Data Architectures for Data Science Using Data Virtualization

A Whitepaper

Rick F. van der Lans
Independent Business Intelligence Analyst
R20/Consultancy

October 2021

Sponsored by

Table of Contents

1 Summary

Introduction – *Data science* is a profession with a long history. Throughout its history, it has faced a wide variety of changes, such as new algorithms, new tools, new use cases, and the availability of much more data. Data science has never been a static discipline. The latest changes and challenges that data scientists face are a fast-changing data storage technology landscape, new restrictive regulations for data privacy, and time-consuming data preparation tasks.

Data scientists are facing some new, complex challenges.

The need for big data systems has led to the introduction of numerous new technologies for storing data, including file systems, such as Amazon S3, Apache Hadoop, and Microsoft Azure Data Lake, and also modern SQL systems, such as Amazon Athena, Google BigQuery, and Snowflake. The data storage landscape keeps changing for data scientists.

New regulations for data privacy and protection, such as *GDPR,* define and limit what organizations are allowed to do with their data, which type of data can be stored, how long it can be kept and for which purposes it may be used. This complicates and restricts the analytical work of data scientists.

Studies have shown that data scientists spend only 20% of their time on actual analytical work and as much as 80% of their time on *data preparation* tasks[1]. Data preparation tasks involve determining which data elements and data values are required, identifying from which systems that data needs to be extracted, developing programs to extract the data from those systems, transforming the data into meaningful data, loading the extracted data in some system, and making the data easily available for analytics.

Simplifying Data Access for Data Scientists – Organizations need *data architectures* that enable data scientists to develop analytical models in the easiest and fastest way possible. To cope with changing data storage technologies, new data privacy regulations, and to shorten the data preparation tasks, such data architectures must face the following challenges:

- Data is everywhere
- Data is stored in a heterogeneous set of source systems
- Data is stored in a schema-on-read form
- Data is 'hidden' behind applications
- Data is cryptic
- Data history is missing
- Data quality is poor
- Metadata is missing
- Data security rules are restrictive
- Data privacy rules are restrictive

Cloud Platforms and Data Lakes – A popular solution that helps data scientists to deal with these challenges is to copy the data relevant for them to a centralized environment on a *cloud platform*. Others opt for a *data lake*. Both solutions have strong, but also weak points. For example, both handle the first group of challenges, but the others are not adequately dealt with.

Cloud platforms and data lakes are popular solutions to help data scientists deal with the challenges.

[1] A. Ruiz (InfoWorld), *The 80/20 Data Science Dilemma*, September 2017; see https://www.infoworld.com/article/3228245/the-80-20-data-science-dilemma.html

Data Architectures and Data Virtualization – A third solution is to implement a data architecture in which *data virtualization technology* is applied. This architecture meets all the challenges and, therefore, offers data scientists easy and fast access to data. For data scientist such an architecture simplifies data access and consequently shortens the data preparation phase allowing them to spend more time on analytics.

Using data virtualization offers data scientists easy and fast access to data.

The Whitepaper – This whitepaper describes in detail the challenges that data scientists face when accessing, querying, and analyzing data. Cloud platforms and data lake solutions are explained and for each, the challenges are described. A short description of data virtualization is given. Amongst other aspects, building blocks, and data security rules are explained. It also shows how data virtualization features can speed up the work of data scientists, and how it helps to deal with all the challenges. A flexible data architecture, called the *logical data lake*, is described in which data virtualization acts as the general entry point for data scientists to access data.

Note: Most of the challenges described in this whitepaper are faced by other data consumers as well. This whitepaper focuses on data scientists as the importance of data science for organizations continues to increase.

Target audience – The main target audience of this whitepaper consists of data architects, enterprise architects, solutions architects, IT architects, data warehouse designers, technology planners, IT managers, and chief data officers. Additionally, it is meant for data scientists, statisticians, actuaries, and quants.

Basic knowledge of data virtualization is required. For detailed explanations of data virtualization please refer to the books *Data Virtualization for Business Intelligence Systems*[2] and *Data Virtualization: Selected Writings*[3].

2 The Ever-Changing World of Data Science

This section describes the current changes and challenges that data scientists face: a fast-changing data storage technology landscape, new restrictive regulations for data privacy, and time-consuming data preparation tasks.

Fast-Changing Data Storage Landscape – Before the big data era, most enterprise data was stored on premises using SQL databases and new data storage technologies were hardly ever adopted. Big data changed all that. Organizations moved data to cloud platforms using new data storage technologies, such as Hadoop, Amazon S3, and Microsoft Azure Data Lake, all supporting massively parallel processing. This change of the data storage landscape complicated the work of data scientists. Lately, with the arrival of new analytical SQL databases, such as Amazon Athena, Google BigQuery, and Snowflake, more data storage platforms were deployed which, again, changed the data storage landscape. And again, data scientists had to work with different data storage technologies.

Data scientists need to work with a fast-changing set of data storage technologies.

[2] R.F. van der Lans, *Data Virtualization for Business Intelligence Systems*, Morgan Kaufmann Publishers, 2012; see https://www.r20.nl/DataVirtualization_V1.htm

[3] R.F. van der Lans, *Data Virtualization: Selected Writings*, Lulu.com, September 2019; see http://www.r20.nl/DataVirtualizationBook.htm

Note that this not a plea for rejecting new technologies. Most of them offer real benefits, such as increased performance, up and down scalability, more autonomous processing, and enriched analytical capabilities.

New Regulations for Data Privacy – To simplify data access for data scientists, concepts such as data lakes, data hubs, and data sandboxes have been introduced. What all three have in common, is that data from various source systems is copied to these centralized environments. This centralized data storage concept clashes somewhat with new regulations for data privacy and protection, such as *GDPR*. For example, they define and limit what organizations are allowed to do with their data, which types of data they are allowed to store, how long the data can be kept, and for which purposes it may be used. One of the effects of GDPR is that certain combinations of data elements are not allowed to be stored together. This prohibits the concepts of data lakes in which data storage is centralized. Some of these issues can be bypassed by anonymizing data through masking, scrambling, filtering, aggregating, or compressing. It is the responsibility of the data architects to make sure that data scientists have access to all the data they require but that some data has been masked and scrambled.

The Data Preparation Tasks of Data Science – Developing data science models is not just a simple matter of switching on an analytical tool, pointing it to a dataset, and before you know, several descriptive or prescriptive models pop up. Developing data science models is a complex, multi-step process; see Figure 1. These steps can be divided into two groups. The first group is responsible for retrieving the data required for the data science exercise and transforming it into the form required for analytics. The second group of steps deals with the actual analytical work. Evidently, this whole process is highly iterative.

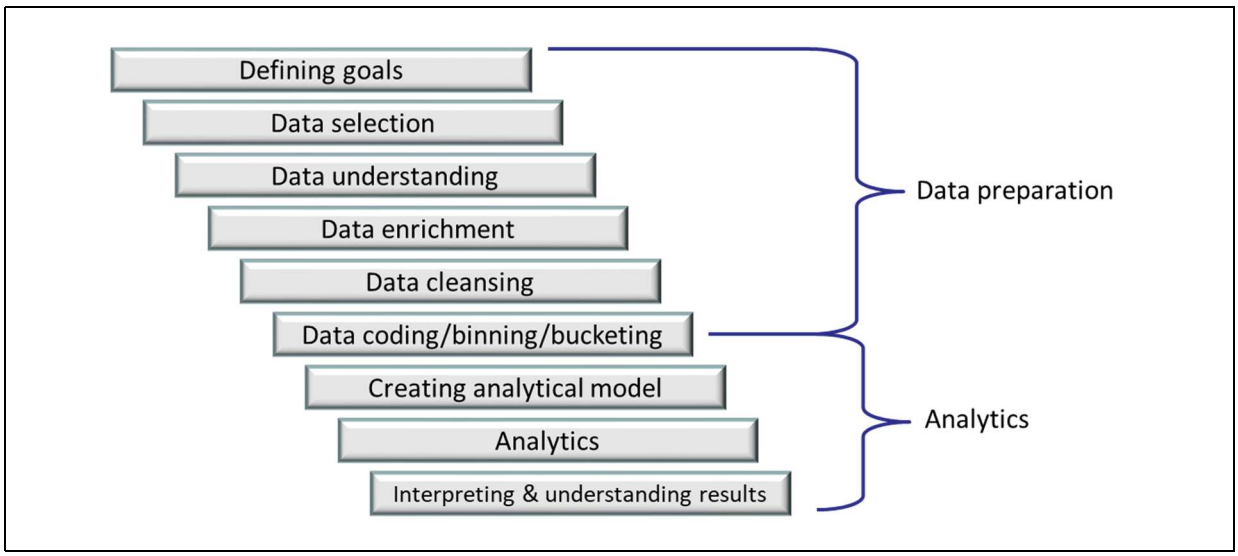

Figure 1 *Developing data science models is a multi-step process.*

Data preparation is a time-consuming data science task.

Together, the first group of steps is commonly referred to as the *data preparation phase*. As indicated, studies[4] show that data scientists spend no less than 80% of their valuable time on data preparation of which a considerable amount of time is devoted to the *data selection* step. During data selection, data scientists determine which data elements they need, identify from which systems it needs to be extracted, develop programs to extract the data from those systems, transform the data into meaningful data, load all the extracted data in some system, and finally, make the data easily available for analytics. Although this sounds

[4] A. Ruiz (InfoWorld), *The 80/20 Data Science Dilemma*, September 2017; see https://www.infoworld.com/article/3228245/the-80-20-data-science-dilemma.html

easy and straightforward, data selection can be a time-consuming exercise. One reason is that data scientists are dependent on the availability of other developers to help them extract the data.

Shortening the Data Preparation Phase – This time-consuming data preparation phase reduces the time that data scientists can allocate to analyze data and develop predictive or descriptive models. Solutions are needed that shorten the time they must spend on the data preparation phase. Roughly three groups of solutions exist:

- **Deploying Data Preparation Tools:** Data preparation tools are available that simplify and/or automate some of the data preparation tasks. According to Gartner[5], these "tools are also used by citizen integrators and data engineers for data enablement to reduce the time and complexity of interactively accessing, cataloging, harmonizing, transforming and modeling data for analytics in an agile manner with metadata and lineage support." Data preparation tools exploit machine-learning algorithms to analyze the data and subsequently recommend and/or automate actions to transform the data to make them analytics-ready.

- **Offloading Work to Data Engineers:** Many data preparation tasks are highly technical and may require in-depth knowledge of the source systems. To save data scientist's time, it is recommended to leave most of the data preparation work to so-called *data engineers*. They have the right qualifications for the technical part of data preparation. Here is a common definition and interpretation of what they do[6]: "Data engineering is the aspect of data science that focuses on practical applications of data collection and analysis. […] Data engineers focus on the applications and harvesting of big data. Their role doesn't include a great deal of analysis or experimental design."

- **Extending the Data Architecture:** Extend and adapt a data architecture in such a way that data selection becomes much simpler and more flexible. This is the topic of the whitepaper.

3 Challenges For a New Data Architecture

A modern *data architecture* helps data scientists to deal with the fast-changing data storage landscape, the new regulations for data privacy and protection, and it shortens the data preparation tasks. For this, data architectures need to face the following challenges.

Challenge 1: Data is Everywhere – The data needed by data scientists is not often stored in one specific source system, but spread across a multitude of systems. It is stored in financial systems, sales systems, data warehouses, website logs, and so on. Additionally, data scientists want to analyze data coming from external data sources, such as a public data marketplace. Because organizations can have thousands of applications and databases, determining which systems contain the required data can be an exhausting exercise by itself. It is like looking for a needle in a group of haystacks.

Challenge 2: Data is Stored in a Heterogeneous Set of Data Sources – It is most unlikely that the systems containing the required data are all developed with the same technology. It is more likely that different storage technologies are used, ranging from simple flat files, via SQL databases to Hadoop and NoSQL. And some data may be stored in spreadsheets, Word documents, and legacy systems. All these technologies support different interfaces and database concepts, which means that data scientists need to use many different languages for extracting relevant data.

[5] Gartner, *Data Preparation Tools Market*; see https://www.gartner.com/reviews/market/data-preparation-tools

[6] Data Science Graduate Programs.com, see https://www.datasciencegraduateprograms.com/data-engineering/

As indicated in Section 2, the introduction of big data has resulted in a fast-changing and heterogeneous market of data storage technologies.

Challenge 3: Data is Stored in a Schema-On-Read Form – With the introduction of Hadoop and cloud platforms, it has become common to store data in systems that do not 'understand' the structure of the data. This approach for data storage is called *schema-on-read*. For example, if customer address data is stored this way, applications are not able to ask for the names of the customers from the Northern region. The structure of the data is not known to the storage system, instead applications themselves need to know how to interpret the data. In other words, the *schema* (structure) of this *structure-less* data is determined by the applications when they read the data, hence the term schema-on-read.

Offering structure-rich data to data scientists saves them time.

When schema-on-read is used, data scientists need to spend time on transforming structure-less data into *structure-rich* data, else it is difficult to process. Offering structure-rich data to data scientists saves them time.

Note that from a data storage perspective, storing structure-less data offers some benefits, the most important one being data storage flexibility. If the structure of the original data changes, the definition of the data structure doesn't need to be changed. The data already loaded, which has the old data structure, can remain unchanged.

Challenge 4: Data is 'Hidden' Behind Applications – A large portion of enterprise data is stored in databases that are part of *packaged applications*. In this case, not the organization but the vendor of the application designed the data structure. Although it is technically possible, vendors often discourage direct access to these databases. Instead, it is recommended to access the data through a supplied interface. These are proprietary and unique interfaces that need to be studied by the data scientists.

Challenge 5: Data is Cryptic – Data extracted from source systems can be highly cryptic which complicates the work of data scientists. The data is cryptic because many codes are used to represent real data values. Some examples why working with cryptic codes is difficult:

- The meaning of the codes is not documented properly, so the source code of the applications needs to be studied for explanation.
- Complex multi-column constructs are used, such as, if column C_1 contains the value v_1, then the value v_2 in column C_2 means something specific, but if column C_1 contains the value v_2, then v_2 in column C_2 means something else.
- So-called *surrogate key values* are used to represent real data values making it difficult to understand the data. For each surrogate value, data scientists need to find the corresponding real data value.
- In some source systems, data is organized in such a way that a straightforward copy is not an option. For example, in some mainframe database systems the stored data is connected through pointers. In this case, a simple copy to a file will not work. Application code must be developed to transform it into flat data that can be copied to files.

Challenge 6: Data History is Missing – Not all transactional source systems keep track of history. In such systems, when data changes, the old version is overwritten with the new version. These older versions of the data may be useful for data scientists, especially when they want to analyze the data historically. Building a solution to start keeping track of that history is quite an engineering task.

Challenge 7: Data Quality is Poor – Data may require cleansing before it can be analyzed. Incorrect data can lead to incorrect data science models. Some incorrect data values are easy to spot for everyone. For example, for identifying a misspelled city name and recognizing an impossibly high personnel age,

common sense is sufficient. But some incorrect data is not easy to identify and may require in-depth knowledge of the business and the source systems before it can be corrected.

It is the responsibility of data owners to make sure the data stored in their systems is correct. But if it is not, data scientists need to spend time on correcting the data and upgrading it to a quality level that results in reliable models.

Challenge 8: Metadata is Missing – Some data sources are poorly documented. Even if data is stored in a structure-rich style, its meaning may still be unclear. For example, if a column in a SQL table is called revenue, it still leaves many questions unanswered. Is it gross or net revenue? Is it total revenues per store, city, region, or province? Descriptive metadata is indispensable for data scientists. Incorrect interpretations of data lead to incorrect data science models and eventually to poor business decisions.

Challenge 9: Data Security Rules are Restraining – Data security rules may be in place that do not allow data scientists to access the data, or maybe they only have partial access to the data. They may also have access to data, but that data may not be allowed to be copied and stored outside its original security realm.

Challenge 10: Data Privacy Rules are Restraining – As indicated in Section The Ever-Changing World of Data Science, not all the data that an organization stores can be used by data scientists without modifications. Regulations may require sensitive data, such as *personally identifiable data*, to be anonymized, filtered, masked, scrambled, or aggregated before it is made available to them. This may involve anonymization of partial or full data values. Note that when data is anonymized, certain data elements may become useless for data scientists.

4 Solution 1: Copying Data to a Cloud Platform

Storing data on a cloud platform to support data scientists.

Advantages of Storing Data on a Cloud Platform – An architectural solution to help data scientists with their data needs and to solve some of the challenges described in the previous section is based on copying all the required data from the source systems to a cloud platform. Platforms, such as those hosted by Amazon, Google, and Microsoft, allow massive amounts of data to be stored. This solution offers the following benefits:

- Most of the cloud platforms offer fast data access (even on big data) that speeds up the data scientists' queries.
- Storage is relatively cheap.
- Schema-on-read and schema-on-write formats are both supported.
- Data scientists need to access only one centralized environment to retrieve data.
- Data scientists can add their own data to this environment, for example, data coming from specific tests and analysis exercises. This solves the problem of collecting the data.

Not All Challenges are Overcome by Cloud Platforms – Copying data to cloud platforms doesn't solve all the challenges described previously. Looking closely at this solution, these challenges remain:

- Challenge 3: data is stored in a schema-on-read form
- Challenge 5: data is cryptic
- Challenge 6: data history is missing
- Challenge 7: data quality is poor
- Challenge 8: metadata is missing
- Challenge 9: data security rules are restraining
- Challenge 10: data privacy rules are restraining

Centralizing all the data on a cloud platform offers data scientists some important benefits but is not a complete solution.

5 Solution 2: Copying Data to a Data Lake

A Data Lake for Storing Raw Data – Another architectural solution to speed up the data selection task is the *data lake*. The data lake was originally introduced to support data science and similar forms of exploratory and investigative forms of analytics as is reflected in James Serra's[7] description of data lake: "A data lake is a storage repository […] that holds a vast amount of raw data in its native format until it is needed. It is a great place for investigating, exploring, experimenting, and refining data, in addition to archiving data."

A data lake is a storage repository to store raw data for data scientists.

The data lake is somewhat similar to the solution described in the previous section. Data lakes are commonly implemented using file systems on cloud platforms. The difference with the previous solution is that a data lake is a more managed environment than 'just' a file system.

The data lake tackles the same challenges:

- Challenge 1: data is everywhere
- Challenge 2: data is stored in a heterogeneous set of source systems
- Challenge 4: data is 'hidden' behind applications

The Zones of a Data Lake – What a data lake entails, has changed somewhat over the years. For some it is an environment that consists of *zones*, or sometimes called *layers* or *tiers*; see Figure 2. In this case, a data lake is divided into areas. Data is first copied to the *landing zone* (the original data lake). Next, the data is slightly processed and copied to the *curated zone*, and subsequently it is processed and copied to the *production zone*. Step by step the data becomes more meaningful for and needs less and less transformational work by data scientists.

Adding zones to data lakes solves two more challenges:

- Challenge 3: data is stored in a schema-on-read form
- Challenge 5: data is cryptic

[7] J. Serra, *What is a Data Lake?*, April 2015; see http://www.jamesserra.com/archive/2015/04/what-is-a-data-lake

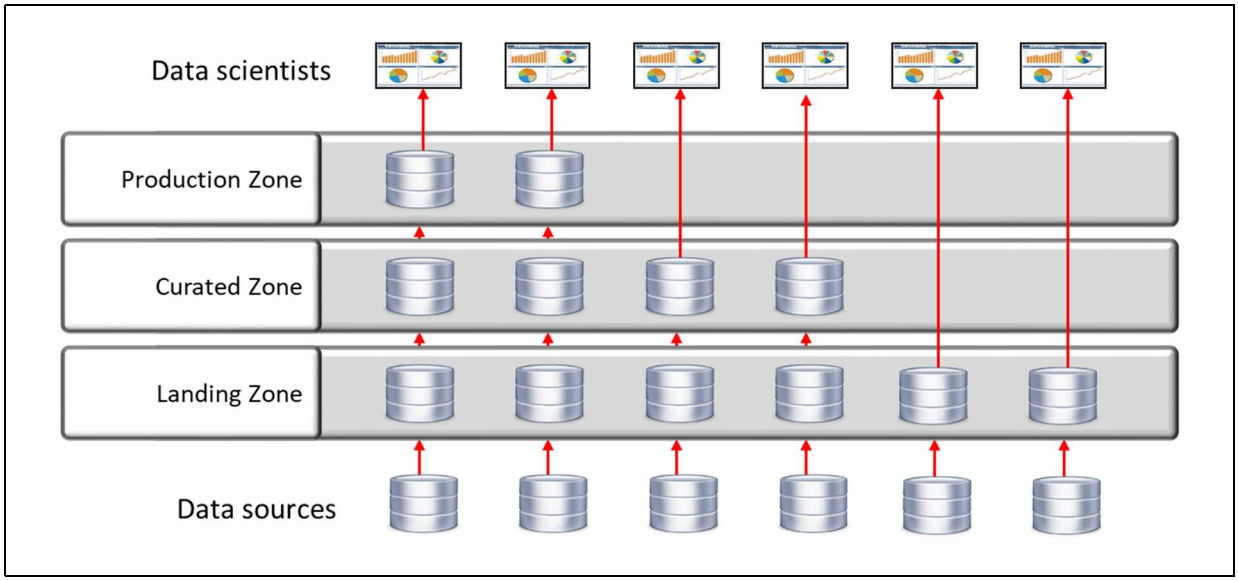

Figure 2 *The zones of a data lake.*

Dealing with the other Challenges – The following challenges are not dealt with:

- Challenge 6: data history is missing
- Challenge 7: data quality is poor
- Challenge 8: metadata is missing
- Challenge 9: data security rules are restraining
- Challenge 10: data privacy rules are restraining

The Multi-Purpose Data Lake – James Serra indicates that a data lake serves only a limited purpose: to support investigative and experimental forms of analytics. Currently, many organizations see this differently and consider the data lake as a *multi-purpose data store*. Here, all the data is copied to the data lake, including data that is not relevant to data scientists. From there it is copied to and used by other data consumers, including data warehouse environments, transitional systems, and websites. In a way, a multi-purpose data lake resembles what was called the *operational data store*[8].

The single-purpose data lake versus the multi-purpose data lake.

When a multi-purpose data lake becomes the source for more controlled forms of data consumption, such as data warehouse environments and websites, it must be professionally managed and governed. In this case, the data lake cannot be an experimental environment anymore. For the data science style of data consumption, an additional data store must then be developed to which relevant data from the data lake is copied. This additional data store, sometimes called a *data sandbox*, is similar to the original data lake and has similar advantages and disadvantages.

6 Solution 3: Data Virtualization to the Rescue

The two solutions described in the previous sections cannot handle all the challenges that data scientists face. The third architectural solution described in this whitepaper does cover all of them. The solution is based on *data virtualization* technology. This section provides a brief introduction to data virtualization for those not familiar with it.

[8] Wikipedia, *Operational Data Store*; see https://en.wikipedia.org/wiki/Operational_data_store

For more detailed explanations of data virtualization please refer to the books *Data Virtualization for Business Intelligence Systems*[9] and *Data Virtualization: Selected Writings*[10].

Data virtualization operates as a data abstraction layer.

Introduction to Data Virtualization – Data virtualization operates as a *data abstraction layer* between source systems and data consumers; see Figure 3. It can access almost any kind of source system using almost any kind of technical interface and can make that data available to all types of data consumers, including simple dashboards, mobile apps, spreadsheet users, websites, and data scientists. Data consumers extract data from the source systems via the data virtualization server that integrates, transforms, filters, and aggregates the data. In other words, all the *data processing specifications*, which are commonly spread across an entire data warehouse environment, are now defined centrally within the data virtualization server. Metadata to describe and define the data resides within this server as well.

The data abstraction layer also serves as a central access layer for data scientists to governed data that can be easily queried and modeled. This makes the whole data preparation process easier for all involved parties.

Whereas most traditional technologies, such as ETL, require data to be stored after the data processing specifications have been processed, data virtualization is optimized to execute these specifications *on demand*. This offers more agility for data delivery.

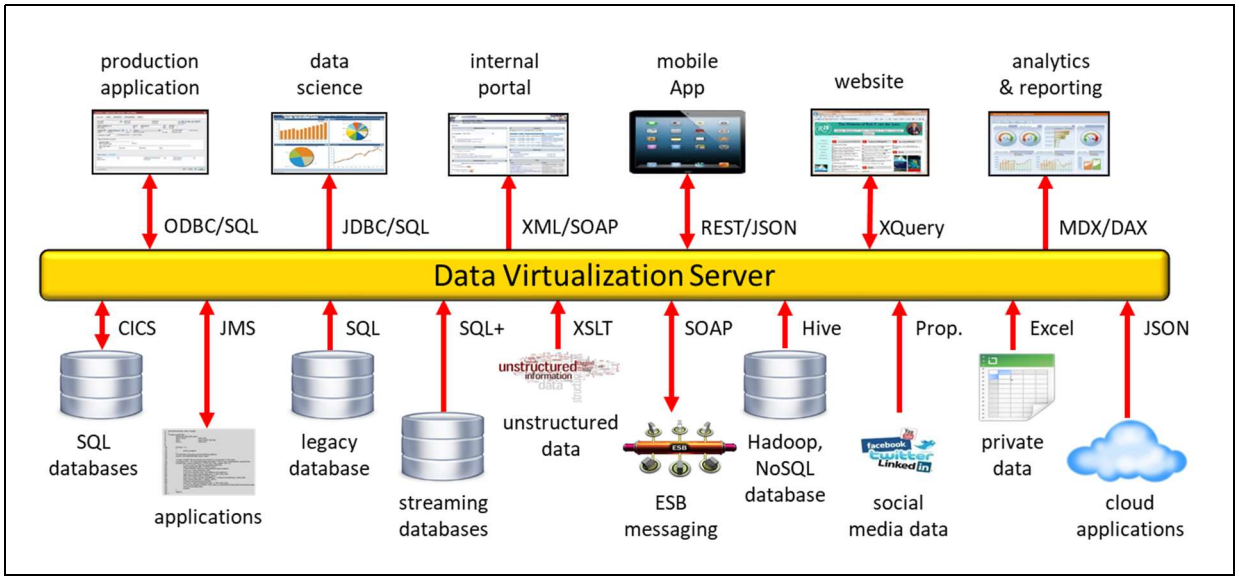

Figure 3 *Data virtualization operates as a data abstraction layer between source systems and data consumers.*

Data processing specifications are defined within views using SQL.

Views – The core building block of a data virtualization server is the *view* (or *virtual table*). The definition of a view specifies which *data processing operations* must be applied to source data when it is retrieved by data consumers. Examples of data processing operations are aggregations, filters, joins, calculations, and concatenations. Through these views, source data can be transformed into whatever business users need.

[9] R.F. van der Lans, *Data Virtualization for Business Intelligence Systems*, Morgan Kaufmann Publishers, 2012; see https://www.r20.nl/DataVirtualization_V1.htm

[10] R.F. van der Lans, *Data Virtualization: Selected Writings*, Lulu.com, September 2019; see http://www.r20.nl/DataVirtualizationBook.htm

The language used to specify the data processing operations is the popular SQL language. In fact, most views are defined using SQL queries.

When data is accessed through views, all the operations are applied to the data. For example, when a view with a definition containing a filter and aggregation operation is accessed, both operations are applied to the data before it is returned. In other words, the operations making up the definition of a view are applied on demand.

Stacking of Views – Views can be stacked on top of each other; see Figure 4. In this figure three layers of views are defined. Note that it is not mandatory but optional. Here, each layer of views has certain tasks. The bottom layer of the views is commonly responsible for extracting data from the source systems. The middle layer takes care of data integration and, if required, data cleansing. In the top layer, the views have a data structure that meets the requirements of the data consumers. Together, the three layers transform the data coming from the source systems to the form required by the data consumers.

For different forms of data consumption, different views can be defined. For example, a data scientist may want to see data differently from a more traditional business user. The former may want to have all the required data organized as one wide, flat table, while the latter wants it organized as a star schema. This can be implemented by defining different views for different users.

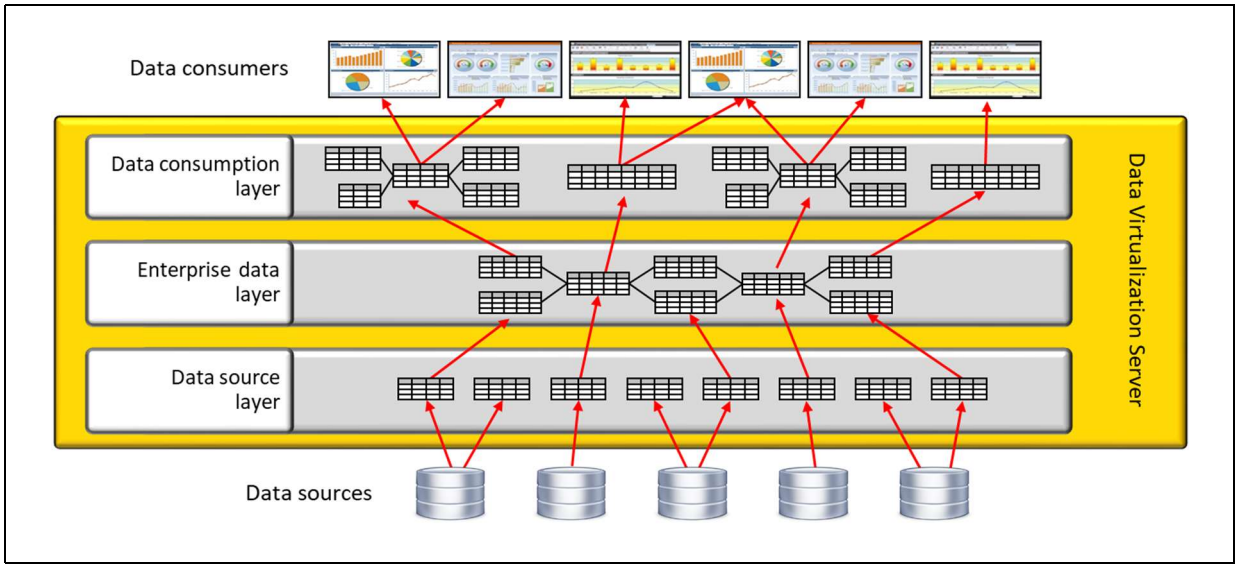

Figure 4 *Stacking of views.*

Materialization of Views – Most data virtualization servers offer an extensive and flexible mechanism for the *materialization of views*. When a view is materialized, the virtual contents of that view is retrieved from the underlying source systems and stored in a database. From then on, when the materialized view is queried, the stored contents is accessed instead of the underlying source systems. In addition, no transformations are required anymore, because the operations have already been applied. The side effect of materialization is that when materialized views are queried, the data returned may no longer be 100% up-to-date. For each materialized view, a refresh specification can be defined that indicates when and how often the virtual contents needs to be updated. Every view can be materialized.

Centralized Data Security and Privacy – Security in any form is of crucial importance. This is especially true when it relates to data. It is of the utmost importance that data accessible via a data virtualization server is protected against any form of deliberate or accidental unauthorized use. Therefore, data

virtualization servers offer a rich set of authentication, authorization, and encryption features to protect the data in the data sources.

Regarding *authentication*, when accessing a data virtualization server, data scientists must present credentials (such as their user name and password) to identify themselves. The data virtualization server checks whether the user is really who (s)he says (s)he is. In this case, the data virtualization server is responsible for authentication checking.

Data security and privacy rules can be centrally defined for all the data.

Authorization rules specify which data scientist is allowed to do what with which data elements. Note that it is not likely that every data scientist needs access to or is allowed to access all the data.

Rules related to data privacy such as data anonymization can be defined in the views as well. For example, data can be scrambled, masked, or aggregated if needed.

Data security and privacy rules defined within a data virtualization server do not 'turn off' the ones implemented in the source systems, those remain active. Nor do data virtualization servers bypass the security systems of source systems. Data virtualization gives organizations a choice. Rules can be defined within the data virtualization server, within the source systems, and within both.

All the data security and privacy rules are centrally defined within the data virtualization server. This centralized security system can be used to protect all the data stored in all the source systems.

Query Performance Through Query Pushdown – Data virtualization servers are designed specifically to access source systems. It is important that they access these source systems efficiently. Therefore, they offer several advanced internal techniques for executing queries on such systems. One of the key features is *query pushdown*. With query pushdown, as many data processing operations as possible, which are making up a query, are 'pushed down' to the source systems. This allows the data virtualization server to maximize the speed of the source system. For example, if the source system is a Snowflake database server, the data virtualization server will let Snowflake do most of the query processing to use its parallel query processing power. On the other hand, if the source system is a simple file system, almost no processing operations are pushed down and the data virtualization server will do most of the processing. The data virtualization server determines dynamically how much query processing is pushed down.

Data Virtuality Platform's Unique Features – The above features are typical for most data virtualization servers. The product called *Data Virtuality Platform* offers some additional features beneficial to data scientists. It offers all the features described in this section, plus the following ones:

- **Supported Database Server:** *Data Virtuality Platform* can use almost any SQL-based database server, including Oracle, Snowflake, and Amazon Redshift, to physically store data. This database server can be used for storing, for example, reference tables, tables with some master data, and also for storing the materialized and replicated views. Data scientists can use this database server for storing test results, training data, sample data, and intermediate data results. This can be done within the confines of the data virtualization server.

- **Replication of Views:** Next to the materialization of views, *Data Virtuality Platform* supports *replication* of views. When replicated, the view becomes a physical table of which the content is initially equal to that of the source. What is special is that the history of the data is stored. When records are updated, the old and the new versions of the record are stored in the table, and when data is deleted, it remains available in the table. So, this content of a replicated table may hold more

Replication of views allows for keeping track of the history of data.

data than its equivalence in the original source system. This is an easy way for data scientists to implement a mechanism to keep track of history in case a source system doesn't.

Note: Data scientists do not 'see' the difference between virtual views, materialized views, and replicated views when accessing them.

- **Procedural SQL Language:** *Data Virtuality Platform* comes with an extensive *procedural language*. In some cases, the structure of the source systems is very complex which may be inefficient to access using just one SQL query. Procedural SQL can then be used to define *user-defined functions* and *procedures* that can be invoked by views (and other procedures). Also, procedural SQL can be used to develop functions that are used by views to access application data while completely hiding the complexity and proprietary-ness of the application's API.

7 Data Virtualization and Data Science

This section describes how data virtualization and specifically *Data Virtuality Platform* can help to overcome all the challenges described in Section 2.

Challenge 1: Data is Everywhere – Data virtualization is all about abstraction. It hides where (location transparency) and how data is stored and how it needs to be accessed. It can show all the data in an integrated way. To data scientists, it feels as if all the required data is stored in one integrated system. Data virtualization also hides whether data is stored in on-premises or cloud-based systems, or whether the data is coming from an internal or external source.

Challenge 2: Data is Stored in a Heterogeneous Set of Source Systems – Data virtualization can show all the data in an integrated fashion through one language and interface. Data virtualization supports query optimization techniques to efficiently execute distributed heterogeneous joins. Data stored in non-flat data sources can be flattened to make them more accessible for data science tools.

Challenge 3: Data is Stored in a Schema-On-Read Form – When data is stored in a structure-less way, the structure can be defined within the views. The definition of the view contains all the logic required to turn the structure-less data into structure-rich data. Note that the data is still stored in a structure-less form, but it is viewed by data scientists as if it has structure. If the structure of the data is highly complex, procedural SQL can be used to transform these complex data structures into virtual, flat data structures.

Challenge 4: Data is 'Hidden' Behind Applications – Data virtualization servers unlock data hidden behind packaged applications. Views can be defined on the interfaces allowing the data to be accessed through the views. Data scientists do not have to deal with the complex, proprietary APIs of the applications and that saves time.

Challenge 5: Data is Cryptic – Views can be defined to transform cryptic data into meaningful data. Additionally, reference tables can be defined in the database server containing the cryptic values and their explanations. These can then be joined with the cryptic data coming from the source systems.

Surrogate key values can be transformed to meaningful data values by joining multiple source system tables and presenting all that data with one view (without surrogate key values) to the data scientists.

Similarly, source systems with complex data structures can be transformed into simple, flat data structures. If the complexity is high, procedural SQL can be used.

Challenge 6: Data History is Missing – In case a source system does not keep track of history, the data scientists can transform a view into a replicated table. Automatically, *Data Virtuality Platform* collects the history of that data. This is an easy mechanism that data scientists can use to keep track of history when required.

Challenge 7: Data Quality is Poor – With views, cleansing operations can be defined that carry out data cleansing operations. Complex cleansing logic can be implemented using procedural SQL and can be invoked from multiple views. Finally, data lineage features show in detail how data flows all the way from the source systems to the data they receive. This offers data scientists a complete audit trail that shows where the data originated from and how it was manipulated and by whom.

Challenge 8: Metadata is Missing – In *Data Virtuality Platform*, the views and their columns can be documented, described, and defined. This metadata, although it may be missing in the source systems, can be added to the views. Data scientists can access all the metadata. Data virtualization also supports *lineage* features allowing data scientists to see how the views are stacked on top of each other and from which source systems the data originates. Note that besides integrating data, Data Virtuality also unifies metadata from all the source systems; one place for the data scientists to find all the metadata.

Challenge 9: Data Security Rules are Restrictive – Data virtualization offers highly detailed and extensive forms of data security. Which views and columns data scientists are allowed to access, can be specified for each of them separately. The advantage of this solution is that all the data security specifications (who is allowed to see what) are centrally stored and managed.

Challenge 10: Data Privacy Rules are Restrictive – Additionally, when data scientists do have access to specific data, but they are not allowed to see the real data, anonymization rules can be defined.

8 The Logical Data Lake: A Data Architecture for Data Science

High-Level Overview – Figure 5 presents a high-level overview of a data architecture in which data virtualization forms the heart. All the data used by data scientists is managed by the data virtualization server. For data scientists, the data virtualization server is their system for accessing data. This architecture is called the *logical* or *virtual data lake*. This naming is in line with the term *logical data warehouse*[11] that has become a popular and flexible alternative to the *classic data warehouse architecture* in which working with multiple physical copies of the data is common.

> The logical data lake is a flexible data architecture that gives data scientists easier and quicker access to all the data.

Section 5 describes the zones of data lakes. These zones can be easily simulated by developing layers of views. Instead of developing physical files in the landing, curated, and production zone, landing views, curated views, and production views are defined within the data virtualization server showing the same data. The difference is that data is not physically copied in the logical data lake. The data processing specifications needed to physically copy the data from one zone to another now end up inside view definitions. This makes the entire environment much more flexible.

Depending on the needs of specific data scientists, access can be granted on the landing, curated, or production views. For example, if they prefer to work with raw data, they can access the landing views, and if they prefer to work with processed data, they access the top layer.

[11] R.F. van der Lans, *Data Virtualization: Selected Writings*, Lulu.com, September 2019; see http://www.r20.nl/DataVirtualizationBook.htm

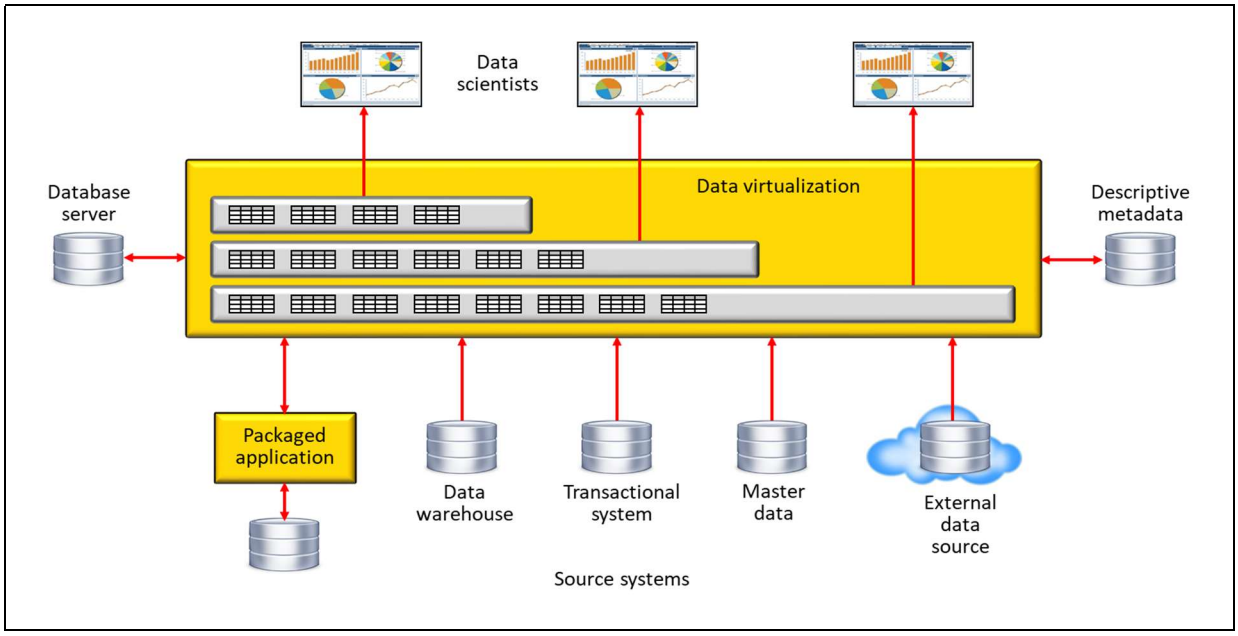

Figure 5 *A data architecture for data science based on data virtualization.*

Characteristics of the Logical Data Lake Architecture

- To data scientists, the logical data lake looks and feels like a central system in which all the data has been stored.
- Data scientists can use one and the same language or interface to access all the data.
- The data virtualization server uses three (or more) layers (the zones) of views to allow different data scientists to access data on different levels of processing.
- The data virtualization server technically organizes access to all the source systems. It integrates data from multiple systems if needed.
- Data processing operations that need to be applied to the data are defined in views and can be reused and shared. For example, if a data scientist has developed a structure-rich view on top of a structure-less file, this view can be reused by all others.
- All the rules for data security and privacy are defined on the views inside the data virtualization server.
- The data virtualization server hides how the data from the source systems is made available. There are a few options. First, data is accessed live when the data scientists ask for it when querying a view. Second, ETL capabilities copy data from the source systems to a separate database that is accessed by the data virtualization server. Third, a view is materialized and the related source data is stored in the database. Fourth, a view is replicated and the data virtualization server starts to collect data history. But whichever option is selected, it is not visible to the data scientists. This also means that changing to another option later does not affect the work of the data scientists.

- The data virtualization server may need access to a system containing *master data*. Master data may be required to determine what the correct data is or to combine data from different source systems. Again, for data scientists, the master data looks like a set of views that is as easily accessible as the source data.

- Data scientists can use metadata to determine the meaning of certain data elements and data values.

- Data lineage functionalities enable data scientists to see a complete audit trail to trace back the data flow all the way to the source systems.

9 Closing Remarks

The business value of data science is undisputed. Descriptive and predictive data science models can really improve and optimize business processes and decision-making processes. Unfortunately, when developing these models, data scientists need to deal with the following challenges:

- Data is everywhere
- Data is stored in a heterogeneous set of source systems
- Data is stored in a schema-on-read form
- Data is 'hidden' behind applications
- Data is cryptic
- Data history is missing
- Data quality is poor
- Metadata is missing
- Data security rules are restrictive
- Data privacy rules are restrictive

By simply copying data to cloud platforms or data lakes, only the first four or five of these challenges are addressed.

The logical data lake architecture with data virtualization at the heart does address all the challenges. It is a flexible data architecture that gives data scientists easier and quicker access to all the data they need. It helps data scientists to deal with current changes: it can hide the fast-changing data storage technology landscape, it can handle the new restrictive regulations for data privacy, and it shortens the time-consuming data preparation tasks.

About the Author

Rick van der Lans is a highly respected independent analyst, consultant, author, and internationally acclaimed lecturer specializing in data architecture, data warehousing, business intelligence, big data, database technology and data virtualization. He works for R20/Consultancy (www.r20.nl), which he founded in 1987. In 2018, he was selected the sixth most influential BI analyst worldwide by onalytica.com[12]. He has presented countless seminars, webinars, and keynotes at industry-leading conferences.

Rick helps clients worldwide to design their data warehouse, big data and business intelligence architectures and solutions and assists them with selecting the right products. He has been influential in introducing the new logical data warehouse architecture worldwide which helps organizations to develop more agile business intelligence systems. He introduced the business intelligence architecture called the *Data Delivery Platform* in 2009 in several articles[13] all published at B-eye-Network.com.

He is the author of several books on computing, including his new *Data Virtualization: Selected Writings*[14] and *Data Virtualization for Business Intelligence Systems*[15]. Some of these books are available in different languages. Books such as the popular *Introduction to SQL* are available in English, Dutch, Italian, Chinese and German and are sold worldwide. Over the years, he has authored hundreds of articles and blogs for newspapers and websites and has authored many educational and popular white papers for a long list of vendors. He was the author of the first available book on SQL[16], entitled *Introduction to SQL*, which has been translated into several languages with more than 100,000 copies sold.

For more information, please visit www.r20.nl, or send an email to rick@r20.nl. You can also get in touch with him via LinkedIn and Twitter (@Rick_vanderlans).

Ambassador of Axians Business Analytics Laren: This consultancy company specializes in business intelligence, data management, big data, data warehousing, data virtualization and analytics. In this part-time role, Rick works closely together with the consultants in many projects. Their joint experiences and insights are shared in seminars, webinars, blogs, and whitepapers.

About Data Virtuality

Data Virtuality is a data integration and management platform for instant data access, easy data centralization and data governance. It empowers companies to get fast and direct insights from scattered data. Data from multiple data sources can be integrated and managed in one interface. This not only simplifies data management but also drastically reduces data integration efforts - by up to 80%. A whole range of renowned businesses around the world, such as Bosch, Audi and DHL, financial institutions like Crédit Agricole and Vontobel, as well as many SMEs are already using the *Data Virtuality* platform to make better use of their data.

[12] Onalytica.com, *Business Intelligence – Top Influencers, Brands and Publications*, June 2018; see http://www.onalytica.com/blog/posts/business-intelligence-top-influencers-brands-publications/
[13] See http://www.b-eye-network.com/channels/5087/view/12495
[14] R.F. van der Lans, *Data Virtualization: Selected Writings*, Lulu.com, September 2019; see http://www.r20.nl/DataVirtualizationBook.htm
[15] R.F. van der Lans, *Data Virtualization for Business Intelligence Systems*, Morgan Kaufmann Publishers, 2012; see https://www.r20.nl/DataVirtualization_V1.htm
[16] R.F. van der Lans, *Introduction to SQL; Mastering the Relational Database Language*, fourth edition, Addison-Wesley, 2007.