



Data Virtuality Platform

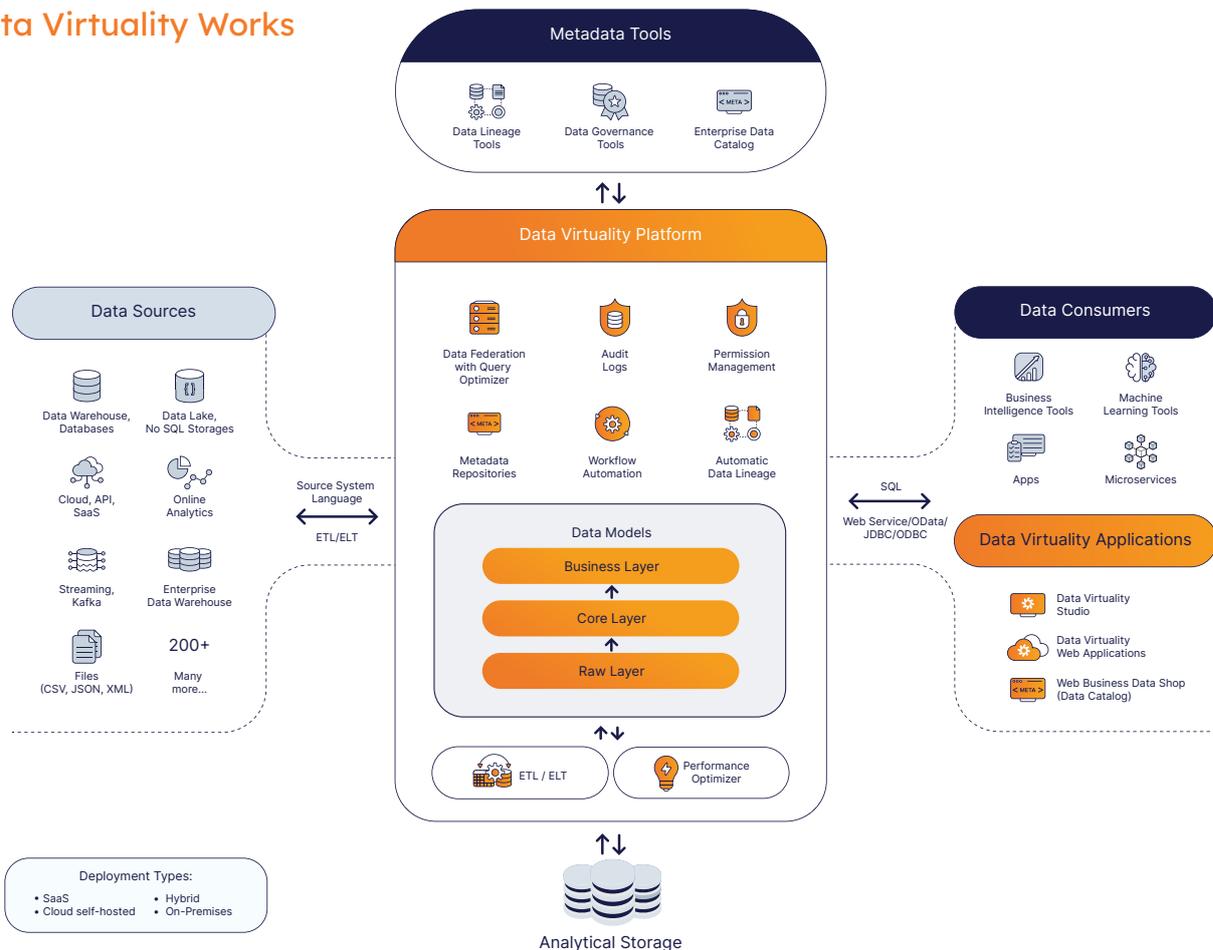
DATASHEET

The Data Virtuality Platform is a step up in the evolution of data integration, combining the best of data virtualization and ELT/ETL in a single, comprehensive data management solution deployed as SaaS (software as a service), hybrid, or on-premises:

- The data virtualization engine allows to retrieve data using a single query language in near-realtime and assemble different data models or views of the data from disparate data sources. The engine contains a large number of optimizations to ensure best query performance on distributed data.
- Physical data integration is the part of the Data Virtuality Platform that ensures fast query response on large data sets while decoupling performance from the source data stores and moving the data to the physical repository. It also facilitates data historization, data preparation and data cleansing.
- The software provides cloud, on-premises, and hybrid deployment options, enabling you to choose the best-fit for infrastructure management, compliance, and data utilization.

In this manner, the effort-intensive physical transfer of the data is minimized and simplified, effectively removing lengthy data movement delays from the critical path of data integration projects. The final result is easy data access without fundamentally changing the existing environment.

How Data Virtuality Works



1

Connect to your data sources

Data Virtuality Platform connects to multiple data sources and allows querying data from there by using SQL. Data sources can either be relational or non-relational.

2

Create a central data logic

Data Virtuality Platform also allows you to create a central data logic that covers the business logic as well as the logical connections between the different systems. This layer can easily be implemented by using SQL views and/or stored procedures.

3

Get your data in a data warehouse

Data Virtuality Platform copies/replicates data for you using SQL commands. It transforms these SQL commands into the respective native data source commands, such as SQL commands, API calls, access to various file systems, or access to CDC (Change Data Capture) interfaces of different systems and databases. Data Virtuality Platform provides predefined jobs for the most common transformations. It gathers statistics about the usage of the data and uses these statistics to create a data warehouse in a special database, on a database server of your choice. This is done through automatic recommendations for the internal structure of the data warehouse that are then proposed to the database administrator. The data is then updated in your data warehouse with different time schedules.

4

Make your data accessible

Finally, Data Virtuality Platform supports the standard interfaces (JDBC, ODBC, REST, OData) to deliver data to the data consumers. This could be, for example, reporting tools, advanced analytics tools, or custom programs in various programming languages.



Data Virtuality Platform Features

- **DEPLOYMENT TYPES**

- . SaaS
- . Cloud self-hosted
- . On-premises
- . Hybrid

- **DATA VIRTUALITY SERVER**

- . Windows Server 64bit
- . Linux 64bit (Redhat, CentOS, Ubuntu, and others)
- . Deployment in Docker Container
- . Kubernetes

- **DATA VIRTUALITY STUDIO and WEB UI**

- . SQL Editor code completion on column level
- . Metadata dependency viewer (Data Lineage)
- . Metadata catalog and search
- . Graphical view builder
- . Wizards for easily connecting generic data (files/ (S)FTP/S3/Webservices) using formats XML, JSON, CSV, xSV
- . Wizard for querying Google Analytics APIs
- . Google Chrome, Microsoft Edge, Firefox, Safari, and other modern browsers (*Web UI relevant only*)
- . Windows 64bit, Linux 64bit, Mac 64bit (*Data Virtuality Studio relevant only*)

- **ACCESS DATA**

- . Via JDBC Via ODBC
 - . Windows (32bit/64bit)
 - . Linux (unixODBC 32bit/64bit)
 - . Mac (unixODBC 32bit/64bit)
- . Via REST API (REST-JSON)
- . Via OData

- **DATA FEDERATION**

- . Cross-database joins
- . Nested loop
- . Merge join
- . Dependent semi-join
- . Cross-database unions
- . Cross-database SELECT INTO, INSERT INTO
- . Dynamic cost-based query optimization

- **STREAMING**

- . Kafka connector
- . Change Data Capture (CDC) for selected data sources
- . Continuous record sets

- **DATA GOVERNANCE**

- . Automatic data lineage
- . Column-level data lineage
- . Column masking

- **BUSINESS DATA SHOP**

- (self-service data catalog)

- . Metadata catalog and search
- . Self-service data access for business users
- . Write and run queries
- . Download data

- **SQL DIALECT**

- . ANSI-92 with extensions
- . DDL, DML, procedural SQL
- . Nested subqueries
- . Common Table Expressions (CTEs)
- . Window functions/Framing clauses
- . XML/JSON parsing
- . Web service access
- . Scripting languages (server-side javascript)
- . Native query syntax

- **MOVE AND EXPORT DATA**

- . One query language: SQL
- . Permission-based INSERT, UPDATE, DELETE statements on all relational databases, Salesforce, SAS
- . Push-export via FTP, SFTP, SCP, email, S3, Azure Blob storage, web services (REST, SOAP, plain HTTP), file system and others
- . Export data using Data Virtuality Studio/SQL

- **SCHEDULE TYPES**

- . Once with optional delay
- . On time interval (every X minutes, hours etc.)
- . Daily at certain times of day
- . Weekly on certain weekdays
- . Monthly
- . Using custom cron expressions
- . Depending on other jobs or schedules (on success/failure/always)

- **DATABASES AND CONNECTORS**

- . More than 200 ready-to-use connectors. All our connectors can be found: <https://datavirtuality.com/en/connectors/>

● STRUCTURE OPTIMIZATION

- . Self-learning recommendation engine for automatic structure optimization
- . Materialized source tables and (virtual) views
- . Precalculated joins
- . Precalculated aggregations
- . Automatic index creation

● SECURITY, AUTHENTICATION, AUDIT

- . Kerberos
- . Row-based security
- . Git integration
- . Built-in user/role based permission system
- . Permission granularity on schema, table, column level
- . LDAP authentication (Active Directory, ForgeRock, etc.)
- . History of changes (versioning) for all custom metadata
- . Access to audit information and usage statistics using SQL from external tools
- . Security protocols: SSL/TLS, HTTPS

● JOB TYPES

- . Full copy with different cleanup options
- . Batch update (optionally with overlap cleanup)
- . History update (slowly changing dimension type 2)
- . Upsert with optional surrogate keys
- . Custom SQL jobs
- . External programs and scripts

● MATERIALIZATION ALGORITHMS

- . Full copy (used with materialized tables, views, joins, aggregations)
- . Incremental replication based on timestamp/id fields (used with materialized tables and views)

● IN-MEMORY CACHING FOR EVEN FASTER RESPONSES

- . Session scope
- . User scope
- . Virtual database scope

● WHAT ELSE?

- . Mail notification on job and replication status
- . Multi-tenancy (Sandboxing)
- . Graphical web-based performance monitoring
- . Password encryption
- . Smart data movement approaches (Snowflake & Redshift S3 load, Azure DWH Blob storage load, Salesforce Bulk API)
- . Programmatic access to all server functionality using Data Virtuality Management API

● About Data Virtuality

● **Founded:**
2012 by Nick Golovin (PhD) in Leipzig, Germany after 8 years of research

● **Offices:**
Munich, San Francisco, Leipzig

● **Solutions:**
Data Virtuality Platform SaaS
Data Virtuality Platform On-Premises
Data Virtuality Pipes Professional
Data Virtuality Pipes

● **Acknowledgements:**
Honorable Mention in 2022 Gartner Magic Quadrant for Data Integration Tools

● **Awards:**
Most Innovative Data Management Provider 2022, 2021 and 2019 (A-Team Insights)
2020 and 2019 Deloitte Technology Fast 50

Message: info@datavirtuality.com

Visit: datavirtuality.com

Request Demo: demo@datavirtuality.com

Data Virtuality Platform SaaS Free Trial: <https://eu.platform.datavirtuality.com/#/start-trial>

