R20
CONSULTANCY

Data Virtualization for Business Intelligence Agility

A Whitepaper

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

February 9, 2012

Sponsored by

COMPOSITE
SOFTWARE

Some of the material used in this whitepaper is based on the upcoming book *Data Virtualization for Business Intelligence Architectures* by Rick F. van der Lans; to be published by Morgan Kaufmann Publishers.

Table of Contents

1 Summary

Organizations are changing. The decision process is changing. As a result, business intelligence has to change. It has to adapt to the new reporting and analytical needs of business users. All these needs can be summarized with one word: agility*. Business intelligence solutions have to become more agile. This whitepaper describes how* data virtualization technology *makes business intelligence systems more agile.*

In 1964, Bob Dylan wrote the song 'The Times They Are A Changin'. He was (and is) probably not a business intelligence expert, although, with Dylan you never know. Still, this text is very much true for the current world of business intelligence. In the beginning, business users were satisfied with simple tabular reports showing what had happened. This was followed by the wish to present data more graphically. Next, users wanted to have more dynamic capabilities, they wanted to 'play' with the report data, they wanted to be able to do so-called drill-downs and roll-ups. And since then, the new demands kept coming. Their wishlist keeps changing. They demand new forms of reporting and analytics.

The biggest challenge facing the business intelligence industry today is how to develop business intelligence systems that have an agility level that matches the speed with which the business evolves. If the industry fails in this, current business intelligence systems will slowly become obsolete and will weaken the organization's decision-making strength. Now that the economic recession is not going to pass soon and now that businesses have to operate more competitively, the need to increase the *agility* of the business intelligence systems should be number one on every organization's list of business intelligence requirements. Agility is becoming a crucial property of each and every business intelligence system.

Agility is becoming a crucial property of each and every business intelligence system.

This whitepaper discusses how data virtualization technology can help to make business intelligence systems more agile. This technology simplifies the development process of reports through aspects such as unified data access; data store independency; centralized data integration, transformation, and cleansing; consistent reporting results, data language translation, minimal data store interference, simplified data structures, and efficient distributed data access.

Data virtualization supports the ability to decouple reports from data structures, the ability to integrate data in an on-demand fashion, and the ability to manage meta data specifications centrally without having to replicate them. This makes data virtualization the ideal technology for developing agile business intelligence systems. This is the primary reason for the increased agility.

Data virtualization is the ideal technology for developing an agile business intelligence system.

Typical business intelligence application areas that experience an increased level of agility are virtual data marts, self-service reporting and analytics, operational reporting and analytics, interactive prototyping, virtual sandboxing, collaborative development, and disposable reports.

2 Business Intelligence in Support of Decision-Making

The success of most organizations is highly dependent on the quality of their decision-making. Undoubtedly, aspects such as the quality and price of the products and services delivered by an organization have a major impact on whether they are successful as well. For example, releasing a faulty car model on the market could lead to bankruptcy of a car manufacturer; or services that are priced too high, could place an organization out of business. But even if the quality is perfect and the price is acceptable, it's still not a guarantee for success. The reason is that one incorrect decision can destroy a product, even a whole organization. For example, marketing a product incorrectly or distributing products inefficiently could lead to business disasters.

The field of business intelligence focuses on supporting and possibly improving the decision-making process of organizations by supplying the organization with the right data at the right time and in the right form.

The field of *business intelligence* focuses on supporting and possibly improving the decision-making process of organizations. This is done by supplying the organization with the right data at the right time and in the right form. Note that business intelligence is not the process of making decisions, but about supporting the process of making decisions.

Many definitions of business intelligence exist. In this whitepaper we use the definition introduced by Boris Evelson of Forrester Research[1]:

> *Business intelligence is a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making.*

From this definition we can derive that business intelligence is not a tool nor some design technique, but it's everything needed to transform and present the right data in such a form that it will improve the decision-making process.

In the first years of business intelligence, the focus was very strongly on supporting the decision-making of the two top management levels: strategic and tactical. Very typical of this form of business intelligence is that the data used by these management levels is aggregated and the data doesn't have to be 100% up-to-date. Data that is one day, week, or month old, could still be useful.

But like anything else, the world of business intelligence is changing. The main reason is that organizations are changing and their way of decision-making is changing: decisions have to be made faster, thus there is less time for the decision-making process. In other words, commercial and non-commercial organizations have to react faster.

[1] B. Evelson, *Topic Overview: Business Intelligence*, November 21, 2008.

If businesses have to operate faster and react faster to business events, more agility is expected from their IT departments because they're responsible for delivering the right data at the right time and in the right form. But, as James Kobielus of Forrester Research[2] so appropriately indicated: "The recession has forced companies to lay off or stop hiring IT staffers and business analysts, forcing everyone to do more with less. As a result, many IT staffs face growing backlogs of information requests from increasingly frustrated end users."

3 Are Business Users Satisfied?

If the decision-making process is changing, the million dollar questions are, first, are the decision makers satisfied with the business intelligence systems that have been developed by the IT departments? And second, are these systems able to support these faster forms of decision-making? Unfortunately the answer to the first question is not always yes. There are apparent signs indicating this dissatisfaction:

- In 2009, renowned analyst organization Gartner[3] predicted that by 2012, business units would control at least 40% of the total business intelligence budget. A few years ago this percentage was definitely not so high. IT departments would still control at least 90% of the business intelligence budget. They would determine which tools to use, which reports to run, and when those reports would be developed. Clearly, business units are shifting away from the IT department for making decisions concerning their business intelligence strategy. They are taking it in their own hands.

- The popularity of *BI in the cloud*, sometimes referred to as *BI as a Service*, is increasing. With these solutions, organizations park their data in the cloud and analyze it via internet browsers. Their primary advantage is that they let the cloud vendor handle most of the technical aspects. Most cloud vendors report that their customers are not the IT departments, but the business units. In many cases these cloud-based solutions are developed in addition to the current corporate business intelligence systems and are not regarded as replacements. Managers of those business units decide to go for BIaaS, because they know if they ask their own IT department that it will take too long. The consequence is that business intelligence solutions are developed outside the control of the IT department.

- A third trend worth mentioning is the increasing popularity of self-service business intelligence. The reason this is becoming so popular is that it allows business users to develop their own reports easily and quickly. A study performed by The Aberdeen Group in March 2011[4] showed that self-service business intelligence was high on the agenda of organizations. Of the 170 respondents 64% indicated that enabling business users to be more self-

Many organizations indicate that enabling business users to be more self-sufficient is their primary strategy for delivering agile business intelligence.

[2] E.Horwitt, *Self-Service BI Catches On*, CSO Online, January 24, 2011.
[3] Gartner, *Business Intelligence Summit 2009* in The Hague, The Nederlands, January 20-22, 2009.
[4] The Aberdeen Group, *Agile BI Benchmark Report*, March 2011.

sufficient was their primary strategy for delivering agile business intelligence. In short, the tools are there and the market is there, there is no stopping of self-service business intelligence.

- Another clear trend relates to the decision-making processes of organizations themselves: there is less and less time for making decisions. That same study by The Aberdeen Group also shows that 43% of the organizations report that making timely decisions is becoming more difficult. Managers increasingly find they have less time to make decisions after business events occur. Less time for making decisions implies there is less time available for the IT department to develop the supporting reports.

To answer the first question, it's clear that business users are not always satisfied with the speed with which their IT departments react on their new demands. More and more they don't rely on the IT department anymore to develop their business intelligence system. This automatically answers the second question as well: if the first question is answered with no, then the answer to the second question is definitely no as well. To get out of this impasse, our business intelligence systems have to become more agile. Business intelligence systems should be able to 'follow' the speed of the business.

Business intelligence systems should be able to 'follow' the speed of business.

4 Why are Business Intelligence Systems not Agile?

A recent study by TDWI[5] showed plainly that many business intelligence systems are not agile. Here are some figures from that study.

- The average time needed to add a new data source to an existing business intelligence system was 8.4 weeks in 2009, 7.4 weeks in 2010, and 7.8 weeks in 2011.
- 33% of the organizations needed more than 3 months to add a new data source.
- Developing a complex report or dashboard with about 20 dimensions, 12 measures, and 6 user access rules, took on average 6.3 weeks in 2009, 6.6 weeks in 2010, and 7 weeks in 2011. This shows it's not improving over the years.
- 30% of the respondents indicated they needed at least 3 months or more for such a development exercise.

It's not simple to pinpoint why most of the current business intelligence systems are not that agile. It's not one aspect that makes them static. But undoubtedly one of the dominant reasons is the database-centric solution that forms the heart of so many business intelligence systems.

The architectures of most business intelligence systems are based on a *chain of data stores*; see Figure 1. Examples of such data stores are production databases, a data staging area, a data warehouse, data marts, and some personal data stores (PDS). The latter can be small files or spreadsheets used by one or two business users. In some systems an operational data store (ODS) is included as well. These data stores are chained by transformation logic that copies

[5] TDWI, *2011 TDWI BI Benchmark Report; Organizational and Performance Metrics for Business Intelligence Teams*, 2011.

data from one data store to another. ETL and replication are commonly the technologies used for copying. Systems with this or a comparable database-centric architecture are called *classic business intelligence systems* in this whitepaper.

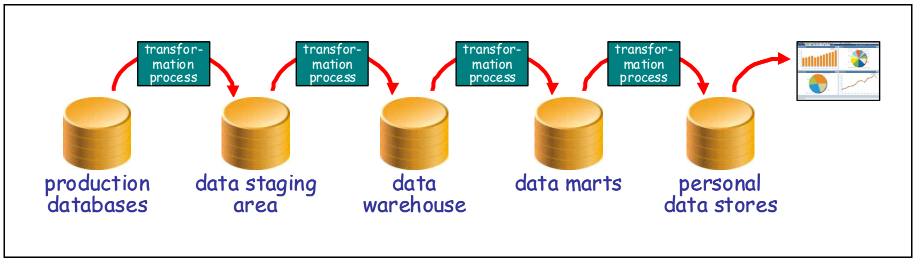

Figure 1 *Many business intelligence systems consist of a chain of data stores linked with ETL, ELT, and replication jobs*

The reason why so many business intelligence systems have been designed and developed in this way, has to do with the state of software and hardware of the last twenty years. These technologies had their limitations with respect to performance and scalability, and therefore, on one hand, the reporting and analytical workload had to be distributed over multiple data stores, and on the other hand, transformation and cleansing processing had to be broken down into multiple steps.

These classic systems have served us well in the last twenty years. However, considering the need for more agility, they have some disadvantages:

Disadvantage 1 – Duplication of data: The first disadvantage relates to the huge amount of duplicate data stored in these systems. Most of the data stores contain data derived from one or more other data stores. For example, 100% of the contents of a data mart might be derived from that of a data warehouse. The same applies to the PDSs which might be packed with duplicate data. Even a data staging area and a data warehouse will contain masses of overlapping data. Additionally, even inside a data store a lot of duplicate data can be found. This duplicate data is hidden in indexes, materialized query tables, columns and tables with aggregated data, staging areas, and so on. Most of this is there to improve the performance of queries, reports, and ETL scripts.

The amount of duplicate data stored in business intelligence systems is phenomenal.

We know that data warehouses take up huge amounts of storage, in fact, terabytes, and sometimes even petabytes of storage. But how much original data is there really? An extensive study done by UK-based analyst Nigel Pendse[6], shows that a business intelligence application needs approximately 5 gigabytes of original data. This is the median value. This number sounds realistic, but how does it match the results of many other studies indicating that the average data warehouse is 10 (or more) terabytes large? If this would all be original data, according to the study of Pendse, 2,000 different business intelligence applications would be needed with no overlapping data elements to get to 10 terabytes, which is highly unlikely. These numbers prove that the amount of duplicate stored data is phenomenal.

[6] N. Pendse, *Keynote at Trends in Business Intelligence*, 2006, Array Publications.

Obviously, there is a reason for storing all this duplicate data: it's performance. To speed up queries, we need those indexes, materialized query tables, columns with aggregated data, derived data stores, and so on.

Storage is not that expensive anymore, so what's the issue? The issue is agility. The more duplicate data is stored, the less flexible the architecture is. Every change made, requires an extra change of the duplicate data. There will be costs involved in keeping duplicate data synchronized. Storing duplicate data might also lead to data inconsistencies. Business intelligence systems could be simplified considerably if most of the duplicate data is removed.

Storing duplicate data drastically lowers the agility of a business intelligence system.

Disadvantage 2 – Non-shared meta data specifications: The second disadvantage of the classic systems can be described with the term *non-shared meta data specifications*. Many reporting and analytical tools allow us to enter meta data specifications to make report definitions more data source independent. For example, in the Universe concept of SAP's Business Objects, we can define specific terms and relationships between tables. This is perfect for all the reports created with that tool, but what if we want to develop other reports, for example, with Excel or with the IBM/Cognos tools? It means these meta data specifications have to be replicated in those tools.

To summarize, many meta data specifications entered in tools are non-shared. Organizations tend to setup heterogeneous environments in which different tools are used for different tasks, so the need for shared specifications does exist. This example of non-shared specifications relates to reporting and analytical tools, but examples of other non-shared specifications exist within ETL tools and database servers as well. Non-shared meta data specifications decrease agility and can lead to inconsistent results.

Disadvantage 3 – Limited flexibility: An important disadvantage relates to flexibility. The world of software engineering has taught us that if we design a system, we have to separate the storage structure from the application structure. Because if we separate them cleanly, changes on the storage structure do not always require changes to the application structure, and vice versa. This is good for maintenance and flexibility. David L. Parnas[7] was one of the first to recognize the importance of this concept. He called it *information hiding*. Later on, this concept became the basis for more popular concepts such as object orientation, component based development, and service-oriented architectures.

Every software engineer regards information hiding as a very fundamental concept, but it looks as if the business intelligence specialists don't. Unfortunately, most business intelligence systems are not at all based on information hiding. Almost all the reports are tied to a particular database server technology. Take a simple example. Imagine we are using a reporting tool in which we can write our own SQL statements to access a specific database server and we are using all the bells and whistles of that product to get optimal performance. What

Business intelligence systems should be based on the concept of information hiding to improve their agility.

[7] D.L. Parnas, On the Criteria to Be Used in Decomposing Systems Into Modules, C*ommunications of the ACM* Volume 15, no. 12 (December 1972). More recently republished in *Software Fundamentals, Collected Papers by David L. Parnas*, Addison-Wesley Professional, 2001.

happens if we want to replace our database server by another, one that supports a slightly different SQL dialect? Or, what if we want to switch to an MDX-based database server, or maybe we want to access an external database that does not return a table but an XML document? In probably all those situations, we have to change our report definitions dramatically.

It's important that the concept of information hiding is adopted in business intelligence systems to improve their agility and to make it easier to implement changes and adopt new emerging technologies.

Disadvantage 4 – Decrease of Data Quality: When multiple copies of the same data exists, there is always a risk that data becomes inconsistent. In other words, storing duplicate data, which, as indicated, is done extensively in classic business intelligence systems, involves a data quality risk. David Loshin[8] formulates it as follows:

> *Each time data is copied, it is also subjected to any number of data transformations, each of which provides an opportunity for the introduction of data errors. Each subsequent copy resembles the original data source less. Copying data can only lead to entropy and inconsistency.*

In every business intelligence system one of the goals should be to minimize duplication of data to minimize data quality risks.

Disadvantage 5 – Limited support for operational reporting: Another disadvantage relates to *operational reporting and analytics*. More and more organizations show interest in supporting this new challenging form of business intelligence. With operational business intelligence the reports the decision-makers use, have to include more up-to-date data. Refreshing the source data once a day is not enough for them. Especially, decision-makers who manage the core business processes, need 100% up-to-date data. But how is this done? One doesn't have to be a technological wizard to understand that if data has to be copied four or five times from one data store to another in order to get from the production databases to the reports, doing this in just a few microseconds is close to impossible. Most business intelligence systems are not designed in such a way that these operational reports are linked to the operational data. Supporting them would involve a major re-design, because they don't have the flexibility. We have to simplify to support operational business intelligence. The bottom line is that the architecture has to be simplified by removing data stores and minimizing the number of copy steps.

Architectures of business intelligence systems have to be simplified by removing data stores and minimizing the number of copy steps.

Disadvantage 6 – Limited support for reporting on unstructured and external data: The growing interest for analytics and reporting on unstructured and external data, brings us to the last disadvantage. Most data warehouses are filled with structured data coming from production databases, but rarely ever do we find unstructured and external data in there. To allow reporting on these new data sources, many specialists propose to handle these two data sources in the same way internal production data has been handled: if it's not structured, make it structured and copy it

[8] D. Loshin, *Effecting Data Quality Improvement through Data Virtualization*, Knowledge Integrity, Inc., June 2010.

into the data warehouse so that it becomes available for analytics and reporting. In other words, the proposals are based on copying data. They want to shape these data sources so that they fit in a classic business intelligence architecture.

But why not run the analytics straight on the unstructured data sources itself, and straight on the external data sources? In a way, that would be the internet-style solution. If we search something on the internet, we don't first copy all the pages needed to one of our own databases. No, the data stays where it is. Likewise, more and more of the document management systems do allow us to analyze their databases straight on, thereby reducing the need to copy their data to a data warehouse first. Unfortunately, many business intelligence tools don't support access to those systems. With respect to external data, doing business intelligence over the internet on external data can be done today in a very sophisticated way with mashup tools.

To summarize, classic business intelligence systems have supported us well for many years, and for many organizations it will still be the right solution for a few more years to come. But it is time for a new architecture, first of all, because the business is demanding more agile solutions (the focus of this whitepaper), secondly, because it should be easy to adopt available new technologies, and thirdly, because the user requirements are evolving. Business users have raised their expectations levels with respect to business intelligence and are asking for new forms of reporting and analytics.

5 What is Data Virtualization?

In this whitepaper, we limit ourselves to a short summary of data virtualization. In a nutshell, when data virtualization is applied, an abstraction layer is provided which hides for applications most of the technical aspects of how and where data is stored; see Figure 2. Because of that abstraction layer, applications don't need to know where all the data is physically stored, how the data should be integrated, where the database servers run, what the required APIs are, which database language to use, and so on. When data virtualization technology is deployed, to every application it feels as if one large database is accessed.

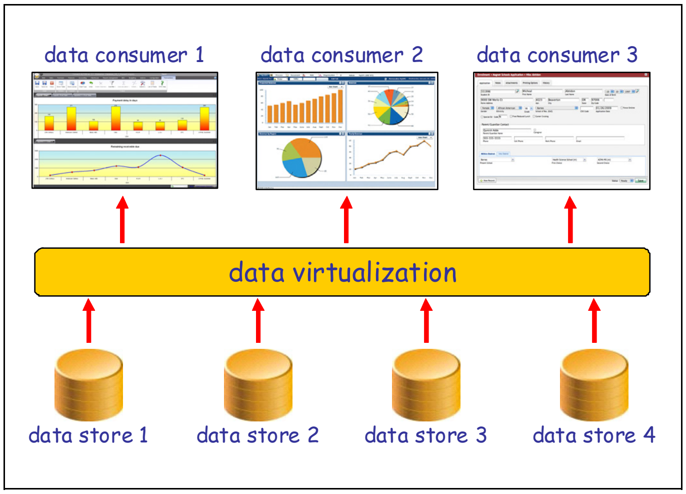

Figure 2 *When data virtualization is applied, all the data sources are presented as one integrated data source*

In Figure 2 the terms *data consumer* and *data store* are used. The neutral term data consumer refers to any application that retrieves, enters, or manipulates data. For example, a data consumer can be an online data entry application, a reporting application, a statistical model, an internet application, a batch application, or an RFID sensor. Likewise, the term data store is used to refer to any source of data. This data source can be anything, it can be a table in a SQL database, a simple text file, an XML document, a spreadsheet, a web service, a sequential file, an HTML page, and so on. In some cases, a data store is just a passive file and in others it's a data source including the software to access the data. The latter applies, for example, to database servers and web services.

The concepts of data consumer and data store are key to the definition of data virtualization:

> *Data virtualization is the technology that offers data consumers a unified, abstracted, and encapsulated view for querying and manipulating data stored in a heterogeneous set of data stores.*

So, even if the data stores in Figure 2 use different storage formats and technologies, to the data consumers, the data virtualization layer presents them all as one integrated set of data. And in addition, for the same set of data different languages can be used to access it.

6 Advantages of Data Virtualization

In most cases, accessing a data store directly is easy for a data consumer. For example, if data from a table in a SQL database has to be retrieved, the data consumer supplies a user id and a password, the name of the database, and some other technical details, and immediately it can access the required data through SQL statements. The same applies when a data consumer needs to access other data stores such as MDX databases, XML documents, or a spreadsheet.

If accessing data stores is so easy, then why is data virtualization needed? Using a data virtualization server offers numerous advantages. Below, these advantages are classified in three groups. The first group of advantages apply if only one data store is accessed:

The advantages of data virtualization:

- *Unified data access*
- *Centralized data transformation*
- *Centralized data cleansing*
- *Centralized data integration*
- *Consistent reporting results*
- *Data store independency*
- *Efficient distributed data access*
- *Simplified table structures*
- *Database language and API translation*
- *Minimal data store interference*

- **Database language and API translation:** It could be that the database language and API offered by a data store are not ideal for some data consumers, or maybe it's a language not supported by them. Maybe the database server supports SQL, while the data consumer would prefer to use a language such as MDX, XQuery, CQL, or a set of Java classes. A data virtualization server is able to translate the language supported by the data store to one convenient for the data consumer.

- **Data store independency:** Many SQL database servers support the SQL standard, meaning they have all implemented comparable SQL dialects. Still, differences exist. Data virtualization could hide those differences, making it possible to replace the current database server by another, if needed. Data virtualization makes data consumers independent of a particular data store technology. It will become easier to use the right data store technology at the right time.

- **Minimal data store interference:** A data consumer might cause interference on the data store it's accessing. Its queries might be so resource intensive that other data consumers experience some form of performance degradation. Most data virtualization products support a caching mechanism. If this is switched on, a copy of the required data of the data store is kept and managed by the data virtualization server. From then on, the data consumer accesses the data in the cache instead of the data in the data store, thus minimizing interference on the source data store.

This all means that data virtualization simplifies application development, because it reduces the amount of code required for accessing the necessary data in the right way and form; see Figure 3.

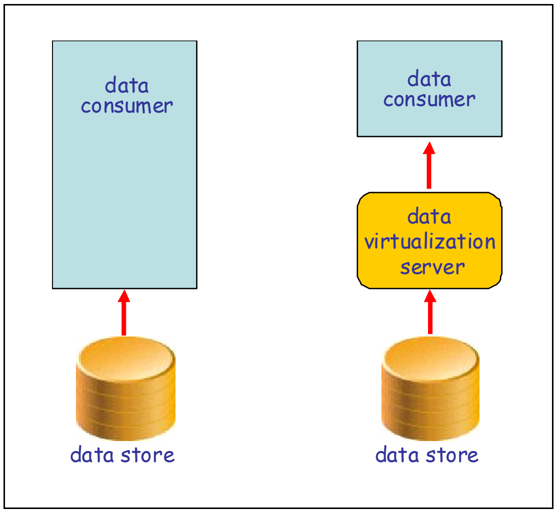

Figure 3 *Data virtualization reduces the amount of code required for accessing data; the size of the blue boxes represent relatively how much code has to be included for retrieving and processing data*

The second set of advantages relates to the fact that if data virtualization is used, meta data specifications are implemented within the data virtualization server only once, and no need exists to replicate them to multiple data consumers. In other words, data consumers share these specifications:

- **Simplified table structures:** The table structures implemented in a data store might be complex to access for the data consumers. This leads to complex queries for retrieving data and that complicates application development. Data virtualization could present a simpler and more appropriate table structure, simplifying application development and maintenance. Every data consumer can benefit from those simplified table structures.

- **Centralized data transformation:** Particular data values in a data store might have formats that aren't suitable for some data consumers. Imagine that most data consumers want to process telephone number values as pure digits and not in the form in which the area code is separated from the subscriber number by a dash. A data virtualization server could implement this transformation and all the data consumers will use it.

- **Centralized data cleansing:** Some data values in the data store might not be correct. Imagine that the column Gender in a table contains three different values to indicate Female. In this case, all the data consumers accessing these incorrect data values have to include code to correct them. It would be better if this is handled by a data virtualization server that only shows the correct values to the data consumers. This solution is better than replicating this data cleansing logic to all the data consumers.

If multiple data consumers use the same data virtualization server, they share the same meta data specifications; see Figure 4. This simplifies application development, will result in a more consistent application behavior and more consistent results, and it will make specifications easier to change and thus improving flexibility.

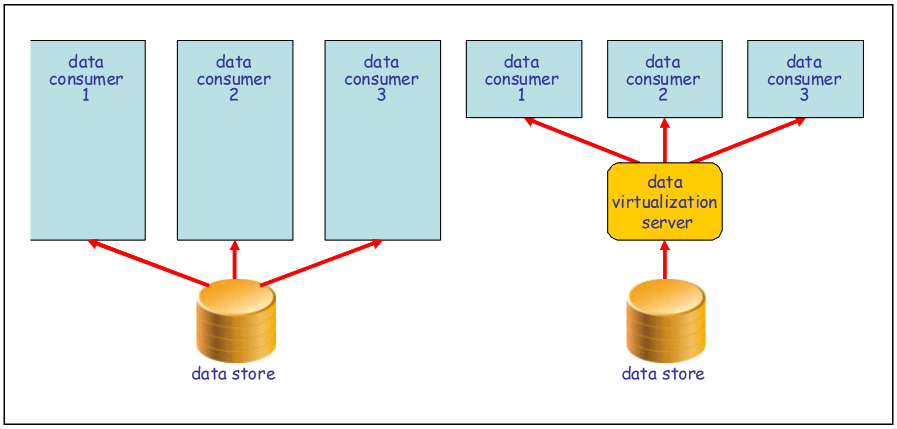

Figure 4 *Multiple data consumers share the same meta data specifications if data virtualization is applied; simplifying application development and making the system more flexible*

The third group of advantages relates to integrating data from multiple data stores:

- **Unified data access:** Different data stores might be using different storage formats. For example, some of the data might be stored in a SQL database, some in Excel spreadsheets, some in index sequential files, some in databases supporting other database languages than SQL, some in XML documents, and some of the data might even be hidden in HTML-based webpages. A data virtualization server can offer one unified API and database language to access all these different storage formats, therefore simplifying data access for the data consumers. They will only have to support one language and API.

- **Centralized data integration:** If multiple data consumers have to access multiple data stores, each and every data consumer has to include code that is responsible for integrating those data stores. The consequence is a lot of replication of data integration solutions; see Figure 5. A data virtualization server centralizes integration code and all data consumers will share that integration code.

- **Consistent reporting results:** If each data consumer includes its own integration solution, how can we guarantee that they all integrate data in the same way and according to the same rules? If this is not guaranteed, the odds are that data consumers will deliver different and inconsistent results. If all the integration solutions are handled by a data virtualization server, it's likely that results will be consistent.

- **Efficient distributed data access:** When data from multiple data stores is joined, a performance question is where and how the join is processed: is all the data first shipped to the data consumer, will the latter process the join, or should the data from one data store be shipped to another, and will that other data store process the join? Other processing strategies exist. A developer should not be concerned with such an issue. Therefore, this is a task taken over by a data virtualization server.

All these advantages prevent that data integration specifications are replicated across multiple data consumers; see Figure 5. Applying data virtualization centralizes these integration specifications as indicated in Figure 6.

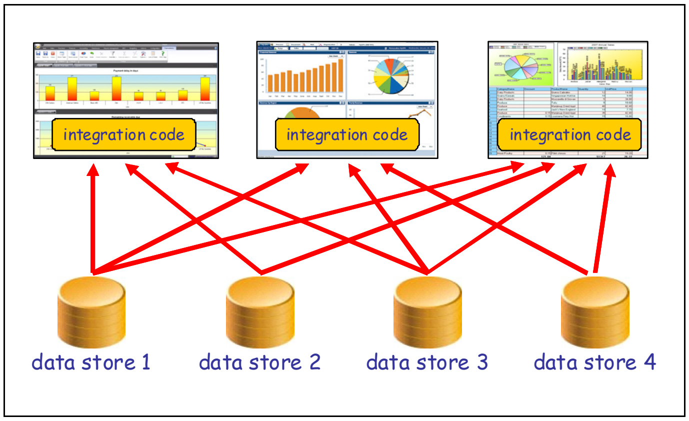

Figure 5 *Integrating data stores without a data virtualization server leads to replication of data integration logic*

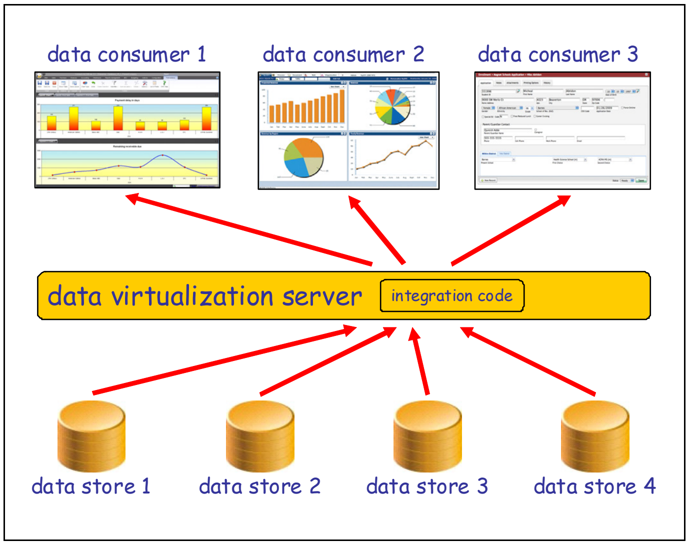

Figure 6 *Integrating data stores using a data virtualization server leads to centralization of data integration logic*

7 Data Virtualization and On-Demand Data Integration

A data virtualization product can integrate data live. So, when a data consumer queries data, only then is data from the data stores retrieved and integrated. Compare this to ETL solutions which integrate data in a more scheduled fashion. The result of an ETL integration process has to be stored before it can be used for reporting. Live data integration is called *on-demand data integration* whereas ETL delivers *scheduled data integration*.

Evidently, on-demand data integration requires more processing power, because integration is repeated everytime a query is executed whereas with ETL the integration takes place periodically. However, on-demand data integration results in a more light-weight and therefore more agile architecture; there are fewer data stores. With ETL a more heavy-weight and less agile architecture has to be developed.

On-demand data integration results in a more light-weight and therefore more agile architecture.

If, due to the on-demand style of processing, queries become slow and take up too many resources, caches can be defined. The effect is that the source data stores are accessed once and the result is stored in a cache managed by the data virtualization server. This cache is then accessed in order to answer the queries from the data consumers. Caches can be refreshed periodically. The existence of a cache is completely transparent to the data consumers.

8 Data Virtualization and Business Intelligence Systems

Deploying data virtualization in a business intelligence system means that data consumers are accessing all the data stores through a data virtualization server; see Figure 7. For the data consumers the data virtualization server hides from which data stores the data is retrieved. Data consumers have no idea from which data store the data is coming (nor do they have to know).

Figure 7 might suggest that when a data virtualization server is deployed, all the data stores normally found in business intelligence systems are still there. This doesn't have to be the case, on the contrary, the intention should be to remove as many of those data stores as possible so that a simpler architecture evolves. This figure only illustrates that all those data stores can be accessed, but it's not a requirement. For example, if all the data in a personal data store (PDS) is also stored in a data mart or data warehouse, it can be removed. Only the queries accessing the PDS have to be redirected to the other data store. The same might apply for a report retrieving data from a data mart. If all the data of a data mart is available in the data warehouse as well, the data mart might not be needed anymore. Redirecting the queries from the data mart to the data warehouse will have no impact on the reports themselves, they'll still see the same table structures. The key benefit is that if data stores are removed, the storage structure of the business intelligence system becomes simpler and thus more agile.

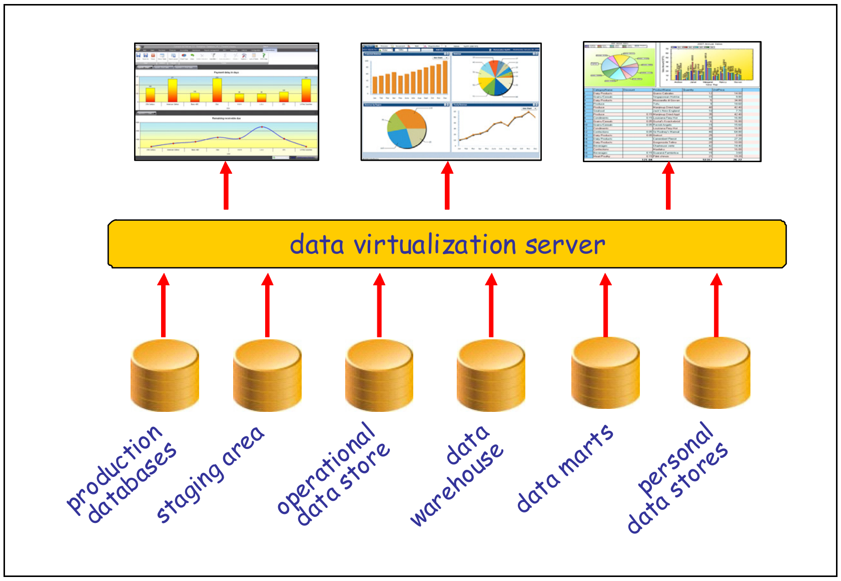

Figure 7 *Data virtualization and a business intelligence system*

The impact of introducing a data virtualization server in a business intelligence system is manyfold. First, the developers of the reports won't know which data store is accessed. They see this one large, integrated database. They develop their reports on top of the virtual tables, but they won't have to know what the underlying data stores look like.

Second, when a data virtualization server is deployed, meta data specifications related to the data, which would normally end up in the reports, will be stored centrally in the dictionary of that data virtualization server. By letting these specifications be managed by the data virtualization server, all reporting tools (even those from different vendors) accessing the data via a data virtualization server use them. So, all reports share the same specifications.

Third, in a classic business intelligence system quite some data transformation and data integration logic has to be developed to get the data in a form suitable for the reporting tools. Normally, most of that logic ends up in ETL scripts. When a data virtualization server is used, that logic is implemented in the data virtualization server. Nevertheless, ETL will not disappear completely. For example, data still has to be copied periodically from production databases to the staging area, and possibly from there to the data warehouse as well.

As long as the data is the right data, has the right quality level, and is returned with an acceptable performance, or in other words, as long as the data meets all the business users' requirements, they will be satisfied.

Something that won't change is what the users see and experience. In this new architecture, users don't see that data access is handled by a data virtualization server. They see their familiar reports and they don't see whether the accessed data store is a data mart loaded with data using ETL scripts or whether it's data accessed through a data virtualization server. The good thing is that it's not important for business users to know from which data store the data is retrieved. As long as the data is the right data, has the right quality level, and is returned with an acceptable performance, or in other words, as long as the data meets all their requirements, they will be satisfied.

9 Increasing Agility by Deploying Data Virtualization

There are various business intelligence-related application areas that benefit from deploying data virtualization. In all these areas the agility level will increase.

Virtual Data Marts – In most business intelligence systems, if data marts exist, they usually handle the majority of all the reporting and analytics. The most dominant reason for organizations to develop data marts is query performance. Without data marts all the queries have to be executed on a data warehouse or some other central data store. A second reason is that some reporting and analytical tools require the data to be organized in a particular way. If the tables in the data warehouse have been normalized, such a tool won't be able to access them. In this case, a data mart is used to present the same data in the right structure. A third popular reason is if users are geographically dispersed. In this case, accessing a central data store might result in considerable network delay when the query results are returned, slowing down the reports. In this situation, it might be necessary to move data physically closer to where the users are located, taking network traffic out of the equation.

These are all excellent reasons for developing a data mart, but there are also disadvantages, namely, these data marts have to be designed, developed, and managed. And all these activities cost time and money. In fact, in 2008 Gartner[9] already indicated that consolidating siloed data marts into a single data warehouse, reduces cost and complexity of the data integration processes feeding the data marts. They estimated that organizations can save approximately 50% of the total cost spend on data marts

With a data virtualization server the existence of a data mart can be simulated. Such a data mart is called a *virtual data mart*; see Figure 8. In a data virtualization server, virtual tables are defined with the same structure as those developed for a *physical data mart*. However, with a data virtualization server the data is not physically stored. When the virtual data mart tables are accessed, data is retrieved from, for example, a central data warehouse through on-demand transformation. Instead of actually developing a data store for a physical data mart and writing ETL scripts for periodically refreshing the data mart, virtual table structures and transformations are defined in the data virtualization server.

[9] Gartner, *Gartner Recommends Key Cost-Cutting Tactics in Data Management and Integration*, Press Release, March 10, 2008.

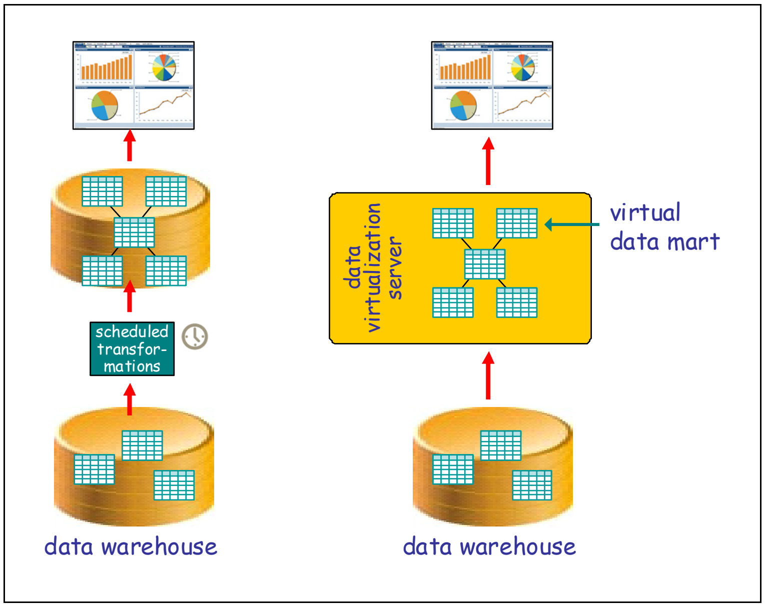

Figure 8 *Virtual data marts (right-hand side) as an alternative to physical data marts (left-hand side)*

The principal advantages of virtual data marts over physical data marts are development speed, lower costs, and, most importantly, agility.

Agility: Changing a virtual table in a virtual data mart involves altering a specification and nothing more. Making a comparable change in a physical data mart might lead to changing the data structures, redefining ETL scripts, unloading and reloading data, possibly changing the existing index structures, and re-tuning and re-optimizing the database. In other words, there is more work to do which reduces agility.

The principal advantages of virtual *data marts over* physical *data marts are development speed, lower costs, and, most importantly, agility.*

Development speed: Table 1 lists most of the activities for setting up a physical data mart versus a virtual one. It clearly shows that setting up a physical data mart requires a lot more work. Some activities have to be done, regardless of whether a physical or virtual data mart is developed, such as designing the required data structures and defining the needed transformations. But for a physical data mart environment, that's not where it stops. Database servers have to be installed and tuned, data has to be reloaded periodically, and so on. More activities means more development time.

Lower costs: Developing and changing physical data marts is more expensive and time-consuming than developing and changing virtual data marts. The main reason is, again, it involves less work. In addition, due to the extra copies, physical data marts have a higher total cost of ownership because of ongoing management and supports costs.

Note: A potential advantage of a physical data mart is that it can offer a better performance than a virtual alternative. However, the performance of a virtual data mart can be improved by enabling caching. In this case, a comparable performance can be obtained.

Activities for setting up a physical data mart	Activities for setting up a virtual data mart
Define data structures (probably with a star schema)	Design data structures
Define ETL logic to copy data from data warehouse to data mart	Define mappings
Prepare a server	—
Install a database server	—
Create a database	—
Implement the tables	Define virtual tables
Physical database design, tuning, and optimization	Enable caching (if needed)
Load the tables periodically	—
Tune and optimize the database regularly	—
Make a DBA responsible	—

Table1: *Activities required to setup a physical and virtual data mart*

Self-Service Reporting and Analytics **–** Business users don't always want to wait for the IT department to develop their reports. With self-service reporting, business users can develop their own reports. Even if they're not that tech-savvy, they should still be able to work with this new category of tools.

In the ultimate form of self-service, business users are given access to any data store possible. They can determine by themselves which tables in which data stores they want to access. The risk of this approach is whether the users are manipulating the data in a correct way. Are they combining tables on the right columns? Do they know how history has been implemented in particular tables? Do they understand the values used in some columns? Do they know that some columns might contain incorrect codes and do they know how to transform them?

A data virtualization server allows the IT department to meet the users halfway. By pre-defining the proper virtual tables with the right transformation, integration, and cleansing rules, users are given access to a consistent set of tables. If the users need more tables, or if table structures have to be changed, access can quickly be given in a controlled manner. In other words, with a data virtualization server, virtual tables can be set up and adjusted quickly, because they are virtual concepts and not physical ones. Overall, the IT department can react more quickly, and this ability to react more quickly fits perfectly with the nature of self-service BI.

In a nutshell, by deploying data virtualization, users can have their flexible self-service environment while still giving the IT department control over data access without losing agility. Data virtualization turns self-service reporting in *managed self-service reporting* where the needs of both parties are fullfilled.

Data virtualization turns self-service reporting in managed self-service reporting.

Virtual Sandboxing – The term *sandbox*, besides being a box full of sand usually meant for kids, in the world of business intelligence refers to a stand-alone and somewhat isolated environment setup for analysts and data scientists to study data and find answers to unclear or poorly defined questions. In such a sandbox, business users can execute any form of analysis or reporting without disturbing other users. They can change table structures if they think that's necessary, they can run queries that can take hours of processing, and so on. It's a playground. It's not uncommon to throw the sandbox away when the analysts have their results.

Currently, in most organizations when a sandbox is needed, a physical environment is setup, consisting of a server, a database server, a data store, analytical software, and so on. The data store is loaded with all the data that the business users think they need. Setting up such a physical sandbox costs time and money. In addition, if more data is needed than initially planned for, special ETL scripts have to be written to get and load that new data.

With a data virtualization server, a *virtual sandbox* can be set up in a fraction of the time. This solution demands less work upfront and a smaller investment than when an actual physical sandbox is created. In addition, if other types of data are needed, only virtual tables have to be defined. To summarize, a virtual sandbox is more agile than a physical sandbox, which fits the temporary nature of a sandbox better.

A virtual *sandbox demands less work upfront and a smaller investment than when an actual physical sandbox is created.*

Interactive Prototyping – Developing a new business intelligence architecture or extending it with new functionality, can be quite time consuming. Existing data stores have to be studied, table structures have to be designed, the required transformation and integration operations have to be determined, ETL scripts have to be designed and implemented, and so on. All these activities have to be completed before any result can be shown to a business user for verification. Then, if the user discovers that data has been transformed or handled incorrectly, or if the wrong definition of a business object has been implemented, many changes have to be applied. In fact, a considerable amount time might get lost purely by going back and forth between development and talking to users.

Valuable time can be saved when a *prototype* is developed first, but one that doesn't use ETL scripts, nor any extra data stores. A prototype can more easily be developed using data virtualization. Virtual tables and transformations can be defined, and the result can be shown to the users right away. If something is not to their liking, the specifications can be changed right away. Very quickly, such a prototype gives a clear insight in the problems that exist with regards to transformations, integration, and cleansing.

Data virtualization can be used as a prototyping instrument that offers a more interactive and dynamic environment in which users can participate.

Later on, the final version can be developed by using the insights that have been gained by developing the prototype. This could even mean that a business intelligence system with a more classic architecture is developed consisting of a physical data warehouse and physical data marts. In this case, data virtualization has been used as a prototyping instrument that offers a more interactive and dynamic environment in which user demands can be implemented easily.

***Operational Reporting and Analytics* –** Users who need reports based on operational data, need access to operational data, meaning they need access to the production databases, because these contain the most up-to-date state of the data.

Roughly three approaches exist to implement *operational reporting* and *analytics*. The first is the most straightforward approach: let the reports access the production databases directly. With the second approach, data is copied from the production databases to other data stores using replication. These replicated data stores are the ones accessed by the reports; see Figure 9. With this approach data is copied 1:1. This means that the structure and the contents of the tables in a replicated data store are identical to those of the original production database; the replicated data is not transformed.

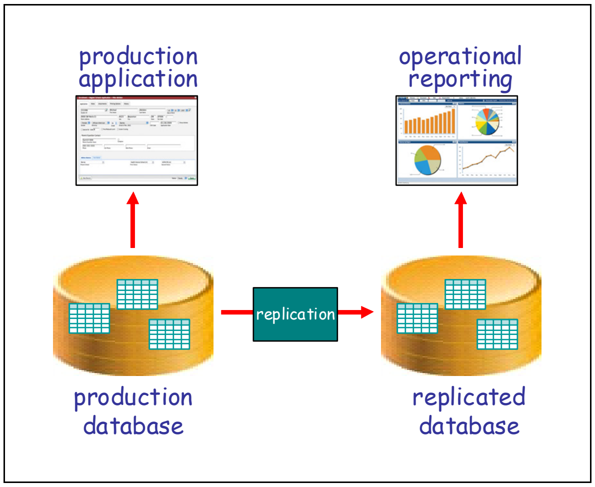

Figure 9 *Operational reporting using replication*

Although both approaches sound viable, they lead to the following complexities:

- The table structures in a production database may not be suitable for reporting. They might have been designed to support production applications, but not operational reports. These structures could lead to complex reporting code.
- The tables might contain incorrect data that needs to be corrected or recoded. Usually, this is handled by ETL scripts. With operational reporting the reporting tools have to handle this; they become responsible for on-demand cleansing and on-demand transformations. Most reporting tools have not been designed to handle such issues.
- If data from multiple production systems has to be integrated, it's the reporting tool itself that has to include all the integration logic. Again, most reporting tools are not properly equipped for this.
- The queries on production databases could cause performance problems, especiallly if they are complex, statistical queries. The query might cause so much I/O, that operational users who try to insert or update data, have to wait. Note that this problem applies to the first approach only, and not to the second, because here the queries are run on the replicated data store.
- Reporting on production databases could also cause concurrency issues. Running a report might lead to locking of data that results in operational users waiting for data to become available. Vice versa, updates by operational users might lead to queries that have to wait before they can be executed.

In fact, this list is also a short summary of the reasons why data is normally copied to a data warehouse before it is used by the reporting tools. All the issues we normally deal with during the copying process have to be handled by the reporting tools. Most of them do not support sufficient functionality to deal with these issues easily and elegantly. For some, it will even be impossible.

The third approach for implementing operational reporting is by deploying data virtualization. A data virtualization server gives the reports direct access to the production databases. It's responsible for handling the complexities listed above. In fact, these products have been designed to handle these issues. By defining the right virtual tables and transformations, they can present more suitable data structures, transform incorrect data, integrate data from different production databases, and they can optimize access in such a way that the problems of performance and concurrency are minimized or are resolved completely. An additional benefit of using data virtualization is that all the meta data specifications managed by the data virtualization server are shared by all the tools that need access to the operational data; they become re-usable.

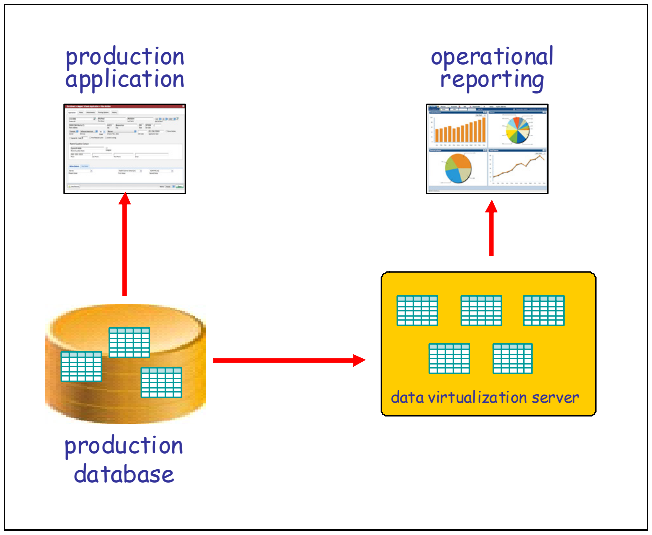

Figure 10 *Operational reporting using data virtualization*

The queries of a data virtualization server might have a negative impact on the on-going operational work. Those same queries might also lead to concurrency problems. Caching may not be the solution here, because if the reports really need to access the most up-to-date state of the data, they have to be executed on the production databases. In this case, an approach where data virtualization and replication are used together, might be preferred; see Figure 11.

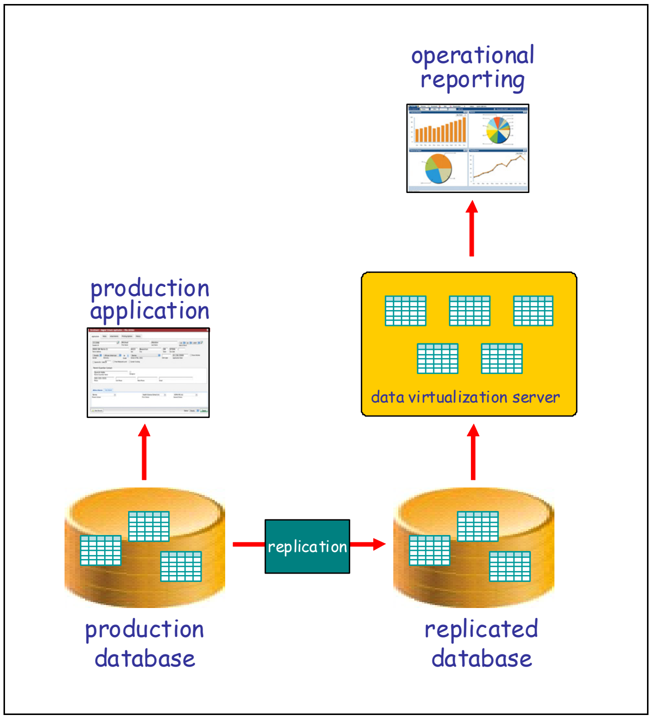

Figure 11 *Operational reporting using data virtualization and replication*

Collaborative Development **–** Various activities that are part of the overall design and development process of a business intelligence system require the expertise of IT specialists as well as business users. When data virtualization is deployed, many of these activities become a more collaborative exercise between IT specialists and business users, and in addition they become more agile. Let's take database design as an example.

Usually, logical database design is quite an abstract exercise. The designer comes up with a set of table definitions presented in some diagram. For users those are abstract concepts, especially if they don't have a computing background. For them, it's sometimes hard to see how those tables together represent their information needs. The reason is that they don't think in terms of data structures, but they think in terms of the data itself. For example, a designer thinks in terms of customers and invoices, while a user thinks in terms of customer McGuinn based in London, and invoice 6473 which was sent to customer Metheny Metals Inc.. Therefore, it can be hard for a user to determine whether the table structures resulting from logical database design really represent his needs.

With data virtualization, database design becomes a more collaborative and a more agile process.

It would be better if, when data structures are shown to users, they are shown together with samples of the real data, so that they can understand better what those tables represent. When data virtualizaion is used, a database design can be implemented as virtual tables. When a virtual table is defined, its (virtual) contents can be shown instantaneously. In other words, when a virtual table is defined, the analyst and the user together can browse the contents and the user is able to confirm that what he sees is according to his information needs. This way, database design becomes a more collaborative and a more agile process.

Disposable Reports – More and more often business intelligence departments are confronted by business users who need to run a new report that only has to be run once or twice to address a specific, one-time issue. In most cases, these reports have be developed urgently. We call these *disposable reports*.

Given the urgency, there is no time for designing a complete physical environment consisting of a data mart, ETL scripts, and so on. Plus, if the time for all that work would be available, it would make it too expensive.

More fitting is to use data virtualization. Here, a few virtual tables can be developed quickly and easily to support the report. After the report has been created and executed, the virtual tables can be removed. This requires a minimal investment, especially compared to a more physical solution.

Performance problems are not an issue here. Even if executing this report takes five hours, it wouldn't make sense to optimize it afterwards, because probably the report won't be executed again. Optimizing and tuning of queries only makes sense for queries that are executed repeatedly.

Summary – Many application areas of business intelligence exist where data virtualization definitely increases the agility level.

10 Developing True Agile Business Intelligence Systems

A business intelligence system can be seen as if it consists of two major parts, where the first part consists of all the reporting and analytics and part 2 of all the data stores and the transformation logic needed to provide data. In Figure 12 these two parts are presented as two layers.

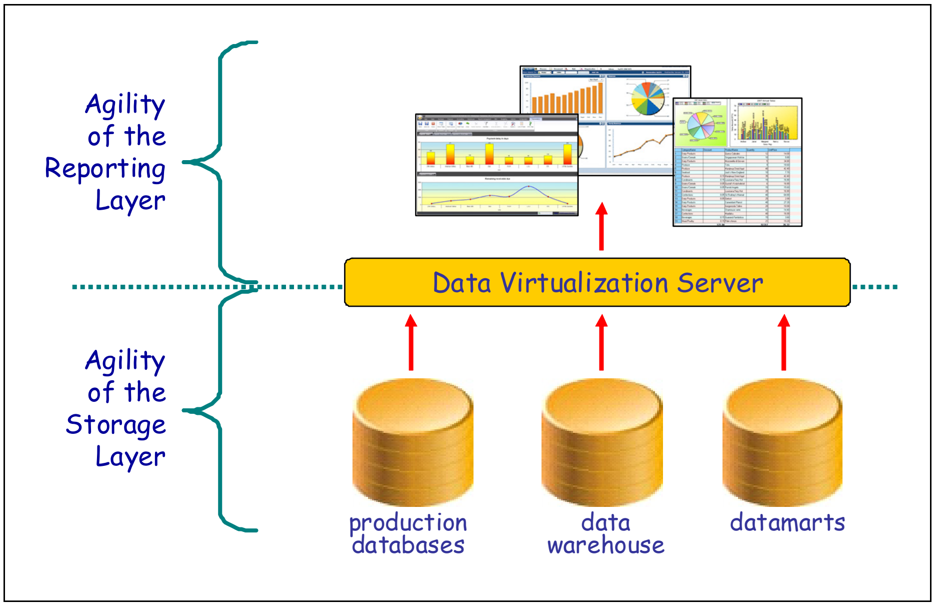
Agility of the Reporting Layer
Agility of the Storage Layer
Data Virtualization Server
production databases
data warehouse
datamarts

Figure 12 *A business intelligence system can be seen as consisting of two layers: the storage and the reporting layers*

Popular self-service business intelligence tools, such as QlikView, Spotfire, and Tableau, unquestionably increase the agility of a business intelligence system. However, they only increase the agility of a part of such a system, namely, the *reporting layer*. Introducing self-service tools leaves the *storage layer* untouched and its level of agility remains the same. Compare this to wrapping a colossal block of concrete in a thick layer of rubber. When finished, the block of concrete is still as flexible as it was before. It's still concrete.

The same applies to business intelligence systems. True agility is reached by making all parts of a business intelligence system agile. If only the agility of the reporting layer is increased, its agility is still restricted by the storage layer. This is where data virtualization comes to the rescue. Data virtualization raises the agility level of the storage layer.

To summarize, what self-service reporting has done to the reporting layer (making it more agile), that's what data virtualization will do to the data storage layer of a business intelligence system. Together they will help to develop a true agile business intelligence system. And besides making it more agile, data virtualization upgrades self-service business intelligence to *managed self-service business intelligence*, limiting the risks of this new, somewhat unmanaged style of reporting.

> *What self-service reporting has done to the reporting layer (making it more agile), that's what data virtualization will do to the data storage layer of a business intelligence system.*

Besides implementing a business intelligence system with agile technology, it's also recommended to adopt more agile design techniques. Therefore, it's recommended to adopt an *agile design technique* such as Scrum and eXtreme Programming. However, only applying an agile design technique is not enough. Here the same applies as what applies to self-service tools. The effect of applying agile design techniques in a business intelligence system developed with non-agile technology, is minimal. Both the design technique and the tools have to be agile.

11 Two Case Studies

This section contains two brief case studies of organizations using the data virtualization server of Composite Software called Composite Information Server. For more detailed information on both case studies, we refer to the book on data virtualization by Judith Davis and Robert Eve[10].

Qualcomm – Qualcomm Inc. is a global leader in next-generation mobile technologies. They manufacture chipsets, licenses technology, and provide communications services worldwide. They're based in San Diego, California. The company has more than 17,000 employees and had an annual revenue in 2011 of close to $15 billion.

Their business problem was that in their fast-moving market, they are constantly being challenged to get things done faster to maintain their leadership position in the mobile

[10] J.R. Davis and R.J. Eve, *Data Virtualization; Going Beyond Traditional Data Integration to Achieve Business Agility*, Nine Five One Press, 2011.

technology market. They needed more agility. So they implemented data virtualization to provide the level of IT agility their business requires.

In [10] Mark Morgan of QualComm indicates "Without data virtualization, a project that involved moving data around, including building the process to do that [...], might take months. In contrast, because data virtualization leaves data in place, it would enable us to prototype and get feedback to customers much faster and get applications up and running in a matter of weeks." In other words, their development times shortened with a factor of 4 and thus improved their agility, the speed at which they can react to a changing business.

Their primary use case for data virtualization is the ability to make data available to multiple applications without having to copy and move the data. For example, bringing in a new application that needs access to five different systems, might take six months. With Composite Data Virtualization virtual tables presenting the same data could be developed in days, again, improving development speed and agility.

With their agile data virtualization foundation in place, Qualcomm has successfully grown their deployment to enterprise wide scale. In just two years, Qualcomm implemented over 24 projects, 19 reusable data services, and 17 entity models using data virtualization.

Northern Trust – Chicago-based Northern Trust is a leading provider of innovative fiduciary, investment management, private banking, wealth management and worldwide trust and custody services. Clients include corporations, institutions, families, and individuals in more than 40 countries. The company has over 13,000 employees. At the end of 2010, the company had assets under custody of $4.1 trillion, assets under management of $643.6 billion, and banking assets of $83.8 billion. Annual revenues in 2010 were almost $3.7 billion.

An important line of business for Northern Trust is providing outsourced investment management operations for corporate customers. These institutional customers provide front-office investment management functions for their end clients, but outsource all or some of the mid-office and back-office functions to Northern Trust. By contracting these functions out to Northern Trust, the institutions don't have to invest in the resources, assets and skills necessary to provide the functions internally. In return, Northern Trust provides guaranteed levels of quality, service, resilience, and value-to-cost criteria and management.

The business problem for Northern Trust was that it had a large number of new customers in the outsourcing pipeline and simply could not implement them fast enough. The existing client reporting capabilities, based on a classic infrastructure, were inefficient and more importantly, inflexible. They desperately needed the ability to bring on new customers without having to make technology changes for each one.

They considered two alternative architectures. One proposed solution was to physically integrate the data using a classic business intelligence system based on ETL and data warehousing technology. Their estimate was that this would be a massive project that could easily take up to a year and a half.

Their second alternative was based on data virtualization. Data would be integrated virtually using the Composite Data Virtualization Platform. By placing this data virtualization server in the middle between the client reporting tool and the back-end data sources, Northern Trust would be able to quickly abstract the data into the data virtualization layer, reuse the existing business logic to access the data, and generate all the required data for client reporting. This approach required just seven months of development.

With time-to-solution the critical success factor, Northern Trust selected the data virtualization alternative. Data virtualization has dramatically reduced the time it takes Northern Trust to implement a new outsourcing customer by 50%, from 6-9 months to 3-6 months. Moving customers through the pipeline faster, improves overall customer satisfaction and gives them the capacity to bring in even more business and revenue.

About the Author Rick F. van der Lans

Rick F. van der Lans is an independent analyst, consultant, author, and lecturer specializing in data warehousing, business intelligence, service oriented architectures, and database technology. He works for R20/Consultancy (www.r20.nl), a consultancy company he founded in 1987.

The last three years he has focused on applying data virtualization in business intelligence system resulting in his new book entitled *Data Virtualization for Business Intelligence Architectures* which will be released in the summer of 2012.

Rick is chairman of the annual European Data Warehouse and Business Intelligence Conference (organized in London), chairman of the BI event[11] in The Netherlands, and he writes for the B-eye-Network[12]. He introduced the business intelligence architecture called the *Data Delivery Platform* in 2009 in a number of articles[13] all published at BeyeNetwork.com.

He has written several books on SQL. His popular *Introduction to SQL*[14] was the first English book on the market in 1987 devoted entirely to SQL. After more than twenty years, this book is still being sold, and has been translated in several languages, including Chinese, German, and Italian.

For more information please visit www.r20.nl, or email to rick@r20.nl. You can also get in touch with him via LinkedIn (http://www.linkedin.com/pub/rick-van-der-lans/9/207/223) and via Twitter (http://twitter.com/Rick_vanderlans).

About Composite Software

Composite Software, Inc. is the data virtualization gold standard at ten of the top 20 banks, six of the top ten pharmaceutical companies, four of the top five energy firms, major media and technology organizations; and multiple government agencies. These are among the hundreds of global organizations with disparate, complex information environments that count on the Composite to increase their data agility, cut costs and reduce risk. Backed by nearly a decade of pioneering R&D, Composite is the data virtualization performance leader, scaling from project to enterprise for data federation, data warehouse extension, enterprise data sharing, real-time and cloud computing data integration. Composite Software is a privately held, Silicon Valley-based corporation. For more information, please visit www.compositesw.com.

[11] See http://www.bi-event.nl/59857
[12] See http://www.b-eye-network.com/channels/5087/articles/
[13] See http://www.b-eye-network.com/channels/5087/view/12495
[14] See http://www.amazon.com/Introduction-SQL-Mastering-Relational-Database/dp/0321305965/ref=sr_1_1?ie=UTF8&s=books&qid=1268730173&sr=8-1