CONSULTANCY

Overcoming Cloud Data Silos with Data Virtualization

A Technical Whitepaper

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

May 2020

Sponsored by

TIBCO®

About TIBCO
TIBCO fuels digital business by enabling better decisions and faster, smarter actions through the TIBCO Connected Intelligence Cloud. From APIs and systems to devices and people, we interconnect everything, capture data in real time wherever it is, and augment the intelligence of your business through analytical insights. Thousands of customers around the globe rely on us to build compelling experiences, energize operations, and propel innovation. Learn how TIBCO makes digital smarter at www.tibco.com.

Table of Contents

1 Summary

Cloud Platforms Misconceptions – Organizations no longer consider the cloud an exotic solution and are moving their systems and data to cloud platforms from providers such as Amazon, Google, and Microsoft. Cloud platforms offer valuable benefits, such as unburdening of the IT organization, improved performance and scalability, and a flexible fee structure.

Despite the popularity and widespread use of cloud platforms, many persistent misconceptions exist:

- The cloud is one centralized environment
- The cloud stores all the data
- The cloud is a homogeneous environment
- The cloud handles data integration

Like most on-premises environments, the cloud is a highly distributed and heterogeneous environment. It is distributed, because most organizations have or will have a *multi-cloud strategy* (deploying multiple cloud platforms) and a *hybrid cloud strategy* (running systems on cloud platforms and on premises).

Numerous technologies for data storage and analytics are available on cloud platforms. This heterogeneity has been deliberately introduced. With so many different, specialized data storage technologies available, a wide range of use cases can be supported, including cases where massive amounts of data, transactions, queries need to be processed. These technologies allow organizations to exploit the full power of cloud platforms.

Cloud Data Silos – For all types of data consumption, such as reports, dashboards, analytics, mobile apps, portals, and websites, data from different systems still needs to be integrated. *Data integration* was, is, and will always be a complex process of transforming all the source data into meaningful data. The cloud does not simplify this data integration challenge.

Cloud data silos have exacerbated data integration challenges.

What has happened on pre-cloud environments over the years, the development of data silos, has already happened on cloud platforms, the emergence of *cloud data silos*. In the cloud, data is also stored on different cloud platforms, in different systems, using different data storage technologies. In the cloud era, the need for data integration remains. In fact, adoption of cloud platforms may even exacerbate data integration challenges as organizations deploy hybrid and multi-cloud strategies.

Approaches for Data Integration – This whitepaper describes and compares four approaches for integrating cloud data silos in a sensible and practical way:

- Data integration with applications
- Data integration with a data lake
- Data integration with a data warehouse
- Data integration with data virtualization

Each approach is evaluated based on four aspects: data integration, query performance, data security and privacy, and development productivity of new reports and applications.

Comparing Approaches for Integrating Cloud Data Silos – With the first two approaches, most of the development work is done by *business developers* and not by *data engineers*. As a result, in both approaches, data integration specifications for different applications are independently developed using different tools and techniques. In other words, the wheel is re-invented over and over again. This reduces productivity, complicates maintenance, and can lead to different data consumers seeing different report results.

In a data warehouse environment, the data engineers do most of the development work. The data integration specifications are developed once by data engineers and reused many times. The drawbacks are that it is not a flexible environment, because data is copied several times before it becomes available for data consumption; it has not been designed and optimized to support all forms of data consumption, but primarily traditional forms of reporting, analytics, and dashboarding; and the data latency is generally high, which is unacceptable for data consumers who require near real-time data.

Using Data Virtualization to Integrate Cloud Data Silos – The fourth approach is based on *data virtualization* technology. It combines the best of the previous approaches. As in a data warehouse environment, with data virtualization all the data integration specifications are developed once by data engineers and reused many times. This improves development productivity. There is less need to create redundant copies of the data which makes supporting near-real-time data consumers a reality, because of the lower data latency. Because a data virtualization server offers access to all types of data storage technologies, organizations can take advantage of all the specialized technologies available on cloud platforms. But the key benefit of data virtualization is that it decouples data consumers from data storage. It hides the fact that a heterogeneous, hybrid, multi-cloud environment is in use.

Data virtualization hides a heterogeneous, hybrid, multi-cloud environment.

Summary – With data virtualization, it is relatively easy to overcome the cloud data silos and to transform them into one integrated and flexible environment whilst preserving the advantages of cloud platforms and leveraging all the new data storage technologies.

2 The Misconceptions of Cloud Platforms

Advantages of Cloud Platforms – The advantages of cloud platforms from providers, such as Amazon, Google and Microsoft Azure, are beyond dispute. Three groups of benefits exist:

- **Unburdening the organization:** When an organization moves data processing to a cloud data platform, countless everyday hardware and software-related tasks are taken care off. Examples of such tasks are: managing a fully equipped data center, dealing with failing hardware components, organizing a perfect secure data center, implementing for power outage solutions, organizing fallback environments, and establishing backup and restore mechanisms and procedures.

- **Data processing scalability and performance:** Cloud platforms seem to offer an endless level of scalability with respect to processing power, storage, and memory. All the resources that organizations need are potentially available. This is also true when organizations are faced with an (unexpected) peak workload, allowing them to (temporarily) increase the required resources. Additionally, if the workload of new applications is unpredictable, starting small and scaling up later is easy.

- **Flexible fee structure:** The fee of cloud platforms and many software services is based somehow on the intensity of usage; informally called 'pay by the sip.' For example, it can be based on usage of data storage, usage of processor cores, or the number of queries executed. This flexible fee structure is attractive to many organizations.

Evidently, drawbacks exist, but for most organizations, they outweigh the benefits.

Misconceptions of Cloud Platforms – Despite the popularity and widespread use of cloud platforms, many misconceptions exist:

- **Misconception 1 – The Cloud is One Centralized Environment:** In many diagrams, cloud platforms are drawn with one simple cloud icon suggesting that a cloud platform is a centralized and integrated data store. Nothing could be further from the truth. On cloud platforms data is also stored in a multitude of data storage systems. In addition, most organizations deploy a so-called *multi-cloud strategy*, which means that multiple cloud platforms are used. Flexera[1] reports that "in 2019, 84% of the organizations rely on a multi-cloud strategy relying on one." And according to InformationAge[1] "60% of all businesses in the sector expect their IT environment to be multi-cloud, integrating both on-premises and externally hosted cloud infrastructure. Only 18% say they will solely rely on the public cloud." Organizations may have chosen for a multi-cloud strategy on purpose, but it may also be 'enforced' upon them due to, for example, an acquisition.

- **Misconception 2 – The Cloud Stores All the Data:** An almost endless amount of data can be stored on cloud platforms. This does not mean that organizations will store all their data there. The ones that have are exceptions. For several reasons, there still remains a great deal of data that is not or cannot be migrated to the cloud. One reason may be that the data resides in a platform that cannot operate on a cloud platform; regulatory rules prohibit the storage of data in cloud platforms; or, the data is produced in massive quantities and the network delay to push it to the cloud is too long. In other words, for now and for the foreseeable future, some data will still be stored on premises. This is commonly called a *hybrid cloud strategy*. Flexera[1] reports that "the adoption of a hybrid cloud strategy grew from 51% to 58% between 2018 and 2019."

- **Misconception 3 – The Cloud is a Homogenous Environment:** All the cloud platform vendors offer many different technologies for storing data. They offer well-known SQL database servers, such as Oracle, MySQL and PostgreSQL, their own, more proprietary and non-portable database servers, such as Amazon RedShift, Google BigQuery, and Microsoft SQL Azure; file systems for storing data such as Hadoop; proprietary file systems, such as Amazon S3 and Microsoft Azure Data Lake; and NoSQL systems, such as MongoDB and Cassandra. Because all these systems have their use cases, no organization will end up with storing all their data on one cloud platform using one data storage technology. For some use cases, a file system is preferred, while for others a generic SQL database engine, a NoSQL system, or something even more specialized as a graph database server, would be more suited. The result is that, from a data storage and data access perspective, a cloud platform is not a homogeneous but a highly heterogeneous environment. More on data storage technologies in Section 3.

---

[1] Hostingtribunal.com, Cloud Adoption Statistics for 2020; See https://hostingtribunal.com/blog/cloud-adoption-statistics/#gref

- **Misconception 4 – The Cloud Implies Data Integration:** Moving data, systems, and applications to the cloud does not integrate them, even when everything is moved to one cloud platform. It does not even make them easier to integrate. Data integration remains a big challenge.

  More and more types of data consumption require access to many data elements that reside in a multitude of systems. This data needs to be integrated before it can be used. For example, data scientists and more traditional forms of business intelligence require data coming from many different systems and needs to be integrated before it can be used. Copying all the data from these systems to one cloud platform or to one file system on a cloud platform does not qualify as data integration. Data integration entails much more than copying all the relevant data to one platform. Additionally, moving all the data is not always allowed, nor is storing certain types of data.

3 Cloud Data Storage Technologies

This section describes the heterogeneity of the data storage technology landscape.

The New Diverse World of Data Storage Technologies – Each cloud platform offers a plethora of technologies for storing, manipulating, and querying data. They range from simple file systems to the most advanced database servers. For every imaginable use case a suitable data storage technology exists. Many are designed and optimized to process *big workloads*, such as storing and managing massive data volumes, fast ingesting of immense data streams, and processing workloads consisting of highly complex queries.

To show the diversity of the market, Table 1 lists some of the many alternatives available on the three popular cloud platforms. Besides these solutions, organizations can also choose from numerous products that are available on most of these cloud platforms, such as Hadoop, Kinetica, MemSQL, MySQL, PostgreSQL, and SnowflakeDB.

Cloud Platform	Data Storage Technology
Amazon AWS	Aurora DocumentDB DynamoDB Elasticache for Redis RDS Redshift S3 Timestream
Google	BigQuery Cloud Bigtable Cloud Firestore Cloud Spanner Cloud SQL
Microsoft Azure	Cache for Redis Cosmos DB Data Lake SQL Database Synapse Analytics

Table 1 *Some alternative data storage technologies available on cloud platforms.*

From General Purpose to Specialized Data Storage Technologies – How did this market become so extensive and diverse? There was a time when most of the products for storing and manipulating data were *general purpose products*. They were suitable for almost any kind of workload, including a transactional workload and an analytical workload, and they could support data warehouses, websites, portals, and so on. They were good at everything, but did not excel at anything.

For the new generation of big data systems, being general purpose is not always good enough. Products had to support e.g. massive BI workloads, massive transactional workloads, massive graphical analytical workloads, and massive data ingestion workloads. Many of the general purpose systems were unable to handle these workloads, opening the market for more specialized products.

Many new data storage technologies are specialized technologies.

The current market of data storage technologies can be compared to track and field. In this sport you have specialists in sprinting, high-jumping, and throwing discusses, and next to that you have the overall athletes that are good at everything, the decathletes. Most decathletes are good at everything, but they do not excel at one specific sport. The latter group can be compared to general purpose database servers and the sprinters and high-jumpers to the more specialized data storage technologies.

This need for specialized products has also increased the heterogeneity of the market of data storage technologies. For example, a graph database server such as Neo4j excels at graph analytics; Redis excels at processing massive lookup tables in memory for fast look up queries; and object storage systems, such as Dell EMC Elastic Cloud Storage, excel at handling large objects, such as images, videos, and documents. And most of them support proprietary interfaces.

Unique Features – Besides being able to process massive workloads, some of these products offer very valuable and unique features. For example, SnowflakeDB offers end-to-end encryption of the data, Neo4j supports graph analytical capabilities not offered by most other products, Edge Intelligence offers a distributed database for storing and querying massive amounts of remote sensor data, and some products offer unique self-managing features, shortening the time needed to optimize and manage the product.

Exploit Specialized Data Storage Technology – When adopting cloud platforms, organizations can do what they have always done and try to use one and the same data storage technology for all their use cases. This would undoubtedly result in selecting a general purpose technology. In this case, the special features and scalability levels of specialized data technologies are not exploited. This may result in unnecessary productivity, performance, and scalability problems.

Do not miss out on the benefits of specialized data storage technologies. They can be enormously beneficial. In fact, one of the reasons why organizations should adopt cloud platforms is to be able to deploy these newer and highly specialized technologies. Organizations need to accept that their data architectures will most likely include multiple different data storage technologies, even if only one cloud platform is selected.

Specialized data storage technologies can be enormously beneficial.

Additionally, the last ten years have shown numerous new groundbreaking data storage technologies, especially on cloud platforms, and it is unlikely that this avalanche of new technologies is going to dry up in the foreseeable future. Faster, more scalable, and more self-managing technologies will become available. Organizations need to design data architectures that are able to exploit all these new and upcoming data storage technologies available on cloud platforms.

4 The Era of Cloud Data Silos

Data Silos and the Need for Data Integration – Before the cloud era, every organization had its data scattered across multiple machines, multiple systems, and many different data storage technologies, including SQL databases, mainframe databases, files in all kinds of formats, and spreadsheets. Most of these systems were developed, maintained, and managed in isolation. Data was spread across *data silos*.

For all forms of data consumption, such as reports, dashboards, analytics, mobile apps, portals, and websites, data from the silos must be integrated first. *Data integration* is a complex process of transforming all the source data into meaningful data. It involves filtering, aggregating, cleansing, joining, masking, calculating data, and many more operations. For example, data warehouse specialists and data scientists spend most of their development time (called data preparation) on developing, maintaining, and managing the appropriate data integration specifications. Data warehouse projects and, more recently, data lake projects have been started to help with the data integration challenge of data silos.

The more data is siloed, the more complicated data integration is. Because data is being used more intensely and more widely by organizations, the need for data integration is increasing.

Cloud Data Silos – Moving data and systems to cloud platforms will not change the data integration challenge. What has happened on pre-cloud environments, the development of data silos, has happened on cloud platforms as well, the development of *cloud data silos*. In fact, it started to happen on the first day cloud platforms were adopted. In the cloud, data is also stored on different cloud platforms, in different systems using different data storage technologies. By moving data to the cloud, the need for data integration remains. It may even be more complex, because source systems are now scattered across multiple cloud platforms *and* on premises systems.

Cloud data silos have existed since the birth of the cloud.

5 Business Developers and Data Engineers

Data Integration is a Complex Process – Data originating in source systems needs to undergo many changes before it has the right form, quality level, and level of detail. In other words, data needs to be integrated before it becomes *consumption-ready*. *Data integration* is a complex process that involves filtering, decoding, and aggregating data; keeping track of the data history to support historic analysis, but also for governance and auditing requirements; correcting data; anonymizing data to adhere to data privacy regulations; securing data against incorrect use; and so on. In actual systems, many *data integration specifications* need to be designed, developed, and maintained.

Two Groups of Developers – Two groups of developers for data integration specifications exist: *data engineers* and *business developers*.

Data engineers work on general, foundational modules and systems that are accessed and used by many data consuming applications. In other words, multiple applications make use of what data engineers develop. Examples of data engineers are DBAs, ETL developers, system administrators, data security experts, data warehouse designers, and data administrators.

Data engineers work on general systems, and business developers work on data consuming applications.

Business developers are responsible for developing the applications that consume the data. Examples of such applications are Excel spreadsheets, data science models, BI dashboards, and balanced scorecards. They are responsible for the last mile that data has to travel. They normally work on one application or application environment at a time. Examples of business developers are data scientists, super users, power users, citizen analysts, self-service BI users, and spreadsheet developers.

Business developers use the data delivered by the products developed by data engineers. The more data integration work is performed by the data engineers, the more the data is consumption-ready for business developers and the less time they have to spend on development work. For example, if data engineers have developed an entire data warehouse architecture, including staging areas, central data warehouses, and data marts, minimal development work is required by the business developers to convert that data into valuable reports and dashboards. Much of the data integration work has been done for them. While, when data is copied from source systems and copied to flat files, business developers need to spend a lot of development time on data integration.

Four Approaches to Data Integration of Cloud Data Silos – Cloud data silos also require data integration. The following four approaches for integrating cloud data silos are described and compared:

- Data integration with applications
- Data integration with a data lake
- Data integration with a data warehouse
- Data integration with data virtualization

Each approach is evaluated based on four aspects: data integration, query performance, data security and privacy, and development productivity. With regard to data integration, the approach is evaluated based on the division of development work between data engineers and business developers. Figure 1 shows an example of how this is represented. The grey dots in the diagram represent data integration operations, such as filtering, joining, masking, and decoding. The data sources are shown on the left. In this example, two sources run on cloud platforms and the third on premises. The data consumer applications are shown on the right, which can be spreadsheets, self-service BI tools, mobile apps, and data science tools. The sizes of the large green boxes reflect the amount of development work performed by each developer group. In this example the data engineers do most of the development work and the business engineers do some work.

Note that each of the approaches described in this whitepaper has its merits and may be the appropriate choice depending on the use case.

6 Data Integration With Applications

Introduction – Cloud data silos can be integrated by implementing all the required data integration specifications within the data consumer applications. Figure 2 shows the effect of this approach on the distribution of work among data engineers and business developers. Here, business developers implement almost all the data integration specifications. Business developers may be using different reporting, analytics, or development tools. For example, some may use TIBCO Spotfire, others TIBCO JasperSoft or Microsoft Excel, and the data scientists may use Python.

In this approach data engineers are responsible for the source systems, but not for data integration. They may develop special applications to extract data from the source systems and copy it to special files or databases. In this case, business developers do not have to access the source systems directly, nor will their applications interfere with the workload on the source systems.

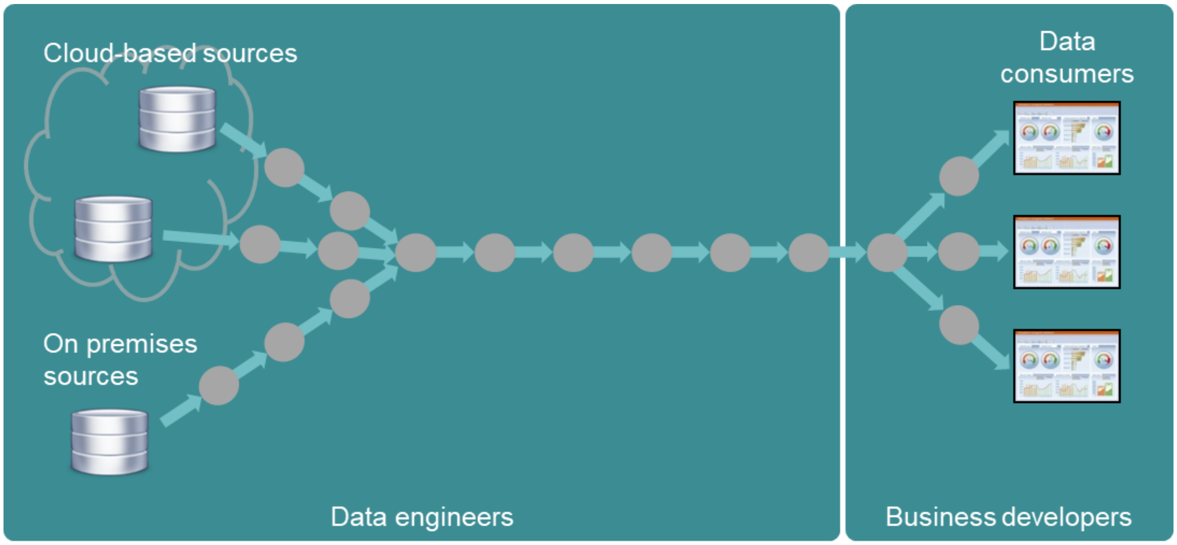

Figure 1 *Dividing the development work of data integration operations among data engineers and business developers.*

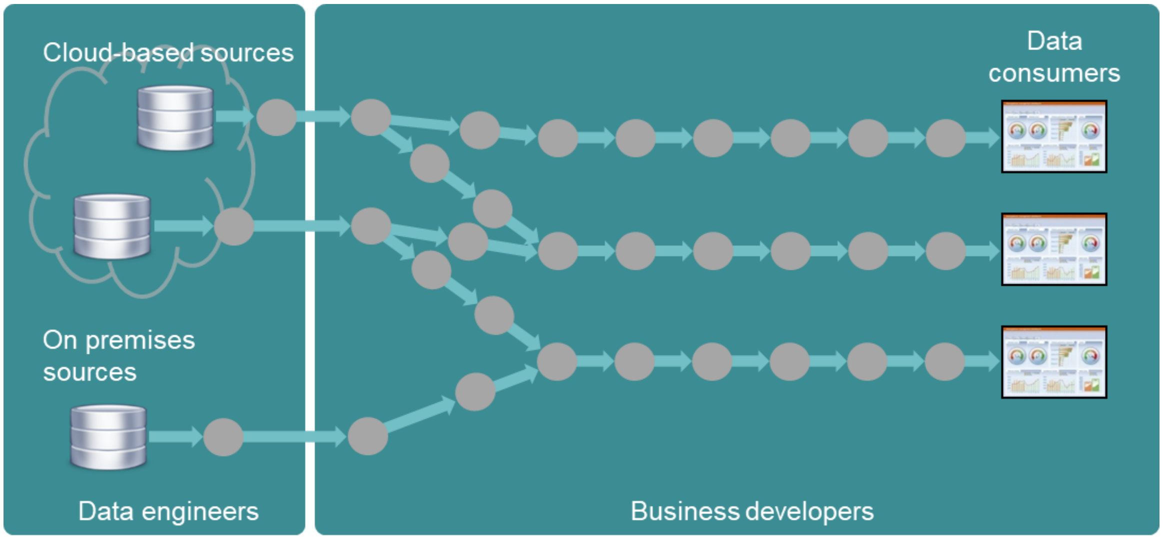

Figure 2 *Data integration by applications.*

Data Integration – As indicated, in this approach almost all the data integration specifications are developed by business developers. They need to develop all the data integration specifications. The work of data engineers may be restricted to developing small applications that extract data from the source systems and copy it to temporary data stores. Typical for this approach is that there is minimal or no reuse of the data integration specifications. Specifications developed for one data consumer application are not reused by another. Business developers may not even be aware that certain specifications have already been implemented somewhere else. And even if they know, it may be developed with a different tool and language.

This approach is technically feasible, but not ideal from multiple perspectives. First, it is very development-intensive, because the wheel is reinvented over and over again. Second, there is no guarantee that the same specifications are implemented consistently, and if it is not, it leads to inconsistent data results. For example, business users of a dashboard may see slightly different sales data, then those using a specific spreadsheet. Third, keeping track and maintaining all the different implementations of the data integration specifications is complex and cumbersome.

Query Performance – Every time a data consumer application requires data, it is extracted from the source systems, the data integration specifications are applied, and results are presented by the applications. This is inefficient, because the source systems are continuously queried. The same query may be executed over and over again. Also, the same integration specifications are executed repeatedly.

Integration of cloud data silos with applications can be inefficient.

When data needs to be retrieved from multiple source systems, the tools used by business developers usually do not have the features to optimize network traffic, which could lead to too much data being transmitted to the applications, as no distributed join optimization takes place.

If applications retrieve data directly from the source systems, it may cause an unacceptable level of interference. To minimize interference, data can be extracted periodically and stored in some temporary data store. However, this data store needs to be developed, maintained, and managed, and applications must be developed to refresh them periodically.

Data Security and Privacy – Different source systems use different data security mechanism. Because the data consumer applications access the source systems directly, business developers need to understand all the complicated ins and outs of these different security systems.

When a temporary data store is deployed and operated outside the realm of the source systems, the question is how that data is protected against unauthorized usage.

When data from the source systems must be anonymized before usage, the applications need to incorporate solutions. Besides the fact that it requires development time, the question is whether the tools support professional features for anonymization and comparable aspects.

Development Productivity – Developing the correct data integration specifications can be highly complex and time-consuming. It requires a deep understanding of the business developers on how the data values and data structures are organized in the source systems.

Data integration solutions are not or minimally reused. It is highly unlikely that different business developer groups (probably using different development tools) share their solutions. The wheel is reinvented over and over again. This decreases productivity and complicates maintenance.

Development of the correct data integration specifications by business developers can be highly complex and time-consuming.

Minimal reuse of data integration solutions can also lead to inconsistent data results. For example, business users of a dashboard may see slightly different sales data, then those using spreadsheets.

7 Data Integration With a Data Lake

Introduction to the Data Lake – A popular approach for integrating cloud data silos is by developing a *data lake* or *data hub.* Data is extracted from the on-premises and cloud-based source systems and copied practically *unchanged* to the data lake; see Figure 3. All the required data is copied from the cloud data silos to one data storage system. In most cases, this data storage technology is a file system, such as, Azure Data Lake or Amazon S3.

When all the data is stored in a data lake, it can be accessed with one technical interface and with one data security system. This makes it easier for applications to access the data than when the individual source systems need to be accessed separately. The workload on the data lake will not disturb that of the source systems.

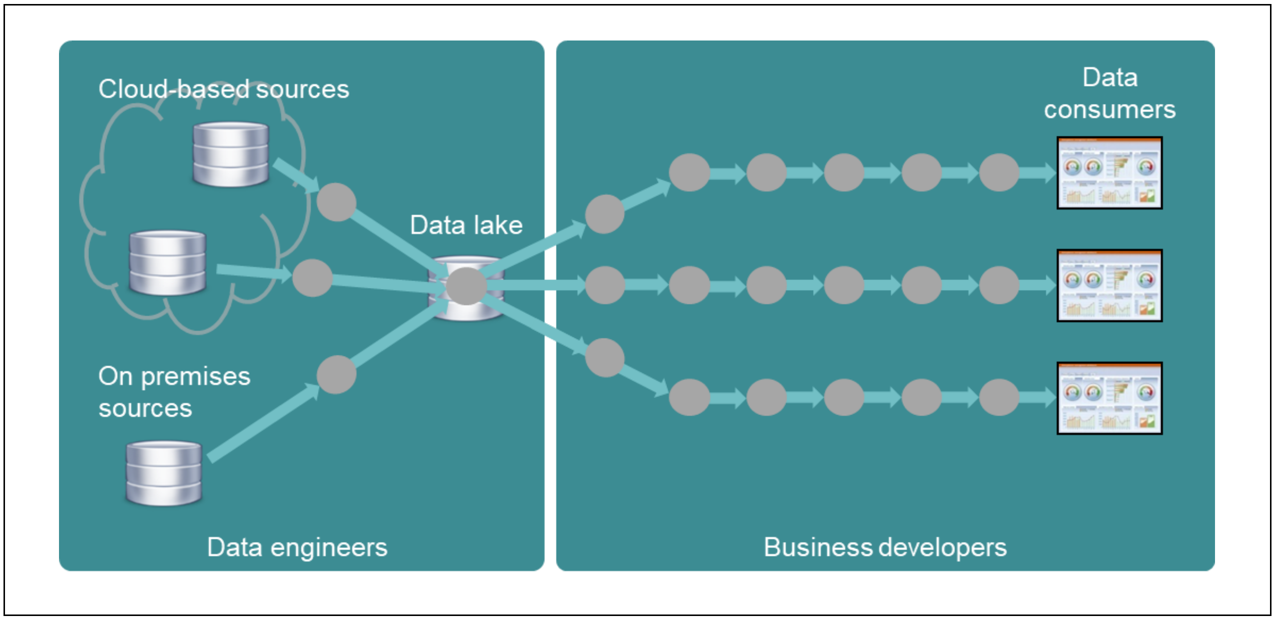

Figure 3 *Data integration with a data lake.*

When data is moved to a data lake it is not or minimally integrated.

Data Integration – Figure 3 illustrates how development work is divided between data engineers working on generic solutions and business developers working on the data consumption applications. When data is moved to a data lake it is not or minimally integrated. Retrieving all the data from one (the data lake) rather than multiple sources does simplify data integration somewhat. With a data lake, almost all the development work on data integration is done by business developers. Sharing solutions is complex and the risk of inconsistent data results is high.

Query Performance – Many file systems on cloud platforms used for data lakes support massively parallel query processing. This accelerates access to data even when it concerns big data. This means that copying all the data to a data lake using such a file system, may provide the right query performance.

Unfortunately, it is very common to select one specific file system technology for developing a data lake. Each file system is designed to support a limited number of use cases well. The question is whether the use case of the expected workload matches that of the file system. For example, if a specific application needs to access individual records by key, a general file system may not be optimal, while a key-value data storage technology might be. In such a situation, some data consumers will experience a perfect performance, while others will not. The performance may be so unacceptably slow that data needs to be extracted from the data lake and copied to a specialized data storage technology that offers the right performance. In general, a data lake does not exploit specialized data storage technologies; see Section 3.

The wide range of data consumption forms is one of the key reasons why cloud platforms offer so many different data storage technologies. An ideal technology exists for most use cases. With a data lake you miss out on all the strengths and benefits of the available technologies. In other words, by placing all the data into a data lake using one centralized data storage technology, many of these specialized technologies are bypassed.

Data Security and Privacy – Consumer applications only need to deal with the data security system of the data lake. This simplifies data access. The challenges are, first of all, whether such a data lake delivers a rich set of data security capabilities, and secondly, whether suitable anonymization capabilities are supported, or else the business developers need to develop those themselves.

Development Productivity – With a data lake, business developers are responsible for most of the development work. This is detrimental to overall productivity and maintenance, because over and over, similar solutions are developed for different applications. There will be no or minimal sharing of solutions, and definitely no sharing of solutions across different solutions. The responsibility of the data engineers is limited to developing solutions that periodically extract data from the source systems and copy it into the data lake.

Data Herding – As an aside, the data lake and similar approaches where data is copied unaltered to a centralized storage environment can be called *data herding* approaches. Wikipedia defines herding as follows: "Herding is the act of bringing individual animals together into a group (*herd*), maintaining the group, and moving the group from place to place." Data herding can similarly be defined as "the act of bringing data from multiple systems together into a data store, maintaining the data, and moving the data from place to place."

With data herding business developers are responsible for data integration.

When cows are herded to another pasture, the cows are not changed, they remain the same cows, with all their characteristics, peculiarities, and errors. There was no structure in how they were organized in the pastures they came from, not is there one in the new pasture. It does not reduce the amount of work it requires to process the cows and convert them into consumable meat. It is the same with data. When data is herded to a data lake, it does not reduce the amount of work required to turn data into consumption-ready data. Data herding puts the development of data integration specifications entirely in the hands of the business developers.

8 Data Integration With a Data Warehouse

Introduction – A traditional way of integrating systems is by developing a *data warehouse environment*. This also applies to integrating cloud data silos. In a data warehouse environment, data is extracted from the data source systems, copied to a *staging area*, then it is extracted from the staging area and copied to a *central data warehouse*. In this process, data is commonly integrated and processed. In many data warehouse environments, data is then again extracted and copied to *data marts*. Such a data warehouse environment can reside fully or partially on premises or on a cloud platform. Figure 4 illustrates how the development of a data warehouse environment is divided between data engineers and business developers.

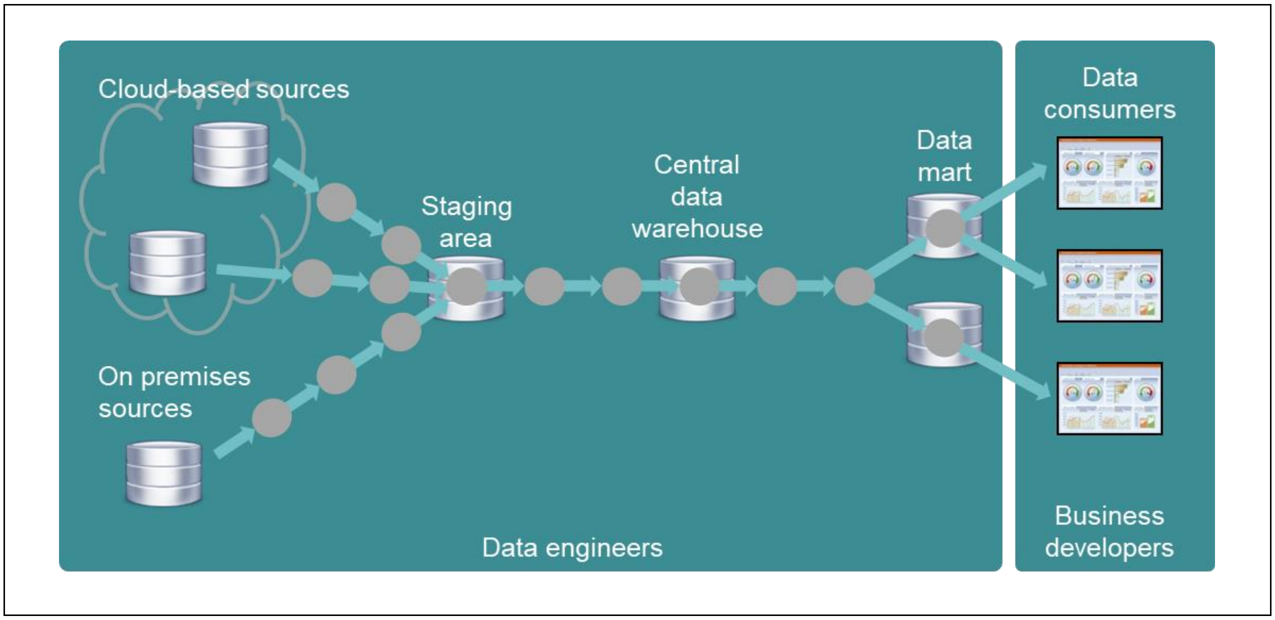

Figure 4 *Data integration through data warehousing.*

Data Integration – Whether they access the data warehouse or a data mart, with data warehousing business developers experience an integrated data environment. Data is delivered to the data consuming applications in the required form.

A data warehouse environment is a proven solution for cloud data silo integration in general, but not for every form of data usage. For example, it is not the right solution for mobile apps used by thousands of customers concurrently, for data scientists developing models using exploratory techniques, or for applications that require access to near-real-time data because the data latency is too high.

Also, if the source systems contain voluminous amounts of big data, it may be too slow and expensive to copy all the big source data to the data warehouse environment.

Query Performance – Query performance is fully dependent on the data storage technology selected, the amount of data, the query complexity, and the amount of consumer queries. With regard to the data storage technology, a different one can be selected for each data mart, allowing the right technology to be used depending on the nature of its workload. This enables exploitation of the diversity of technologies available for data storage. For example, one data mart can be developed with a GPU-based, analytical SQL database server and another with a specialized graph database server.

In a data warehouse environment, data must be copied a few times before it becomes available for consumption. This increases the data latency and makes it unsuitable for certain use cases.

Data Security and Privacy – Most data warehouse environments are developed with SQL database servers that support advanced mechanisms for data security. Most of the data security rules are defined once by data engineers. When data is copied to the data marts, it can be anonymized. Business developers do not concern themselves with data security and privacy aspects.

The complexity of a data warehouse environment limits it flexibility.

Development Productivity – Because data engineers are responsible for a large portion of the data integration work, not much development work on data integration specifications is left to the business developers. Their productivity is high, because they can focus purely on consumption of the data.

It is a different story for data engineers. Developing the entire data warehouse environment (staging area, central data warehouse, and several data marts) is quite an exercise. To periodically refresh all the data, applications need to be developed and scheduled and the proper and reliable execution must be managed. The complexity of such an environment is the main reason why they are not very flexible. Ostensibly, implementing simple changes to reports can lead to a waterfall of changes throughout the entire environment. Making changes to the data structures and ETL programs can be time-consuming. This lack of flexibility is a well-known problem of many data warehouse environments.

9 Data Integration With Data Virtualization

The fourth approach for integrating cloud data silos is based on using data virtualization.

Note: Several data virtualization servers are available on the market. Most of the statements about data virtualization made in this whitepaper apply to all, and, where mentioned, some apply specifically to the *TIBCO Data Virtualization* product (TDV).

Introduction – The key aspect of data virtualization is *abstraction* or *decoupling*. With data virtualization, applications are decoupled from data stores and data producers. Data consumers see data, but they have no notice of where and how data is stored. They are shielded from those aspects. Data integration specifications are defined within the data virtualization server. A data virtualization server can be implemented on-premise and in the cloud itself. For a detailed description of data virtualization see the book *Data Virtualization for Business Intelligence Systems*[6].

Figure 5 illustrates how development is divided between the data engineers and business developers when data virtualization is deployed. Most of the development work is performed by the former group. Depending on their technical skills, business developers can also define data integration specifications within the data virtualization server. The consequence is that both development groups are using one and the same platform for implementing the data integration specifications.

Data Integration – Data virtualization supports *on-demand data integration*. Storing integrated data redundantly, as in the data warehouse environment, is optional. Data stored in the source systems is directly accessible, but can also first be copied to other systems and accessed there. Whatever the solution, it is hidden from all the data consumers.

Due to its abstraction capabilities, data can be migrated from one data storage technology to another without impacting the data consumers, even if the new home deploys a different technology. Migrating data to another cloud platform can also be hidden from all the consumers. Implementing a hybrid cloud strategy using specialized data storage technologies becomes much easier to implement.

Data virtualization simplifies the implementation of a hybrid cloud strategy using specialized data storage technologies.

To all the data consumers, data virtualization transforms the hybrid cloud into one logical database. It unifies all the source systems.

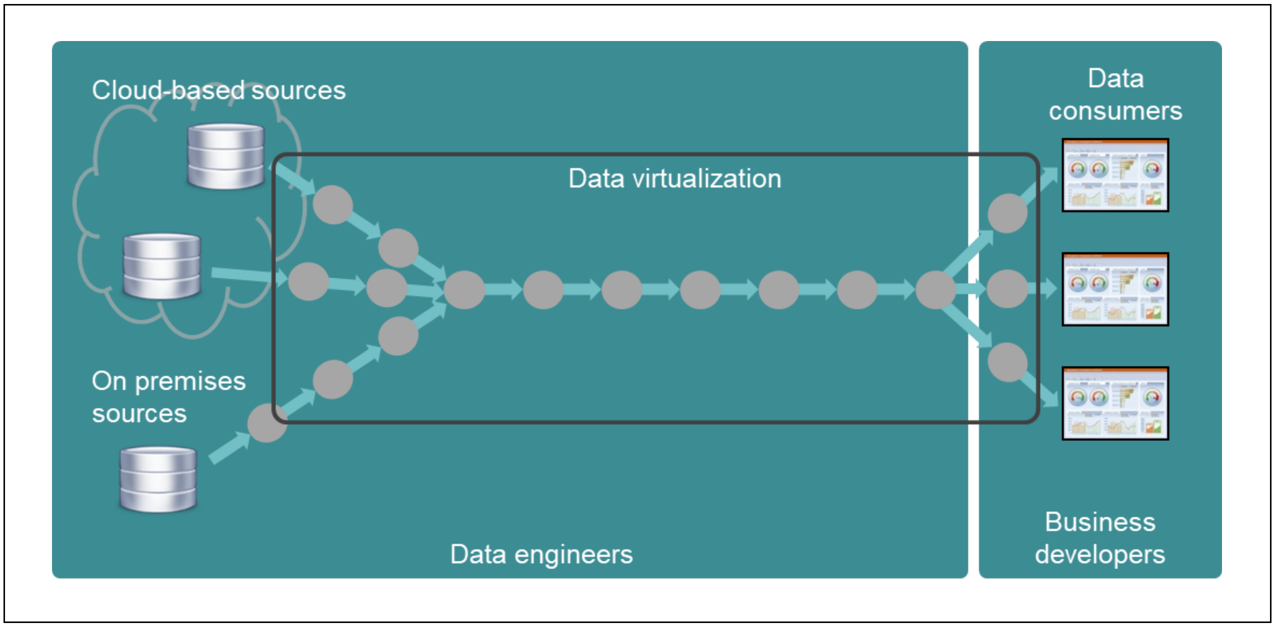

Figure 5 *Data integration through data virtualization.*

Query Performance – Data virtualization servers offers several features to improve query performance:

- **Query Pushdown:** An important aspect that determines the performance of data virtualization servers is *query pushdown*. In general a data virtualization server processes a query coming from a data consumer in a few steps. First, the incoming query is extended to include all the operations necessary to apply all the required data integrations specifications (e.g. transformations, aggregations, filters, calculations, and joins). Next, this extended query is 'pushed down' to the underlying source system. The source system executes the query and the result is sent back to the data virtualization server which then passes it on to the data consumer.

  This is how it works when the source system is able to process the entire query. If not, then the data virtualization server pushes down only those operations supported by the source system. The result is sent back to the data virtualization server, which then executes the remaining operations after which the result is sent back to the data consumers. In other words, a data virtualization server tries to push as many of the query operations down to let the source systems do most of the processing. It does this to fully exploit the strengths of the data source. In other words, it uses the full power that systems, such as SnowflakeDB, Oracle, and Amazon S3, have to offer.

  With data virtualization, the data remains in the source systems. Queries are executed on those platforms. This is very different from the other approaches, such as the data lake where all the data is herded into one and the same system.

- **Caching Query Results:** Most data virtualization servers support a mechanism called *caching*. All the data integration specifications are defined within *views* (aka virtual tables). The views are the objects that business developers see when accessing a data virtualization server. When caching is activated for a view, its virtual contents is determined and physically stored in a database server. Data engineers can select in which database server the cached contents needs to be stored. When a data virtualization server receives a query on a cached view, the source systems are not accessed, but the cached view. All the specifications that make up the view have already been executed, so working with caches can significantly improve performance.

A refresh scheme can be defined to indicate how often and when the cached contents needs to be updated to stay synchronized with the source systems.

Note that the cached views are managed by the data virtualization server.

- **Parallel Query Processing:** More and more data storage platforms support parallel query processing. They can take advantage of the massively parallel hardware available on cloud platforms. TDV supports parallel query processing. This avoids that TDV becomes the performance bottleneck in a massively parallel environment. Its internal architecture matches the new big data technologies. TDV can distribute data processing across hundreds of nodes in a similar way as some of the big data technologies. The internal workings of parallel query processing by TDV is described in detail in the whitepaper[2] "Data Virtualization in the Time of Big Data".

Data Security and Privacy – Almost every data storage technology comes with its own data security system for defining authorization, authentication, and encryption rules. They support different concepts, different capabilities, and different interfaces. Any data integration solution needs to work with all these differences when integrating data.

Data virtualization can simplify all this. It can act as a centralized data security layer on top of all the source systems. What they are allowed to do with the data can be specified for each user separately. Also, authorization rules can be defined on different levels of detail: on the virtual table, column, row, or individual value level. In other words, for data storage technologies with poor data security capabilities, data virtualization acts as a more sophisticated data security layer.

Many source systems do not yet support anonymization capabilities. A data virtualization server can anonymize the data.

Development Productivity – To business developers, working with a data virtualization server is very similar to working with a data warehouse environment. The data from the cloud silos has been integrated and is consumption-ready, allowing them to focus on the requirements of the data consumer applications.

The amount of development work of business developers when using data virtualization is comparable to when a data warehouse environment is used. There is, however, an important difference. Defining the data integration specifications is easier with data virtualization than with a data warehouse environment, because there is no need to develop, maintain, and manage additional databases. Data does not need to be herded first and physically copied. Developing data copies is optional, not mandatory. Mechanisms, such as caching, can be used to create temporary copies.

When every new application and each new report needs to discover and define how data from the cloud data silos has to be accessed, much time is lost. In fact, the wheel is reinvented over and over again. They also need to deal with all the different interfaces, languages, and so on.

With data virtualization, many specifications to integrate data silos are defined once and reused many times.

With data virtualization, many specifications for data integration, data security, data filtering, data aggregation, data calculations, and data anonymization are defined once, and, more importantly, can be reused many times. Every data consumer, whether it is a dashboard, report, data scientist, or a mobile app, can access the same views. A specification that defines how to integrate customer and invoicing data needs to be defined only once, but can be reused by anyone. This seriously improves the

---

[2] R.F. van der Lans, *Data Virtualization in the Time of Big Data*, December 2017; see https://www.tibco.com/resources/whitepaper/data-virtualization-time-big-data

productivity of data engineers. The define-once approach also simplifies maintenance, because if a specification needs to be changed, it only needs to be changed in one place.

10 Closing Remarks

From day one, the cloud was a heterogeneous, distributed, non-integrated environment. Cloud data silos have always existed. For many forms of data consumption, the cloud data silos need to be integrated. Table 2 contains a high-level comparison of the four approaches for integrating cloud data silos described in this whitepaper.

Aspects	Data integration with applications	Data integration with a data lake	Data integration with a data warehouse	Data integration with data virtualization
Data Integration	*	*	***	***
Query performance	*	**	***	***
Exploitation of specialized data storage technologies	*	**	**	***
Data latency	***	**	*	***
Data security	*	*	**	***
Data privacy	*	*	***	***
Development productivity business developers	*	*	***	***
Development productivity data engineers	***	***	*	***

Table 2 *High-level comparison of approaches for integrating cloud data silos.*

Each of the approaches has its merits and can be the appropriate choice depending on the use case. But overall, of the four approaches, data virtualization scores best. With data virtualization, it is relatively easy to overcome the cloud data silos and transform them into one integrated and flexible environment while preserving the advantages of cloud platforms and leveraging all the new data storage technologies.

With data virtualization, it is relatively easy to overcome the cloud data silo.

About the Author Rick F. van der Lans

Rick van der Lans is a highly-respected independent analyst, consultant, author, and internationally acclaimed lecturer specializing in 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[3].

He has presented countless seminars, webinars, and keynotes at industry-leading conferences. For many years, he has served as the chairman of the annual *European Enterprise Data and Business Intelligence Conference* in London and the annual *Data Warehousing and Business Intelligence Summit*.

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 a number of articles[4] all published at B-eye-Network.com.

He is the author of several books on computing, including his new *Data Virtualization: Selected Writings*[5] and *Data Virtualization for Business Intelligence Systems*[6]. Some of these books are available in different languages. Books such as the popular *Introduction to SQL* is available in English, Dutch, Italian, Chinese, and German and is 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[7], 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 TIBCO Software Inc.

TIBCO Software Inc. unlocks the potential of real-time data for making faster, smarter decisions. Their Connected Intelligence platform seamlessly connects any application or data source; intelligently unifies data for greater access, trust, and control; and confidently predicts outcomes in real time and at scale. Learn how solutions to their customers' most critical business challenges are made possible by TIBCO at www.tibco.com.

---

[3] Onalytica.com, *Business Intelligence – Top Influencers, Brands and Publications*, June 2018; see http://www.onalytica.com/blog/posts/business-intelligence-top-influencers-brands-publications/

[4] See http://www.b-eye-network.com/channels/5087/view/12495

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

[6] R.F. van der Lans, *Data Virtualization for Business Intelligence Systems*, Morgan Kaufmann Publishers, 2012.

[7] R.F. van der Lans, *Introduction to SQL; Mastering the Relational Database Language*, fourth edition, Addison-Wesley, 2007.