dremio

WHITEPAPER

Best Practices for Efficient and Productive Analytics

Introducing the Dremio Semantic Layer

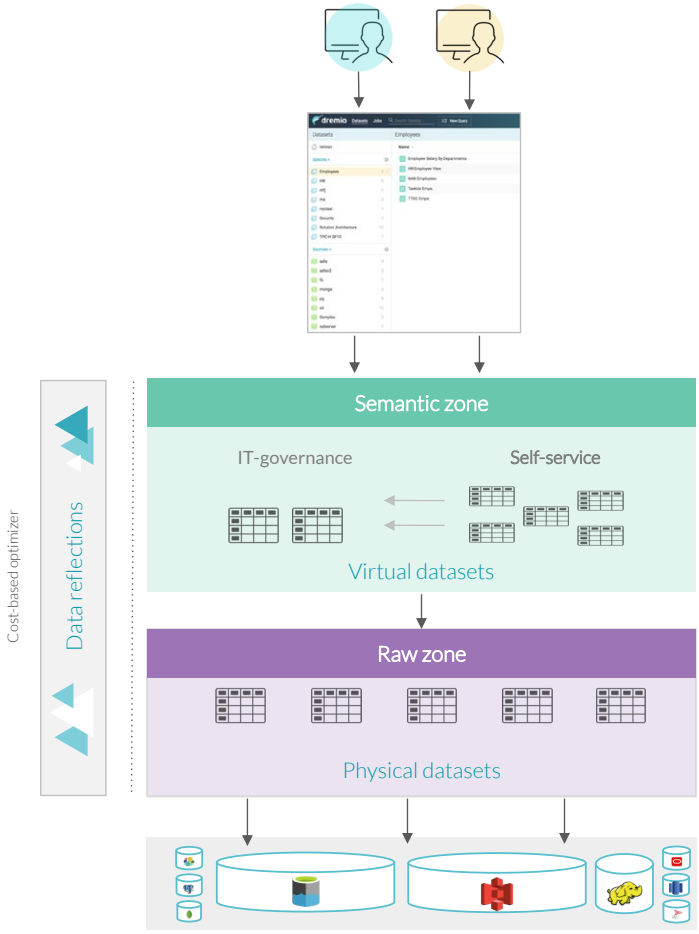
June 2020

The Self-Service Semantic Layer

The Dremio data lake engine incorporates a semantic layer that provides business analysts and data scientists with a consistent view of their data, regardless of what tool they are using (e.g., Tableau, Power BI, Jupyter).

The purpose of a semantic layer is to expose a business representation of an organization's data assets so that it can be accessed using common business terms. It is an abstraction layer, whereby business users can interact with the objects created in the semantic layer without regard for the complexities of where and how the data is physically stored and organized. Semantic layers establish views called virtual datasets (VDS) into an organization's data assets without the overhead and complexity of copying data.

In Dremio, the semantic layer is fully virtual, indexed and searchable, and the relationships between data sources, virtual datasets and transformations as well as all queries are maintained in the Dremio data graph, making it easy to identify exactly where each virtual dataset came from. Role-based access control makes sure that everyone has access to exactly what they need (and nothing else), and SSO enables a seamless authentication experience. The semantic layer finally makes secure, self-service access to data sets possible for data analysts and data scientists.



Query acceleration reduces cloud infrastructure costs by

>75%

Dremio's self-service semantic layer is a powerful means of improving productivity for analytics initiatives in enterprise organizations, and as a result can significantly reduce the total cost and time of service delivery to lines of business. With a self-service semantic layer, data engineers can quickly provision new virtual datasets, and data consumers can quickly put them to use and even share with others to avoid duplicate effort.

Virtual datasets and spaces, where virtual datasets are saved, make up Dremio's semantic layer. The semantic layer also features an integrated, searchable catalog that indexes all of the metadata, so business users can easily make sense of the data.

This document describes Dremio's best practice approach to organizing a semantic layer to enable interactive performance for business users. Dremio offers a number of acceleration technologies such as data reflections and a Columnar Cloud Cache (C3) which allow the logical semantic layer and physical optimization of the data to be designed and implemented independently. This document provides guidance on designing and implementing the logical semantic layer.

Spaces – Endless Customization

Dremio has the ability to customize spaces and folders, and can create arbitrary levels of hierarchy to categorize and organize semantic layer objects. However, with the nature of self-service analytics, as the volume of virtual datasets in Dremio increases and as more people start to work in the system to create virtual datasets, the arbitrary hierarchies of resources can quickly become difficult to navigate and manage if proper process and design isn't first established.

Dremio's built-in dictionary and tagging functionalities help to alleviate some of this burden because they enable users to search for specific resources, but they are not a substitute for good design and don't provide a clear view of all objects in an organized manner.

Further, without a proper design in place, each developer that creates folder structures for their own needs risks duplication of effort, potentially creating multiple equivalent versions of a virtual dataset.

4-100x

faster performance compared to Presto

Use Dremio to Architect a Semantic Layer

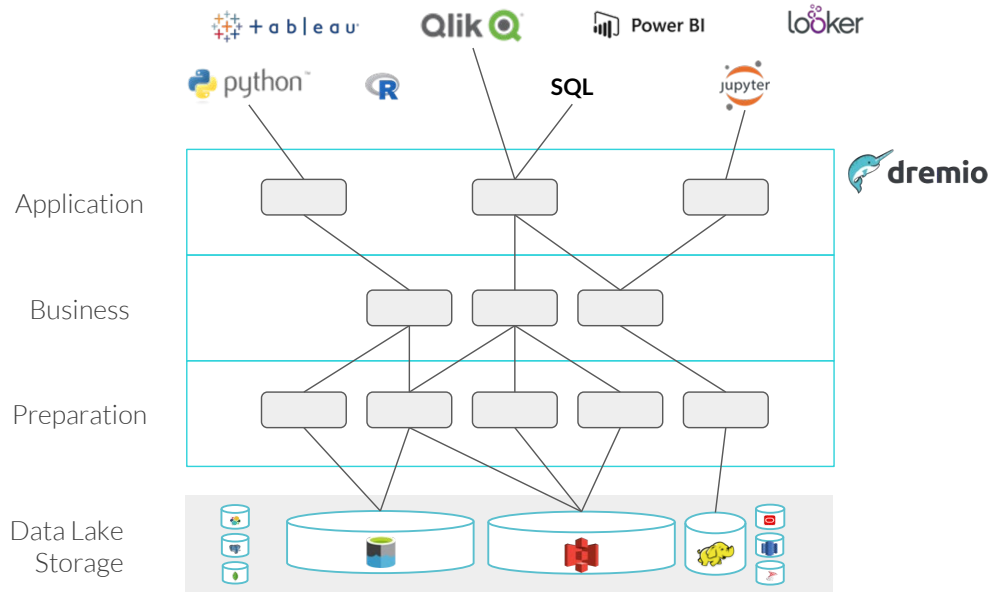
Dremio is the perfect platform for developing an enterprise-scale semantic layer. It provides a clear boundary between physically stored datasets and the logical IT-governed and self-service virtual datasets. It seamlessly provides

data engineers and semantic data modelers the ability to create virtual datasets based upon physical datasets without having to make copies of the physical data.

Since interactive performance for business users is a key capability of the semantic layer, when appropriate, Dremio is able to maintain physically optimized representations of source data known as *data reflections*. When queries are made against VDSs that have data reflections enabled, the query optimizer can accelerate a query by utilizing one or more data reflections to partially or entirely satisfy that query, rather than processing the raw data in the underlying data source.

With Dremio, you can create layers of VDSs that enable you to present data to business consumers in a format they need without having to worry about which physical locations the data comes from, or how the data is physically organized to provide interactive performance using data reflections. A layered approach enables you to create sets of re-usable VDSs for multiple projects. It promotes a more performant, low-maintenance solution that provides agility to development teams and business users and delivers better control over data.

The diagram below illustrates Dremio's best practice for layering VDSs in the semantic layer:



As shown in the diagram above, the physical data sources are in the data lake storage along the bottom; these are the same physical data sources that are represented in the sources section of the *Dremio documentation.* The consuming applications are represented at the top. The challenge is to take the physical data and transform it into the structure that business consumers require in their applications.

Dremio's best practice employs a methodology whereby VDSs created in the lower layers are used by VDSs that are created in the higher layers. VDSs in the lower layer serve as building blocks and can be reused many times.

The first layer that is closest to the data source is called *preparation*. This layer provides a simple way to organize and expose only the required datasets from a source, instead of the entire physical data sets (PDSs) in a source. There is one sub-layer per source connection with VDSs mapping one-to-one with their PDS equivalent. This is the layer that applies all column aliasing, column data type casting, and data cleansing, as well as the creation of some derived columns. No joins to other VDSs occur in this layer. Typically, a data engineer is responsible for creating the VDSs in this layer.

All VDSs in the second layer, dubbed the *business* layer, must be built by querying either 1) resources in the preparation layer or 2) other resources in the same business layer. This layer is predicated on the idea that the business has a standard or canonical way to describe key business entities (such as a customer, product or an order). It is the first layer where joins among and between sources occur. Typically, a data modeler works with business experts and data providers to define the VDSs that represent the business entities.

It is anticipated that many sub-layers could be created inside the business layer, with each one consisting of VDSs for different subject areas or verticals. These VDSs are reusable components that can and should be shared across business lines.

The business layer enables an additional layer of abstraction over the raw data sources, which facilitates important use cases such as data migration, among others. Common migration patterns include an on-premises data lake to a cloud data lake, and a non-data lake source, such as a data warehouse, to an on-premises or cloud data lake. While data is in the process of being migrated, business consumers are initially exposed to VDSs from the original source. Once the data is fully migrated, the VDS is updated to select its data from the new source, and the business consumer continues to receive the data without impact. In this use case, the business consumer always queries the same VDS throughout the entire migration process; all that changes is where the VDS gets its data from. This also applies to any applications that use this data

The ultimate goal of the business layer is to provide a holistic view of all the data across an enterprise.

The *application* layer serves to map the business layer VDSs into VDSs that output data in the format that each business consumer expects. The business layer derives the canonical VDSs that describe key business entities; and the application layer is used to tailor those canonical VDSs for the needs of individual business consumers, organization departments, etc. For example, one set of business consumers might want to see a “customer” VDS aggregated on field A, whereas a different set of business consumers might want to see the same “customer” VDS aggregated on field B.

If given access, business consumers such as analysts or data scientists can work directly in the application layer of the Dremio UI to create and modify VDSs for use in their own dashboards and can leverage the VDSs built inside the business

layer to help them achieve this. Joining multiple VDSs from the business layer to produce a new VDS in the application layer is a perfectly valid approach.

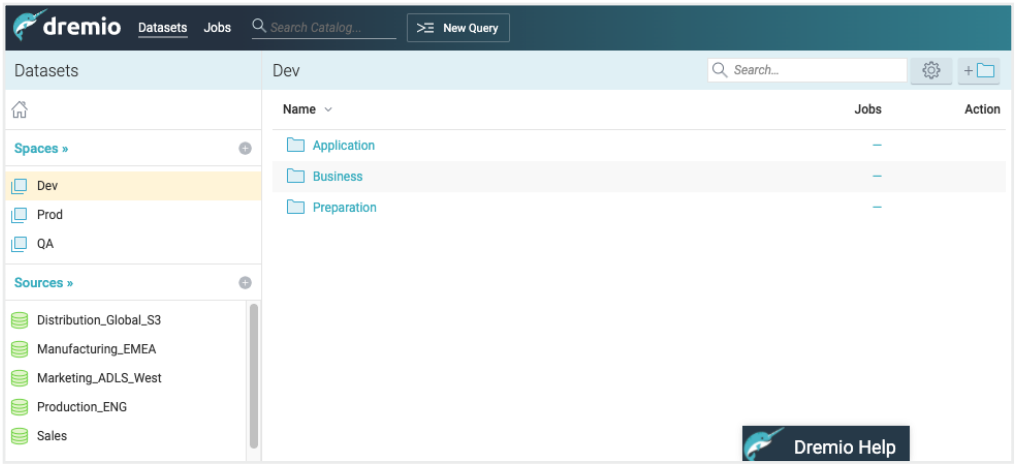
Alternatively, if business consumers do not have direct access to the Dremio UI, then it becomes the responsibility of designated Dremio developers who do have access to the application layer to create and modify VDSs on behalf of the business consumers via a formal change process. Change management discussions are outside of the scope of this document; however, best practice is to pair a Dremio developer with an Application SME.

Many sub-layers could be created inside the application layer in order to organize the resources exposed by Dremio into subject areas or verticals. Ultimately, consuming applications (e.g., Tableau, Power BI, Jupyter, etc.) will only ever have visibility into application layer resources and will not be exposed to any lower-level VDSs.

Dremio's semantic layer best practices assume the layers described above translate directly into folders in Dremio. Spaces are reserved for environmental or organizational segregations. For example, a single Dremio cluster might be used for development work, for QA work, and for production work; therefore, spaces are used to segregate our resources into these environments. Please note that we rarely recommend this particular scenario. Instead, we recommend keeping development, QA and production workloads in isolated Dremio clusters where possible.

Typically, if an organization has distinct environments for development/QA/production, it is recommended that the spaces represent business units within the organization instead. This scenario results in clean, easy-to-follow spaces with a nested layer of folders inside those spaces.

The image below illustrates how the above concepts are mapped into a coherent space and folder structure inside Dremio:



What About Security?

By arranging spaces and folders into the layered structure described above, it is easy in Dremio to apply a role-based access control (RBAC) approach to securing the resources; there are clear distinctions on which groups (roles) are able to interact with the spaces and folders.

Dremio leverages an external LDAP server (such as Microsoft Active Directory) to manage permissions, whereby users and groups are first created, and then users are assigned to groups. Dremio connects to the LDAP server to identify users and groups, and assign permissions for the groups against the objects in Dremio.

DEVELOPMENT ENVIRONMENTS/SPACES

When working in a Dremio development environment or in a development space within a single Dremio environment, Dremio recommends setting permissions on each of the layer folders at the highest level based upon a set of roles and then propagating the permissions down through any subfolders and VDSs as required. Note that some roles have multiple possible permissions defined; this is because you may want to give nested subfolders different privileges depending on who owns the folder (e.g., the owner will be able to edit the VDSs in a folder but other roles can only view the VDSs in the folder):

	User Role		
Layer	Data Engineer/SME	Semantic Data Modeler	Report Developer/Analyst/Data Scientist
Application	Edit or View	Edit or View	Edit
Business	Edit or View	Edit	View
Preparation	Edit	View	None

QA/TEST AND PRODUCTION ENVIRONMENTS/SPACES

When working in a Dremio QA\test or production environment or in a QA\ test\ production space within a single Dremio environment, no user role should be given edit permissions to any VDSs. Changes should only be permitted in a development environment or development space. Dremio recommends setting permissions on each of our layer folders at the highest level based upon a set of roles and then propagate the permissions down through any sub-folders and VDSs as follows:

	User Role		
Layer	Data Engineer/SME	Semantic Data Modeler	Report Developer/Analyst/Data Scientist
Application	View	View	View
Business	View	View	View
Preparation	View	View	None

Business User Experience

Dremio's functionality for curating data via tagging VDSs and adding wiki content plays a very important role in enabling business users to quickly find the resources they are interested in using – either as-is or as the basis for a new VDS. A business user is curious to know what resources are already available to them, so the ability to search for tag keywords or search on table or column names is very useful. In addition, business users want to understand in more detail the meaning of a set of VDSs in a particular space or folder, what their purpose is, and what data they expose, etc., which is where wiki entries are especially useful. More details on data curation can be found in *Dremio documentation.*

Based on Dremio's security recommendations above, business users will not have visibility into VDSs in the preparation layer, and since these are typically one-to-one mappings with their PDS equivalents anyway, there is little value to business users in adding tags and wiki entries to the preparation layer resources. However, data engineers and semantic data modelers may find tags and wiki content useful in the preparation layer.

By contrast, the business layer lies at the heart of the self-service semantic layer, therefore, tags and wiki entries should be used extensively in this layer. It is not the responsibility of business users to add tags and wiki content; this is done by the semantic data modelers with assistance from data engineers. Business users will spend a reasonable proportion of their time searching for relevant metadata in the business layer and subsequently use the results of their findings to create new VDSs or alter existing VDSs in the application layer. To that end, it is recommended that business users add tags and wiki entries themselves to the VDSs and folders they create respectively in the application layer.

Conclusion

Dremio's semantic layer is an integrated, searchable catalog that indexes all of your metadata, so business users can easily make sense of your data. Virtual datasets and spaces make up the semantic layer, and are all indexed and searchable. By managing data preparation in a virtual context, Dremio makes it fast, easy, and cost effective to filter, transform, join, and aggregate data from one or more sources. And virtual datasets are defined with standard SQL, so you can take advantage of your existing skills and tools. The ultimate result of implementing Dremio's semantic layer on the cloud data lake is to increase architectural efficiency and improve productivity, giving data architects and engineers more time to focus on what matters most.

ABOUT DREMIO CORPORATION

Dremio's Data Lake Engine delivers fast query speed and a self-service semantic layer operating directly against data lake storage. Dremio eliminates the need to copy and move data to proprietary data warehouses or create cubes, aggregation tables and BI extracts, providing flexibility and control for Data Architects, and self-service for Data Consumers. For more information, visit www.dremio.com.

Founded in 2015, Dremio is headquartered in Santa Clara, CA. Investors include Cisco Investments, Lightspeed Venture Partners, Norwest Venture Partners and Redpoint Ventures. Connect with Dremio on GitHub, LinkedIn, Twitter and Facebook.