

Contents

1	Overview	1
2	How Can Data Virtualization Help My Business?.....	1
3	How Can I Tell My IT Environment Needs Agile Integration?	2
4	Why is the Integration Infrastructure so Inflexible?	4
5	What are the Root Causes of this?.....	5
6	What is Data Virtualization?	6
7	Enterprise Enabler: Agile Integration Software with Data Virtualization	7
8	Integration Integrity Manager	8
9	Conclusion.....	8
10	Glossary.....	9
11	Glossary Continued	10

Your integration software, called “[middleware](#),” is supposed to make data flow easily throughout the organization, but too often, finding data is more like a scavenger hunt. Symptoms of brittle legacy integration include:

Not everyone is working off the same numbers. In your weekly cross-department planning meeting, everyone brings their reports for production, operations, planning, and billing, and you spend most of the time debating who has the right numbers. It’s hard to run a company when you can’t get everyone on the same page.

Long wait times to get access to new data you need. You want to start seeing the financials of the new acquisition in the context of the big picture, but have to deal with a totally different perspective from the acquisition’s existing reporting structures. How can it possibly take six months to get the new data into the [data warehouse](#) or reporting dashboard?

Manpower costs for building integration are exorbitant. You either have to send your own people to two three-month training sessions to learn how to use the big name middleware you have in place, or you have to hire unbelievably expensive consultants to do it. And of course, the consultants are always happier the longer it takes to accomplish anything! Why can’t you just use a reasonably good programmer? Because it’s built on a secret ancient scripting language, and your IT people have to conjure up twenty-year-old environments to run it on.

More than 40% of new projects and software implementation costs are for integration and data access. Typically, for every dollar spent on licensing and implementing a new application, the integration costs at least two dollars.

Data you get is not up-to-the-minute. [Data latency](#) is due to a number of things, all of which are mitigated by Data Virtualization (DV) with agile data integration. If your data comes from the data warehouse, chances are that the data is at least a day old, and sometimes considerably more, because it is captured on a schedule from the original source and massaged to combine with other data for the data warehouse. Data virtualization removes this problem.

You have business processes that are highly manpower intensive. Almost any manpower-intensive process can benefit from being automated with a combination of data virtualization/federation with [data workflow](#), validation, and notification software.

Your customers or partners are not getting what they need as quickly and accurately as they would like. The information they need comes from several different sources and each customer wants different information provided in totally different ways. Even though you provide it electronically, it takes weeks to get the information set up for a new company, and if they need a change, it’s another few weeks... not the best for customer/partner relations.

You may be moving forward with Cloud, Big Data, and [Master Data Management](#), but the rest of your infrastructure and IT teams can’t keep up with everyday project work. You should feel like you are on the leading edge of technology, but still the list of new business initiatives and internal projects cannot seem to get off the ground and executed quickly.

4 WHY IS THE INTEGRATION INFRASTRUCTURE SO INFLEXIBLE?

Multiple copies of the same data. Often, data that is needed by more than one system or report ends up with many copies. Too often, separate databases are created for each use, so that each application can find all the data it needs in a single place. That means that there may be four or five or twenty places where the same data can be found. You guessed it... this is a problem! Data changes, so how can you be sure that everyone is working off the same data? That's not as easy as it seems. You can do data synchronization across all these, but that requires an effort because each of these databases may need the information in a different format or context. You are not just broadcasting data multiple places whenever something changes, you are customizing the data for each use. Which one is the best?

Most information is a composite of data from multiple places. One answer to that is to create a data warehouse where all the data resides. You, no doubt, have experienced the typical long delays and resistance to having new information you need put into the data warehouse. This is partly because it is important to make sure the data is appropriately incorporated into the [data model](#), and the bigger it gets, the more difficult it is to maintain reasonable models, and IT people become reluctant to make additions and changes without considerable analysis and validation. The delays also have to do with the tools used to construct the data integration from the sources into the data warehouse. By the way, contrary to what you would expect, usually the application or report that needs the data does not consume it directly from the data warehouse, but instead there is a separate mechanism in place to get the data out of the data warehouse and post it to the application or to the separate database that feeds the application. Each of these data movements is developed by a set of tools that virtually always require [custom programming](#), which is always an impediment to speed and agility.

Complex interdependencies across data throughout the enterprise. With the data flowing as depicted in the "scavenger hunt" picture above, many data stores, applications, etc. are connected via integration. Each data store (database) and application has what is called "[schema](#)," which is essentially the shape of the data. For example, a spreadsheet has columns of data, each with the same format, such as numeric or text. If another application needs that data, the integration that is built will be configured or coded to expect that format. Imagine how many such dependencies must exist throughout the enterprise IT Infrastructure. If a database analyst changes anything in the schema, it could have catastrophic effects reverberating throughout, causing errors in reports, bad electronic transactions, and system crashes. Detecting what has happened can take many days, and no CIO wants to admit it has ever happened. One of the natural impacts of this delicate set of interdependencies is a reluctance to apply changes that are needed unless plenty of study and testing has been performed. Stone Bond's Enterprise Enabler has a patented process for ensuring the integrity across schemas.

5 WHAT ARE THE ROOT CAUSES OF THIS?

Legacy baggage. Apart from the inherent data interdependencies, most of the causes go back to the middleware tools themselves. Because of their age and history, the truly [enterprise-ready](#) middleware companies (e.g., IBM Websphere, Informatica) unfortunately bring a huge amount of baggage and brittleness to integrations across the organization. Over the years, new functionality was necessary as technology evolved, and these companies simply bought newer integration product companies to nominally incorporate functionality. The same growth pattern has brought companies like SAP and Oracle into the integration business. Each of these companies had its own development environment and components, and the purchased products had separate ones. The result is that there is no continuity across the products they offer. Building integration may require using four or five different products with different user interfaces and different underlying philosophies.

Over the last few years new integration platforms have showed up, such as Dell Boomi, and others that are designed for a specific type of integration like Salesforce or general [cloud integration](#). None of these can be considered enterprise grade solutions, and they actually suffer from some of the same ills as the legacy tools since they require custom coding for everything outside their sweet spot. Even the Data Virtualization specialty players (e.g., Denodo, Cisco Composite) do not handle other [integration patterns](#).

Custom coding. Cryptic scripting languages. Plenty of hand coding is required to implement integration logic with data transformation and validation logic. Custom coding epitomizes the core of inflexibility, as it immediately becomes baggage that is hard to support. This is “[Tech Debt](#)” right out of the gate.

Multiple Tools. Little reusability. Programmers and Integration specialists have to use many different tools to create an integration. For the most part, and for the reasons cited above, there is no sharing of configurations and business rules across the tools so redundant work is necessary.

Complexity induced by the tools. All of the above idiosyncrasies of the tools instill fear and doubt in the next team of programmers that need to support the solutions.

Staging Databases. Data from each source is put in a staging data base, with a model that is designed to satisfy the destination or consumers. If you look around, the number of staging databases will probably astound you.

Fortunately, with Enterprise Enabler, [Agile Integration Software](#) has come of age as a truly enterprise-ready platform that can respond readily to change, since these root causes have been eliminated.

6 WHAT IS DATA VIRTUALIZATION?

Not the same as virtualized data. First, it's important to know that Data Virtualization is not the same as virtualizing data. The latter is simply putting data in the cloud or on a virtual machine. Data Virtualization ("DV") is a mechanism for accessing data from multiple totally different sources, aligning it so it makes sense together without making a copy of it anywhere.

Just one of many [integration "patterns."](#) Data Virtualization is a relatively new integration "pattern," that joins the age-old patterns of Extract, Transform and Load (ETL) as one of the ways to get data from its sources to its destination. ETL, and the later patterns of EAI (Enterprise Application Integration) and ESB (Enterprise Service Bus) all are in the business of getting data from where it is to where it needs to be.

Federating data. The key concept about DV is that it is an approach that can bring data live from multiple sources ("federating") without putting it in a staging area or format, and any application that needs data from that set can simply [query](#) the "virtual" database exactly as if it were a physical database. The other patterns physically copy all the data into a physical database and query that. Of course, that approach brings not only a significant amount of overhead, but it carries latency because of the extra steps. DV delivers the very latest data aggregated directly from the sources of record. These integrations can be built in a matter of minutes and made available in many different form like [ODBC](#), [JDBC](#), [OData](#), [SOAP](#) and [REST web services](#), and [native SharePoint BCS](#).

Virtual model, virtual data. There are two reasons that "virtualization" is in the name. First, when data is federated, there is no physical database, but instead, there is a "Virtual" data model that is effectively called upon. The second is that since DV is triggered by the requesting application, the new data that is extracted may never be physically saved anywhere. For example, a website may make a call to get data, and you may see it on the screen, but as soon as the screen is refreshed, it is gone, never having been stored anywhere. This point is of great interest in situations where the security of the data is such that copying it to the cloud, for example, is unacceptable.

Write back to the sources. Suppose the website mentioned above uses DV to display your information such as name, address, etc. If you want to change it, with bi-directional DV, you simply correct it on the screen, and the data will be automatically written back to the original source. While Enterprise Enabler is the only DV platform that does write-back without considerable custom coding, it is a very powerful way to interact with dashboards and portals without building heavy framework.

Federation with other integration patterns. Clearly, the ability to federate data *en route* is a powerful and efficient way to serve what is necessary in almost every data activity. Unfortunately, most DV platforms are limited to the DV pattern; that is, they cannot move data physically to destinations or handle transactions and full data workflows. Stone Bond's Enterprise Enabler is the only truly enterprise integration platform that leverages the live federation across all the integration patterns, and to share the definitions across multiple uses. This capability can be leveraged to dramatically reduce data latency and eliminate almost all of the staging databases in your IT infrastructure. We've seen earlier how much overhead and rigidity are associated with these databases, and leading businesses are beginning to do so, driving toward an agile IT and business infrastructure.

And even Big Data needs Data Federation and Virtualization. If you need to feed Big Data to Hadoop, why wouldn't you want to get the data as fresh as possible, align it with other relevant data and feed the beast with highly palatable data just to the beast's taste? Data federation, virtualization, and validation compose a huge part of what ends up being hand coded into the processing steps for Big Data. Why not save a big chunk of money and add flexibility there, too? Simultaneously DV plays a tremendous role in simplifying gathering and analyzing Social Media and integrating Cloud Apps.

Cautionary Note. It is important to note that most Data Virtualization products only handle sources that are either relational databases or XML, which is a severe handicap that can even negate the possibility of DV when another source is involved. Enterprise Enabler is the only DV platform that combines any sources at all, in their native forms, including databases, applications, web services, electronic devices, spreadsheets, ERPs, and hundreds of others.

7 ENTERPRISE ENABLER: AGILE INTEGRATION SOFTWARE WITH DATA VIRTUALIZATION

The diagram below shows critical key features for a truly agile integration environment. When you look at the matrix, you will see that if any single feature is missing, the tool cannot bring the agility you need to move forward without the constraints of your legacy middleware solutions. Enterprise Enabler provides the full range of capabilities and scalability expected of an enterprise middleware, but with the agility one would expect today.

IF ANY ONE OF THESE IS MISSING....								THEN IT CANNOT BE AGILE	
	IDE	EAI/ETL/SOA/DV	Transformation Engine	Virtual Federation	DV with writeback	Data Workflow	Change management	Impact	Why?
Single IDE (with Runtime engines)	X	✓	✓	✓	✓	✓	✓	5X - 10X increase in implementation time; instant tech debt	Need to move among different tools and need to leave and set up separate environments for testing
EAI/ETL/SOA/DV	✓	X	✓	✓	✓	✓	✓	No reuse of metadata and rules across patterns. Increased development time, increased maintenance/tech debt	No sharing of metadata across patterns. Must learn other products, redundant programming, configuration, installations
Native Transformation Engine	✓	✓	X	✓	✓	✓	✓	More time configuring, custom programming, more processing at run time.	Must convert all data to some central format, e.g., XML, before transformation, then from XML to destination format
Virtual Data Federation (for all patterns)	✓	✓	✓	X	✓	✓	✓	More design, configuring. Significant maintenance/tech debt. Write-back may not be possible.	Staging required. What could be one transformation becomes multiple transformations.
Data Virtualization with writeback	✓	✓	✓	✓	X	✓	✓	Eliminates the possibility of interactive use of data virtualization.	Cannot support end user input from endpoint application to feed back to sources
Complex data workflow logic	✓	✓	✓	✓	✓	X	✓	Can't do realistic event-driven ETL scenarios.	No validation, error handling, parallel data flow
Pro-active monitoring for change	✓	✓	✓	✓	✓	✓	X	Risk of catastrophic impact from changes in endpoint schemas. Risk whenever reusable metadata is modified.	Not taking advantage of the patented pro-active metadata monitoring built into Enterprise Enabler.
With All Features	✓	✓	✓	✓	✓	✓	✓	Agile Integration Software (Enterprise Enabler™)	

8 INTEGRATION INTEGRITY MANAGER

Leveraging the knowledge Enterprise Enabler® has about integration “touch points” and all the schemas involved, the patented IIM ensures that any changes applied to the integration will not cause a conflict with any other parts of the integrated environment. For example, if any of the data being accessed is changed, and that template is used other places, IIM validates that the change will not impact another use. If IIM discovers that it will, it prohibits saving/deploying the change and presents the information of exactly what would be impacted and how.

IIM also monitors endpoints for change at integration touch points. If a change is made to a field data type in a database, for example, it is automatically detected and a notification is sent identifying the change and the impacted integration objects.

9 CONCLUSION

While over the last ten years, consumer technologies have become increasingly easier to use and more powerful, the same is not true for corporate infrastructures. The same products and technologies that came to the forefront fifteen years ago during the Y2K scare continue to be the backbone of data movement throughout the enterprise today. There is a great fear of even touching what’s in place because of the brittle nature of the hard-wired integrations and their interdependencies, and because of the huge time and effort required to make even a small change.

Adopting Data Federation and Virtualization will tremendously reduce your Tech Debt, streamline your data flows to directly address the business needs and reduce the maintenance you have been dealing with on the extraneous staging databases. Most of all, it will unfetter your business to be a responsive and agile company.

10 GLOSSARY

Agile Integration Software (AIS) – A platform for integration that is configured as opposed to programmed, with a single Integrated Development Environment for Design, Development, Testing, and Deploying integrations without ever leaving the environment. Using AIS brings a consolidated agile IT infrastructure, which in turn supports the maximum agility for the business itself.

Cloud integration – integration technologies that perform data movement among cloud-based applications, and between cloud and on-premise applications.

Custom programming (custom coding) – time-consuming work done typically by a computer programmer to define business logic. Programming is behind the scenes of any system or application. Instead of requiring programming, some platforms can be “configured” instead, often by a business user, leaving the technical activities to be handled behind the scenes by the platform itself.

Data Federation – Bringing together data from multiple sources, historically by making copies of the data in a data warehouse or a staging database. With the rise of Data Virtualization, federation is accomplished by extracting and aligning data live from multiple different sources for querying by external consuming applications, without physically moving the data. Only Enterprise Enabler leverages live data federation across all integration patterns, not just on-demand Data Virtualization.

Data latency – the time between when data is generate and it is made available. Typically, other than intentionally historic data, the lower the latency, the better. One wants the most up-to-date information.

Data model – the “shape,” or more technically, the “schema” of a set of data. This includes the data types, data formats, relationships across the data, and other information that describes the data, but not including the data itself.

Data Virtualization (“DV”) – a data integration pattern that creates relationships across separate data sources without saving a copy of any of the data. Sometimes called a “virtual data model,” applications, dashboards and Business Intelligence software can request data from the DV. This integration pattern eliminates much of the overhead of classic data handling, delivering live data directly from the sources, with proper contextual alignment.

Data Warehouse – a central data repository primarily for reporting and analysis. Historic and current data is uploaded into the repository periodically from operational, accounting, and other systems, requiring considerable effort to design construct and maintain. All data from a data warehouse carries a latency; that is, it cannot ever provide real-time data.

Data workflow – the orchestration of logical steps associated data movement; it may include data validation, transformation, filtering’ notifications, with conditional paths and error handling. Data workflow is typically associated with physical data movements.

11 GLOSSARY CONTINUED

Enterprise Enabler® - The only enterprise-ready Agile Integration Software that provides all integration patterns (Data Virtualization, Extract-Transform-Load, Enterprise Application Integration, Enterprise Service Bus, SOAP, and data workflow) in a single product with a single Integrated Development Environment, sharing data federations and configurations across all patterns and complex combinations of these patterns, and including the full range of sources such as databases, data warehouses, relational, XML, cloud, electronic instruments, ERPs, Geo data, binary, and any data standard.

Enterprise-ready – the ability to handle Enterprise scale situations. There are few truly enterprise-ready integration platforms, that are able to scale in size, range of coverage, and perform with the speed, and security required of a fortune 100 company. Few integration platforms, Enterprise Enabler being one, are able to meet the standards for being enterprise-ready.

Integration patterns – patterns such as ETL (Extract-Transform-Load) which moves data in bulk from a source to a destination, utilizing a staging step to store the data so it can be moved from there to the ultimate destination. Other classic patterns are EAI (Enterprise Application Integration), ESB (Enterprise Service Bus), and there are others. Data Virtualization is the newest pattern, which eliminates many of the difficulties and overhead of older patterns

Middleware – an enterprise scale platform designed to handle data movement throughout the enterprise and the cloud, to exchange data with business partners, and take responsibility for the integrity and security of the data.

Master Data Management – a system for managing and standardizing the definitions of corporate data sets, with the objective of having everyone using the same data definitions and ultimately the same data.

ODBC, JDBC, OData, SOAP and REST web services – all are ways to

Query – the way to request a specific subset of data from a data source, usually referring to a relational database

Schema – see data model. Enterprise Enabler automatically discovers the schema of any application, web service, database or other endpoint and presents it to the user for selection of relevant information.

SharePoint BCS – Microsoft SharePoint’s Business Data Services, which include “external lists” and the built-in mechanisms to interact with virtual data bi-directionally, via something call Business Data Catalog. Stone Bond Technologies worked closely with the Microsoft SharePoint development team as they developed this rich set of features. Enterprise Enabler was the only way they could test their new features since they had nothing that could contemplate Data Virtualization. Stone Bond had already been doing data federation for many years, and was able to quickly configure for federation and “write-back” to the sources in just a few minutes.

Staging database – a database that exists for the primary purpose of bringing data together from different sources so that it can be aligned and used together.

Tech Debt – the immediate accumulation of future maintenance whenever certain technologies are used, particularly where multiple tools and significant custom coding are used. The ability and cost of keeping these implementations running over time become prohibitive.