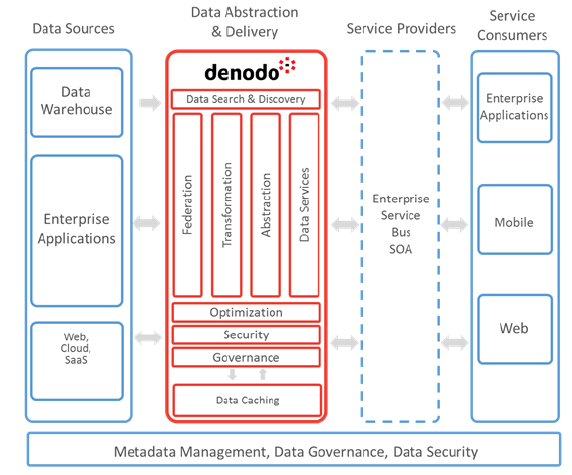
**Notes About Denodo’s Toolkit**

### 1. Overview

Denodo is a leader in data management that empowers everyone to integrate, manage, and deliver data securely in order to make better decisions. Denodo helps business users to immediately and easily access the data they need without worrying about where it is housed, how it is formatted, or how quickly it changes. We make this vision a reality for our customers, transforming the way they innovate and operate their businesses.

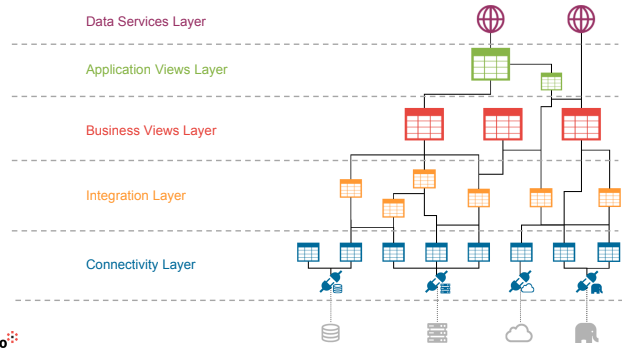
What makes Denodo different is the capacity to develop **data virtualization** tasks. Data virtualization is the core technology that enables modern data integration and data management solutions without replicating or moving the data. Allows you to integrate, manage and deliver data from across multiple sources through a logical layer.



DV is a critical part of logical data warehouse architecture enabling queries to be federated across multiple data sources – both traditional structured data sources, such as databases, data warehouses, etc., and less traditional data sources, such as Hadoop, NoSQL, Web Services, SaaS applications, and so on – while still appearing as a single ‘logical’ data source to the user.

The route that Denodo follows is: connect, combine and consume data. Connect to data sources from a wide range of possibilities, combine and integrate base views using relational operations and build reusable derived views to create data entities needed by users and last but not least, delivering these views to the consumer in required format and using required protocols. In order to follow this route, Denodo executes an architecture called bottom up that is based on connecting to data sources and then building upwards (a layer at a time) so as to finally create views exposed to users. This workflow has 5 layers:

* **Connectivity layer:** contains connections to data sources and base views that are closely related to the physical systems.
* **Integration layer:** contains derived views with all combinations and transformations needed for the business entities.
* **Business Views layer:** canonical model exposed to all users that use associations to represent relationships between entities.
* **Application Views layer:** views consumed directly by applications and tools (application-specific models, transformations and formatting and pre-built reports and analysis frequently consumed by users).
* **Data Service layer:** contains web services that publish views from other levels. Might include intermediate views needed for web service-related formatting and manipulation. Interface views can act as an API definition, isolate consumers from changes in the model.



### 2. Metadata management

Denodo is not an 'inventory-style' Data Catalog solution like others in the market. Instead, it integrates heterogeneous, distributed, structured, and semi-structured data sources in real-time without storing the data.

The platform serves as the primary access point for "certified data" within an organisation. This certified data is governed and secured by IT, ensuring compliance with internal policies and governance frameworks.

#### Metadata management capabilities

Denodo offers robust metadata management features comparable to traditional data catalogues but tailored to its unique delivery model. These include:

* **Discovery and Import**: Automatic discovery of metadata from a wide array of sources (databases, web services, XML, JSON, LDAP, Salesforce, etc.) with the ability to import these metadata structures into Denodo's platform.
* **Data Relationships**: The platform allows defining and managing relationships between data entities, including foreign and primary key associations and programmatically inferred associations between data sources that do not share schema-level relationships.
* **Metadata Cataloging**: Denodo maintains a catalog of data entities and their metadata, offering tools for graphical management and discovery.
* **Data Lineage and Impact Analysis**: Denodo enables tracing the transformation of data from source systems to its final delivery, allowing for an understanding of how data flows and changes through the platform. It also provides tools for analyzing the impact of changes in underlying source schemas on the data delivery process.
* **Third-Party Metadata Integration**: Denodo's metadata can be shared with external tools such as business glossaries, data quality tools, and other metadata management solutions.

#### Realtime metadata synchronization

Denodo supports the synchronization of logical models with actual data sources, automatically detecting schema changes and allowing users to propagate these changes within the Denodo platform if desired. Through tools like the **Denodo Model Bridge**, users can import data models from external modeling tools (ERwin, Embarcadero E/R Studio, InfoSphere Data Architect, etc.) into Denodo, creating logical "interface views" that map to physical data sources.

#### Data Governance

Denodo provides extensive data governance capabilities by leveraging data virtualization. This allows the platform to centralize data access, governance, and security policies across diverse and distributed systems. Whether data resides on-premises or in the cloud, Denodo ensures consistent governance practices. Key governance tools include:

* **Data Lineage:** Users can visualize data lineage to understand the source and transformations applied to data fields and views, aiding in optimizing query performance and ensuring compliance with governance policies.
* **Impact Analysis:** Changes in data sources can be visualized and assessed for their impact on the data flow, ensuring that downstream data consumers are not negatively affected by schema modifications.

#### Denodo Data Catalog

The Denodo Data Catalog is a self-service, web-based tool designed to enhance the discovery and usability of data for business users. It offers several important features:

* **Google-Like Search**: Users can perform free-text searches across indexed views without needing to understand the underlying schema. Searches can be refined using filters based on databases, categories, or tags.
* **Graphical Representations**: Users can access visual representations of business entities, associations, and data lineage. These visuals can be exported as images for further use.
* **Query Creation and Sharing**: Users can create their own queries, save them, and share them with others via links. Queries can also be exported to different formats such as CSV, Tableau workbooks, or HTML.
* **Metadata Editing and Management**: The catalog allows users to manage metadata descriptions for views and view attributes, promoting reusability and collaboration.

#### Integration with Third-Party Tools

Denodo supports integration with external metadata management solutions via its **Metadata API**, facilitating seamless interaction with tools such as:

* **Collibra Data Governance Center (DGC)**, **IBM Information Governance Catalog (IGC)**, and **Informatica Metadata Manager (IMM)**. This integration allows for enterprise-wide metadata management, extending Denodo's capabilities by incorporating data lineage from other systems like ETL jobs that are outside Denodo's scope.
* Denodo also provides specific tools, such as the **Denodo Governance Bridge**, to synchronize Denodo metadata and data lineage with IBM IGC, enabling governance of Denodo’s virtual databases within the broader enterprise data governance framework.

#### Advanced features for developers and administrators

**APIs for Metadata Management**: Denodo provides a robust **Metadata API** that allows programmatic management of metadata through built-in stored procedures. These APIs can be invoked via SQL (using JDBC, ODBC, or ADO.Net clients) or as web services (SOAP/RESTful) for lightweight client interactions. This flexibility allows seamless integration with corporate processes or third-party tools.

* **Metadata Retrieval**: Through stored procedures like CATALOG\_VDP\_METADATA\_VIEWS and VIEW\_DEPENDENCIES, developers can programmatically retrieve metadata details about views, their fields, and their dependencies.
* **Column Lineage**: The COLUMN\_DEPENDENCIES procedure allows tracing the dependencies of individual fields in views, providing visibility into how columns are derived or transformed as they move through various data sources and views.
* **Automatic View Generation**: Denodo provides stored procedures to automate the generation of views, such as GENERATE\_VQL\_TO\_CREATE\_JDBC\_BASE\_VIEW, which can be used to create views over JDBC data sources programmatically.
* **Impact Analysis**: The platform also supports tools to visualize the impact of schema changes, which can be triggered automatically using the **Source Refresh API**.

**Graphical Tools**: In addition to APIs, Denodo offers graphical tools within its **Design Studio** that enable users to manage metadata, create and manipulate views, and analyze data flows visually.

* **Tree View for Data Lineage**: This feature allows users to see the composition of views, the operations performed on them (joins, selections, etc.), and the data sources involved. Users can click on nodes to view detailed properties of combining operations or data sources.
* **Impact and Usage Analysis**: Denodo enables users to assess which views depend on a particular element, providing insight into how data changes could impact downstream users and systems.

**Data Lineage and Change Tracking**: Denodo supports both view and column-level lineage tracking, allowing organizations to monitor the flow and transformation of data across various systems. This is critical for auditing, performance optimization, and regulatory compliance.

#### Security and Customization in Denodo Data Catalog

The **Denodo Data Catalog** provides a secure and customizable environment for users to explore both data and metadata. It inherits the security restrictions defined at the virtualization layer, ensuring that users only see data they are authorized to access.

* **Customization Options**: Administrators can tailor the catalog to suit the organization’s needs by configuring which elements are visible (views, web services, etc.), customizing the branding (logos, background images), and setting limits on the export of results.
* **Role-Based Access**: The platform allows fine-grained control over who can access, view, and edit metadata and data assets based on user roles. Administrators can also define metadata properties like ownership, responsibility, and validation statuses, ensuring proper data governance.
* **Metadata Management for Compliance**: Denodo’s support for custom properties in the metadata catalog can help address data quality and privacy regulations, by associating data elements with responsible individuals, validation statuses, and other regulatory requirements.

### 3. Denodo's Performance

Denodo 8 is designed to offer high performance through sophisticated query optimization techniques, which include dynamic execution plans and extensive caching capabilities. The platform optimizes queries by moving as much processing as possible to the data sources, thereby reducing network traffic and minimizing the load on Denodo’s virtualization engine.

Key features:

* **Query Delegation**: By processing queries at the data source, Denodo minimizes the data that needs to be transferred, improving efficiency.
* **Dynamic Query Rewriting**: Denodo rewrites queries dynamically to maximize query pushdown, optimizing for source processing power.
* **Advanced Optimization Techniques**: Includes aggregation pushdown, star schema optimization, and various join strategies.
* **Cost-Based Optimization**: The query optimizer uses a cost model based on factors like network latency, source indexing, and source constraints to choose the best execution plan.

#### Query Optimization Techniques

Denodo 8 incorporates numerous optimization strategies for maximizing performance:

* **Aggregation Pushdown**: Full and partial aggregations are pushed down to data sources to reduce the data load processed by Denodo.
* **Join Optimization**: Denodo uses multiple join strategies (merge, hash, nested) and reorders them dynamically to optimize execution.
* **Data Movement Optimization**: Denodo moves small datasets to larger datasets to optimize joins.
* **Partition Pruning**: For union views across multiple data sources, Denodo only processes the relevant branches, improving performance.
* **Alternative Sources**: The optimizer can select between multiple data sources for the same data, choosing the one that allows the most query pushdown.
* **Constraint Propagation and Query Rewriting**: Selections and projections are pushed down as deeply as possible to avoid unnecessary data processing.

#### Denodo Query Execution Model

Denodo’s query execution process follows several steps:

* **Execution Plans**: The platform generates multiple candidate execution plans for each query and selects the optimal one based on cost-based optimization.
* **Sub-Query Execution**: Queries are decomposed into sub-queries executed in parallel across data sources.
* **Result Streaming**: Denodo provides asynchronous operation, streaming results back to the client as they become available without waiting for the full query to complete.

#### MPP Query Acceleration

Denodo supports native integration with massively parallel processing (MPP) engines like Spark, Hive, Impala, and Presto to accelerate query execution in Big Data scenarios. This integration enables:

* **Parallel Processing**: Query processing is distributed across MPP clusters for improved performance.
* **Dynamic Query Optimization**: Denodo dynamically chooses between in-memory execution and delegation to MPP engines, automatically generating Parquet files for optimized data transfer to MPP systems.

#### Caching

Denodo’s caching system supports flexible, distributed caching strategies to boost performance:

* **Partial Cache**: Caches query results based on frequently accessed data, optimizing for temporal locality.
* **Full Cache**: Functions like materialized views, caching entire datasets to avoid querying the data source.
* **Incremental Cache**: Combines cached data with real-time data from the source, merging results to ensure up-to-date information.
* **Distributed Caching**: Supports high availability and shared cache across multiple Denodo servers in clustered environments.

#### Smart Query Acceleration

Denodo includes the **Smart Query Acceleration** feature, which uses precomputed summaries of transactional data to speed up large-scale analytical queries. This feature:

* **Cost-Based Analysis**: Denodo intelligently determines whether to use precomputed summaries for specific queries, improving execution times significantly.
* **Performance Gains**: Analytical queries benefit from up to 100x performance improvement when using these summaries.

#### Resource Management

Denodo includes a **Resource Manager** that allows administrators to define workload management policies:

* **Workload Plans**: Administrators can define plans that restrict query execution based on resource consumption (e.g., memory, CPU) or user priority.
* **Throttling**: Denodo supports client, server, and data source throttling to manage concurrency, limit resource usage, and prevent overloading.

#### Performance Monitoring

Denodo provides detailed performance monitoring, capturing metrics like query response times, memory, and CPU usage at a granular level. This allows administrators to proactively identify bottlenecks and manage server load effectively.

#### Performance Benchmarks

Denodo benchmarks focus on comparing performance in real-world scenarios (such as TPC-DS queries) rather than simple throughput metrics. Denodo consistently outperforms other distributed engines in multi-source query federation and complex data virtualization scenarios.

#### Scalability

Denodo supports both vertical and horizontal scaling:

* **Vertical Scaling**: Denodo can leverage multi-core architectures for parallel processing within a single server.
* **Horizontal Scaling**: In clustered environments, Denodo distributes workload across multiple servers with shared cache for higher availability and performance. The scalability is nearly linear in horizontal configurations, with minimal overhead added by the load balancer.