Denodo Security Overview

Revision 20220427

Denodo Inc.: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA www.denodo.com

DENODO UNIFIED SECURITY

Security, data privacy, and data protection represent concerns for organizations that must comply with policies and regulations that can vary across regions, data assets, and personas.

Data virtualization offers a single logical point of access, avoiding point-to-point connections from consuming applications to the information sources. As a single point of data access for applications, it is the ideal place to enforce access security restrictions that can be defined in terms of the canonical model with a very fine granularity (e.g., access to “Bill,” “Order,” and so on).

Denodo has been successfully deployed in many organizations worldwide with strict security requirements. Those organizations benefit from Denodo's capabilities to customize security policies in the data abstraction layer, centralize security when data is spread across multiple systems residing both on-premises and in the cloud, or control and audit data access across different regions.

Denodo secures access from consumer applications to final data sources end-to-end. Typically this is established via SSL/TLS connections between the consumer and the Denodo Platform and by the specific data source security protocol between the Denodo Platform and the data sources (e.g., SSL, HTTPs, or sFTP). When SSL (TLS) is enabled on the Denodo servers, the version of TLS used depends on the configuration of the components involved in the communication. Although for clarity purposes we refer to this as SSL, SSL is not actually used, only TLS. Denodo uses TLS 1.2

The Denodo Platform supports user and role-based authentication and authorization mechanisms with both schema-wide permissions (e.g., to access Denodo databases and views) and data-specific permissions (e.g., to access the specific rows or columns in a virtual view). Denodo offers very fine-grained access up to the cell level (applying both row-based and column-based security) including the possibility of masking specific cells (e.g., managers are not allowed to view the “salary” column of higher-level management, so those cells would appear masked in the results). Denodo row-based security does not require any coding, and it can be defined graphically with the Denodo Administration tool.

When there is an authentication mechanism in place, Denodo can delegate authentication to an external LDAP or Active Directory server, so there is no need to define users in the built-in user management system, and the LDAP/AD system would provide the role for the user, which would be used to constrain the user’s access to any database or view within the data virtualization server. In addition, this option allows you to leverage password management policies as defined in the corporate LDAP/AD.

As a third alternative, it is also possible to connect to external custom entitlement services (through Custom Policies).

To grant access to a given data source through the data virtualization layer, the user can choose whether to make use of a service account for the source, in which case the Denodo Platform would always use the service account credentials to access the source, or to apply pass-through authentication and authorization in such a way that the security guardrails in the Denodo Platform are bypassed, and user credentials are directly used in the data source.

SECURITY ARCHITECTURE & PROTOCOLS

We include here a description of the security support in Denodo Platform. The following diagram highlights the Denodo Platform Security architecture.

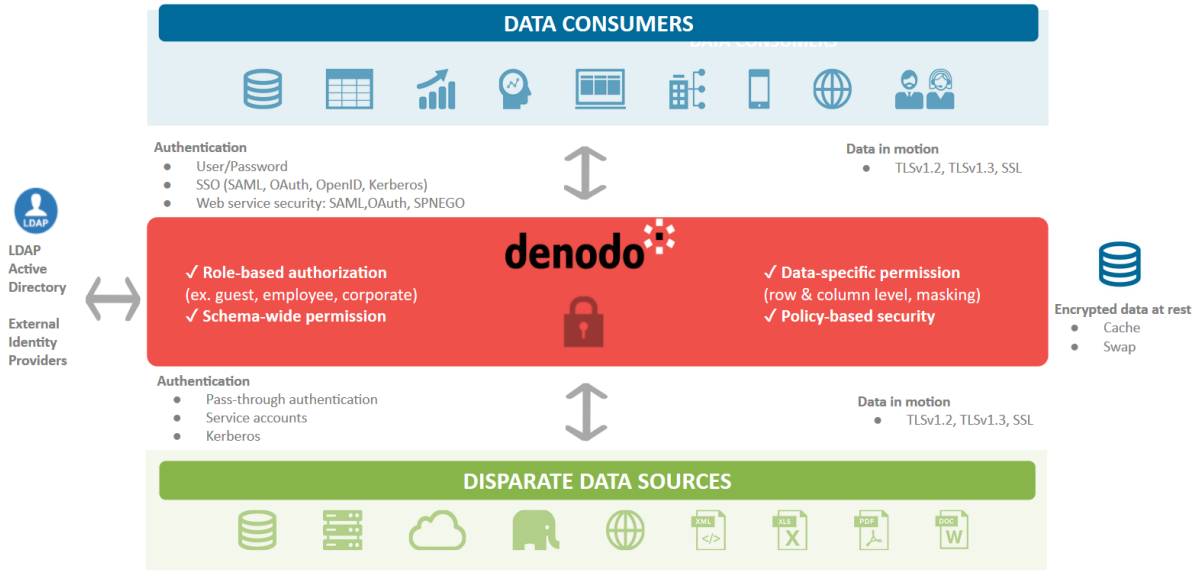


Figure 1: Denodo Platform Security Architecture and Protocols

The Denodo Platform Security Architecture has the following features:

- Unified Security Management offers a single point to control the access to any piece of information.
- Role based security access: Different users (e.g. User A, User B, and Guest) are only allowed to access either ‘filtered’ or ‘masked’ data by using the Denodo role-based security model. Denodo offers a low granularity level with regard to security that can be established in the following way:
    - Schema-wide permissions: each user/role can be assigned permissions (e.g. connect, create, read and write) to specific schemas (Denodo databases and views) so that different user profiles obtain different levels of control on the virtual schemas and the data they access.
    - Data-specific permissions: Denodo supports access permissions to specific rows (row-based security) or columns (column-based security) of virtual schema views, so that users/roles can be restricted from accessing specific pieces of data in the system. For example, non-managers may not be allowed to view the “salary” column on a virtual “Employee” view.
- Authentication credentials can be stored either internally in Denodo Platform in a built-in repository or externally in a corporate entitlements server such as an LDAP / AD repository. When stored in the built-in repository sensitive metadata such as users and passwords are stored encrypted.

- Also it is possible to use ‘custom policies’ to connect to any other corporate security system.
- Security protocols used to connect to data sources (southbound) or to publish data to consuming applications (northbound) are shown in the figure. In particular Denodo supports the following standards:
    - Northbound / Publish Interface:
        - Standard JDBC, ODBC, ADO.Net security mechanisms (username/password, SSL).
        - Kerberos support for JDBC, ODBC, ADO.Net, SOAP (SPNEGO), REST (SPNEGO), RESTful (SPNEGO) and OData (SPNEGO) interfaces and for the VDP administration tool.
        - HTTP-based authentication pass-through for the OData interface.
        - Web Service security with HTTPS, HTTP Basic/Digest, SAML 2.0, OAuth 2.0, HTTP SPNEGO (Kerberos) and WS-Security protocols.
    - Southbound / Data Source Connection:
        - Pass-through authentication including Kerberos.
        - Kerberos support for JDBC, Web Services (SPNEGO).
        - X509v3, SSL, WS-Security, HTTPS, HTTP Digest authentication, HTTP Basic Authentication, HTTP NTLM, OAuth 1.0a and 2.0 (JWT)
        - sFTP, FTPS
        - Anonymous Web browsing: through the use of an anonymizer server (i.e. anonymous proxy).
        - Denodo Virtual DataPort provides support to obtain the credentials of JDBC data sources from an external Credentials Vault.

- Denodo offers different alternatives to integrate with identity, authentication and authorization services:
    - Denodo built-in security.
    - Integration with external entitlement service (LDAP/AD).
    - SSO using OAuth, Kerberos and SAML
    - Integration with external custom entitlement service with specific security policies (Custom Policies).
- The Denodo Platform provides an audit trail of all the information about the queries and other actions executed on the system. Denodo will generate an event for each executed sentence that causes any change in the Denodo Catalog (store of all the server metadata). With this information it is possible to check who has accessed specific resources, what changes have been made or what queries have been executed. All Denodo components include configurable multi-level logs based on the Log4J standard, and they can be configured to log specific information that can be later used for compliance and auditing purposes
- Data in Motion (securely accessing and delivering data)
    - All communication between the Denodo Platform and the Data Consumers/Data Sources, as well as between the different modules within the Denodo Platform, can be secured through SSL/TLS at the connection level.
        - When SSL (TLS) is enabled on the Denodo servers, the version of TLS used depends on the configuration of the components involved in the communication. Although for clarity purposes we refer to this as SSL, SSL is not actually used, only TLS. Starting with version 8, Denodo uses TLS 1.2. If TLS is enabled on the web components, only TLS 1.2 and TLS 1.3 connections are allowed as

previous SSL/TLS protocol versions are disabled as they are no longer considered secure.
        - Starting with Denodo 8, the SSL/TLS configuration is automatized via the SSL/TLS Configurator Script
        - It supports any encryption algorithm supported by the default Java Cryptography Providers of the Denodo Platform JRE (Java 8), or by any additional provider, registered as part of the Denodo Platform JRE.
        - You can look here for supported cipher suites. If you want to enable the strongest ciphers available to JDK 8 you need to install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files (here)

    - If security at the connection level is not required, Denodo´s built-in functions for encryption/decryption can be selectively applied to sensitive fields to prevent unauthorized access.

- Data at rest (secured caching of sensitive data or storage in staging area)
    - When working in cached mode, Denodo will transparently leverage any encryption mechanism available in the selected Cache System. For example, Oracle Transparent Data Encryption (TDE) allows sensitive data to be encrypted within the data files to prevent access to it from the operating system. This would be transparent to Denodo.
    - When security at the dataset level is not required, it's possible to selectively apply encryption/decryption only to sensitive fields using Denodo's built-in functions. These functions support any encryption algorithm supported by the default JCE of the Denodo Platform JRE, or by any additional provider registered as part of the Denodo Platform JRE.
    - Swapping: when the allocated memory buffers for query execution are full, Denodo swaps data to disk. There are two possibilities to deal with this data:
        - Use your OS file system encryption for the folder where Denodo manages those swap files. It will be transparent to Denodo
        - Disable swapping. Only advisable when dealing with small result sets. Configurable by view and database

USER & ROLE MANAGEMENT

Denodo´s fine-grained, role-based access control (RBAC) includes row- and column-level permissions, full integration with LDAP/Active Directory for sourcing user identities and group-based authorizations to virtual views.

Instead of creating the roles manually, you can import them from an LDAP server. Denodo Roles, defined in a Denodo Virtual Database, aggregate permissions on individual users (defined externally in LDAP/AD or built-in as Denodo virtual database users) for accessing virtual database schemas (data sources, views, web services, stored procedures), etc.

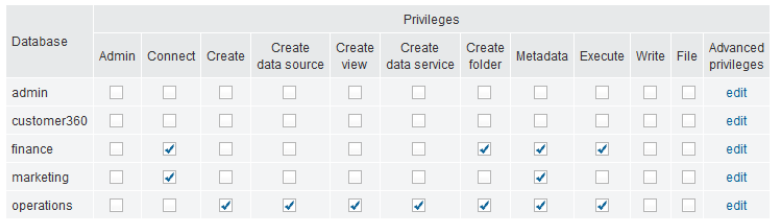
Denodo distinguishes two types of users:

- ‘Administrators’ can create, modify and delete databases without any limitation. Likewise, they can also create, modify and delete users. When the server is installed, a default administrator user is created with user name admin. There must always be at least one administrator user. The default administrator user can be deleted as long as there is at least one administrator user.
- ‘Normal users’ cannot create, modify or delete users or databases. Administrators can grant them connect, metadata access, execute, create, write privileges to one or several databases or to specific views contained in them. An Administrator can promote a normal user to ‘local administrator’ of one or more databases, which means that this user will be able to perform administration tasks over these databases.

Denodo users can have roles, i.e. sets of access rights over databases, virtual views and stored procedures. Roles allow administrators to manage user privileges easily because by changing the privileges assigned to a role, they change the privileges of all the users assigned that role.

Denodo´s access rights are applied to a specific user or a role, to delimit the tasks they can perform over databases, views and stored procedures. Access rights can be applied globally to a database or specifically to a view/stored procedure in a specific database:

- Schema wide permissions: each user/role can be assigned permissions to specific databases, views and stored procedures so that different user profiles obtain different levels of control on the virtual schemas and the data they access. Denodo supports the following types of global database access rights: Connect, Create (different levels including data sources and/or views), Metadata (i.e. access to the schema definition but not to the data), Execute, Write and File. Denodo also supports individual privileges to specific views and stored procedures within a given database. The privileges that can be applied to a specific view and/or stored procedure of a database are: Metadata, Execute, Write, Insert, Update, and Delete.



Database	Privileges											
	Admin	Connect	Create	Create data source	Create view	Create data service	Create folder	Metadata	Execute	Write	File	Advanced privileges
admin	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	edit
customer360	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	edit
finance	☐	☑	☐	☐	☐	☐	☑	☑	☑	☐	☐	edit
marketing	☐	☑	☐	☐	☐	☐	☐	☑	☐	☐	☐	edit
operations	☐	☐	☑	☑	☑	☑	☑	☑	☑	☐	☐	edit

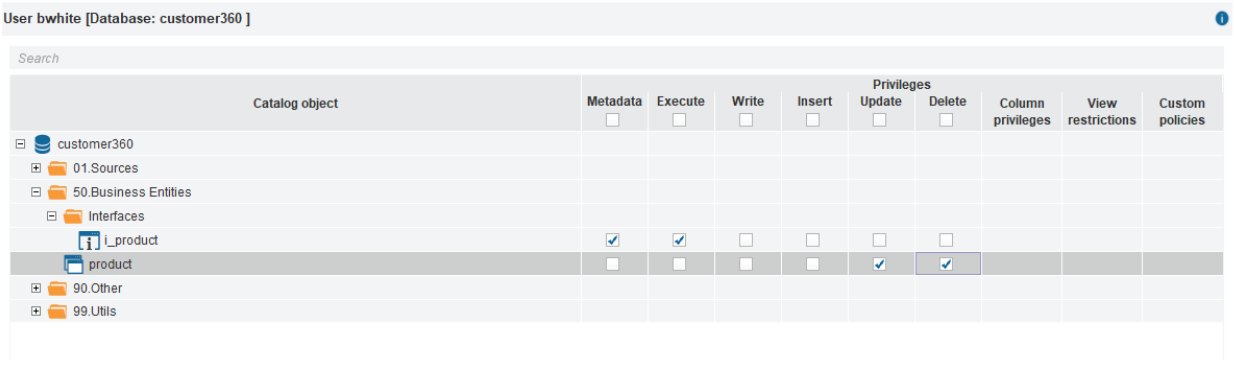


Figure 2: Schema-wide and data specific privileges assignment

- Data specific permissions: with the above Read privilege on a view, a user (or a role granting same), can query this view to obtain all its data. However, if certain users or roles should have access to only some of the columns of a specific view, Denodo supports 'Column privileges' and 'Row privileges' to further constrain Read access. For example, non managers may not be allowed to view the "salary" column on a virtual "Employee" view.

In addition, Denodo server defines some special roles that are created during the installation process and cannot be modified or deleted. Those roles allow you to grant users for, for example, modifying the settings of the Denodo server, the Data Catalog (different roles for granting privileges for administrators, content editors, classifiers, catalog exportation, etc.), the Solution Manager (different roles for granting privileges for administrators, promotions, etc.), the Diagnosing and Monitoring tool or the Scheduler, monitoring the Denodo server via JMX, or granting/revoking privileges to other users.

Starting with Denodo 8, the users and roles management for Solution Manager users can be done in an unified interface to manage roles and users available in the *Solution Manager Administration Tool,* while in previous versions it is needed to connect to the Virtual DataPort Server of the Solution Manager for these tasks.

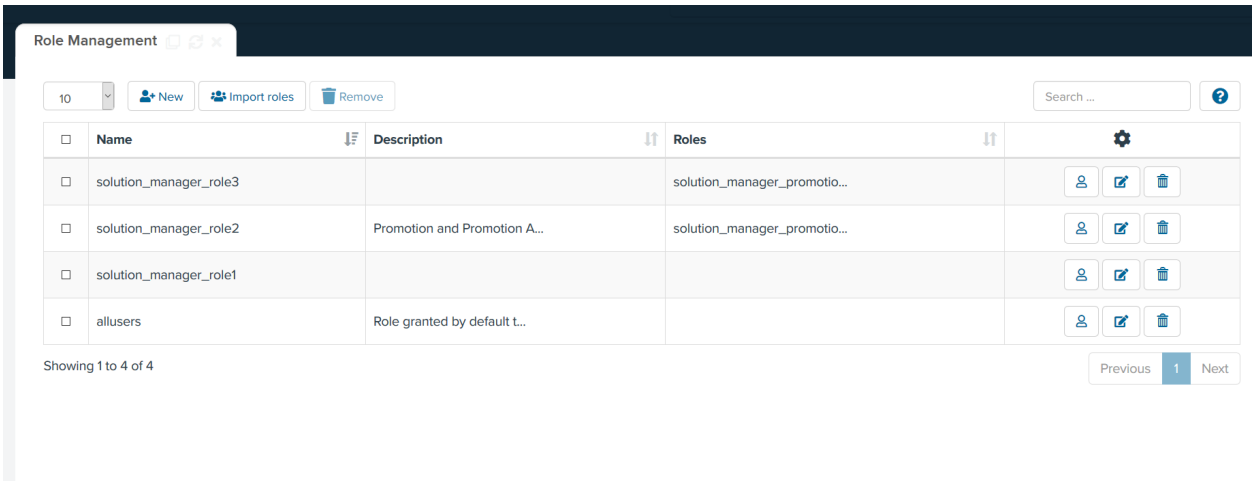


Figure 3: Role Management interface in the Solution Manager

SOLUTION MANAGER - USER & ROLE MANAGEMENT

Introduced in Denodo 7, the Solution Manager is the centralized administration console of Denodo installations. It allows the management of the nodes of a Denodo cluster, license operations and code promotions just to name a few important operations. For an overview of its role and architecture please refer to this article.

Denodo 8 introduced more fine grained privileges in the Solution Manager giving administrators the control over the tasks users can do over each environment.

There are two types of privileges:

- Global privileges: granted to a user or a role over all the environments or over all the environments of a certain type (for example global administration, promotion administration ...)
- Environment wide privileges: granted to users over a specific environment (for example connect, read, write, deploy ...).

HIERARCHICAL ROLES

Roles can be “hierarchical”. Once a baseline role is established, another role can be created which “inherits” and refines it. These “role hierarchies” can be built to any depth within Denodo.

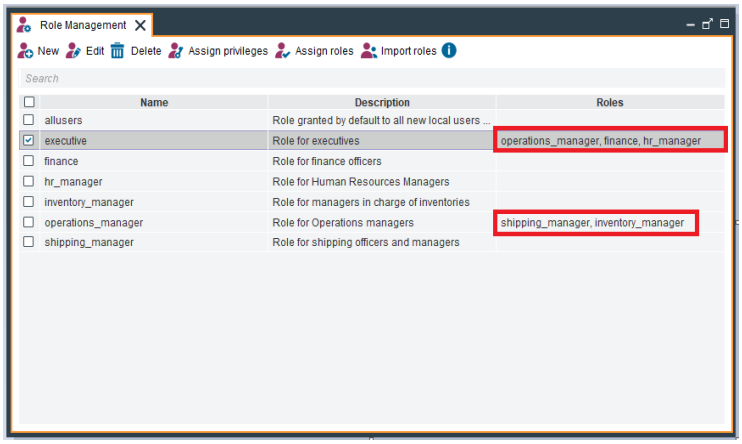


Figure 3 : Creating role hierarchies

AUTHENTICATION THROUGH LDAP/AD

Denodo can delegate the authentication to the corporate LDAP/Active Directory system. Roles can also be imported from the corporate LDAP/Active Directory at any time, and then configured to constrain access to Denodo's virtual databases and views.
A database with LDAP authentication delegates the authentication of users to an LDAP server. The benefit over the Denodo built-in authentication is that you can rely on an LDAP server such as the Microsoft Windows Active Directory, to authenticate users without having to create them in Denodo. In addition, this option allows you to leverage password management policies as defined in the corporate LDAP/AD.
Once authenticated, Denodo gets the names of the roles that the users belong to, from the LDAP server and uses them to check which actions users can do according to the security policies defined in Denodo.
When a user tries to connect to an LDAP database, the Server checks first if the user is a Virtual DataPort "administrator". If not, it connects to an LDAP server to check the credentials and obtain the roles of the user.

Starting with Denodo 8, the LDAP configuration can be done at the server-level so that the databases inherit by default this configuration. If a database needs a custom LDAP configuration, it can be configured to overwrite the default one.
Finally, the latest version Denodo 8 allows modification of OAuth 2.0, SAML 2.0 and Kerberos settings without server restart.

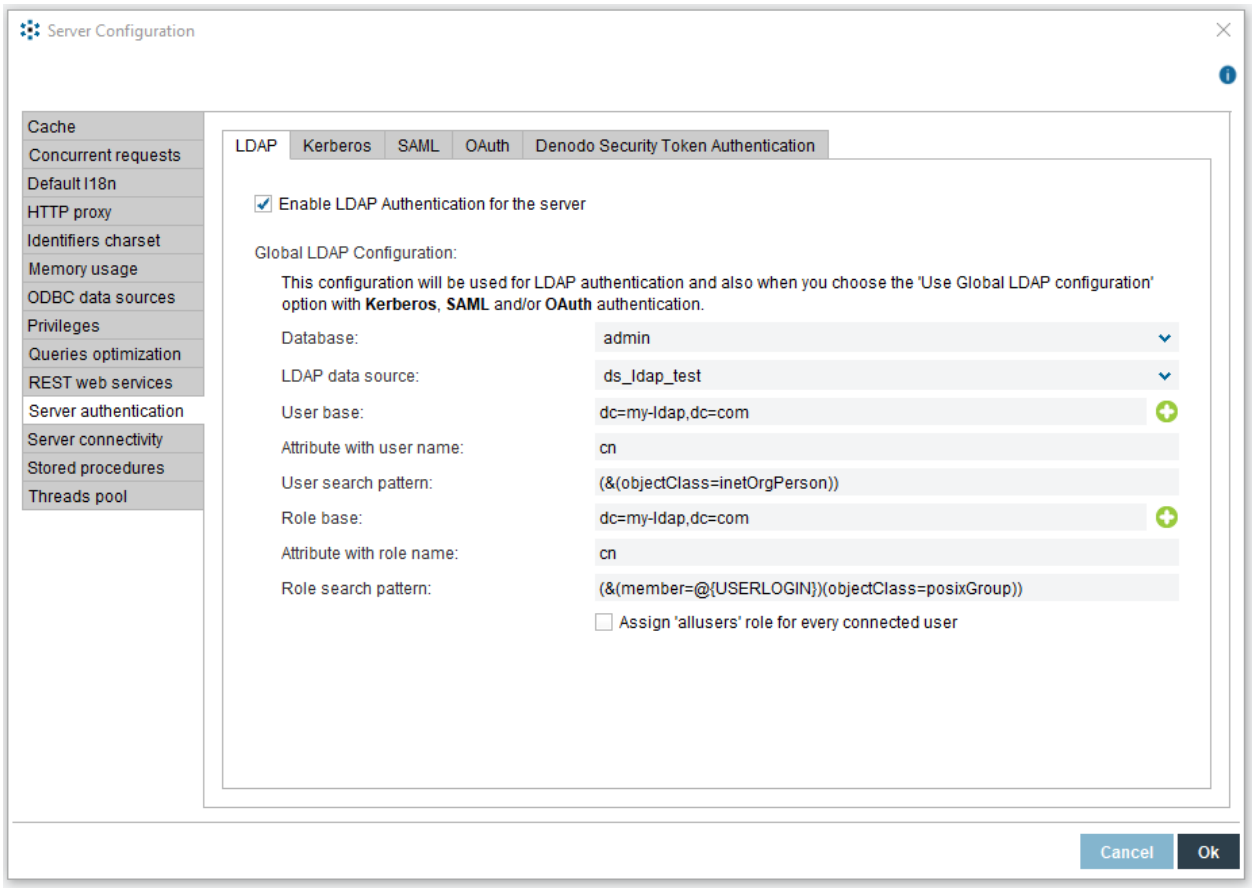


Figure 4 : Configuring a server-wide LDAP authentication

MULTI-FACTOR AUTHENTICATION

Denodo supports multi-factor authentication in published web services (Denodo 7 and 8) and web applications (Denodo 8). It is done by integrating, via SAML or OAuth2, Denodo with Identity Providers providing 2FA.
Denodo currently supports Okta, Duo, PingFederate, Azure AD and AWS SSO, although it could be integrated with any other Identity Provider compliant with SAML and/or OAuth2.

ROW AND COLUMN LEVEL SECURITY AND DATA MASKING

Denodo can enforce strict, fine-grained user and role-based permissions for each and every element defined within it. Denodo´s authorization policies can implement row and column security. These policies are specified by view, by row (specified with a selection condition), by column or by row-column combination (i.e. rows restricted but only for restricted columns), and are evaluated prior to each query execution to determine if the customer is allowed to see particular results or not. The non authorized data can either be 'filtered' (i.e. removed from the results of the query) or 'masked'.
For example, Denodo´s row-level security would allow hiding the salary of people with position=’manager’ when querying a ‘salary’ view from users with an ‘employee’ role (thus not entitled to access salary information).

SINGLE SIGN-ON

Denodo supports SSO using OAuth, OpenID, Kerberos and SAML. Denodo currently supports Okta, Duo, PingFederate, Keycloak, Azure AD and AWS SSO, although it could be integrated with any other Identity Provider compliant with SAML and/or OAuth2.

When SSO authentication is delegated to an external Identity Provider, starting with Denodo 8 the Denodo Security Token system will take care of delegating the authentication, extracting the roles from the delegated authentication object and issuing temporary credentials.

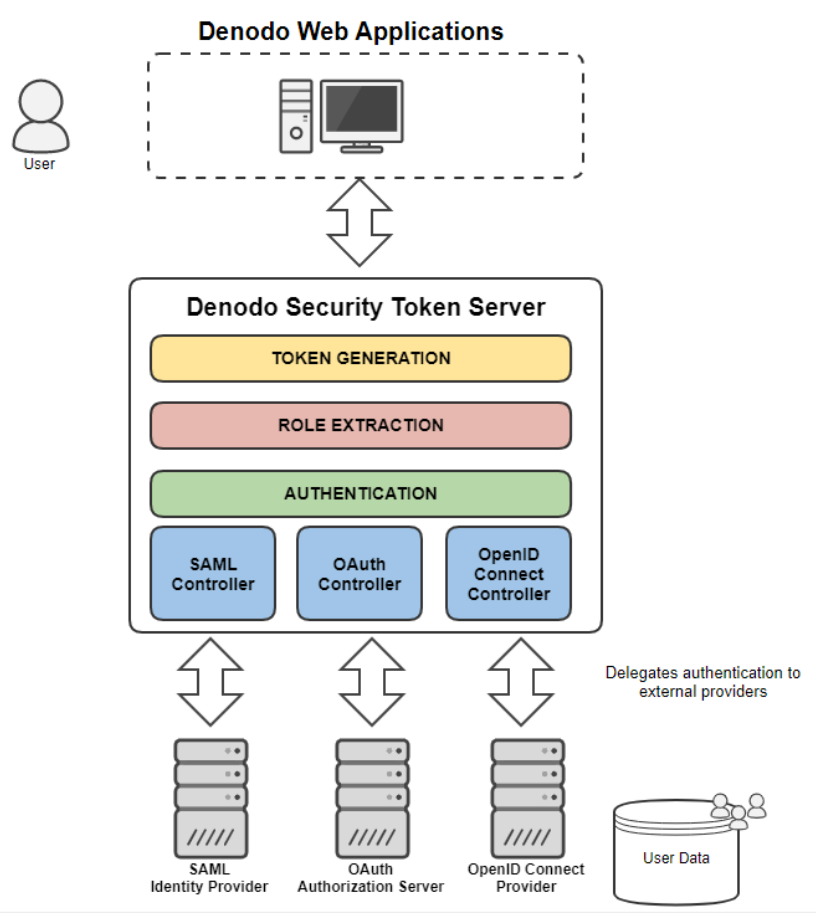


Figure 5: Denodo Security Token Architecture

CACHE

When accessing cached data, the same security restrictions defined in Denodo of the user/role on a given database, view, columns and/or rows are taken into account.

POLICY BASED SECURITY

Custom Policies allow developers to provide their own access control rules. Similar concepts already exist in some systems such as Oracle (Virtual Private Database Policies). This feature would work as follows: developers can code their own custom access control policies and the administrator can assign them to one (or several) users/roles in a view in Denodo.
Custom policies can also be used to implement Attribute Based Access Control (ABAC).

When a user queries a view with a custom policy assigned, the policy can take one of the following actions:

1. Reject the query.
2. Accept the query without applying any restrictions.
3. Or, accept the query but impose some restrictions such as limit the rows returned by the query, add a filter condition, etc.

Custom policies have access to the full context of the query (user, projected fields, query executed, interface, IP from where it connects to Denodo, etc.) in order to accept or reject it.
Custom policies can be used to integrate an external Policy Server (e.g. Axiomatics) to provide dynamic authorization based on policies defined in that server.

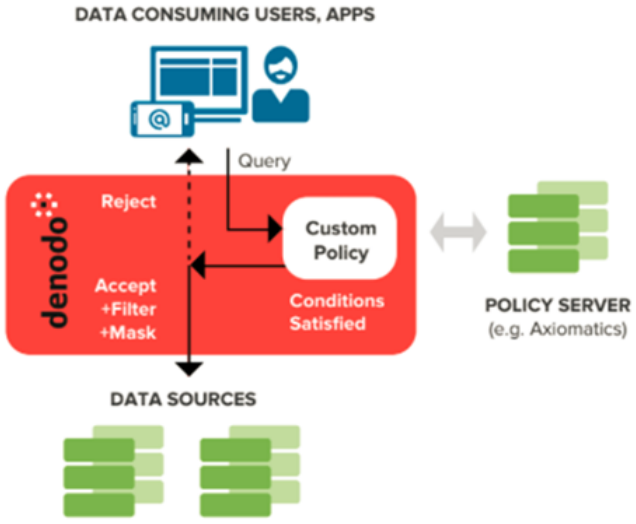


Figure 6 : Policy-based security

Examples of possible uses: set limits to the number of queries executed by a certain user/role; determine if a query can be executed depending on the time of the day or leveraging the access policies in an external policy server.

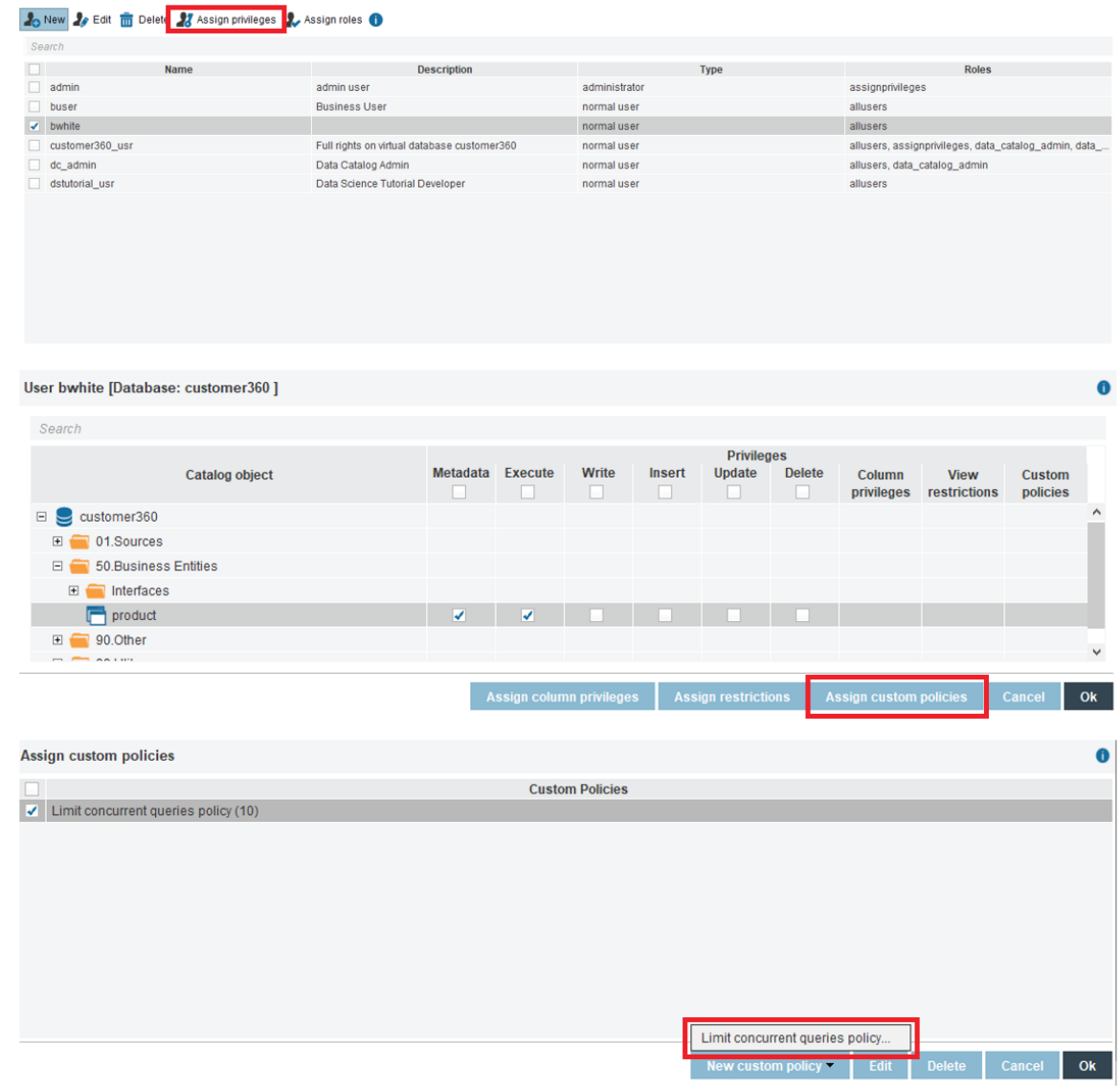


Figure 7: Dynamic Authorization based on policies

ENCRYPTION

Denodo's hybrid approach to data integration, allows different data access & delivery modes, all of which may involve securely accessing sensitive data: real-time from the data sources; from the Denodo cache; or from a staging area (i.e. ETL-like process where data is moved from its original data source to an external repository).

In order to cover all possible scenarios, Denodo supports the application of strategies on a per view basis to guarantee secure access to sensitive data through encryption/decryption at different levels.
Data at rest (secured caching of sensitive data or temporal storage)

- When working in cached mode, Denodo will transparently leverage any encryption mechanism available in the selected Cache System. For example, Oracle Transparent Data Encryption (TDE) allows sensitive data to be encrypted within the data files to prevent access to it from the operating system.
- When full encryption at the dataset level is not required, it's possible to selectively apply encryption/decryption only to sensitive fields using Denodo's built-in functions. These functions support any encryption algorithm supported by the default JCE of the Denodo Platform JRE, or by any additional provider registered as part of the Denodo Platform JRE.
- Swapping: When the allocated memory buffer for query execution is full, Denodo might swap part of the data to disk. In order to protect such data from unauthorized access you must encrypt the swap folder by using OS file system encryption.

Data in motion (securely accessing and delivering data)

- All communication between the Denodo Platform and the Data Consumers/Data Sources, as well as between the different modules within the Denodo Platform, can be secured through SSL/TLS at the connection level.
    - When SSL (TLS) is enabled on the Denodo servers, the version of TLS used depends on the configuration of the components involved in the communication. Although for clarity purposes we refer to this as SSL, SSL is not actually used, only TLS. Denodo uses TLS 1.2
    - It supports any encryption algorithm supported by the default Java Cryptography Providers of the Denodo Platform JRE (Java 8), or by any additional provider, registered as part of the Denodo Platform JRE.
    - You can look here for supported cipher suites. If you want to enable the strongest ciphers available to JDK 8 you need to install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files (here)
- When full encryption at the transport level is not required, Denodo's built-in functions for encryption/decryption can be selectively applied to sensitive fields to prevent unauthorized access.

AUDITING

The Denodo Platform provides an audit trail of all the information about the queries and other actions executed on the system. Denodo will generate an event for each executed sentence which causes any change in the Denodo catalog (which stores all the server metadata). With this information it is possible to check at any time who has access to which resources, what changes have been made or what queries have been executed, and when it happened.
The information is stored centrally in log files or to an auditing database. Denodo provides a web tool, the Denodo Monitor Reports to visualize this data in a variety of reports. Denodo log files are standard CVS files, and it supports SNMP, JMX and WS-Management standards. This enables Denodo to integrate with third party Security Information and Event Management (SIEM) solutions.
In addition, Denodo provides a metadata API which allows to create reports on access privileges per user/role for regulatory compliance.

GDPR COMPLIANCE

The Denodo Platform does not store data but only metadata, used to access the heterogeneity of customer's data sources to provide integrated near real-time data for business users. The Denodo Platform is data agnostic. The nature of the data accessed through Denodo is customer’s sole decision.
With that said, the functionality provided by Denodo had helped our customers to comply with GDPR, HIPAA and other data security and privacy regulations. Denodo helps in ensuring secure transfer of data, consistently applying access privileges across data sources, providing data lineage to understand from where data is queried, providing full audit capabilities to understand who is accessing which data, masking sensitive data to ensure it is not accessed by unauthorized users, etc.
This infographic explains how data virtualization can comply with some of the most important principles of the GDPR.

SENSITIVE DATA ENCRYPTION

Sensitive information, such as service accounts to access data sources and users accounts are stored encrypted or hashed in the Denodo metadata repository (embedded Derby database).
In addition, you can enable transparent data encryption to encrypt the Derby database so it would apply not only to sensitive data, but to all the metadata. After enabling this feature, the metadata is transparently decrypted when it is accessed so the users do not need to be aware that the metadata they are accessing is encrypted, nor they have to change any setting on their end.
The Transparent Metadata Encryption is unrelated to how the data is transmitted across the network.
Sensitive data is exported in encrypted format (e.g. during migrations between environments or backup). Optionally, administrators can use a custom password for sensitive data encryption. If selected, the given password will be used to encrypt sensitive data in the generated VQL file. A VQL file generated using this option will require the password while importing it in another environment.