define and manage roles and permissions tailored to each DDSIBB service. This careful assignment of roles and access rights is essential for maintaining our system's security and ensuring that users have the access they need for their specific roles, without overstepping into areas they shouldn't.

1. Realm Roles in Keycloak

Before diving into the specifics of realm roles, it's important to first set up and define the realms. In Keycloak, a realm is essentially a space where resources, roles, and users are managed separately and securely. Each realm corresponds to different aspects or services of the system, ensuring that access control is both efficient and compartmentalized. In this case we are working with the **openepcis** realm which is used for all EPCIS 2.0 Repository related authorization requests.

Once realms are established, the next step is to define the realm roles. These roles are central to managing access rights within each realm. In Keycloak, realm roles list the specific roles that can be assigned to users within that realm. For example, a role dedicated to laboratory data management might have roles such as 'Lab Technician', 'Data Analyst', or 'Quality Controller'.

To provide a clear understanding of how these roles are organized and managed in Keycloak, we will present a screenshot of the Realm roles list in **Figure 2: Keycloak Realm Roles**.

openepcis

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

Realm roles

Realm roles are the roles that you define for use in the current realm. [Learn more](#)

Search role by name → [Create role](#) 1 - 15

Role name	Composite	Description
admin	False	Administrator
capture	False	Capture EPCIS Events
default-roles-openepcis	True	\${role_default-roles}
event-access-medifit-cheese	False	Cheese Event Access Role
event-access-medifit-distributor	False	Distributor Event Access Role
event-access-medifit-honey	False	Honey Event Access Role
event-access-medifit-lab	False	Analytical Lab Event Access Role
event-access-medifit-manufacturer	False	Manufacturer Event Access Role
event-access-medifit-supplier	False	Supplier Event Access Role
event-access-medifit-surveillance	False	Surveillance Authority Event Access Role
masterdata_admin	False	Administer Masterdata REST API
masterdata_schema_admin	False	Administer Masterdata REST JSON Schema namespaces
offline_access	False	\${role_offline-access}
query	False	Query for EPCIS Events
uma_authorization	False	\${role_uma_authorization}

Figure 2: Keycloak Realm Roles

2. Managing Groups and User Attributes in Keycloak for DDSIBB

User and Group Management in Keycloak Keycloak employs a sophisticated system for managing users and groups, which forms the backbone of its access control mechanisms. At a high level, users are individual accounts with specific roles and permissions, while groups are collections of users that share common access rights. This structure allows for streamlined management of permissions, where assigning a user to a group automatically grants them the access rights associated with that group.

Key Attributes for EPCIS Document Access Control

Within the context of MEDIFIT DDSIBB, two specific Keycloak attributes play a crucial role in controlling access to captured EPCIS documents:

1. **epcis-capture-roles-default-allowed:** This attribute is used to define the default roles that are allowed to query a captured EPCIS Document. It sets a baseline for access control, ensuring that only users with specified roles can retrieve information from these documents.
2. **epcis-capture-grant-roles-allowed:** This attribute restricts the range of roles a user can select from when setting permissions for querying a captured EPCIS Document. It adds an extra layer of control, allowing administrators to limit the roles that can be granted query access, thus tightening security measures around sensitive data.

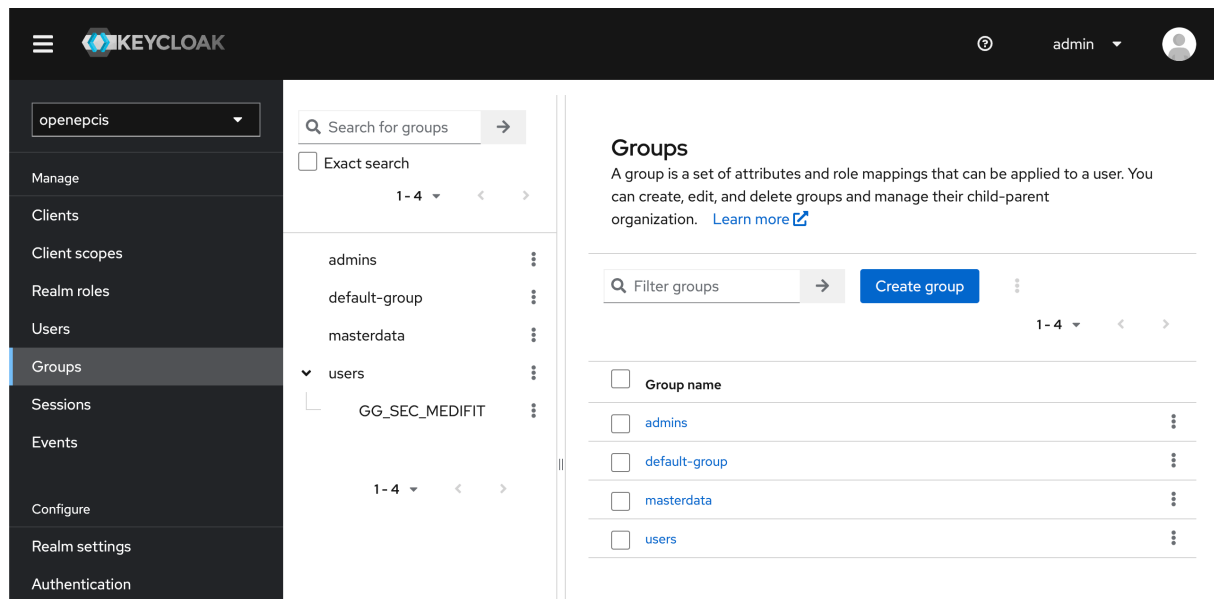
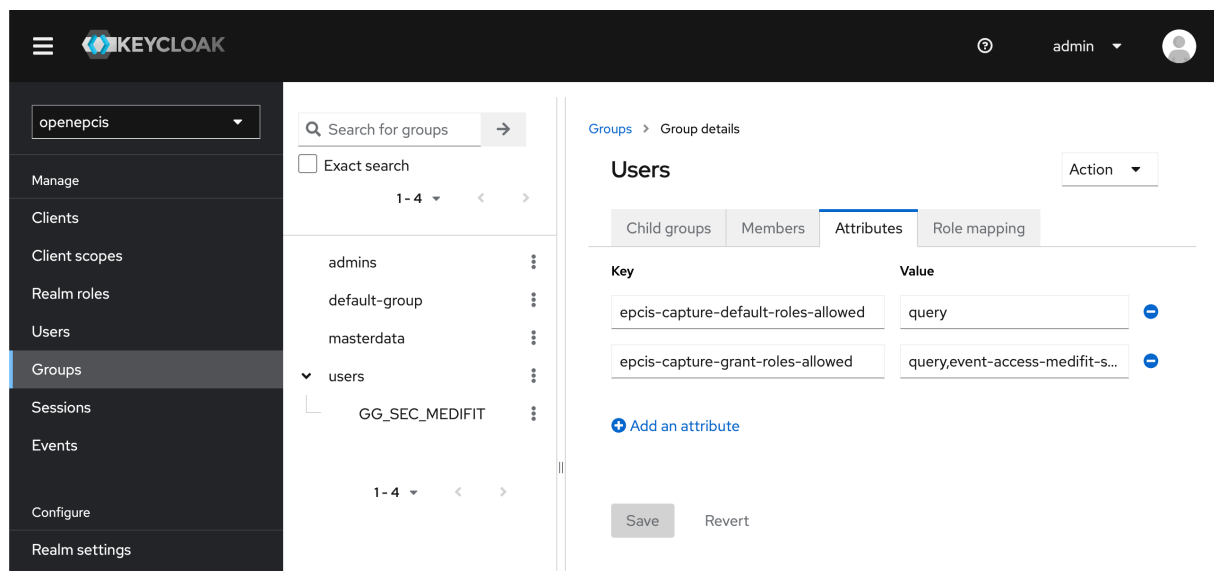
These attributes exemplify the level of control and flexibility that Keycloak offers in managing access to critical data within the EPCIS 2.0 repository. Leveraging these attributes ensures that access to EPCIS documents is both secure and compliant with the data access policies.

Visualization of Group Structures For a clearer understanding of group management in Keycloak, take look at the screenshots of the groups list and the editor interface for managing group attributes. In **Figure 3:** Keycloak Groups and **Figure 4:** Keycloak User Attributes you can observe how groups are organized within Keycloak and the ease with which administrators can modify group attributes to align with specific security and operational requirements.

3. Integrating LDAP User Federation in Keycloak for MEDIFIT

In the MEDIFIT project, federated user management and access control are a key requirement of the system's overall security posture. One aspect of this is the integration of LDAP (Lightweight Directory Access Protocol) for user federation, exemplified by the benelog active directory.

Background: LDAP and User Federation LDAP is a widely used protocol for accessing and managing directory information services over an IP network. In the context of user federation, it allows external

**Figure 3:** Keycloak Groups**Figure 4:** Keycloak User Attributes

user directories, like Active Directory (AD), to be integrated with Keycloak. This integration means that Keycloak can use these directories as sources for user data, streamlining the process of user management and authentication.

Active Directory: A Federated Source for MEDIFIT Users In MEDIFIT's DDSIBB, benelog's active directory functions as a federated source for user information. All MEDIFIT project users are managed within this active directory and are made accessible via Keycloak. This setup simplifies user management, as it centralizes user information in a familiar and widely-used system while leveraging Keycloak for access control within the DDSIBB.

To control access to MEDIFIT's resources, the Active Directory group GG_SEC_MEDIFIT is utilized. Users who are members of this group in Active Directory are granted access rights to specific resources within the DDSIBB framework.

Keycloak Group Mapping from LDAP A crucial step in this integration process is the mapping of AD groups to Keycloak groups. This is achieved by creating a group-ldap-mapper in Keycloak, which in this case is named 'medifit-group-mapper'. This mapper ensures that all users who are members of the GG_SEC_MEDIFIT group in Active Directory are automatically mapped to the user parent group in Keycloak.

This approach highlights Keycloak's ability to not only federate with LDAP sources but also to map these federated users to specific groups within Keycloak. This capability provides a flexible and powerful mechanism for access control and management, as it allows users from different LDAP sources to be grouped and managed in a consistent manner within Keycloak.

Flexibility in User Federation The MEDIFIT implementation of LDAP federation in Keycloak demonstrates the system's adaptability and scalability. Keycloak's ability to federate with multiple LDAP sources and map users from these sources to specific groups paves the way for highly flexible and efficient user access control. This is particularly beneficial in environments like MEDIFIT's DDSIBB, where user management needs to be both secure and adaptable to the varying needs of the project.

In conclusion, the integration of LDAP user federation in Keycloak for the MEDIFIT project is a strategic choice that enhances the system's security and operational efficiency. By centralizing user management in a familiar environment and seamlessly integrating it with Keycloak's robust access control capabilities, MEDIFIT ensures a secure, efficient, and flexible user management system.

[User federation](#) > [Settings](#) > Mapper details

medifit-group-mapper Action ▼

ID	8bf9210a-b118-4127-a3f8-ef3940416d0c
Name * ⓘ	medifit-group-mapper
Mapper type * ⓘ	group-ldap-mapper
LDAP Groups DN ⓘ	CN=Users,DC=company-group,DC=dir
Group Name LDAP Attribute ⓘ	cn
Group Object Classes ⓘ	group
Preserve Group Inheritance ⓘ	<input type="checkbox"/> Off
Ignore Missing Groups ⓘ	<input checked="" type="checkbox"/> On
Membership LDAP Attribute ⓘ	member
Membership Attribute Type ⓘ	DN ▼
Membership User LDAP Attribute ⓘ	userPrincipalName
LDAP Filter ⓘ	(CN=GG_SEC_MEDIFIT)
Mode ⓘ	READ_ONLY ▼
User Groups Retrieve Strategy ⓘ	LOAD_GROUPS_BY_MEMBER_ATTRIBUTE ▼
Member-Of LDAP Attribute ⓘ	memberOf
Mapped Group Attributes ⓘ	
Drop non-existing groups during sync ⓘ	<input type="checkbox"/> Off
Groups Path ⓘ	/users

Save Cancel

Figure 5: Keycloak LDAP Attribute Mapping

4. Enhancing EPCIS 2.0 Events with RBAC in DDSIBB: A Focus on Metadata and Flexibility

In the realm of the MEDIFIT project's DDSIBB, augmenting EPCIS 2.0 event data to align with Role-Based Access Control (RBAC) models presents a unique integration of security and data management. This article explores how OpenSearch JSON indexing and the addition of custom metadata attributes enhance the management and security of EPCIS event data within the framework.

Integration of Metadata in EPCIS 2.0 Events A key aspect of this integration is the augmentation of EPCIS 2.0 event data with additional metadata attributes. This involves incorporating elements like `capturedBy` (the identifier of the user who captured the data), `rolesAllowed` (specifying which user roles are permitted to access the event data) and `captureID` (a unique identifier assigned for every capture request). These metadata attributes play a crucial role in aligning the data with the RBAC model, ensuring that access to each event is precisely controlled based on user roles within the DDSIBB.

OpenSearch for Flexible and Efficient Data Management OpenSearch, employed for JSON indexing of the EPCIS event data, offers a robust mechanism for managing and accessing this augmented data. It enables efficient indexing and retrieval of event data, allowing for quick and accurate queries. While OpenSearch also offers mechanisms for access control, the decision in MEDIFIT to not use these features was deliberate. The aim was to maintain technology agnosticism, ensuring that the RBAC implementation could work seamlessly with various data backends—be it SQL databases like PostgreSQL, graph databases, or others—without being tied to the specific capabilities or limitations of OpenSearch.

A Practical Look at EPCIS Event Data To illustrate this integration, consider the following example of an EPCIS event stored in OpenSearch:

```
{
  ...
  "metadata": {
    "visible": true,
    "rolesAllowed": [
      "event-access-medifit-manufacturer"
    ],
    "capturedBy": "e4a3dcc1-b852-46b1-8a73-f4a7f1e831d2"
  },
  "captureID": "a9af2245-b829-4e4c-bf9a-bdaa057f9999",
  "type": "ObjectEvent",
  "bizStep": {
```



```
    "asURN": "urn:epcglobal:cbv:bizstep:inspecting",
    "asCaptured": "inspecting",
    "asURI": "https://ref.gs1.org/cbv/BizStep-inspecting",
    "asBareString": "inspecting"
  },
  "disposition": {
    "asURN": "urn:epcglobal:cbv:disp:in_progress",
    "asCaptured": "in_progress",
    "asURI": "https://ref.gs1.org/cbv/Disp-in_progress",
    "asBareString": "in_progress"
  },
  "recordTime": "2023-11-10T11:38:57.164Z",
  "sensorElementList": [
    {
      "sensorReport": [
        {
          "uom": "CEL",
          "type": {
            "asURN": "gs1:Temperature",
            "asCaptured": "Temperature",
            "asURI": "https://gs1.org/voc/Temperature",
            "asBareString": "Temperature"
          },
          "value": 46.142
        }
      ]
    }
  ],
  ...
}
```

In this snippet, the `rolesAllowed` attribute within the metadata section is particularly noteworthy. This attribute specifies the roles that are permitted to access this specific event. The roles listed here are matched against the roles assigned to users within the DDSIBB, ensuring that only authorized users can access the event data. This matching process is a fundamental part of the RBAC implementation, ensuring that access to sensitive data is strictly controlled and aligned with user permissions.

The enhancement of EPCIS 2.0 events with RBAC-compatible metadata within the MEDIFIT project's DDSIBB framework represents an advanced approach to data security and access management. By leveraging the capabilities of OpenSearch for efficient data indexing and retrieval, and by embedding critical metadata attributes like `rolesAllowed`, the DDSIBB ensures that data access is both secure and aligned with the specific roles and responsibilities of users in the system.

5. Extending the EPCIS REST API with Roles-Allowed HTTP Request Header

In the era of RESTful APIs, which form the backbone of modern web services, HTTP request headers play a crucial role in defining and controlling the behavior of API requests. The MEDIFIT project's enhancement of the EPCIS 2.0 REST API, specifically through the addition of the Roles-Allowed header, exemplifies a sophisticated approach to integrating Role-Based Access Control (RBAC) within the API framework.

The Role of HTTP Request Headers in RESTful APIs HTTP request headers are an integral part of RESTful API design, carrying metadata that defines the parameters of an API request. These headers instruct the server on how to process the incoming request, ranging from specifying the desired data format to controlling access and behavior of the API. In essence, they are the directives that precede the actual data payload, setting the stage for how the API interaction will unfold.

Standard Headers in EPCIS 2.0 Capture REST API The EPCIS 2.0 Capture REST API, as defined by the GS1 Standard, includes several headers that dictate how the API should process event capture requests^{3 4}. These include:

- **GS1-Capture-Error-Behaviour:** This header determines the error handling behavior during the capture process. Options include `rollback`, where the capture job must be entirely successful or otherwise all events are rejected, and `proceed`, where the system attempts to capture as many events as possible despite errors. The default is `rollback`, mirroring the behavior in EPCIS 1.2.
- **GS1-EPCIS-Version & GS1-CBV-Version:** These headers specify the versions of the EPCIS and Core Business Vocabulary (CBV) standards being used, ensuring compatibility and understanding between the client and server.
- **GS1-Extensions:** This header identifies any specific EPCIS or CBV extensions supported in the capture request, allowing for expanded functionality beyond the base standards.

Introducing the Roles-Allowed Header for RBAC The MEDIFIT-specific extension, the `Roles-Allowed` header, is a significant enhancement to the EPCIS 2.0 REST API. This header allows for the specification of access levels based on user roles at the time of event capture. Essentially, it sets the stage for who can access the event data once it is stored in the system. By including this header in the API request, the client can specify a list of roles that are permitted to access the captured event. This

³MEDIFIT Project, "EPCIS 2.0 REST Bindings", 2023. Online Available at: https://epcis.medifit-prima.net/q/swagger-ui/index.html#/Capture/post_capture

⁴GS1 EPCIS 2.0 Standard, "EPCIS Standard", Chapter 12.6, 2022. Online Available at: <https://ref.gs1.org/standards/epcis/>

mechanism ensures that access to event data is tightly controlled and aligned with the RBAC policies set within the DDSIBB.

For example, if an EPCIS event is captured with the Roles-Allowed header specifying `event-access-medifit-manufacturer`, only users assigned this role in the DDSIBB will be able to query and view this event. This approach ensures that sensitive data is only accessible to authorized users, enhancing the security and integrity of the data within the MEDIFIT ecosystem.

IV. Demonstrator of the RBAC component in DDSIBB

As previously highlighted, upon capturing the EPCIS document, users have the capability to specify the Roles-Allowed header. This functionality allows for the restriction of access to the captured EPCIS document. For instance, in the subsequent example, the intent is to capture the EPCIS document with the designated `event-access-medifit-manufacturer` role. Consequently, upon successful capture, solely users possessing the `event-access-medifit-manufacturer` role will be granted access to these specific EPCIS events. The process of this capture is illustrated in the accompanying screenshot within the Hoppscotch REST API platform **Figure 6: Capture event with roles**.

Furthermore, the system accommodates the capture of EPCIS events with multiple roles by allowing their specification as comma-separated values. This flexibility enables the inclusion of diverse role designations, such as `event-access-medifit-manufacturer`, `event-access-medifit-supplier`, facilitating an advanced approach to access control. This capability to assign multiple roles expands the granularity of access management, granting selective access to a broader spectrum of designated user categories.

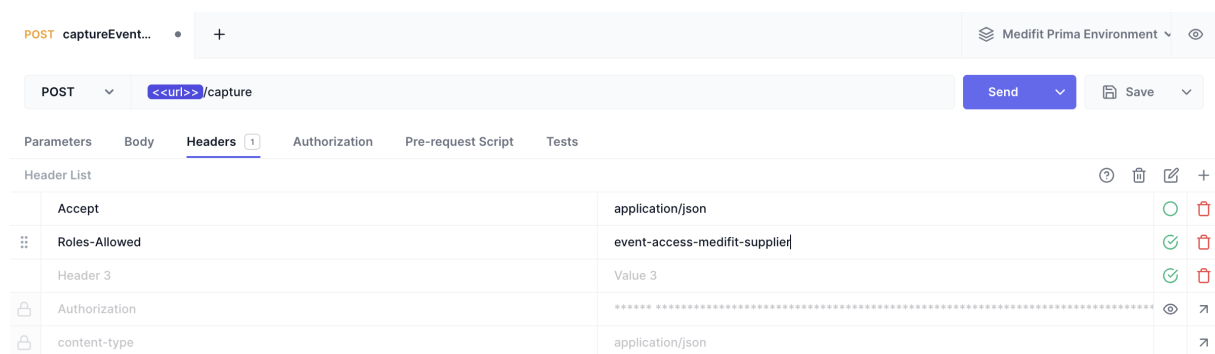


Figure 6: Capture event with roles

The association of roles with users within Keycloak assumes a critical role in event querying processes. This pivotal linkage dictates that when a user initiates a query for events, only those events linked to

the user's specific role will be returned. Conversely, any event not aligned with the user's assigned role will be intentionally excluded from the query results. This mechanism effectively imposes strict visibility limitations on events, allowing only those events that correspond with the roles assigned to the querying user to be accessible. By selectively filtering and disclosing events based on the user's roles, it not only facilitates a more streamlined and personalized user experience but also fortifies the overall data security measures in place. This process has been demonstrated within the Hoppscotch REST API platform **Figure 7: Query event with roles**, as illustrated in the accompanying screenshot.

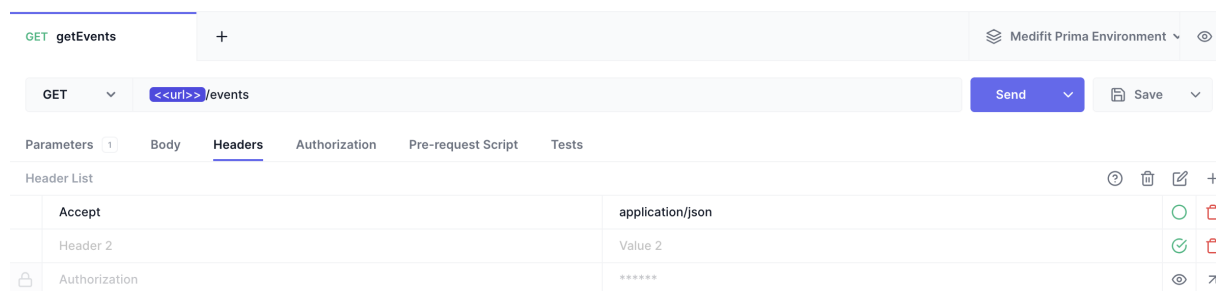


Figure 7: Query event with roles

V. Conclusion: Access Control in the MEDIFIT Project with Keycloak and EPCIS

A. Keycloak and EPCIS 2.0 Integration in the MEDIFIT Project

The MEDIFIT project's implementation of Keycloak and the enhancement of the EPCIS 2.0 REST API are pivotal steps in addressing the intricate challenges of managing access to vital food integrity data.

Keycloak's ability to manage user identities and its integration with various authentication providers form the foundation of MEDIFIT's security infrastructure. The application of Keycloak's realms, alongside user and group management with LDAP federation, creates a comprehensive identity and access management system. This system is not just secure; it is also tailored to be user-friendly and adaptable to the varying roles and needs within the food supply chain.

B. Advancing Dynamic Access Management in Supply Chains

However, the dynamic nature of supply chain roles, where entities often assume multiple roles such as supplier, manufacturer, and distributor at different times or contexts, underscores the need for more nuanced access control. The current use of RBAC, while effective, points towards the potential benefits of integrating Contextual RBAC (C-RBAC). This advanced approach could offer dynamic access

control, adapting permissions based on contextual factors like the specific role in the supply chain, point-in-time, or product context.

Looking ahead, integrating machine learning and artificial intelligence could further refine this system. AI/ML algorithms could analyze patterns and make informed decisions about access rights, adapting in real-time to the evolving contexts within the supply chain. This would not only enhance the precision of access control but also add a layer of proactive security management, potentially identifying and responding to anomalies or shifts in user behavior and supply chain dynamics.

In conclusion, while the MEDIFIT project has made commendable progress with Keycloak and EPCIS 2.0, exploring advancements like Contextual RBAC and AI-driven access control could be invaluable in addressing the multifaceted and dynamic nature of supply chain roles.

C. Verifiable Credentials: Enhancing Supply Chain Verification in Food Supply Chains

In the dynamic landscape of supply chain and decision support frameworks like MEDIFIT, the integration of verifiable credentials, particularly in conjunction with Keycloak and EPCIS 2.0, presents a significant opportunity to improve the verification processes within the food supply chain. Verifiable Credentials complement EPCIS 2.0 in food supply chains by providing secure, cryptographically verifiable proof of product information, enhancing traceability and trustworthiness at every step.

Leveraging Verifiable Credentials for Enhanced Verification

Verifiable credentials are revolutionizing the landscape of digital authentication, as highlighted in the W3C's Verifiable Credentials Data Model v2.0⁵. These credentials provide a secure and cryptographically verifiable way to prove various claims digitally. The integration of verifiable credentials into GS1 Standards marks a significant advancement, especially relevant in supply chain contexts. The GS1 Repository on Verifiable Credentials⁶ offers a wealth of information, including formal documents and artifacts essential for implementing these credentials.

Notably, GS1 identifiers such as GTIN (Global Trade Item Number) and GLN (Global Location Number) are particularly suitable for use as verifiable credentials. GTINs are unique identifiers for products, commonly used worldwide for inventory tracking and point-of-sale identification. GLNs serve as unique identifiers for locations, enabling precise identification of companies, warehouses, and other entities within the supply chain. The inherent characteristics of GTINs and GLNs – their uniqueness and global

⁵The World Wide Web Consortium (W3C), "Verifiable Credentials Data Model v2.0", 2023. Online Available at: <https://www.w3.org/TR/vc-data-model-2.0/>

⁶GS1 Global Office, "GS1 documents and artefacts related to Verifiable Credentials", 2023. Online Available at: <https://ref.gs1.org/gs1/vc/>

recognition – make them ideal for use in verifiable credentials, significantly enhancing the security, trustworthiness, and transparency of transactions and tracking in supply chains.

Credential Issuance for Non-GS1 Identifiers

In the future scenario, Keycloak's role could be significantly expanded to include support for OpenID for Verifiable Credentials, as per the OpenID specifications⁷. This extension would enable Keycloak to serve as a provider for verifiable credentials, particularly for non-GS1 identifiers. Such an enhancement would offer additional verification support in areas like analytical measurements, predictive models, and other datasets, further improving the capabilities to manage and authenticate diverse elements within the supply chain.

EPCIS 2.0: A Platform for Harmonizing Data with Verifiable Credentials

EPCIS 2.0, integral to MEDIFIT's infrastructure for capturing and sharing product information, can effectively harmonize with verifiable credentials. This integration allows for a broader and more inclusive verification system, where product data, irrespective of the identifier system used, is authenticated and validated. The synergy between EPCIS 2.0 and the verifiable credentials enhances the overall trustworthiness and reliability of the supply chain data.

The Path Forward: Building a Trusted Ecosystem

The future in trusted data sharing and verification is anchored in the seamless integration of diverse identity systems, the adoption of a universal data language, and the establishment of a robust ecosystem of trust. The combination of GS1's verifiable credentials, Keycloak's authentication capabilities, and the data management potential of EPCIS 2.0 creates a comprehensive framework. The chosen frameworks not only ensure that every product is verified and authenticated efficiently but also aligns with the project's commitment to upholding the highest standards in food safety and traceability.

⁷OpenID Connect. T. Lodderstedt, K. Yasuda, T. Looker, "OpenID for Verifiable Credential Issuance", 2023. Online Available at: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html