The New Generation of Self-Service BI

Avoiding Typical Self-Service BI Pitfalls With an Integrated BI Platform

A Whitepaper

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

August 2014

Sponsored by

Information Builders

Table of Contents

1 Management Summary 1

2 The Battle for Data is On 2

3 Self-Service BI Tools 3

4 Requirements for BI Tools 4

5 BI Systems and Centralized Metadata Specifications 6

6 Self-Service BI and Metadata Specifications 8

7 Big Data and Self-Service BI 10

8 Mobile BI and Customer-Facing Analytics 12

9 One Integrated BI Platform 12

10 Information Builders’ WebFOCUS BI and Analytics Platform 15

About the Author Rick F. van der Lans 17

About Information Builders 17

1 Management Summary

Self-Service BI – The *battle for data* is on. Having the right data at the right time in the right form is going to be vital for organizations to survive and compete. But it's not just about data; it's also about having the right tools. It's not that having the best tools guarantees victory in the battle for data, but bad technology definitely assures defeat.

The battle for data is on.

Self-service BI (Business Intelligence) is one of the new emerging tool categories to assist organizations to exploit their data freely. Through their user-friendly, intuitive, and graphical interfaces, they allow users to develop and change their reports and dashboards themselves without assistance from the IT department or BICC (Business Intelligence Competence Center). They are designed for a do-it-yourself approach.

Analytical Silos and Integration Labyrinths – Deploying self-service BI tools is not without problems. For example, a study shows that 64% of organizations struggle with self-service BI. It can lead to *analytical silos* and an *integration labyrinth* where important *metadata specifications*, such as those for data integration, data transformation, data structure, and data cleansing, are not reused, but are reinvented over and over again – resulting in decreased user productivity.

The use of self-service BI can lead to analytical silos and an integration labyrinth.

In addition, although self-service BI tools are very valuable, they *complement* and don't replace all other existing forms of BI. It's one form of BI from a palette of BI forms that an organization needs. It's important from productivity and maintenance standpoints that different types of BI tools, including the self-service ones, share solutions. Only then will the entire BI system be able to support four key requirements: *reporting consistency*, *reporting correctness*, *cross-platform development*, and *high productivity*.

One Integrated BI Platform – If organizations want BI systems that support the four key requirements across the entire BI tool palette, they need one *integrated BI platform* that makes it easy and common practice to reuse metadata specifications. Such an integrated BI platform must offer reuse of metadata specifications, reuse of reporting components, accessibility for many BI tools, universal data access, agility, and centralized security.

Organizations need one integrated BI platform.

The New Generation of Self-Service BI – The new generation of self-service BI tools must not be stand-alone tools. They must be an integral part of and be able to exploit an integrated BI platform. Users want to develop reports and analyze data freely. They're not interested in dealing with the complex technical aspects of how to unravel data from data sources, in developing their own integration solution, or in managing piles of scripts. One integrated BI platform is required to support self-service BI, improve user productivity, and help organizations to compete in the battle for data.

The new generation of self-service BI tools must be an integral part of an integrated BI platform.

Information Builders' WebFOCUS BI and Analytics Platform – In addition to describing the requirements for such a BI platform, this whitepaper also describes *Information Builders' WebFOCUS BI and Analytics Platform*, and how it meets the requirements for an integrated BI platform.

2 The Battle for Data is On

The *battle for data* is on. For organizations to survive and compete, it's no longer sufficient to produce low-cost, sell the best products, streamline business processes, or organize transport and delivery professionally. To survive today, organizations must manage their data impeccably and exploit its business value maximally.

Over the last two decades, many data warehouses and BI systems have been developed to help exploit all the available data. These systems allow organizations to run reports that can, for example, show an up-to-date overview of monthly sales revenues and indicate customer purchases over the last months. All this information is very valuable, but it's not enough anymore to survive, let alone to compete.

Besides knowing what customers have purchased, an organization has to know what they like, what they are likely to buy, to which other customers they're related, what their chance of churning is, what their comments are on the products and the organization on social media platforms, what their socio-demographic characteristics are, and so on. Having the right data at the right time in the right form is going to be vital for business goals, such as raising customer care levels, predicting sales, and personalizing products.

For organizations, it's going to be a battle for data. The organization that owns or has access to the most and best data; the one that integrates data sources perfectly; the one that analyzes the data optimally and uses it for, for example, predictive analytics and data mining; and the one that knows how to use all that data creatively, will have a big competitive advantage and may be successful in this battle. Organizations will compete with the data they have access to and the way in which they exploit it.

> *Organizations will compete with the data they have access to and the way in which they exploit it.*

Data is not valuable if it's just lying around in big databases. It becomes valuable only when it's used. Users across an entire organization, from the strategic to the operational level, need to be able to easily and quickly access the right data with the tool that fits their reporting needs. For some that means using predefined reports that can be invoked on their desktop by simply pushing a button. For others it involves a highly dynamic reporting environment that allows them to study data from every angle and on each level of detail, and for others it's a simple report they can invoke on their mobile phone in the form of an app that can be tweaked a little.

Data is becoming more and more crucial for organizations, but, as indicated, only storing data is not enough. The analytical and reporting technology that is being deployed will, for a large part, determine success. Therefore, the battle for data is also a battle of deploying the right tools. Therefore, this battle has to be fought with the latest tools and technologies.

3 Self-Service BI Tools

Popularity of Self-Service BI Tools – One of the emerging BI tool categories is *self-service BI*. With their user-friendly, intuitive, and graphical interfaces, they allow users themselves to develop and change their reports and dashboards without assistance from the IT department or BICC (Business Intelligence Competence Center); hence the name. They are designed for a do-it-yourself approach. The popularity of self-service BI is understandable from the context of the battle for data, because they allow users to freely exploit and analyze available data.

Self-service BI was in the beginning limited to primarily smart and easy data visualization, charting, and reporting; nowadays, self-service tools exist that support complex analytics and some even offer data integration features. This allows users to determine how data from different sources should be merged together.

64% of organizations struggle with self-service BI.

Problems of Self-Service BI Tools – Deploying self-service BI tools is not without problems. Initially, all these tools look easy to use, but a study by Wayne Eckerson[1] shows that self-service BI tools require more training than expected. This was indicated in the study by 73% of the respondents. In the same study, 61% of the respondents indicated that use of self-service BI tools leads to what's called *report chaos*. To quote the report: 64% of the organizations struggle with self-service BI, giving their self-service BI initiatives a grade of "average" or lower, with 29% rating self-service BI "fair" or "poor."

Source of the Problems – A diverse set of reasons exists for the problems with self-service BI tools:

- **No Trained Developers:** Users are not trained developers. They don't automatically try to develop reusable solutions, nor do they formally test their reports. If they need to develop a report, they just do it, and when it seems to works fine, they're finished.

- **Databases can be Complex:** Having free access to databases does come with a price, namely the users have to understand what all the columns mean, how tables are related, what all the values and codes in the columns mean, and so on. They have to fully understand the structure and the contents of the data sources they access. If not, their reports will show incorrect results.

- **Limited Reuse:** *Reuse* of solutions is not high on the agenda of users of self-service BI tools. For example, when one user has developed his own solution to integrate data from an SAP production database with data from Salesforce.com (which is quite an effort), it's not automatically reused by colleagues. In general, users don't share solutions. Even if they do have that intention, most self-service BI tools can't exploit the solutions developed with other BI tools running in the same BI environment. Each tool is a stand-alone, technological island making reuse of solutions impossible. Every user appears to start from scratch. The consequence is that the wheel is reinvented over and over again. Besides that, this is a waste of resources, and it negatively influences the time-to-market of new reports and dashboards.

- **Bypassing BI Systems:** If users are accessing data marts, they are correct to assume that all the data has been organized in neat star schema structures and that the data is cleansed. But more and more users don't want to be limited to accessing IT-controlled data marts; they want to access all the data that's available. Self-service BI tools make that possible. They allow users to extract data from every source, including external databases, spreadsheets, websites containing social media data, and even production databases. When that happens, users have to know how to integrate these data sources. Suddenly, report development becomes much more complex,

[1] W. Eckerson, *The Promise of Self-Service BI*, April 2013, see http://insideanalysis.com/2013/04/the-promise-of-self-service-bi

resulting in a decrease of user productivity. When databases other than data marts are accessed, the users themselves become responsible for data transformation, data integration, and data cleansing. This style of development can be quite complex, even with the most user-friendly tools.

Summary – Self-service BI tools allow organizations to query data freely without the assistance of IT specialists. This free style of reporting matches the organizational needs to exploit the value of data that has been collected over the years. The challenge is to preserve the user-friendliness and do-it-yourself characteristics of the tools and to make it easy to reuse existing solutions as much as possible. More on this in Section 6.

4 Requirements for BI Tools

BI is More Than Self-Service BI – The overwhelming popularity of self-service BI diverts the attention from other forms of BI that are equally important and valuable for an organization, such as classic executive reporting, mobile BI, operational intelligence, statistics, data mining and analytics, mobile BI, and customer-facing analytics. For example, financial organizations generate monthly reports for regulatory obligations, actuaries need high-end statistical tools to calculate the financial impact of risk and uncertainty to provide assessments of financial security systems, traveling sales people require simple query capabilities on their mobile devices to study the most recent purchases of the client they're visiting, and embedded BI is still needed to include churn chances inside a call center application.

In short, although self-service BI tools are very valuable, they don't replace all these other forms of BI but *complement* them. It's one form of BI from a *palette of BI forms* that an organization needs. These other forms of BI tools can be as valuable for an organization. But this category has made us all aware of how valuable it is to work with agile and user-oriented tools.

Self-service BI doesn't replace all other BI forms but complements them.

Commonalities Between BI Tools – Regardless of the fact that these BI tools differ in how they present and process data, their ease of use, and the reporting and analytical functionality they offer, they still have much in common:

- They all have to extract data from data sources they don't control.
- They all have to integrate data from different data sources.
- They all have to transform and standardize the data structures and data values.
- They all have to cleanse the data coming from the data sources.

For example, a user wants to combine data from an SAP ERP (enterprise resource planning) system with data from a home-made system that stores data in an Oracle database. If he uses a self-service BI tool, besides spending time on the presentation form, he has to develop a solution to standardize values, transform data structures, deal with different APIs, cleanse data, and integrate data. Imagine also a colleague who is using a more traditional reporting tool in which all his push-the-button, select-from-the-menu reports are predefined by IT specialists. He also wants to integrate data from those same two data sources. For his reports that same data has to processed with the same extract, integrate, transform, and data-cleansing logic. This is regardless of the fact that he uses and views the data differently.

Requirements for BI Tools – So, it's important from a productivity and maintenance standpoint that different types of BI tools, including the self-service ones, share solutions. The four key requirements are: *reporting consistency*, *reporting correctness*, *cross-platform development*, and *high productivity*.

- **Reporting Consistency:** It's crucial for an organization that BI environments offer *reporting consistency*. Any form of reporting and analytics must be based on the same set of data and must return consistent results. For example, when users request total sales revenues per month, regardless of the BI tool they use, they should get the same result. One tool may display the results on a handheld device with a simple bar chart, whereas another presents it on a large desktop screen using a pie chart. The presentation form is not important. What is important is that the results are consistent, especially if users switch regularly between BI tools. In fact, this need for reporting consistency was one of the reasons why many years ago data warehouses were initially introduced. Before data warehouses were developed, each report would do its own extraction, integration, transformation, and cleansing of data. This premise still holds: a BI system must offer reporting consistency.

- **Reporting Correctness:** The quality of the data presented in reports depends, for a large part, on the quality of data in the data sources. When data is integrated, transformed, or cleansed incorrectly, its quality decreases. If a complete new solution is developed for every user and for every report, it's close to impossible to guarantee reporting correctness. The more correct the data in the reports is, the more users will trust the reports. Reporting correctness is an important requirement.

- **Cross-Platform Report Development:** Reporting solutions (or components) developed with one tool must be reusable in another. If users or IT specialists develop a report with a particular content and layout, the users must able to request that report from any device and tool. The old Adagio once introduced by Sun for Java is very relevant here as well: *Write once, run everywhere*. Develop the report once and make it available everywhere.

- **High productivity:** Nowadays, user needs can change so quickly that BI tools must support rapid development. Higher productivity improves the time-to-market for any form of reporting and analytics. Of course there are standard reports that users invoke repeatedly and for which development may take some time. But more and more requests for reports are based on a business urgency or operational request. In this case, time is of the essence. So, requirements are much more dynamic than before, demanding that BI tools improve productivity. In addition, what applies to productivity also applies to maintenance: it must be possible to make changes to reports quickly.

Reusability of Specifications as Solution – To offer reporting consistency and correctness, to realize cross-platform development, and to increase productivity, it's important that solutions can be reused easily across all the BI tools, including the self-service BI tools. For example, if some metadata specifications have been developed to integrate two data sources and to cleanse the data, they should be reusable everywhere. No one likes to reinvent the wheel, especially not if it has already been invented fifteen times. Therefore, it must be easy to reuse predefined solutions. To reuse and share metadata specifications improves reporting consistency and correctness, makes cross-platform development possible, and increases productivity.

Solutions must be reusable across all the BI tools, including the self-service BI tools.

5 BI Systems and Centralized Metadata Specifications

Crucial for attaining reusability is to implement and maintain *metadata specifications* in some centralized form. This section discusses these metadata specifications.

> *A BI system can be seen as a BI assembly line.*

The BI Assembly Line – Each BI system consists of many databases and many ETL programs responsible for copying data between these databases; see Figure 1. In a way, most of them resemble an assembly line in which data from the production databases are like raw materials and all the ETL (extract, transform, and load) programs and databases slowly turn them into finished goods that are presented to users. So, a BI system can be seen as a *BI assembly line*.

Crucial in these systems are the metadata specifications. They are like the glue that links everything together. All the metadata specifications together define how raw data is turned into finished data ready for business user consumption.

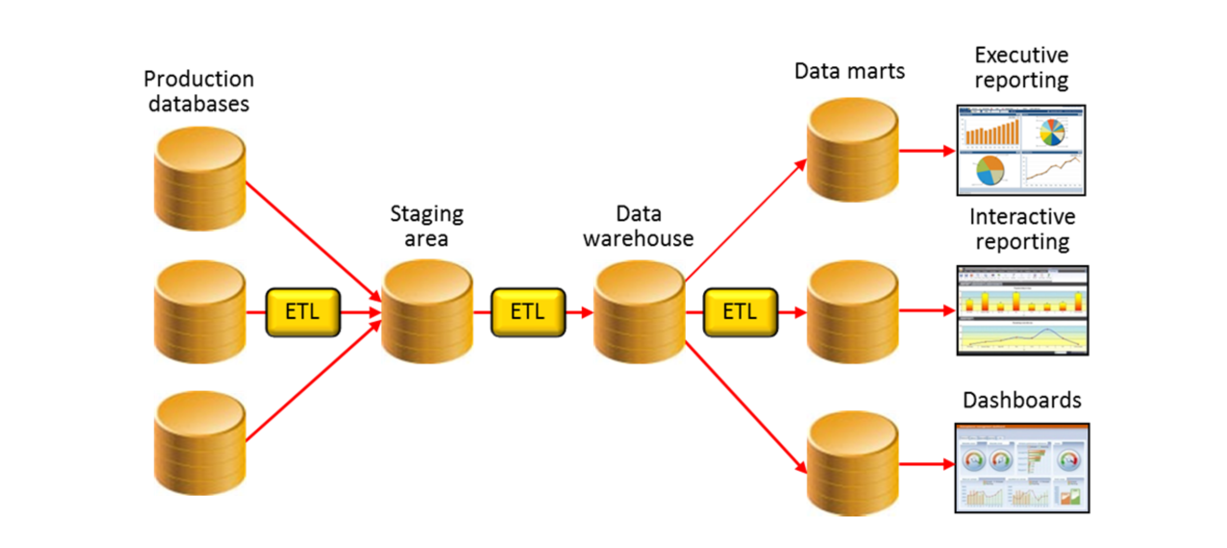

Figure 1 *Classic BI systems resemble assembly lines in which production data form the raw materials and the information in the reports the finished goods.*

Data Integration Specifications – The second group of metadata specifications deals with integration of data sources. Data integration is primarily done when data is copied from the staging area to the data warehouse, and some integration takes place when data is copied from the staging area to a data warehouse. Data integration specifications are usually implemented with ETL programs.

Integration of data sources may sound easy, but in many situations it’s not. For example, older production databases can have data structures that are difficult to map to more normalized structures. Examples of complex structures are repeating groups and conditional columns, where the meaning of a value in a column depends on the meaning of another column.

Data integration should never be trivialized. When a mistake is made with the interpretation of what data in a source system means and how it must be integrated, the outcome is faulty reporting results.

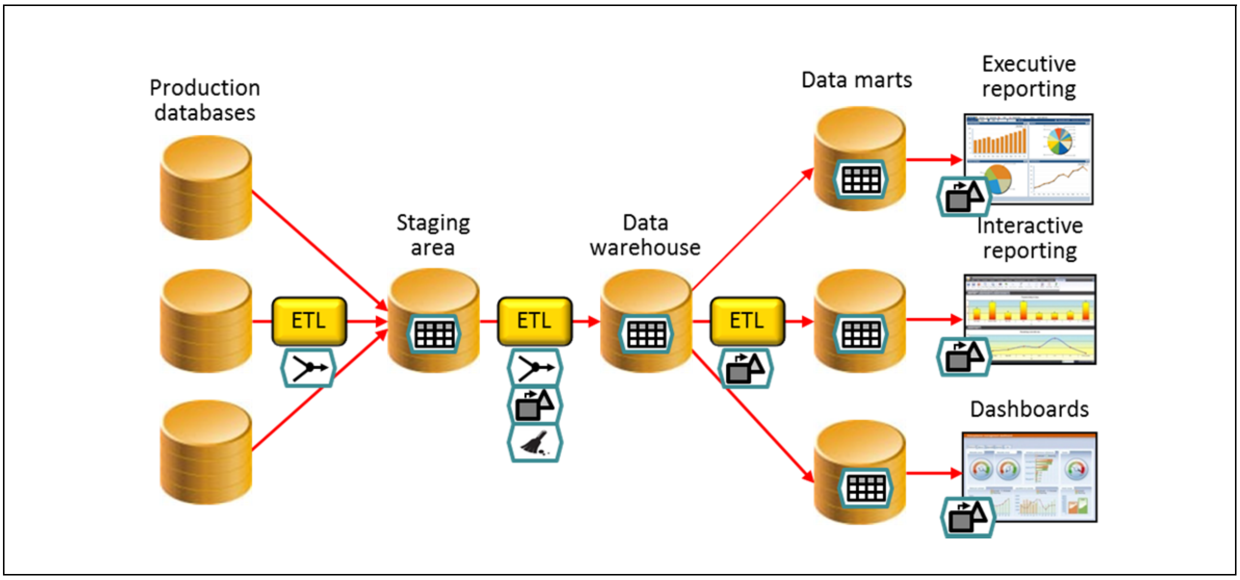

Figure 2 *Metadata specifications can be found throughout a BI system.*

Data Transformation Specifications – Data transformation specifications can be classified in two main groups. The first group transforms data structures and the second data values. Data structure transformation has a close relationship with integration; it refers to taking data from one system and mapping that into another structure. For example, when data is copied from a data warehouse to a data mart, it may have to be transformed from a normalized structure to a star schema structure. In this process, the data values themselves are not changed; it's just that the values are rearranged in another set of tables. Aggregation of data is another example of a data structure transformation.

Transformation of data values has a close relationship with data cleansing as well, except that incorrect data values are not corrected, but correct values are turned into more standardized values. For example, imagine a column that incorrectly contains the value `Male` in the Gender column. This value is not incorrect, it does represent the right gender, but it's not the standard value for it, that's `M`. Data value transformation specifications indicate how to do the transformation. Or, with data transformation specifications a column containing a street name and a street number are split into two separate columns, because that's the way it should be stored. Again, the combined value is not correct; it's just not the right form in which it should be stored.

Transforming data values to more standardized values improves the quality of reports and also user productivity. To use the example of the combined address value again, not splitting them would make it hard to combine it with data from other systems where addresses are organized differently. The user of a self-service tool has to develop the logic itself, which can lead to incorrect reporting code and may slow down development.

Data Cleansing Specifications – Data cleansing specifications indicate how incorrect data can be detected and how it can be turned into correct data to raise the data quality level. If data cleansing takes place within the boundaries of a BI system, it's commonly done when the data is copied from the staging area to the data warehouse. Examples of data cleansing techniques are de-duplication, correction of misspelled names, and completeness checking.

Implementing Metadata Specifications – Together, all the metadata specifications represent an enormous value to the business. A BI assembly line consists of many tools, databases, and technologies. All of them are important, but without the metadata specifications they are useless, because these specifications instruct the tools what to do.

Metadata specifications can be implemented in many different ways. For example, they can be implemented inside ETL tools, as database specifications, with database-stored procedures, or in the semantic layer of the BI tool itself. Whatever the implementation is, it would be a nightmare if the metadata specifications are replicated and end up all over the system. It's important that each specification is defined and implemented only once and reused repeatedly. There should be no *proliferation of metadata specifications*.

There should be no proliferation of metadata specifications.

If two users are studying the same data with different reporting tools, it's crucial that this data is processed with the same specifications. If they view data processed by different implementations of the same specifications, it's hard to guarantee reporting consistency. How do we enforce that both implementations are correct and how do we keep them synchronized when the specifications have to be changed? The key requirement for implementing all the metadata specifications is simple: define once, implement once, and reuse over and over again.

6 Self-Service BI and Metadata Specifications

This section addresses the impact of using self-service BI tools on the use of metadata specifications.

Limited Reuse of Metadata Specifications – If an organization is capable of running all self-service BI tools on IT-controlled data marts and when only a limited set of metadata specifications is implemented in the BI tools, then all the results presented by these tools will (probably) be correct and consistent. But in practice that's not always how the tools are deployed. As addressed in Section 3, there is limited reuse of metadata specifications among users when self-service BI tools are deployed. Reports can easily become *analytical silos,* where each report implements its own metadata specifications for data transformation and data cleansing; see Figure 3. Users of self-service BI tools have a tendency to get careless with these valuable specifications.

An Integration Labyrinth – Most BI tools support data integration features. They allow users to freely integrate data from multiple data sources; see Figure 4. In this case, each report implements its own integration, transformation, data structure, and cleansing specifications. In this situation, guaranteeing reporting consistency and correctness, cross-platform development, and high productivity is extremely difficult. The resulting architecture can be called an *integration labyrinth*.

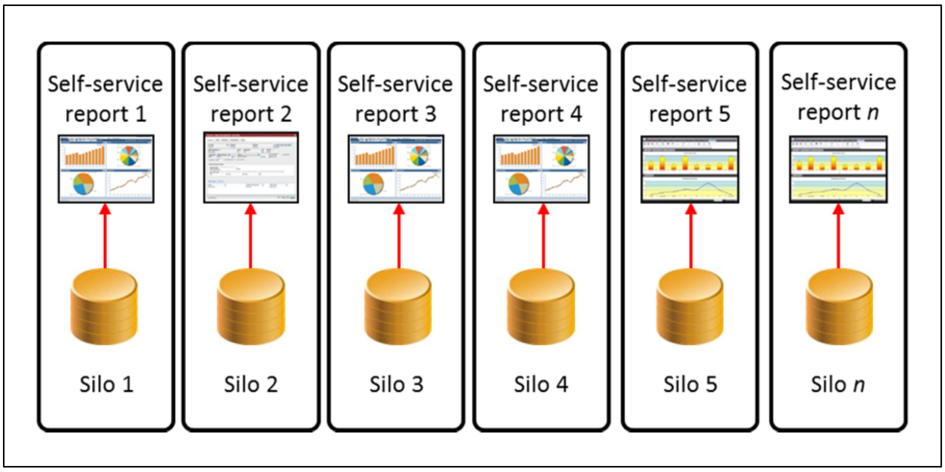

Figure 3 *Self-service BI reports can become analytical silos.*

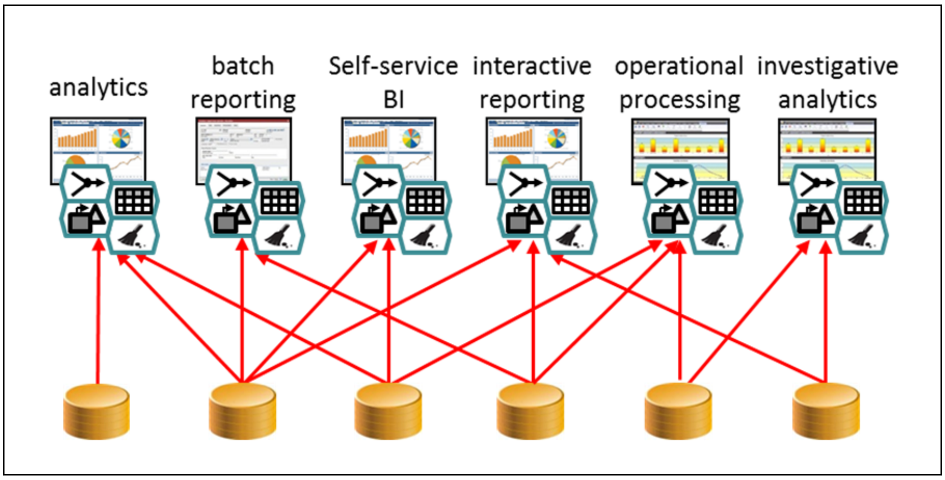

Figure 4 *The integration labyrinth in which each report implements its own metadata specifications.*

Bypassing Metadata Specifications – Users who do not restrict themselves by accessing IT-controlled data marts will access other data sources in the BI systems, including the data warehouse, the staging area, and even the production databases; see Figure 5. At the bottom right-hand side of the diagram, a self-service BI tool is depicted that accesses all databases in a BI system. The consequence is that it's bypassing all the metadata specifications implemented by the IT department. The challenges are how to guarantee that the necessary metadata specifications are implemented in the self-service BI tool and how to guarantee that they are implemented consistently with the existing specifications. In addition, how do we keep them in synch when specifications change?

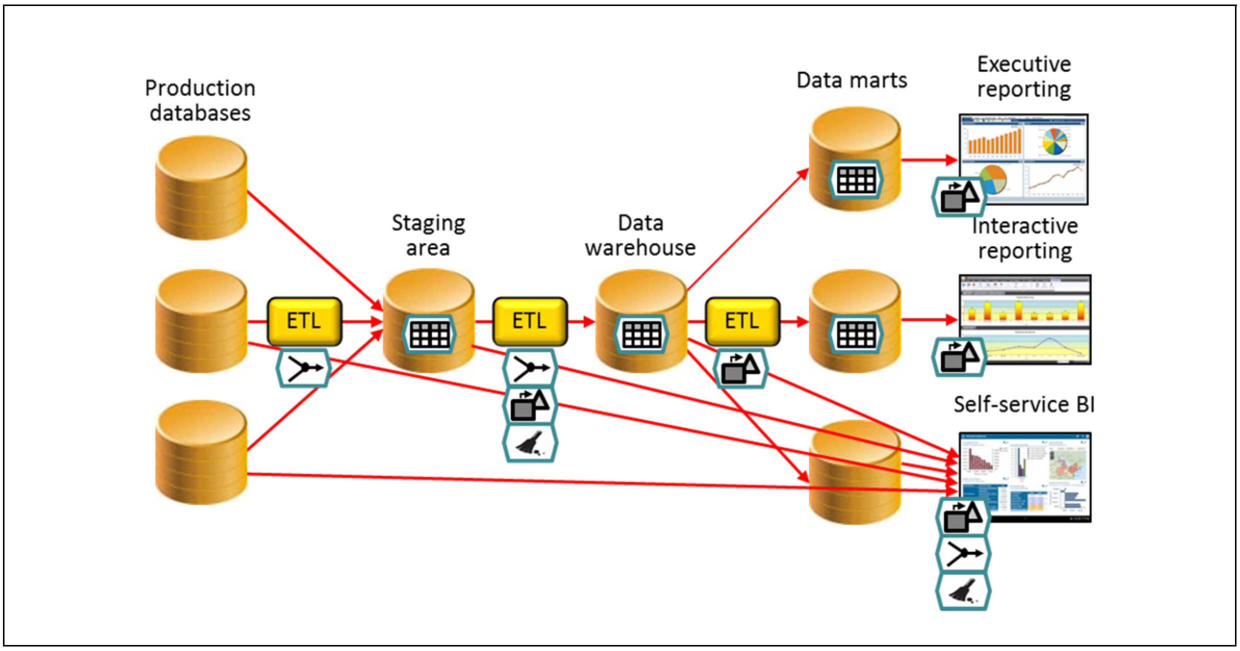

Figure 5 *Metadata specifications are bypassed with self-service BI tools.*

7 Big Data and Self-Service BI

Big data will revolutionize BI systems.

Big Data – Self-service BI has definitely changed the face of BI systems. Another trend that has, is, and will continue to revolutionize our BI systems is *big data*. It's considered the biggest trend in the IT industry. Big data applications store amounts of data magnitudes larger than those in more traditional applications. For example, click-stream applications, sensor-based applications, text-analysis, and image processing applications all generate massive numbers of records per day. The amount of records stored surpasses more often than not hundreds of millions of records.

Data Storage Technology for Big Data – The sheer amount of big data has a direct impact on the database technology used. Therefore, organizations have considered other types of data storage technology that are different from familiar and classic SQL database servers. This convinced vendors and startups to research and develop new database technology, resulting in the market of *Hadoop* and *NoSQL* technologies.

No SQL for Big Data – These technologies allow applications to be developed for the simplest queries to the most complex forms of analytics. Unfortunately, their APIs and languages are quite complex (hence the name NoSQL that is used frequently for these products). Most of them require expertise in Java programming (or a similar language) and in-depth knowledge on how to parallelize query processing efficiently. Another consequence of these special APIs is that many tools for reporting and analytics, including the self-service BI tools, can't access these data stores because they demand SQL and ODBC/JDBC interfaces.

Note: Lately, several SQL-on-Hadoop engines have been released that allow access to Hadoop with SQL. But most of these engines are still young and have to proof themselves in real life systems.

More Analytical Silos – The effect of collecting big data in these new data stores is that dedicated BI tools have to be acquired that are specifically designed for these data storage technologies. Each data store is accessed by a dedicated analytical tool, and vice versa. The result is the next group of analytical silos. In addition, the amount of copying from one big data store to another just to make new data entered in one system available in the others can be horrendous. In other words, it leads to more proliferation of metadata specifications; see Figure 6.

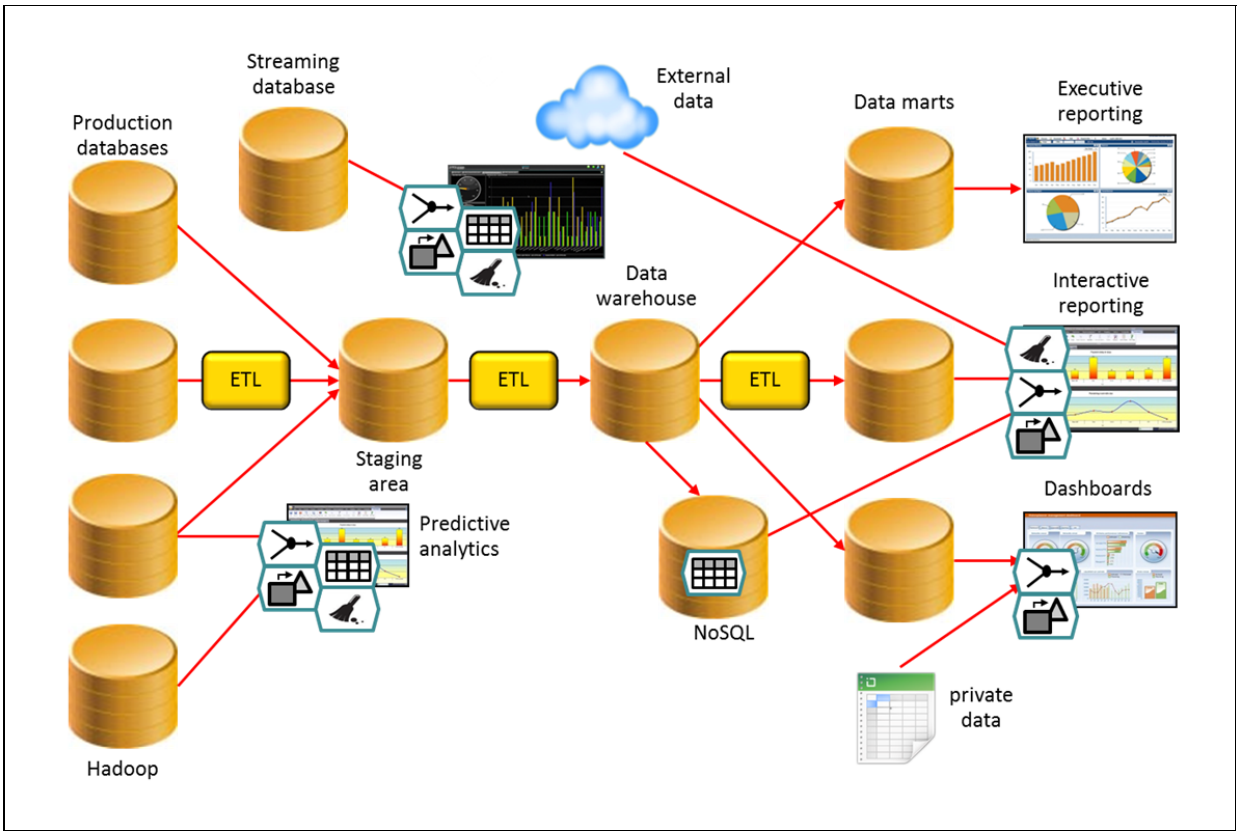

Figure 6 *Bypassing the metadata specifications.*

But most importantly, again we will see no reuse of metadata specifications. Each solution has its own set of integration, transformation, data structure, and cleansing specifications. For some it will be written in non-standard languages such as MapReduce for Hadoop and MongoDB.

These technical and non-open interfaces make it difficult for business users to access big data. It also forces them to use other reporting tools than they prefer and are familiar with. For business users, self-service BI tools should make accessing big data as easy as accessing small data. What applies for classic data applies for big data as well: specifications must be reusable; the wheel should not be reinvented over and over again. For users the storage technologies must be completely transparent.

8 Mobile BI and Customer-Facing Analytics

Two more trends that are impacting BI in general and self-service BI in particular are *mobile BI* and *customer-facing analytics*. Because they're interrelated, both are discussed in the section.

Mobile BI – Another topic that may lead to even more analytical silos and more proliferation of metadata specifications is *mobile BI*. The business value of mobile BI is not being questioned. Numerous users can benefit from having reporting and analytical capabilities available all the time and anywhere. Today, users want to be able to do reporting and analytics on any kind of device, ranging from the smallest smartphone or a tablet to the most powerful desktop machine.

Does the tool for developing mobile BI apps reuse existing metadata specifications?

The link with self-service BI is obvious. Users want to have some lightweight form of report development on their mobile devices. And when, for example, they have developed a bar chart that presents an overview of factory spill levels on their laptop, they must be able to run that on their smartphone as well.

But again the question is, does the tool for developing mobile BI apps reuse the existing metadata specifications? Many companies use a dedicated tool for their mobile BI apps, which is not integrated with the rest of the architecture. In such a case, it's hard to guarantee reporting consistency and correctness across the mobile BI reports and the other forms of BI.

Customer-Facing Analytics – In most organizations, users of self-service BI tools are internal users. More and more a new group of users is accessing an organization's data, namely the customer. This is called *customer-facing analytics*. In the beginning, customer-facing analytics was limited to simple reports that customers could invoke. Nowadays, these users want to be able to freely analyze the data as well, they want self-service reporting and analytical capabilities. For example, customers may want an app for smartphones developed by their bank that gives them insight into their mortgage finances. Another example is a Web-based application that supports a simple form of what-if analysis that allows customers to predict what the effect would be on their monthly mortgage payment if the interest rate increases by 2.5%, or what happens when their salary increases by 15%?

Dedicated tools exist to develop customer-facing analytics. The risk is again that metadata specifications are implemented anew with these dedicated tools, and that there is no reuse of the specifications already implemented in the BI system. Again, more BI silos originate.

9 One Integrated BI Platform

An Integrated BI Platform – There is no question about the value of self-service BI tools. By allowing users to develop their own reports, the productivity improves and reports become available more quickly. But it's not enough. Report development with self-service BI tools should not lead to analytical silos and an integration labyrinth. When organizations want reporting consistency, reporting correctness, cross-platform development, and high productivity, plus, when they want that across the entire BI tool palette – one integrated BI platform is required. This platform must make reuse of metadata specifications easy and common practice. In fact, it must be close to impossible to bypass the metadata specifications. Figure 7 shows a high-level representation of such an integrated BI platform.

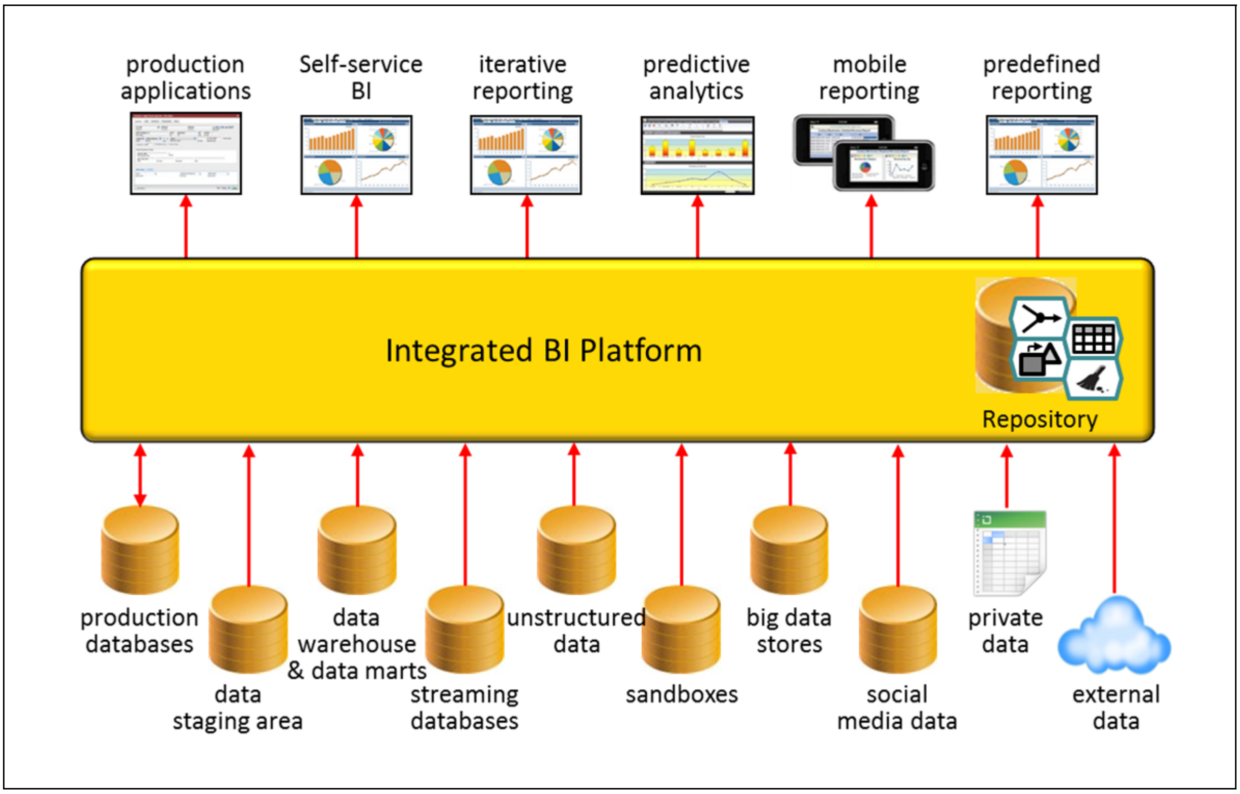

Figure 7 *One open and integrated platform for all forms of reporting and analytics.*

Requirements for an Integrated BI Platform – An integrated BI platform can be developed in different ways. Regardless of the implementation style, the platform must meet the following requirements:

- **Reusable Metadata Specifications:** All four types of metadata specifications must be supported. These specifications must be shareable and reusable. The need to reinvent the wheel should be nil, because the business has no time to reinvent the wheel. For example, when a solution has been developed to integrate two data sources, that solution must be reusable for any user and any report. So, if a metadata specification is defined, it can be used by other reports, such as batch reports, mobile reporting apps, iterative reporting tools, and by self-service BI tools. Only then can reporting consistency be guaranteed. Plus, if a specification changes, it has to be changed in only one place.

- **Reusable Reporting Components:** Users spend a lot of development time on specific reporting components, such as a dashboard or chart. The platform must make it possible to reuse that reporting in a portal, in a standard report, on a mobile device, and so on. This is the cross-platform requirement: develop a report once, and run it on any device.

- **Open for any BI Tool:** The platform must support any form of reporting and analytics, including self-service BI. It must also be open for tools from other vendors, so that users can always use their preferred reporting tool.

- **Universal Data Access:** The platform must hide where and how the data is physically stored. All the data sources should be presented as one, big logical database. This applies also to big data store technology such as Hadoop and MongoDB. It must be just as easy to access big data as small data.

- **Agility:** The platform must be agile. Centralized specifications should be as easy to change as the specifications of a self-service BI tool, because it improves the time to market. If not, users will walk away from such a solution, as they have done from classic solutions.

- **Centralized Security:** By having a centralized solution to data access, implementing data security rules is easier than when every tool accesses the data sources straight on. The platform operates as a gateway to all the data.

Dividing the Work More Equally – Before self-service BI tools existed, IT was doing almost all the development work, and the users were only consuming the data and reports. Today, because of the availability and popularity of the current generation of self-service BI tools, development work has shifted away from the IT department and towards the users. Users have taken development of reports and dashboards in their own hands. They develop the front-end of the BI system and the IT department is responsible for the back-end of the BI system, consisting of the staging area, the data warehouse, and the ETL programs. This shift is presented in Figure 8, where the length of a bar indicates the amount of development work.

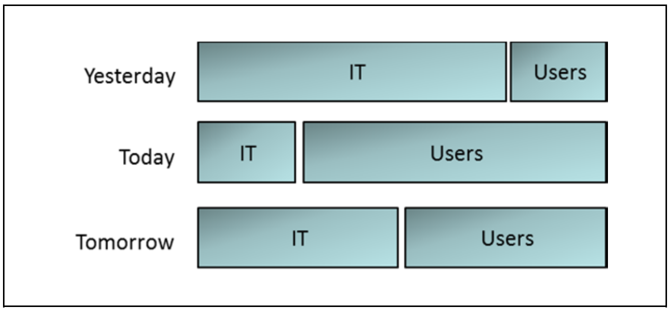

Figure 8 *The shift of responsibilities and work.*

A better distribution of resources is needed. With an open and integrated platform, work on BI systems becomes much more collaborative, where each party is responsible for half of the development work. The IT department becomes responsible for developing reusable components.

The new generation of self-service BI tools must be able to exploit an integrated BI platform.

The New Generation of Self Service BI Tools – The new generation of self-service BI tools must be able to exploit an integrated BI platform. Users are not interested in dealing with complex technical aspects of how to unravel data from data sources, they're not interested in developing their own integration solution, nor are they interested in managing piles of scripts. They just want to be free with respect to developing reports and analyzing data. The new generation of self-service BI tools must support an architecture where the work is divided more equally between IT and users. The architecture must be a more complete system in which self-service BI is just one integrated component, one form of data delivery, and is not seen as the only one.

Summary – A practical advantage of an integrated platform for self-service BI is that user productivity improves. Users don't have to worry about the idiosyncrasies of the data source structures, how to integrate data sources, or how the data must be standardized and cleansed. For all users and all BI tools the same data is available. The user only focuses on what he wants to do with the data that is readily available.

An analogy can be made with cars. If we want a new car, we have two options. First, we can buy a do-it-yourself kit for a hotrod. This kit offers the buyer full freedom. He can build the hotrod the way he wants. But before he can proudly drive his car out of the garage, he has done a lot of welding, wiring, grating, painting, and so on. The only issue is: who checked that car? How safe and trustworthy is it? The second option is to buy a car. The buyer still has total flexibility, he can buy a small and fuel-efficient car, or a large four-wheel pick-up truck. These cars have been checked and tested, and if there is a problem you bring it back to the garage.

Nowadays, self-service BI is too much like tinkering with a hot rod. With an integrated BI platform, it becomes like buying a car and two minutes after paying for it, you're on the road.

10 Information Builders' WebFOCUS BI and Analytics Platform

The BI Modules – Information Builders delivers an extensive and integrated platform for BI called the *WebFOCUS BI and Analytics Platform*. It consists of a rich set of BI modules offering all the different forms of BI, ranging from simple push-the-button reports to advanced forms of predictive analytics:

- Mobile BI
- Ad-hoc query and analysis
- OLAP
- Data discovery
- Predictive analytics
- Dashboarding
- Prescriptive analytics (through a rules engine)
- Parameterized reporting and analytics
- Embedded BI
- Customer-facing BI
- Data discovery and exploratory analytics
- Financial analysis (through SML)

All the BI modules are developed by Information Builders, resulting in a fully integrated BI platform. The platform has not been assembled by acquiring other vendors or products; it's not a set of puzzle pieces that doesn't form one puzzle. WebFOCUS offers organizations *reporting consistency*, *reporting correctness*, *cross-platform development*, and *high productivity*.

Reporting Consistency – In WebFOCUS all the metadata specifications are stored in a central repository, including the security specifications. All the data is processed using the same data structure, integration, transformation, and cleansing specifications. When two users request total sales revenues per month – regardless of whether they use the ad-hoc query and analysis module, the self-service BI module, or the mobile BI module – both will see the same results. One tool may display the results on a hand-held device with a simple bar chart, whereas another presents it on a large desktop screen using a pie chart.

The metadata layer of the repository supports an open API. BI tools from other vendors can use this API to access data. This leads to reporting consistency across all the BI tools deployed by an organization.

Reporting Correctness – Information Builders is considered to be one the dominant vendors in the field of *adapters*. Many vendors resell these adapters because of their quality, performance, and stability. The adapters allow all the BI modules to access almost every data source on this planet, including classic SQL database servers; pre-relational data sources, such as CS IDMS, IBM IMS, and Adabas; new NoSQL systems such as MongoDB; mainframe-based sequential and index-sequential files; the Hadoop file system; and data available in social media networks. If an interface has been defined for one of these data sources all the BI modules can use it, thus all sharing the same specifications. This improves reporting correctness.

In addition, Information Builders was one of the pioneers of *data federation* technology in the 1980s. Through the years this technology has matured and currently offers *universal data access* to the entire set of BI modules. Again, integration specifications implemented in the data federation technology are shared by all the BI modules. For example, if specifications have been defined to integrate a Microsoft SQL Server database and a MongoDB database, they can be used by all the BI modules. All this improves reporting correctness.

Cross-Platform Development – *Active Technology Reports* is the technology used by Information Builders to present reporting results. All BI modules can display their results using this technology. For example, a bar chart developed for a laptop can be presented on a smartphone as well without having to change one line of code. Information Builders offers true cross-platform development.

Another technology that improves cross-platform development is *Hyperstage*. This is a fast, in-memory SQL database server optimized for reporting and analytics. It allows for massive amounts of data to be processed. To use Hyperstage, data is copied from the original data source to Hyperstage and then made available to all the reports. When data is loaded in Hyperstage, it can be used by every BI module, from simple ad-hoc queries to predictive analytics.

High Productivity – Because all the BI modules of Information Builders make use of the same platform and share one central repository, all the metadata specifications can be shared easily. They can be shared amongst users in a self-service environment, but also amongst users and IT specialists. This high level of reuse of specifications improves productivity and eases maintenance.

InfoApps™ – A technology described separately here is *InfoApps*. With InfoApps, simple, easy-to-use, data-driven applications can be developed that support business users in making decisions on the job. InfoApps extends the world of self-service BI. Most BI tools make it possible to develop a report or dashboard. With InfoApps users can pick data from different data sources and present that in an integrated dashboard. InfoApps allows users to develop simple applications with a do-it-yourself approach. As with all the BI modules of Information Builders, InfoApps is completely integrated with WebFOCUS.

The InfoApps module is very well suited for developing customer-facing analytical applications that run on mobile devices or on the Web and are used by customers or other business partners to analyze data and to make decisions.

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, database technology, and data virtualization. He works for R20/Consultancy (www.r20.nl), a consultancy company he founded in 1987.

Rick is chairman of the annual European Enterprise Data and Business Intelligence Conference (organized annually in London). He writes for the eminent B-eye-Network.com[2] and other websites. He introduced the business intelligence architecture called the *Data Delivery Platform* in 2009 in a number of articles[3] all published at BeyeNetwork.com. The Data Delivery Platform is an architecture based on data virtualization.

He has written several books on SQL. Published in 1987, his popular *Introduction to SQL*[4] was the first English book on the market 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. His latest book[5] *Data Virtualization for Business Intelligence Systems* was published in 2012.

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

About Information Builders

Information Builders helps organizations transform data into business value. Our software solutions for business intelligence and analytics, integration, and data integrity empower people to make smarter decisions, strengthen customer relationships, and drive growth. Our dedication to customer success is unmatched in the industry. That’s why thousands of leading organizations rely on Information Builders to be their trusted partner. Founded in 1975, Information Builders is headquartered in New York, NY, with offices around the world, and remains one of the largest independent, privately held companies in the industry. Visit us at informationbuilders.com and follow us on Twitter at @infobldrs and @infobldrsINTL.

[2] See http://www.b-eye-network.com/channels/5087/articles/

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

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

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