

Astera Data Virtualization – Solution Brief

Integrate Disparate Data with Zero Data Replication

Consolidate data from on-premise and cloud sources without the need of moving data physically to save time, cost, and resources.

Serve Curated Datasets to Business Users

Keep up with the evolving data requirements with the capability to build and serve integrated, virtual views of enterprise data.

Consume Integrated Data in Multiple Ways

Expose virtual data views as a virtual database or through ADO.NET/REST APIs to ensure agility and data security.

In the increasingly competitive marketplace, businesses that take advantage of their data are likely to lead their industries. However, with business data getting dispersed across on-premise and cloud systems, gathering data for BI and analysis has itself become a challenge. As a result, business users are spending an increasing amount of time on finding and consolidating relevant pieces of data and lesser time on actually deriving insights from it.

While physical data integration can help unify disparate datasets, the process is not designed to meet the ever-changing business requirements. To deal with the challenge, businesses need to find a way to serve-up integrated data on-demand. Astera Software makes it possible with data virtualization.

Astera Virtualization Server offers businesses the capability to break data silos and address evolving analytical and operational opportunities through flexibility and agility in data access. The solution enables users to integrate data from disparate enterprise systems, in-memory formats, and databases, and present it as a virtual database, thereby shielding them from the technical complexities of connecting to each data source individually, while limiting physical data movement and optimizing the use of developer resources.

Built on Astera's industrial-strength data integration platform, the solution offers support for the design, deployment, and publishing of a virtual data model with robust automated entity relationship building, multi-table query generation, source data caching, and role-based security.

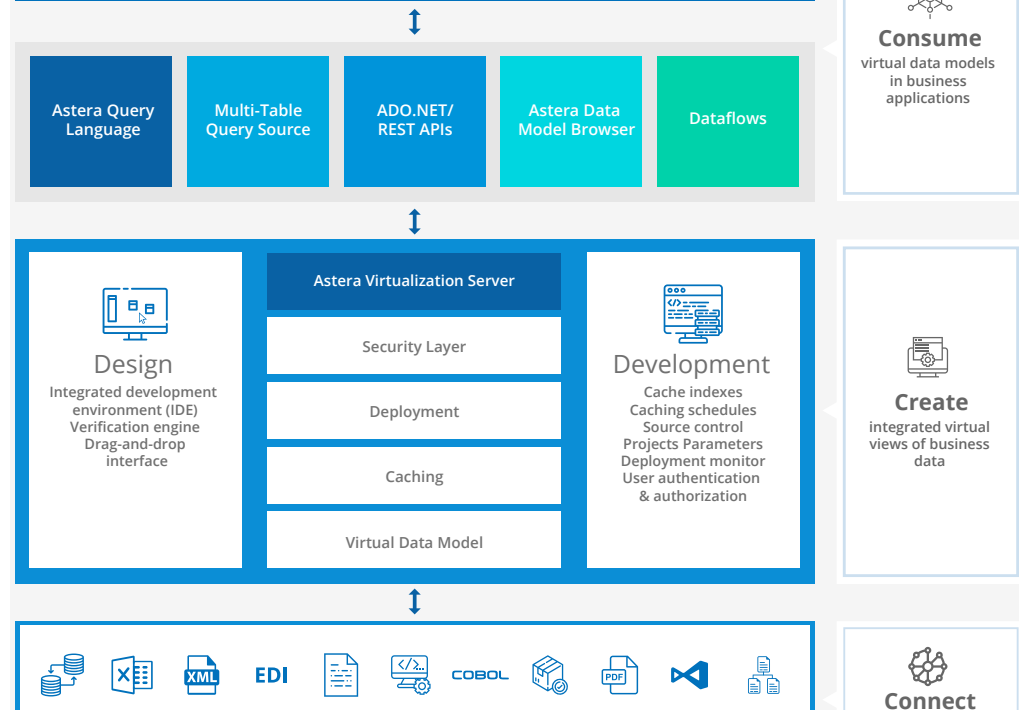


Figure 1 Astera Data Virtualization - Architecture Diagram

Key Features Development Environment

Connectors: Access and consolidate data from disparate sources

Databases	Access popular databases like SQL Server, Oracle, PostgreSQL, MySQL, MS Access, Sybase, IBM Db2 and others via Open Database Connectivity (ODBC) and Object Linking and Embedding, Database (OLE DB)
Cloud Databases	Use cloud databases as a data source in your virtual data model with connectivity to Amazon Redshift and Microsoft Azure*
Packaged Applications	Salesforce, Microsoft Dynamics CRM, SAP
BI and Analytics Tools	PowerBI and Tableau*
Data Warehouses	Teradata, Netezza
Structured, Semi-Structured, and Unstructured File Formats	Excel, delimited files, fixed-length text files, XML, JSON, TXT, RTF, DOC, DOCX, PDF, PDF Forms, COBOL files and copybooks, EDI formats, including ANSI X12, EDIFACT, HIPAA, and HL7
Web Services and other protocols	REST, SOAP, FTP, SCP, HDFS, AS2, ADO. Net, emails, and file system

ER Diagrams: Graphically design a virtual data model using intuitive ER diagrams

Entity Views	Use the different entity views to view all entities and elements, entities and keys, or just the entities comprising a virtual data model
Select Multiple Entities	Easily select all or multiple entities with a single click when working on complex ER diagrams
Additional SQL Queries	Use customized SQL queries to work with filtered datasets instead of a complete database table or entity
Database Entity Layout	This layout represents the structure of the entity in the respective source database
Virtual Entity Layout	Use the Virtual Entity Layout to define the name, column name, data type, and other properties of the attributes in the Virtual Database Table
Cache Database Layout	Visible only to the developers, this layout represents an entity inside the cache database
Create Reference Relationships	Use the drag-and-drop environment of the solution to graphically link nodes and create one-to-one and one-to-many relationships between entities

Integrated Development Environment (IDE) – Intuitive, user-friendly interface to support your development efforts

Drag-and-drop Interface	Add new data sources, apply transformations, or define destinations in simple drag-and-drop actions
Code-free, Graphical Development Environment	Model data, define relationships between entities, build transformations, and more in a graphical, code-free manner
Project Explorer	Keep your data virtualization projects organized with a Project Explorer that allows easy access to schedules, parameters, virtual data models, and shared connections

Data Source Browser: Easily connect and query data on a database server

Access Database Sources	Easily navigate all your database sources and their structure via Data Source Browser
Query Data	Query a given database directly from the Data Source Browser by writing SQL queries
Export Data	Export data from a database to an Excel or delimited file right from the Data Source Browser with a single click

Deployment Environment

Caching: Boost performance and availability with flexible caching options

Scheduled Cache Refresh	Specify hourly, daily, weekly, or monthly frequency to refresh the cache at fixed intervals
One-time Caching	Create a cache repository of sources whose data does not change frequently, such as lookup tables, using the one-time caching option
Continuous Caching	Refresh the cache continuously throughout the day or during specified hours using this option
Incremental Refresh	Update only the most recent data using the incremental cache refresh option by specifying an audit field, such as date or time, version number, or a key field
Full Refresh	Refresh the entire cache once, at specified intervals, or continuously
Event-Based Refresh	Refresh cache repository when a file is dropped in the specified folder or shared drive. You can also define the files to watch for by specifying the file extension, such as .txt or .xlsx.
Submission Caching	Watch an online location on Astera Cloud Portal and update the cache whenever a new file is submitted by a supplier
Cache Indexes	Create cache database indexes to improve the speed of data retrieval. Supports clustered, unique, unique constraint, and primary key indexing methods.

Project Archive – Maintain project backups for ease of re-deployment

Create Project Archive (*.Car)	Maintain a backup of all virtual data models using the archive functionality that automatically generates a *.Car file every time a Virtual Data Model is re-deployed after changes
--------------------------------	---

Source Control: Build, test, and deploy data virtualization projects together

Source Control Explorer	Use Source Control Server to browse different data virtualization projects and their versions
Source Control Server	Connect to a source control system, such as VSTS or Git, to simplify collaboration and keep track of every change made to a data virtualization project
Pending Changes	Organize, manage, and get details about what you have pending in a project with the ability to see and check in or undo pending changes
Compare Versions	Compare two different data virtualization projects and view the differences between the contents of each project
Workspaces	Develop and test your data virtualization projects in isolation on a local or server workspace

Deployment Monitor: Automate and monitor the deployment of virtual data models

Deployment Log	Get information about the deployment of a virtual data model, including initial copying of the archive file to a staging directory, creation and initial population of the cache tables, execution of cache dataflows, and changes to deployment or cache databases in case of redeployment
Entity Cache Info	Access caching details for all the entities comprising your virtual data model at a single place, including the caching schedule, cache status, last update, record count, and scheduled cache jobs
Job Progress	Get complete details about the Jobs that were run as part of a virtual data model deployment along with their progress and SQL script, if any.

User Authentication Engine: Secure communication and data access with advance authentication

Bearer-token Authentication	Make access distinctions and authenticates requests to the server with bearer-token authentication
Secure Domain Communication	Enable secure communication between client machines and the server using HTTPS. Configure an SSL protocol to encrypt the transmission of information and perform authentication using .pfx certificates.

User Authorization: Protect sensitive data with role-based resource access

Define User Roles	Use Roles to define the set of features that a group of users will have access to
Manage User Roles	Add new roles, delete an existing role, or edit role resource to manage access to different APIs and commands
Manage Users	Register new users, assign them to a role, activate or deactivate user accounts, and view the complete list of registered users, along with the roles assigned using a unified user management interface

Parameters: Increase flexibility and reusability with parameterization

Config Parameters	Eliminate the use of hardcoded values and support multiple configurations by specifying parameters in the Config Parameters page
Parameter Values	String representation of the allowed value type or a variable denoted with the \$ symbol surrounded by parenthesis. These are replaced with their assigned value in the Config file during runtime.
Config File	Store project-wide variables and their values in the Config file that can be associated with a particular project deployment

Publishing/Utilization

Astera Data Model Browser: Publish your virtual data models as a virtual database

Access Virtual Databases	Quickly and easily browse through the virtual databases deployed on a server and consume them in a dataflow for transformation or data transfer
Query a Virtual Database	Drag and drop individual tables onto the designer or use multi-table query source to use a virtual database as a source in a dataflow

ADO.NET/REST APIs: Make virtual databases available over intranet and internet

Expose Virtual Data Models over REST	Publish virtual data models in the REST format. REST APIs for create, read, update, and delete (CRUD) operations are supported.*
Expose Virtual Data Models over ADO.NET	Expose virtual data models as a service, available to clients from across the intranet and internet via ADO.NET. These services are accessible over HTTPS using URIs.

Multi-Table Query Source: Query multiple tables together to answer complex needs

Query a Virtual Database	Use the multi-table query source to use a virtual database as a source. Particularly useful for addressing complex business requirements that involve combining data spread out in more than one tables
Query Filtered Data	Apply different filters, such as less than or greater than, is null or not null, is empty string, and others to fetch filtered datasets from a virtual database

Astera Query Language: Use a simplified version of SQL to expedite query writing

Automatically Create Joins	Write AQL statements in the query window or automate complex joins using the multi-table query source
Connect to Non-SQL Sources	Connect to non-SQL sources, such as Excel, PostgreSQL, delimited file sources, and more
Edit Auto-Generated AQL Commands	Edit the auto-generated AQL commands easily to retrieve filtered datasets from a virtual database
AQL Commands	Use common commands, such as SELECT and ORDER BY commands to fetch, join, and sort data from a database table/s