Role Based Access Control With Keycloak in MEDIFIT EPCIS 2.0

Sven Böckelmann, benelog GmbH & Co. KG

Funded By:

PRIMA
PARTNERSHIP FOR RESEARCH AND INNOVATION
IN THE MEDITERRANEAN AREA

European
Commission

Table of Contents

I. Introduction to Role-Based Access Control (RBAC) 1
A. Definition of RBAC 1
B. Importance and Benefits of RBAC 1
C. Overview of Keycloak and its Role in RBAC 2

II. Understanding Keycloak 2
A. Definition and Functionality of Keycloak 2
B. Key Features of Keycloak 3
C. The Role of Keycloak in RBAC 3

III. Setting up Keycloak for RBAC 4
A. Installation and Configuration of Keycloak 4
Using Docker 4
Using Podman 4
Access local Keycloak Container 5
B. Setting up Realms, Roles, and Users in Keycloak 5
C. Configuration of Role-Based Access Control in Keycloak 5

IV. Implementing RBAC with Keycloak 6
A. Assigning Roles to Users in Keycloak 6
B. Managing and Controlling User Access Using Keycloak 6
C. Enhancing Application Access with Keycloak Integration 7

V. Advanced Features of Keycloak in RBAC 8
A. Role Mapping and Role Policies in Keycloak 8
Dynamic Role Assignment 8
Role Policies 8
B. Using Keycloak for Fine-Grained Access Control 8
Attribute-Based Access Control (ABAC) 9
Permission Management 9
C. Keycloak's Support for Multi-Factor Authentication and Single Sign-On 9

VI. Implementing RBAC in an EPCIS 2.0 Business Environment using Keycloak 10
A. Requirements for RBAC in a Supply Chain Environment 11
B. Designing and Implementing an RBAC Solution using Keycloak 11
C. Evaluating the Effectiveness of the RBAC Solution 11

VII. Conclusion **12**

A. Recap of RBAC and Keycloak’s Role 12

B. The Evolution of Access Control with EPCIS 2.0 and Keycloak Integration 12

I. Introduction to Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) is an essential facet of data security architecture, crucial for safeguarding access to organizational resources. It's a strategy integral to maintaining the integrity and confidentiality of data within a corporate environment [1]. RBAC has been a key part of system security architecture for decades, providing a structured approach to managing user permissions based on their organizational roles. This method boasts advantages such as simplified administration and enhanced security through the principle of least privilege; however, it can become cumbersome to manage in highly dynamic environments and may lack the granularity offered by attribute-based access control (ABAC) or context-aware RBAC (CRBAC) access control mechanisms.

A. Definition of RBAC

RBAC operates on a straightforward concept: assign access permissions based on roles within an organization. It's similar to defining clear job descriptions and ensuring employees can only execute tasks pertinent to their positions. This method aligns user privileges with their responsibilities, minimizing the risk of overstepping access boundaries.

The guiding principle of RBAC is the "least privilege" doctrine, which is a cornerstone for bolstering information system security by confining access rights to the bare minimum necessary to perform a job [2].

B. Importance and Benefits of RBAC

In a digital landscape littered with data breaches, RBAC is an essential defense mechanism against unauthorized data exposure. By compartmentalizing access, it significantly mitigates the risk of data falling into the wrong hands.[3]

RBAC delivers multiple organizational benefits. It strengthens security by aligning access with job requirements, minimizes the potential for data mishandling, and streamlines the management of user permissions. Furthermore, it supports regulatory compliance by establishing clear and auditable access controls.[4]

[1] David 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

[2] National Institute of Standards and Technology (NIST). "Assessment of Access Control Systems", 2006. Online Available at: https://csrc.nist.gov/publications/detail/nistir/7316/final

[3] Heise Online. "APIs sicher entwickeln" Heise iX, 07/2022. Online Available at: https://www.heise.de/select/ix/2022/7/2215808090096396564

[4] BSI - Bundesamt für Sicherheit in der Informationstechnik. "IT-Grundschutz-Kompendium." BSI, 2023. Online Available at: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompendium/it-grundschutz-kompendium_node.html

C. Overview of Keycloak and its Role in RBAC

Keycloak stands out as a premier Identity and Access Management solution designed for modern service-oriented architectures. It facilitates application security with minimal coding, providing a suite of capabilities including single sign-on and social login support.[5]

Keycloak's strength in RBAC comes from its capacity to offer detailed role and permission definitions, giving administrators precise access control tools. It supports assigning roles directly to users or through group memberships and enables fine-tuning of permissions for client applications.[6]

Moreover, Keycloak's adaptability to both traditional and microservices-based applications positions it as a versatile choice for RBAC implementation, making it technology-stack agnostic.[7]

In sum, RBAC stands as a formidable instrument for amplifying the security and operational efficiency of information systems. Leveraging solutions like Keycloak, enterprises can effectively govern access to their digital resources, ensuring data protection and regulatory adherence.[8]

II. Understanding Keycloak

A. Definition and Functionality of Keycloak

Keycloak is heralded as an open-source Identity and Access Management (IAM) tool, developed by JBoss, a division of Red Hat. It is engineered to offer a robust suite for managing user identity and facilitating access control within modern applications and services. Keycloak's capabilities extend to include user federation, single sign-on (SSO), identity brokering, and social logins, among others.

This platform abstracts the complexity associated with securing applications, thereby allowing developers to divert their focus toward the development of core features. Keycloak's management of identities, credentials, and permissions is pivotal in safeguarding access to system resources[9].

[5] Keycloak Community. "Keycloak Official Documentation", 2023. Online Available at: https://www.keycloak.org/documentation.html

[6] Heise Online. "IT-Systemzugriffe verwalten: Identity und Access Management mit Keycloak", Heise Online, 2020. Online Available at: https://www.heise.de/ratgeber/IT-Systemzugriffe-verwalten-Identity-und-Access-Management-mit-Keycloak-4961038.html

[7] Heise Online. "Microservices-Architekturen absichern mit Keycloak." Heise iX, 08/2022. Online Available at: https://www.heise.de/select/ix/2022/8/2207607184979801482

[8] European Union. "General Data Protection Regulation (GDPR). Online Available at: https://gdpr.eu

[9] S. Farrell and H. Tschofenig, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," RFC 6750, Oct. 2012. Online Available at: https://tools.ietf.org/html/rfc6750

B. Key Features of Keycloak

Keycloak's suite of features is comprehensive, positioning it as a preferred solution for IAM tasks. Among these features are:

1. **Single Sign-On (SSO) and Sign out**: Incorporating OpenID Connect and Security Assertion Markup Language (SAML) 2.0, Keycloak offers a seamless SSO experience across multiple services, which not only optimizes the user experience but also fortifies security [10].
2. **User Federation**: Keycloak integrates with various user databases such as Lightweight Directory Access Protocol (LDAP) and Active Directory, centralizing user data management and simplifying user administration across different platforms[11].
3. **Identity Brokering**: The platform facilitates easy integration with social media logins, streamlining the process of user authentication and registration[12].
4. **Centralized Management**: A comprehensive management console allows administrators to oversee security aspects such as user roles, sessions, and application settings.
5. **Fine-Grained Authorization Services**: Keycloak supports advanced permission management through policies and access settings that cater to diverse application needs.

C. The Role of Keycloak in RBAC

Role-Based Access Control (RBAC) is a system that assigns access to resources based on the user roles within an organization. Permissions are linked to roles rather than individuals, simplifying the management of user permissions.

Keycloak is integral to implementing RBAC, providing a user-friendly interface for role and permission management. Administrators can define and control access rights, ensuring adherence to the principle of least privilege. The distinction between realm and client roles in Keycloak affords an additional layer of granularity in access control.

In summary, Keycloak stands out as a comprehensive IAM solution. Its extensive feature set and adaptability make it an invaluable tool for ensuring robust application security, for organizations to achieve a secure and efficient user management system.

[10] N. Sakimura et al., "OpenID Connect Core 1.0," OpenID Foundation, Nov. 2014. Online Available at: https://openid.net/specs/openid-connect-core-1_0.html

[11] M. Wahl, T. Howes, and S. Kille, "Lightweight Directory Access Protocol (v3)," RFC 4511, Jun. 2006. Online Available at: https://tools.ietf.org/html/rfc4511

[12] B. Fitzpatrick, "The OAuth 2.0 Authorization Framework," RFC 6749, Oct. 2012. Online Available at: https://tools.ietf.org/html/rfc6749

III. Setting up Keycloak for RBAC

The following section provides a comprehensive tutorial on deploying Keycloak on cloud platforms and configuring it to manage roles and permissions effectively.

A. Installation and Configuration of Keycloak

Keycloak's versatility is evident in its compatibility with a variety of cloud services, such as Google Cloud, Amazon Web Services (AWS), and Microsoft Azure. Although setup steps may slightly differ across services, the core installation process is consistent.

Using Docker

Docker offers an efficient pathway to deploy Keycloak. To begin, ensure Docker is installed on your system by consulting the Docker documentation[13].

Step 1: Pull the Keycloak Docker Image Acquire the latest Keycloak Docker image from Docker Hub by executing:

```
docker pull quay.io/keycloak/keycloak:latest
```

Step 2: Run Keycloak Container To initiate the Keycloak container with an administrator account:

```
docker run -p 8080:8080 -e KEYCLOAK_USER=admin -e
↪  KEYCLOAK_PASSWORD=admin_password quay.io/keycloak/keycloak:latest
```

This command designates port 8080 for Keycloak and creates an admin user with specified credentials.

Using Podman

Podman[14] is a compatible alternative to Docker and operates without a daemon, adhering to the Open Container Initiative (OCI) specifications[15]. To install Podman on your system, please follow the Podman Installation Instructions[16].

[13] Docker Documentation. Online Available at: https://docs.docker.com/

[14] Podman Documentation. Online Available at: https://docs.podman.io/

[15] Red Hat. "What's a Linux container?". Online Available at: https://www.redhat.com/en/topics/containers/whats-a-linux-container

[16] Podman Documentation. "Podman Installation Instructions", 2023. Online Available at: https://podman.io/docs/installation

Step 1: Pull the Keycloak Container Image Retrieve the latest Keycloak image with Podman:

```
podman pull quay.io/keycloak/keycloak:latest
```

Step 2: Run Keycloak Container Deploy the Keycloak container using Podman as follows:

```
podman run -p 8080:8080 -e KEYCLOAK_USER=admin -e
↪  KEYCLOAK_PASSWORD=admin_password quay.io/keycloak/keycloak:latest
```

Access local Keycloak Container

Access the running Keycloak instance via `http://localhost:8080/auth` in your web browser.

Step 1: Log into the Admin Console Log in with the admin credentials established during the container setup phase.

Step 2: Configure Keycloak At this juncture, you can begin configuring realms, clients, and users to suit your organizational needs.

B. Setting up Realms, Roles, and Users in Keycloak

Keycloak employs realms to encapsulate resources. Each realm is an independent entity with distinct users and settings[17].

1. **Create a Realm**: Access 'Add Realm' and submit a name to establish a new realm.
2. **Create a Role**: In the 'Roles' section, opt for 'Add Role', name the role, and confirm.
3. **Create a User**: Under 'Users', choose 'Add User', fill in the details, and allocate roles through the 'Role Mappings' tab.

C. Configuration of Role-Based Access Control in Keycloak

Keycloak administers RBAC by pairing roles with permissions, allowing for fine-grained access control.

1. **Create a Client**: Within 'Clients', select 'Create', input a client ID, and save.

[17] Keycloak - Open Source Identity and Access Management. "Server Administration Guide", 2023. Online Available at: https://www.keycloak.org/docs/latest/server_admin/

2. **Enable Authorization Settings**: Turn on Authorization settings in ‘Authorization -> Settings’.
3. **Formulate a Policy**: Develop a ‘Role Policy’ under ‘Authorization -> Policies’, tying it to a specific role.
4. **Assign Permissions**: Generate permissions linked to the policy via ‘Authorization -> Permissions’.

IV. Implementing RBAC with Keycloak

Keycloak facilitates RBAC functionalities within its administrative functions for managing roles, permissions, and user accounts.

A. Assigning Roles to Users in Keycloak

Assigning roles to users is a fundamental step in leveraging RBAC with Keycloak. A role in Keycloak signifies a collection of permissions that defines the user’s access within the system. The following steps delineate the role-assignment process[18] :

1. **Login to Keycloak**: Initiate by logging into the Keycloak administration console.
2. **Navigate to Users**: Use the left-hand side menu to click on ‘Users’.
3. **Select a User**: Upon display of the user list, select the individual to assign roles to.
4. **Go to Role Mappings**: Located at the top of the user’s details page is the Role Mappings tab.
5. **Assign Role**: Within this section, select the desired role from the ‘Available Roles’ and enact the assignment by clicking ‘Add selected’.

After completion of these steps, the user will possess the selected role and its associated access rights and permissions.

B. Managing and Controlling User Access Using Keycloak

Keycloak is equipped with an advanced authorization service that streamlines user access management. It centralizes the administration of roles, permissions, and policy enforcement, facilitating various permission types, including resource-based, scope-based, and policy-based[19]. For user access management, you define permissions for each resource or scope, correlating them with the necessary

[18] Keycloak Project. “Keycloak Administration Guide - Role Mapping”, 2023. Online Available at: https://www.keycloak.org/docs/latest/server_admin/index.html#_role_mappings

[19] Keycloak Project. “Keycloak Authorization Services Guide”, 2023. Online Available at: https://www.keycloak.org/docs/latest/authorization_services/index.html

policies. Additionally, Keycloak's policy evaluation feature assists in validating the efficacy of your policies and permissions.

C. Enhancing Application Access with Keycloak Integration

Keycloak stands out with its exceptional ability to mesh with a diverse range of application architectures, from modern RESTful services to conventional web applications. It achieves this through a suite of integration connectors and the use of its Representational state transfer (REST) APIs or adapter framework, enabling applications to leverage the sophisticated RBAC capabilities that Keycloak offers. Moreover, its compatibility with established protocols like OpenID Connect (OIDC) and SAML 2.0 allows for seamless interaction with various external services.

Keycloak's features also include Single Sign-On, Identity Brokering, and Social Login, thereby improving the user interaction experience. Its support for prominent protocols such as OpenID Connect, SAML 2.0, and OAuth 2.0 affirms its position as a versatile player in access management.

To uphold the integrity and security of SAML, OIDC, and OAuth 2.0 implementations, the following practices are crucial:

1. **Use HTTPS**: Prioritize the use of TLS/SSL to safeguard data during transmission.
2. **Validate Tokens**: Rigorously check SAML assertions and OIDC tokens to eliminate the risk of fraudulent claims.
3. **Manage Configurations**: Ensure that both identity providers and service providers are configured with security in mind.
4. **Keep Software Updated**: Maintain the most recent software updates to address security flaws promptly.
5. **Employ Strong Cryptography**: Opt for robust, current cryptographic algorithms for the signing and encryption of tokens.
6. **Secure Token Handling**: Handle, transmit, and store tokens with utmost security to prevent unauthorized access or data breaches.

Implementing these measures is critical to prevent potential security threats and safeguarding against vulnerabilities.

V. Advanced Features of Keycloak in RBAC

With its comprehensive support for Role-Based Access Control (RBAC)[20], Keycloak enables administrators to secure applications and services effectively.

A. Role Mapping and Role Policies in Keycloak

Role mapping in Keycloak is a powerful feature that allows for the association of users with roles, thereby determining the permissions they possess within an application or service. Keycloak offers two types of roles: Realm roles, which are global and can be applied across all applications within the Keycloak realm, and Client roles, which are specific to an individual application within a realm.

Dynamic Role Assignment

Keycloak extends the functionality of static role mapping with its dynamic role assignment. This feature enables the conditional assignment of roles based on user attributes, client data, or the results of customized authenticators. This adaptability ensures that roles reflect the current context of the user or the application state, aligning access permissions with real-time scenarios.

Role Policies

Role policies form the backbone of access control in Keycloak, where they define the conditions under which roles can be granted to a user. With the aid of a user-friendly interface or via the API, administrators can create and manage complex role policies that support a granular level of control over user permissions. These policies can be based on attributes, resource types, or even specific actions that users can perform, ensuring that permissions are precisely tailored to the user's context and organizational policies.

B. Using Keycloak for Fine-Grained Access Control

Keycloak moves beyond traditional RBAC[21] with its support for fine-grained access control mechanisms. This means that access can be controlled not just at the level of roles, but also at the level of individual resources and scopes within those resources.

[20] Keycloak Community. "Keycloak Official Documentation", 2023. Online Available at: https://www.keycloak.org/documentation.html

[21] David 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

Attribute-Based Access Control (ABAC)

Keycloak supports Attribute-Based Access Control (ABAC)[22] by allowing the use of user attributes in the evaluation of permissions. For example, you can restrict access to a resource based on user attributes like department, location, or even the time of day.

Permission Management

Keycloak's permission management allows administrators to define exactly what actions a role can perform on a resource. By combining with policies, these permissions offer a nuanced control mechanism. Permissions can be associated with different types of policies in Keycloak, which can be used to evaluate whether a particular permission should be granted during an access request.

C. Keycloak's Support for Multi-Factor Authentication and Single Sign-On

Keycloak enhances organizational security by incorporating Multi-Factor Authentication (MFA, TOTP[23]) with its Single Sign-On (SSO[24]) capabilities, thereby balancing user convenience with robust security measures. Through SSO, Keycloak allows users to log in just once and access several services without repeated sign-ins. This integration not only streamlines user workflows but also tightens security, as Keycloak centrally manages login credentials and session tokens, making it easier to monitor and protect user access across all applications.

Multi-Factor Authentication Keycloak's MFA feature adds an extra layer of security, ensuring that user access to multiple applications is not only seamless but also rigorously verified. MFA requires users to provide two or more verification factors to access their accounts, which significantly reduces the risk of unauthorized access. Keycloak's MFA support includes a range of options, from traditional SMS-based verification codes and email to modern approaches like time-based one-time passwords (TOTP) with apps like Google Authenticator or FreeOTP[25]. Administrators can enforce MFA as a blanket policy or base it on conditional access policies, requiring additional authentication steps in specific scenarios, such as access from an unrecognized device or login attempts from different geographical locations.

[22] National Institute of Standards and Technology (NIST). "Attribute-Based Access Control", 2014. Online Available at: https://csrc.nist.gov/glossary/term/attribute_based_access_control

[23] Keycloak Project. "Configuring authentication", Keycloak Documentation, 2023. Online Available at: https://www.keycloak.org/docs/latest/server_admin/#configuring-authentication_server_administration_guide

[24] Keycloak Project. "Single Sign-On," Keycloak Documentation, 2023. Online Available at: https://www.keycloak.org/docs/latest/server_admin/#sso-protocols

[25] Keycloak Project. "Time-based One-time Password (TOTP)", Keycloak Documentation. Online Available at: https://www.keycloak.org/docs/latest/server_admin/#_otp-policies

Single Sign-On for Unified Access Through SSO, Keycloak allows users to log in just once and access several services without repeated sign-ins. This integration not only streamlines user workflows but also tightens security, as Keycloak centrally manages login credentials and session tokens, making it easier to monitor and protect user access across all applications[26].

Broad Protocol Compatibility The support of protocols like OIDC, SAML 2.0, and OAuth 2.0 ensures that Keycloak's MFA and SSO solutions are compatible with a wide array of applications. This universal approach empowers organizations to implement MFA and SSO without compromising on the security of legacy systems or the integration with modern platforms[27][28][29].

Centralized Session Management With Keycloak's centralized session management, administrators have a clear overview of active sessions across all applications. Users can sign out once to end their sessions across all services, and administrators can manage sessions for security reasons, like enforcing immediate logout in case of suspicious activities[30].

Keycloak's RBAC features are part of a larger suite of tools that make it an indispensable ally for organizations seeking to streamline access management without compromising on security. By providing intricate role mapping, fine-grained access control, and comprehensive support for MFA and SSO, Keycloak stands at the forefront of IAM solutions, balancing security needs with user convenience in an increasingly complex digital landscape.

VI. Implementing RBAC in an EPCIS 2.0 Business Environment using Keycloak

In the interconnected world of supply chains, safeguarding data and ensuring its judicious access is of utmost importance. Incorporating Access Control tailored for supply chains can be pivotal in striking this balance. The requirement analysis took a deeper look at the deployment of RBAC in an EPCIS 2.0[31] environment specifically designed for supply chain roles, leveraging the capabilities of Keycloak and accommodating the nuances of surveillance authorities and analytical labs.

[26] Keycloak Project. "Single Sign-On," Keycloak Documentation, 2023. Online Available at: https://www.keycloak.org/docs/latest/server_admin/#sso-protocols

[27] N. Sakimura et al., "OpenID Connect Core 1.0," OpenID Foundation, Nov. 2014. Online Available at: https://openid.net/specs/openid-connect-core-1_0.html

[28] OASIS. "Security Assertion Markup Language (SAML) V2.0 Technical Overview," 2008. Online Available at: https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[29] B. Fitzpatrick, "The OAuth 2.0 Authorization Framework," RFC 6749, Oct. 2012. Online Available at: https://tools.ietf.org/html/rfc6749

[30] Keycloak Project. "Managing user sessions", Keycloak Documentation, 2023. Online Available at: https://www.keycloak.org/docs/latest/server_admin/#managing-user-sessions

[31] GS1. "EPCIS and CBV Standard," GS1, 2020. Online Available at: https://ref.gs1.org/standards/epcis/

A. Requirements for RBAC in a Supply Chain Environment

1. **Granular Control Over Data Access**: Different supply chain stages demand varying access levels. Surveillance authorities, for instance, need comprehensive data for monitoring, while labs may require specific datasets for analytics.
2. **Scalability**: As the supply chain expands or diversifies, the access system should scale without issues.
3. **Auditability**: An effective RBAC system must transparently record data access patterns, facilitating compliance and monitoring.
4. **Integration**: The RBAC system should meld seamlessly with other supply chain tools and interfaces.
5. **Usability**: A user-friendly interface is essential both for administrators and end-users, including surveillance authorities and labs.

B. Designing and Implementing an RBAC Solution using Keycloak

1. **Role Definition**: Roles such as 'Supplier', 'Manufacturer', 'Distributor', 'Retailer', 'Logistics Manager', 'Surveillance Authority', and 'Analytical Lab' were established, each with tailored access privileges.
2. **Integration with EPCIS 2.0**: The compatibility of Keycloak made its integration with EPCIS 2.0 smooth, ensuring efficient data access and flow.
3. **Policy Creation**: Precise policies were formulated to dictate role-specific access rights within the EPCIS 2.0 environment.
4. **User Assignment**: Depending on their position in the supply chain or their specific surveillance or analytical function, users were assigned corresponding roles.
5. **Testing Phase**: Rigorous tests were run to ensure the RBAC system's security and efficacy.

C. Evaluating the Effectiveness of the RBAC Solution

1. **Audit Results**: The audit capabilities of Keycloak made transparent data access tracking possible, especially crucial for surveillance entities.
2. **User Feedback**: Stakeholders found the system not always intuitive bit still very efficient.
3. **Scalability Test**: The system exhibited flawless scalability in response to the increasing amount of test data created for simulated supply chains.

4. **Integration Success**: Its seamless integration with other tools and services, stood out as a major success point.

VII. Conclusion

Ensuring that the right people have access to the right information is crucial in today's data-rich environments. This becomes even more important in supply chains, where information flow is vital. Role-Based Access Control (RBAC) is a method used to manage this access. When combined with a tool like Keycloak and integrated into EPCIS 2.0 repositories, it can offer a straightforward solution.

A. Recap of RBAC and Keycloak's Role

RBAC – Simplifying Data Access
RBAC is a way to give users access based on their job or role. For example, a manager might have different access compared to an employee. The idea is simple: give people only the access they need.

Keycloak – Making RBAC Easier
Keycloak is a tool that helps manage user identities and their access. When it's used with EPCIS 2.0 repositories, it can control who sees what data.

EPCIS 2.0 – Organizing Supply Chain Data
EPCIS 2.0 repositories are used to store and share data in supply chains. This data is essential for operations, and it's important to control who can see and use it.

B. The Evolution of Access Control with EPCIS 2.0 and Keycloak Integration

Evolving Role Definitions
As we look to the future, it's clear that the static nature of role assignments in traditional RBAC will likely evolve into more fluid and situation-dependent models. This is particularly true in dynamic environments such as supply chains, where user responsibilities frequently shift.

Enhanced Contextual Role Adaptation in Supply Chains
A recognized limitation of conventional RBAC is its rigidity, especially within the fluctuating landscape of supply chain management. Users often oscillate between various roles based on operational needs or the nature of the items they handle. In a standard RBAC framework, this can result in inefficient access control that doesn't reflect the user's current context.

Moving forward, embracing a context-aware approach to role assignment will be crucial. Implementing C-RBAC could be transformative, enabling the system to adjust permissions in response to the user's real-time situation, considering factors like location, time, specific tasks, and the supply chain's present stage. This nuanced approach to permission management ensures that user access is always appropriate, secure, and tailored to the immediate demands of supply chain activities.

Integrating Artificial Intelligence
There is potential for RBAC systems, integrated with solutions like Keycloak, to harness artificial intelligence. AI could significantly enhance these systems by predicting access needs and dynamically adjusting permissions, streamlining the process for both users and administrators.

Innovative Access Control Mechanisms
With the advent of new technologies such as blockchain, the horizon of access management is broadening. Keycloak might become an even more versatile tool by adopting and incorporating these emerging methodologies for securing and managing access.

Refining the User Experience
As federated Access Control and SSO becomes more widespread, the emphasis on user experience is intensifying. Tools like Keycloak are anticipated to evolve, becoming more intuitive and efficient, thereby reducing the complexity and time investment typically associated with access management.

In Conclusion, the integration of Keycloak with EPCIS 2.0 repositories signifies a robust approach to data access governance in supply chains. This pairing is not only effective in today's technological landscape but also prepared for future enhancements as the realms of access control and technology continue to evolve.