USING THE HADOOP ECOSYSTEM **TO MEET BASEL 239 REQUIREMENTS**

By Lowell Bryan and Abhishek Mehta

November 23, 2015

tresata

USING THE HADOOP ECOSYSTEM **TO MEET BASEL 239 REQUIREMENTS**

By Lowell Bryan and Abhishek Mehta

November 23, 2015

"One of the most significant lessons learned from the global financial crisis that began in 2007 was that banks' information technology (IT) and data architectures were inadequate to support the broad management of financial risks. Many banks lacked the ability to aggregate risk exposures and identify concentrations quickly and accurately at the bank group level, across business lines and between legal entities. Some banks were unable to manage their risks properly, because of weak risk data aggregation capabilities and risk reporting practices. This had severe consequences to the banks themselves and to the stability of the financial system as a whole."

Thus begins the white paper called, "Principles for Effective Risk Aggregation and Risk Reporting," issued by the Basel Committee on Banking Supervision in January, 2013. Based in Basel, Switzerland, this committee was commissioned by the Bank for International Settlements, which is the international organization of central banks, and the closest thing the world has to a global bank regulator.

In the paper, the Committee laid out 13 principles describing how risk data aggregation and management should be undertaken by banks and supervised by national regulators. The principles are all common-sensical and the logic behind them is compelling. Since it was issued, these principles have become known as the "Basel 239 requirements."

The problem, however, as bank executives and their boards immediately realized upon reading the white paper, was the major disconnect between the sweeping aspirations represented by the principles and the operating reality of banks' current system for data aggregation and data management at the time.

The deadline for the very large Globally Systemically Important Banks (G-SIBs) to meet these requirements is almost here (January 1, 2016). Despite the three years of lead time, almost all observers informed on the state of the industry's data management capabilities believe that most G-SIBs will not be in compliance by this time. Indeed, as of January 1, 2015, nearly 50% of G-SIBs self-reported that they would be in "material non-compliance" by the target date. Most observers felt that these self-assessments are overly misleading and actually understate the numbers of G-SIBs which will be in material non-compliance as of January 1, 2016. The majority also believe an even greater percent of large national institutions below G-SIBs size will be in material non-compliance, as their national regulators set their respective deadlines.

NEARLY 50% OF G-SIBS SELF-REPORTED THAT THEY WOULD BE IN "MATERIAL NON-COMPLIANCE" OF BASEL 239 REQUIREMENTS

Difficulties in achieving compliance are not for lack of trying by the banking industry. For example, almost all of the large G-SIBs have undertaken massive, "brute force" efforts to get their data aggregation and data management capabilities in shape. Many have assigned top executives to oversee the efforts, who have, in turn, hired armies of outside professionals and spent massively on technology to address their compliance issues.

Much progress has been made, although given how far behind most banks were initially, it is unclear as of now how many of these "brute force" efforts will be judged as in "material compliance" come January 2016, or beyond for that matter.

The feedback from the Federal Reserve's annual "stress tests" has not been encouraging. The Federal Reserve calls these tests "Comprehensive Capital Analysis and Review" (CCAR). In its follow-up discussions with the banks the Federal Reserve has consistently criticized the:

- Data quality (i.e. too high error rates)
- Ability to provide clear data lineage to the original source systems (i.e. too much aggregation of data through semi-manual spreadsheets) and
- Lack of sufficient historical data.

Why is meeting these requirements such a tremendous struggle?

The problem is that in trying to meet the Basel 239 standards large banks are running into fundamental limits in the "Big Iron" technologies underlying their data architectures. Risk data in a G-SIB is sourced today from literally thousands of reporting systems and databases of various sizes and complexity. Trying to aggregate and manage all of this data through semi-manual approaches is a nightmare. The primary "Big Iron" alternative, however, is to create a single enterprise warehouse devoted to managing all risk data. This exposes underlying limits of how much data volume these warehouses can handle and other related issues, such as how much data history can they maintain. Additionally, "Big Iron" technologies are unbelievably expensive. As a result, most banks are attempting to comply with Basel 239 requirements with a patchwork of direct reporting systems, enterprise data warehouses, and semi-manual efforts to fill in the gaps.

The good news is that a superior technology, collectively referred to as the "Hadoop ecosystem" became available about 5 years ago and has reached a state of maturity that allows it to be a viable option for banks to overcome the limitations of the "Big Iron" legacy systems. In fact, Hadoop is already being used by most large banks to store the vast volume of "raw" data being produced today, not only for Basel 239 purposes, but for all purposes.

We are deliberately using the phrase "Hadoop ecosystem" rather than Big Data to describe this technology. The phrase "Big Data" has been hyped to the point that it has lost its meaning. All the major vendors maintain that they deliver solutions to meet Big Data needs. They also maintain that they use Hadoop. In reality, they deliver technologies where most of the aggregation and management of risk data is in data warehouses, not in Hadoop (which they primarily use just for "raw" data storage). As a result, they run into the same limits typical of "Big Iron" technologies.

Later in this paper, we define the Hadoop ecosystem, its components, and how it can help with risk data aggregation and management holistically.

We believe that taking greater advantage of new technologies, like Hadoop, can help banks meet Basel standards in the near team at far more modest costs than trying (and probably failing) to meet those standards using more robust enterprise data warehouses and reporting systems. In the longer term the same Hadoop ecosystem can serve as the foundation of future data architecture for banks. It can meet the challenge of running a 21st century bank that is fit to succeed in the Digital Age.

The starting point to taking advantage of the Hadoop ecosystem to meet Basel 239 standards is by using it to create a total institution-wide, "ready" Risk Data Asset. Or, simply, what we call a "**Risk Data Asset**." By "Risk Data Asset," we mean a single source of clean, consistent data that is made "ready" within the Hadoop ecosystem to provision all the data needed for all risk applications in a manner that is Basel 239 compliant.

In the remainder of this document, we will elaborate on these ideas by describing:

1. **Challenges banks are facing in meeting Basel 239 requirements**
2. **Underlying limits of enterprise data warehouses in meeting these requirements**
3. **Capabilities of a "ready" Risk Data Asset in meeting Basel 239 requirements**
4. **Steps in building a Risk Data Asset**

CHALLENGES BANKS ARE FACING IN **MEETING BASEL 239 REQUIREMENTS**

BASEL 239 PRINCIPLES

THE PRINCIPLES COVER FOUR CLOSELY RELATED SECTIONS:

(i) Overarching governance and infrastructure
(ii) Risk data aggregation capabilities
(iii) Risk reporting practices
(iv) Supervisory review, tools and cooperation

I. OVERARCHING GOVERNANCE AND INFRASTRUCTURE

PRINCIPLE 1

Governance – A bank's risk data aggregation capabilities and risk reporting practices should be subject to strong governance arrangements consistent with other principles and guidance established by the Basel Committee.

PRINCIPLE 2

Data architecture and IT infrastructure – A bank should design, build, and maintain data architecture and IT infrastructure that fully supports its risk data aggregation capabilities and risk reporting practices, not only in normal times but also during times of stress or crisis, while still meeting the other Principles.

II. RISK DATA AGGREGATION CAPABILITIES

PRINCIPLE 3

Accuracy and Integrity – A bank should be able to generate accurate and reliable risk data to meet normal and stress/crisis reporting accuracy requirements. Data should be aggregated on a largely automated basis so as to minimize the probability of errors.

PRINCIPLE 4

Completeness – A bank should be able to capture and aggregate all material risk data across the banking group. Data should be available by business line, legal entity, asset type, industry, region and other groupings, as relevant for the risk in question, that permit identifying and reporting risk exposures, concentrations and emerging risks. For instance, the Basel Committee's Principles for Enhancing Corporate Governance (October 2010) and Enhancements to the Basel II framework (July 2009).

PRINCIPLE 5

Timeliness – A bank should be able to generate aggregate and up-to-date risk data in a timely manner while also meeting the principles relating to accuracy, integrity, completeness and adaptability. The precise timing will depend upon the nature and potential volatility of the risk being measured as well as its criticality to the overall risk profile of the bank. The precise timing will also depend on the bank-specific frequency requirements for risk management reporting, under both normal and stress/crisis situations, set based on the characteristics and overall risk profile of the bank.

PRINCIPLE 6

Adaptability – A bank should be able to generate aggregate risk data to meet a broad range of on-demand, ad hoc risk management reporting requests, including requests during stress/crisis situations, requests due to changing internal needs and requests to meet supervisory queries.

The risk management principles described in the Basel 239 white paper are comprehensive, sweeping and aspirational.

They are also hard to refute.

For example, Principle 3 - Accuracy and Integrity, states, "A bank should be able to generate accurate and reliable risk data to meet normal and stress/crisis reporting accuracy requirements. Data should be aggregated on a largely automated basis so as to minimize the probability of errors".

Or consider Principle 4 - Completeness: "A bank should be able to capture and aggregate all material risk data across the banking group. Data should be available by business line, legal entity, asset type, industry, region, and other groupings as relevant for the risk in question, that permit identifying and reporting risk exposures, concentrations, and emerging risks."

The problem is that almost all large banks' underlying legacy data management architecture is a hodgepodge of thousands of reporting applications, databases, and data warehouses, drawn off of hundreds of source systems. Additionally, each of these systems were built at different times, often by a combination of different vendors and a wide variety of software engineers.

Given the large number of mergers that have taken place in the industry, the underlying source systems were often built by different banks with very different approaches and standards. As a result, most banks have historically chosen to keep many of their systems separate and to aggregate information across them by semi-manual processes.

The "brute force" efforts that G-SIBs have made over the last 3 years have included massive efforts to clean up the underlying source systems and to build even larger reporting systems and enterprise data warehouses designed to aggregate more and more data from more and more source systems. The efforts have been largely focused on making the data more accurate and more consistent. For some banks the investments being made have literally been billions of dollars.

Despite all the investment, major gaps still remain in most institutions' capabilities. These gaps have been patched to meet "stress test" requirements by deploying huge teams (in the hundreds) of analysts, consultants, accountants, and auditors to overcome data limitations. Much of the work is simply reconciling data drawn from the same source systems, but at different times or from different sources that are conflicting. The spending by a large bank annually on such a "stress test" can easily exceed $100 million.

III. RISK REPORTING PRACTICES

PRINCIPLE 7

Accuracy - Risk management reports should accurately and precisely convey aggregated risk data and reflect risk in an exact manner. Reports should be reconciled and validated.

PRINCIPLE 8

Comprehensiveness - Risk management reports should cover all material risk areas within the organization. The depth and scope of these reports should be consistent with the size and complexity of the bank's operations and risk profile, as well as the requirements of the recipients.

PRINCIPLE 9

Clarity and usefulness - Risk management reports should communicate information in a clear and concise manner. Reports should be easy to understand yet comprehensive enough to facilitate informed decision-making. Reports should include an appropriate balance between risk data, analysis and interpretation, and qualitative explanations. Reports should include meaningful information tailored to the needs of the recipients.

PRINCIPLE 10

Frequency - The Board and senior management (or other recipients as appropriate) should set the frequency of risk management report production and distribution. Frequency requirements should reflect the needs of the recipients, the nature of the risk reported, the speed at which the risk can change, as well as the importance of reports in contributing to sound risk management and effective and efficient decision-making across the bank. The frequency of reports should be increased during times of stress/crisis.

PRINCIPLE 11

Distribution - Risk management reports should be distributed to the relevant parties while ensuring confidentiality is maintained.

IV. SUPERVISORY REVIEW, TOOLS AND COOPERATION

PRINCIPLE 12

Review - Supervisors should periodically review and evaluate a bank's compliance with the eleven Principles above.

PRINCIPLE 13

Remedial actions and supervisory measures - Supervisors should have and use the appropriate tools and resources to require effective and timely remedial action by a bank to address deficiencies in its risk data aggregation capabilities and risk reporting practices. Supervisors should have the ability to use a range of tools, including Pillar 2.

PRINCIPLE 14

Home/host cooperation- Supervisors should cooperate with relevant supervisors in other jurisdictions regarding the supervision and review of the Principles, and the implementation of any remedial action if necessary.

Even with these "brute force" efforts, regulators have been highly critical of the results thus far. What they will say when they review where each bank stands on meeting Basel 239 standards after the January 1, 2016 deadline is unknowable. Most observers believe the regulators will be consistent in the severity of criticism based on the "stress tests" already conducted.

For example, the Federal Reserve has been highly critical of data quality which they define as having excessively high error rates, as it has undertaken its annual CCAR stress tests. They have been particularly critical of the lack of clear data lineage - the ability to be able to trace the data from the risk report they see all the way back to the data coming off the original source systems. Specifically, they have criticized that too much of the analysis they see has been done semi-manually using spreadsheets.

The regulators have also wanted much greater data history to be able to understand the basis for the bank's estimates of the effects of past events on losses and to understand the reasonableness of the projections of future losses due to potential "stresses."

Finally, the Federal Reserve has been highly critical of the technology used to do scenario testing (i.e. the modeling has been too "simplistic").

In response to such criticism, and in an attempt to get ahead of the problem, some banks are contemplating building even larger reporting systems and larger enterprise data warehouses to address these data issues.

"Big Iron" technology on a more massive scale is not the answer. Trying to scale data management using such "Big Iron" technology is not only incredibly expensive, it also will not give banks the capabilities they need.

At the end of the day, "Big Iron" technology has reached the limit of its ability to meet the scale of the data aggregation and data integrity requirements demanded by Basel 239 principles.

In explaining the nature of these limits, we will need to use some technical language. Technology discussions are often hard to follow for many business leaders because of the use of jargon. However, behind the jargon are some powerful ideas. We will try to use straightforward language in this paper.

PLEASE NOTE: TO HELP THE READER WITH DATA DEFINITIONS AND JARGON, WE HAVE PROVIDED A BOXED INSERT AT THE END OF THIS PAPER (PAGES 20 & 21).

UNDERLYING LIMITS OF **ENTERPRISE DATA WAREHOUSES**

Enterprise data warehouses are reaching the limits for the volume and variety of data they can handle. Moreover, they have limits in their ability to produce accurate, consistent, clean data—particularly when the data from the underlying source systems are dirty. This presents a problem when clear data lineage and data history are required. Additionally, attempting to build and manage an enterprise data warehouse and related reporting systems of the size and complexity required to meet Basel 239 requirements is prohibitively expensive. To explain what we mean, let's look at why and how data warehouses have reached their limits in terms of volume, data integrity, and spending effectiveness.

LIMITS ON VOLUME

The volume of data implied by Basel 239 requirements is staggering. In effect, Basel 239 requires aggregation of data of vast quantities from nearly every source system in a bank. Every demand deposit system, loan system, payment system, and risk exposure system come into play. Every trading system and treasury system, for all cash and derivative instruments, is involved. Some of these systems are integrated global systems. Most, though, are regional or country-specific systems. In turn, most of these systems come with their own idiosyncrasies.

The volume of "raw data" being created by all these systems can run into the terabytes (i.e. a trillion bytes) each day.

Data volume is not just about daily volumes; what is really critical is how much is retained and for how long.

Indeed, the single greatest driver of the volume of data that needs to be managed is the amount of history being kept for potential future analysis. Over the last couple of decades, the most used measure of data volume has been the terabyte (i.e. a trillion bytes).

The simple decision to keep a terabyte of data daily, for a thousand days (3 years) converts that terabyte of data into a petabyte (i.e. one quadrillion bytes).

When you ask the Chief Information Officer of a G-SIB, "How much data do you have that could be managed?," their estimates range from 50 to 100 petabytes or more. Against this measure, they estimate no more than a petabyte or two is "being managed" today.

There is a fundamental reason why enterprise databases can't handle the volume required by Basel 239. Enterprise data warehouses were built for terabytes, not petabytes. A large enterprise data warehouse managing 25-50 terabytes of data can cost $100 million or more to build and $20 million a year to maintain. Simply storing a terabyte of data for use within the enterprise data warehouse can cost over $100,000 a year, (compared to $1,000 a year on AWS). In other words, just storing (not managing, or processing) 50 terabytes of data in a data warehouse can cost $5 million. Even if they could handle a petabyte of data, the storage costs alone would be substantial i.e. $50 million (storage being no different than storing pictures on Google or Apple.)

ENTERPRISE DATA WAREHOUSES WERE BUILT FOR TERABYTES, NOT PETABYTES

The problem is that enterprise data warehouses are mainframe based (i.e. "Big Iron"). These machines were created for transaction processing versus data management. The underlying technology was developed some 70 years ago and has been implemented and developed ever since. Today's mainframes are strong machines each of which are made up of a powerful central processing unit (CPU), disc storage of data, and random access memory (RAM). The RAM holds the data while it is being processed by the CPU and the flash/disk storage holds the rest of the data until it is ready for processing. A mainframe processes one transaction at a time very, very quickly. Today's mainframes scale by using bigger, faster machines and by splitting data processing loads over multiple machines. Multiple mainframes working together can handle terabytes of data. They are commonly referred to as "scale-up" systems.

These "Big Iron" systems are organized to be extremely reliable and are particularly essential for operating business systems and the processing of related transactions. They will continue to lie at the core of large transaction-based business operating systems for the foreseeable future.

Mainframe-based approaches, however, have reached their limits in their ability to satisfy data needs through direct reporting long ago since most use cases require aggregating data drawn from multiple operating systems and from other sources.

It is for this reason that client-server based data technology systems came into existence in the 1990s. These systems enabled you to manage data that is derived from multiple data sources using common reference data that is managed once and accessed multiple times.

The last two 2 decade data management innovation was to create data warehouses that aggregate data across multiple databases using common reference data. Eventually, these were scaled to become "enterprise" data warehouses intended to provide data to the entire institution, often through a related enterprise management system. Now even these very large "enterprise" data warehouses are reaching their limits as they fundamentally remained "scale-up" systems, like thier mainframe predecessors.

The core issue is that even a large enterprise data warehouse (i.e. 25 terabytes or bigger) that aggregates data from a large number of source systems is running against the limits of what the most powerful mainframes can do. Data warehouses of this size are complex, difficult to build, and impossible to manage. Building one requires long lead times and multiple software development teams under the overall direction of a common leader. Much of the cost and complexity of undertaking such large enterprise data management projects are the costs of coordinating and controlling the work being done by multiple teams.

As the data warehouse grows in relation to the size of the data being managed, the programming itself becomes more complex. For example, programmers need to use techniques such as "sharding" to partition data from an overloaded database into multiple databases to handle the increasing data volumes.

The biggest constraints arising from high volumes in operating enterprise databases are the costs, time, and complexity of moving very large volumes of data multiple times throughout the warehouse. These processes are called "ETL processes." The term ETL stands for Extract, Transform, and Load. In reality, little or no transformation of data is done by ETL processes. Instead, there is lots of extracting, duplicating, moving and loading the of data, repeatedly.

The first ETL process takes data from the source systems and puts it in storage. The next ETL process takes the data from storage, duplicates it, and moves it to "staging"or processing (e.g. structuring the data so it can be placed into a database or data warehouse schema). Once staged, the data is then again extracted, duplicated, moved and loaded into a relational (i.e. enterprise) database. After it is transformed in the relational database, data intended for different populations of users is extracted, duplicated, moved, and loaded into "data marts" (i.e. databases designed to provide access to the same data by a user population for a specific application) for analysis and reporting.

Moving terabytes of data multiple times within a data warehouse is time consuming and expensive. Moving petabytes of data multiple times is simply impractical. For these and other reasons, the practical volume limitations of enterprise data warehouses are around 100 terabytes. So if you need to aggregate and manage petabytes of data to meet Basel 239 requirements, a single, very large enterprise data warehouse is not your answer.

LIMITS OF DATA INTEGRITY

Data has integrity if it is consistent, complete, and accurate. Basel 239 creates high standards for data integrity. Against these standards data warehouses have severe limitations in aggregating and managing data to the required level of consistency, completeness, and accuracy.

It may seem strange to hear that relying on data warehouses creates major data integrity issues in meeting Basel 239, since data warehouses were intended to have high data integrity. The reason why data warehouses are used for purposes such as financial reporting is because they are designed to ensure data integrity.

For example, they operate by "structuring" data through the "staging" processes. In staging, the data from various source systems is organized into a relational structure (rows and columns) in a single database supported by a Database Management System (DBMS). A relational database, which forms the core of an enterprise data warehouse, is supported by a Relational Database Management System (RDBMS) which organizes database tables in schema so that data defined in multiple databases can be analyzed together. The formal definition of "database schema" is a set of formulas, called integrity constraints, within the database that ensure compatibility between parts of the schema.

Enterprise data warehouses have high data integrity if all the data to be used is managed through an internally consistent schema, and if the underlying "raw data" and the related "reference" data and "meta" data are sufficiently accurate for the purposes for which the data is to be used. Unfortunately, large banks are discovering that they cannot always meet these conditions and that existing data warehouses are part of the problem in providing the consistent, complete, accurate data that Basel 239 requires. Next, we will take a look at their limits in providing consistent, complete, and accurate data in a little more detail.

VISUALIZE
ANALYZE
PROCESS
STORE

LEGACY DATA MANGEMENT STACK

Data has been managed in a 'stack' based mode for the better part of the last 50 years. This approach worked well when data under management was in the terabytes. However, the fact that storage, database, analytics and visualization technologies were layered on top of each other imposed a costly data movement 'tax'. Ths model is not designed to scale as data under management moves into the petabytes (1PB = 1000 TB). The stack was also built on the premise to manage critical data assets.

1. Data Consistency Limits

In terms of practicality and cost, enterprise data warehouses reach both practical and cost limits at around 100 terabytes of data. Because of the long lead times, exorbitant budgets involved in building such systems, and their inherent inflexibility, most large enterprises only use large enterprise data warehouses for critical core functions, such as financial reporting or customer relationship management.

Still, user demand for data is insatiable. In addition to building these massive enterprise data warehouses, most large banks have built literally thousands of smaller databases and data warehouses for many other purposes (like analytics) and an equal number of reporting systems that draw information directly from operating systems.

As a result, important risk data in some banks may be on hundreds or even thousands of databases and reporting systems.

Unfortunately, while a single relational database may have internal data consistency, the vast array of systems that provide data to meet Basel 239 requirements across the enterprise as a whole, do not. At a very fundamental level, one of the major reasons for data inconsistency in aggregating risk data is that the data is drawn independently from multiple database and reporting systems at varying times in order to produce aggregated risk reports. This creates enormous data consistency issues because raw data extracted at differing times is different data, even if drawn from the same underlying source systems.

Such inconsistent data must be reconciled using semi-manual processes to ensure analyses are valid. Reconciling data from such inconsistent sources has become a nightmare for many banks. This is not only wasteful, but it also can create fundamental data integrity issues since even with massive amounts of work, sometimes assumptions (guestimation?) need to be made to reconcile the data. Bad assumptions can then lead to bad conclusions, which is why regulators have been so tough in criticizing large banks for the over use of semi-manual processes. Some advocate that the only answer is to build even bigger enterprise data warehouses, but this runs directly into the data volume limits of data warehouses described earlier.

2. Data Completeness Limits

DBMS and RDBMS based data warehouse systems cannot provide sufficient historical lineage to meet Basel 239 requirements, because they are incomplete in the sense they can not hold all the required historic data.

One of the reasons for insufficient history is the volume limits described earlier. Keeping historical data from every source system in a G-SIB for several years quickly gets you to the petabytes of data that greatly exceed the 100 terabyte practical capacity of very large relational databases. For this reason, the primary use case that has already motivated large institutions to evaluate new technologies (like Hadoop, which we will be describing later) is for storing large amounts of historic data. A number of the largest G-SIBs now have ten or more clusters that are primarily devoted to data storage of petabytes of historic data.

There is an even more fundamental reason why traditional systems do not provide sufficiently complete historic data: Such systems are architected to be incremental. That is, data in a field described by a row and a column in a schema has only one value and, as the data changes with the passage of time, the data in that field is updated. Unless the data is consciously kept in a field to make it persistent (i.e. recording the total exposure to a company at a particular moment in time), the data in the field will change and the old data will be lost as the data is dumped or overwritten.

While most of the raw data from source systems is archived and can be retrieved, when the interim data within the warehouse is "dumped," it becomes irretrievable. Once the data is lost, you can not go back to see how that interim data changed over time. This is a problem, if, for example, the interim data represents the total enterprise-wide exposure of a company, or group of companies, to a particular kind of risk or if you want to make year to year comparisons. It also means you can not "back test" the quality of data from one year to the next if the necessary interim data is missing since, you can not duplicate the analyses. It means you can not trace back the data lineage from the final output to the original source systems, if you can't find the interim data that provides the "breadcrumbs" that define the trail back.

In other words, data warehouses have severe limits in enabling institutions to keep the complete data records required by Basel 239.

3. Limits on Data Accuracy

Data warehouses also have limits in their abilities to cure data accuracy issues. Data cleaning is one of the most basic capabilities of a database. Data cleaning in a database can often correct or remove incomplete, incorrect, inaccurate, or irrelevant, data usually by identifying inconsistencies.

After cleaning, a dataset will be consistent with other similar datasets in the system. The inconsistencies detected or removed may have been originally caused by entry errors, corruption of the data in transmission or storage, or by using data dictionaries from different databases that vary in the data definitions that they use.

The problem is that data consistency does not mean the data is accurate. Data warehouses in particular, are designed to make reference data consistent. Reference data is the data used to categorize other data in a database or for relating data to information beyond the boundaries of the enterprise (i.e. name, address, industry, sales, etc. of a corporate customer).

One of the biggest issues in using consistency as a proxy for "clean" data is that the reference data may need to change given the purposes of the analysis. For example, do you define "exposure" to a particular country by companies headquartered in that country, companies with subsidiaries in that country (no matter where the are headquartered), companies importing goods and services from that country, or companies exporting goods and services to that country? Depending on your answer, your "country exposure" will be dramatically different.

This is not a trivial issue from a Basel 239 perspective. Consider paragraph 57 under Principle 8 – "Risk reports should include exposure and position information for all significant risk areas (i.e. credit risk, market risk, liquidity risk, operational risk) and significant components of those risk areas (i.e. single name, country, and industry sector for credit risk).

Perhaps the single messiest data accuracy issue for Basel 239 reporting from a data warehouse perspective is entity resolution.

To understand your exposure to a company or other financial institution, you need to relate every interaction you have with every legal entity related to that company. A very large bank may have tens of millions of such interactions a year with a single company. To understand your exposure, you in turn need to understand the entire "family" structure (i.e. 'grandparent,' 'parent,' 'child') of the corporation, as well as entities such as special purpose vehicles used for financing transactions, joint ventures, guarantees to suppliers, etc. If the data warehouse gets the relationship wrong when making the data consistent, the reported data will also be consistently wrong. If the underlying reference data is inaccurate or missing, as it often is, then the data warehouse will miss the relationship and the exposure data will also be wrong. Furthermore, if there is duplicate data, the exposure to an entity may be listed twice.

This current problem with entity resolution also creates a risk issue driven by government requirements to "Know Your Customer" (KYC). In particular, the US government is increasingly holding banks accountable for knowing whether customers are legitimate risks in regards to Anti-Money Laundering (AML). If you can not accurately resolve the entities you are dealing with, you will either create a lot of "false positives," which are difficult and expensive to track down, or miss "bad actors," which can result in government sanctions and massive fines.

At the end of the day, even if data warehouses had completely accurate reference data, there still remains a fundamental issue in relying on them for data integrity: you can not build a single one that is big enough— or economically. This means that you will have to live with aggregating data semi-manually over multiple enterprise data warehouses or reporting systems. As long as this is true, risk data aggregation and management for the entire institution will require large numbers of professionals to do so semi-manually, which means you will have data integrity and data reconcilement issues.

SPENDING LIMITS

In addition to volume and data integrity limitations, data warehouses are also cost prohibitive. The "brute force" effort to improve risk data management over the last three years has been massive. In some cases, the surge of spending at individual G-SIBs has involved billions of dollars in increased capital spending and hundreds of millions of dollars in increased annual spending. Much of this spending has been to overcome the limits of mainframe-based approaches in handling the volume of data and data integrity requirements implied by Basel 239.

Some banks believe that the biggest surge in spending is behind them, although they acknowledge much is left to be done. For others, spending in risk data systems is still escalating.

"BIG IRON" TECHNOLOGIES THAT CREATE MASSIVE ENTERPRISE DATA WAREHOUSES MAKE LITTLE ECONOMIC SENSE

One thing consistent across all banks is that the surge in spending on risk systems has been hard to afford and avoid. In most banks the surge in risk data spending has squeezed out or delayed other "mission critical" technology and operations spending. For example, efforts like digitizing operating processes or building new applications to better serve customers have taken a back seat. Indeed, risk data spending has encroached on budgets that would allow for banks to take advantage opportunities to better use the vast volumes of "unstructured" data that are being created in the Digital Age from call centers, mobile devices, and social media to better serve clients.

Trying to push the limits of the legacy"Big Iron" technologies further to create even more massive enterprise data warehouses for risk data aggregation and management purposes makes little economic sense.

Instead, the focus should first be on using the Hadoop ecosystem to relieve the limits imposed by data warehouses and the related labor intensity of dealing with their volume, data integrity limits, costs, and effectiveness of managing risk data. The objective should be to improve the availability of high quality risk data while stopping the growth, or better yet, reducing the technology spend and talent costs for risk data aggregation and management. In the medium term, we believe you should be able not only to meet Basel 239 requirements, but also dramatically improve actual risk management practices while fundamentally reducing the technology costs involved in doing so. In the process, the money saved can be invested in getting far more value from the bank's talented people than using that talent to reconcile inconsistent data.

CAPABILITIES OF A "READY" **HADOOP-BASED RISK DATA ASSET**

After a decade of development, the Hadoop ecosystem is finally ready to help very large enterprises take on the challenges of managing the volume and variety of data becoming available as the Digital Age matures. For the purposes of this paper, it has matured to the point that it can help large banks meet Basel 239 requirements.

As previously mentioned, we are using the term, "Hadoop ecosystem" rather than Big Data, to describe our ideas. The term Big Data has been hyped to the point it has lost its meaning. Every vendor that uses it defines the term to fit whatever they want to sell.

For example, the vendors who sell very large enterprise data warehouses might maintain that they enable Big Data management, but, in reality do not.

The core idea in this paper is that the Hadoop ecosystem-based technologies have evolved sufficiently enough to enable the build of a total institution wide "ready" Risk Data Asset composed of multiple petabytes of data. By that we mean a source of easy-to-access, clean, consistent data that is capable of meeting all the data needed by all the users of risk data. We call this population of users "the risk domain." The risk domain embraces all of the managers, analysts, front-line personnel, regulators, and others who need to access risk data.

Let's examine how the Hadoop ecosystem can help build a Risk Data Asset to meet the entire institution's need for risk data. To do this, we will first need to describe the capabilities that the Hadoop platform can bring to bear in aggregating and managing risk data. We will then describe what a "Risk Data Asset" would look like and how it would work.

HADOOP ECOSYSTEM CAPABILITIES

The technology behind Hadoop is founded on using clusters of computers rather than mainframes to massively parallel process, store, and analyze vast quantities of data. It is the technology behind the "cloud." It was pioneered by players such as Google, Facebook, Yahoo, and Amazon, as well as by government agencies (i.e. NSA, CIA, etc.) in the late 1990's and has been invested in and open sourced by these and other players since the early 2000's. Until recently most of the focus of these new technologies has been on managing and transforming petabytes of unstructured data (i.e. unorganized data).

Hadoop is truly "Big Data" technology. Using a Hadoop cluster enables the storage and large scale processing of data in clusters of commodity hardware and the use of machine learning algorithms and population scale analysis and modeling, versus the use of sampling. It is also much, much less expensive and much, much faster, than the "Big Iron" technology banks use in their legacy platform for processing the same amount of data.

A Hadoop cluster can store any type of data (i.e. structured or unstructured, internal or external); it can be scaled to hundreds of petabytes, and uses industrial strength widely available servers. The Apache Software Foundation (ASF) has supported the open source development of Hadoop since Yahoo donated it to the foundation a decade ago. Hadoop evolved into the open source standard of choice for massive parallel processing and storage of data for enterprise purposes. Players such as Intel began to invest heavily into Hadoop several years ago by building into their chips the kinds of features required by corporate users (i.e. security features such as encryption built into chip design). As a result, Hadoop has become a robust, enterprise-ready platform with the necessary capabilities to support enterprise data needs. In fact, it is rapidly becoming a disruptive and fastest growing opensource technology.

We refer to it as an "ecosystem" because there is a set of related software that enables Hadoop to be used for powerful data aggregation, management and analytics. Hadoop itself is a batch system, but Spark, which is an in-memory software that enables "real time" processing in Hadoop clusters, can aggregate very recent data (i.e. last few seconds) with historic data. The ecosystem includes other software such as YARN, HIVE, and HBASE, all of which provide additional functionality to the system.

The Hadoop ecosystem provides far better ways for managing the volume and variety of data as it comes off of the transaction systems than data warehouses do for storage, processing and analysis of data.

Why?

It is because the Hadoop-based ecosystem technologies are designed for the distributed processing of data though multiple nodes, with each node having its own CPU, disc storage, and random access memory, and with fault-tolerance at the software level - almost like assembling your own super computer but for the tenth of what super computers cost. This lets multiple machines process data in parallel by using the same software to reduce all the data in the cluster simultaneously. Such a cluster scales by simply adding more nodes rather than by using bigger, faster machines. Nodes are composed of commodity, lower cost, less reliable machines that achieve reliability through duplicating and backing up data on multiple machines. If one fails, it is simply replaced with no loss of data. If a problem arises in the process of data transformation due to human or machine failure, you simply rerun the job.

Hadoop clusters can be scaled cost effectively to handle petabytes of data. In contrast to mainframes, nodes are cheap. Even a 50 node cluster (roughly able to store and process a petabyte of data) costs only a fraction of the mainframes that are needed to store and process a terabyte of data. While storing a terabyte in a data warehouse for a month can cost up to $10,000, storing a terabyte of data in a node would only cost around $500 a month. The speed advantages of processing large volumes of data in a Hadoop cluster are remarkable. Intel has done tests of Apache Hadoop in its own infrastructure that shows it can reduce the time to sort a terabyte of data (referred to as a terasort) from four hours using a mainframe based approach, to seven minutes, or roughly 35 times faster. In other words, in terms of capacity and processing speed, it is like comparing an ox-cart to a modern truck.

Moreover, the truck costs just a fraction of what the ox-cart costs!

Furthermore, Hadoop can handle an enormous variety of data. Unlike data warehouses, which require data to be structured, the Hadoop ecosystem handles and stores almost any kind of data—structured or unstructured. Unstructured data can come from mobile devices, mechanical devices, social media, call centers, public data, other firm's proprietary data, or even documents, videos, graphs, maps, and so forth.

All you need to do is go to Facebook, LinkedIn, Twitter or Yahoo (all next-gen data companies) and observe the enormous variety of data that they store, access, and analyze in Hadoop.

Data within Hadoop is not transformed through use of rigid schema but is rather "reduced" by applying algorithms to all the data stored in the cluster. In other words, a Hadoop cluster reduces all the relevant data every time a job is run. For very large jobs that can require parsing through petabytes of data, it still never misses a beat.

Hadoop therefore enables you to keep all the data in the cluster itself. It is not stored separately. It is not incremental in that all data is maintained and never updated. Data is never "dumped." You can keep all the history going forward as well as all the interim calculations. New data is simply added. Keeping all data is cost effective because it costs so little to add data storage capacity even by the petabytes (for example it costs $100 for every additional terabyte).

This enables you to do much more with the data. You can produce "movies" rather than "snapshots" of how the data changes over time. You can analyze network efforts. If a bug is found in the data and you need the right answer, you simply rerun the software since you still have all of the original data.

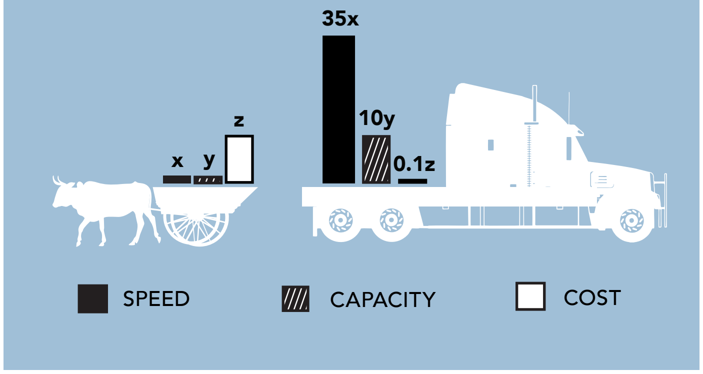
Hadoop has enormous advantages over traditional "Big Iron" approaches, not just in the volume of the data it can handle, but also the variety. It is also far more human (and machine) fault tolerant.

OX CARTS VERSUS TRUCKS

It is not far-fetched to compare the technological difference between Big Data technologies and the legacy technology platform in managing enterprise data to the difference between ox-carts and trucks in carrying loads.

As noted by Richard Gabriel and Karen Metz in their June 1992 *A Short History of War* written for the US Army War College, logistics management had a long history of determining who wins wars. As the size of armies increased in ancient times, armies had to master the task of logistically supporting them. Ramses II revolutionized logistics by introducing the ox-drawn cart, which could carry 1500-pound loads (rather than the previously used donkeys which can only carry about 300 pounds.) This enabled the supply of much larger armies. This ox-cart technology itself had limits (it only travelled two miles an hour,) so it was eventually replaced by teams of horses, which could carry the same loads at more than twice the speed (i.e. 5 miles per hour) at half the cost in in forage to feed the animals.

But all animal based supply, including teams of horses, can't compete against the combustion engine. Trucks today handle box car size loads of anything you want to ship at speeds of 70 miles per hour which is 35 times faster than an ox-cart. Similarly, Intel has done tests of Apache Hadoop on its own infrastructure that shows it can reduce the time required to sort a terabyte of data (referred to as the terasort benchmark) from four hours to approximately seven minutes or roughly 35 times faster. And just as it's easy today to scale the volume of loads carried by trucks (versus being hard to scale by using more ox-carts), it is easy to scale the Big Data volumes and variety by adding more servers, memory, and disc storage (organized into "nodes") to the cluster. In that way, you can increase the cluster up to "petabyte" scale simply by adding more nodes.



OPPORTUNITIES TO BUILD A READY RISK DATA ASSET

Given these differences in power, scalability and cost effectiveness between the Hadoop ecosystem technologies and the traditional "Big Iron" technologies in handling the volume and variety of data, why have these technologies not already been applied to risk data aggregation and management? The reason is straightforward: until now, these technologies were not ready for "primetime" for "mission critical" data management use cases.

More specifically, the data that has been loaded into the Hadoop cluster up until now has not been "ready" for transformation either by existing risk management applications or for building new ones. While the Hadoop clusters have become widely used by large banks to store the "raw" data as it comes from source systems, the data has been "dirty." Moreover, "raw data" lacks reference data. Until recently, the only practical way to clean the data and add reference data was through staging the data and extracting and loading it into databases and data warehouses through the ETL processes described earlier.

To make Hadoop capable of realizing its potential, the data in it must be "ready." We define "ready data" as clean, consistent data, including reference data, that is ready to be loaded into applications systems for transformation.

This is where Tresata, the sponsor of this report, comes into this story. Tresata produces software that operates within the Hadoop ecosystem and convert "raw data" into data that is "ready"- to be loaded into all existing and new applications for predicting analytics.

How do we do it? Tresata software works within the Hadoop infrastructure (withiut moving data out of it) to collect, curatem compute and convert raw data into actionable intelligence. We have built predictive machine learning routines that allow:

1. Determine the quality and references in the data
2. Discover relationships across all data sources (internal and external)
3. Resolve entities and related hierarchies
4. Enrich underlying 'bad' data to make it usable data

We do this without relying on having reference data coming from the source systems that use a common identifying number. Instead, we create new 'unique IDs' to serve as markers for all the reference data. By doing so, we can help the institution create a single source of clean, consistent data for all risk applications. As a result, the data is easy to reconcile since all the data for all applications can draw data from the same source at the same time.

No data is ever dumped. This means you can go about finding the data lineage of any data transformation all the way back to the "raw data" coming off the source system. Plus, you can analyze as much history as you want to use. Our software has analyzed every payment made by all corporate and financial institutions served by a G-SIB over many years (>100 billion rows of data), by creating a Payment Data Asset. This same approach can be used to create a Risk Data Asset. By that we mean a single source of clean, consistent data that can be used to provide data to all existing risk reporting systems, data warehouses, and new applications.

Rather than investing a hundred million dollars or more on a large enterprise data warehouse, building a Risk Data Asset should cost in the tens of millions of dollars. Additionally, annual operating costs should be a small fraction (i.e. one tenth) of the ongoing maintenance and ETL costs of operating a large enterprise warehouse. Hadoop clusters have modest maintenance costs and do not have ETL costs except for loading the "raw data" from source systems into the cluster and extracting the "ready" data to the applications that will use the data.

It is important to note that if the data is simply taken from the "ready" Risk Data Asset and then loaded into a data warehouse, lots of ETL (extract, transform and load) will still be required. To reduce the costs of such enterprise data warehouses banks will need to take steps to change how they use them in a way that reduces the volume of the data they manage in the warehouse and the amount of ETL they therefore have to do.

The efficiencies from transforming most of the data in the Hadoop cluster rather than in a data warehouse are very considerable. It is far more efficient to move an algorithm (with perhaps a megabyte of data) to a Hadoop cluster than to extract, duplicate, and load a petabyte of data from a Hadoop cluster to be staged for an enterprise data warehouse and once staged to move that petabyte of data multiple times again within the warehouse itself!

This gets to another huge advantage of creating a Risk Data Asset: you can build applications to analyze the entire dataset in the cluster without moving the data at all. You transform the entire dataset each time you run a job. This lowers the traditional cost of transforming the data to a small fraction of the costs of doing the same job in a data warehouse. In truth, many of the jobs you may want to do in a Hadoop cluster involving transforming the entire dataset are impossible to do in a data warehouse.

Among other benefits, this means you do not need to construct "statistically valid" samples for analytics. For example, say you are trying to determine the correlation between loan losses on credit cards between people who make direct payroll deposits in your bank and those that do not. Rather than undertaking a regression analysis on a statistically valid sample, you instead can analyze the entire population. In fact, while you are at it, you might also check out the impact of branch usage, call center usage, online mobile-banking usage, home value, home equity size, payroll deposit amount, etc. on credit card losses. Indeed, you can use "machine learning" techniques to discover the relationships, including discovering relationships that are not even intuitive.

Building such new applications is relatively easy because the data you need is "ready." Once you have put "raw" data in the Risk Data Asset for one use case, it is "ready" for any other use case that needs that data without having to go back to the source systems again to extract "raw" data. This greatly reduces the costs of building new applications to a fraction of what it costs to build applications that have to go back to the original source systems one more time.

STEPS IN BUILDING **A READY RISK DATA ASSET**

To reiterate what we mean by a "ready" Risk Data Asset, we mean the single source of all the data, stored and readied within the Hadoop ecosystem, necessary to meet all of the institution's risk data aggregation and management needs. By "all" data, we mean that the Risk Data Asset can include structured or unstructured data, from internal or external sources, in any volumes required (i.e. multiple petabytes). By "ready," we mean that the data is ready to be loaded into business applications for analytics. This involves profiling all of the source data, reference data, and metadata to identify, and then cure, all data issues. When ready, the data can be loaded into all of the institution's existing applications, or it can be transformed through newly built applications within the Hadoop cluster itself.

Creating a "ready" Risk Data Asset removes all the data obstacles involved in complying with Basel 239. For example, such an asset can provide clear data lineage back to the source systems for any data to be reported. It can hold as much history as is required. Moreover, it can enable building much more powerful, much more effective new risk applications to empower the institution to capture far more value from risk management capabilities.

Building a "ready" Risk Data Asset involves five steps:

1. **Define, as Basel 239 requires, who will "own" the Risk Data Asset and the data that it contains**
2. **Load all the raw, reference, and metadata into the Hadoop ecosystem**
3. **Install and use Tresata's software to help make the data "ready"**
4. **Load the "ready" data to all existing risk management applications, including applicable data warehouses**
5. **Improve risk data aggregation and risk management systems fundamentally by building new applications that can use the asset fully, and by eliminating redundant and unnecessary spending on legacy systems**

Let's look at each of these steps one by one.

STEP 1: **DEFINE OWNERSHIP**

Principle 1 of the Basel 239 document states that "a bank's risk data aggregation capabilities and risk reporting practices should be subject to strong governance arrangements consistent with other principles and guidance established by the Basel Committee."

Taking the Risk Data Asset approach makes it relatively straightforward to create governance accountability. By creating this asset, the focus can be on creating data for risk data aggregation and risk management purposes rather than for other purposes. As a result, this makes it easy to name the person (or persons) and the structure (i.e. Risk Committee) responsible for owning all the related risk data issues.

Let us assume for the sake of an example, that the decision is made to create a Risk Data Committee under the oversight of the CEO, Chief Risk Officer, and the Board to "own" the Risk Data Asset, and that the committee is chaired by a single executive reporting to the Chief Risk Officer.

The kind of issues this type of committee, and its chairman, should "own" are:

- What raw data is to be put into the asset? Unstructured? Structured? Internal? External?
- What reference and metadata is to be put into the asset?
- What historic data needs to be retained?
- What are the data lineage requirements?
- What existing applications can be fed with better data?
- What new risk applications should be built?
- How should we manage the cost effectiveness of risk data spending and investments?

At the end of the day this committee and its chairman would need to "own" the target data architecture for risk management, as well as the related projects and tech spend. The committee chair would need to be accountable to the CEO, Chief Risk Officer, and the Board for ensuring that the Risk Data Asset is able to be fully compliant with the Basel 239 requirements. To do this, this same chair and the committee as a whole would be responsible for overseeing steps 2 through 5 described in the next few pages.

STEP 2: **LOAD THE DATA INTO THE HADOOP ECOSYSTEM**

The second step is to install a Hadoop cluster and the related open source software (i.e. HDFS, YARN, SPARK, HBASE). One of the first decisions to be made by the accountable governance body is to determine the size of the initial cluster in terms of terabytes / perabytes of data that needs to be managed and whether or not the system should be dedicated to risk or used for other purposes.

Our strong bias is to dedicate it to risk. Some vendors argue that you should create a total institution-wide central data source for all purposes, but our position is that this is a bridge too far. Such thinking comes from those that have an enterprise data warehouse mentality versus one based in risk.

Not only would an institution-wide approach be prohibitively expensive and time-consuming, it is simply too hard. The core issue is that different users across the institution have very different needs and therefore have very, very different data requirements in terms of data volume and data variety. Trying to create a data asset for the risk domain that would also serve financial reporting users is impractical. As a starting point, trying to do so would confuse ownership of the data itself, violating Basel 239 Principle 1. Financial reporting people might prioritize having real time data available to make it easier to close the books and provide daily income and balance sheets. Risk reporting might prioritize data lineage and data history requirements. The data accuracy requirements of financial data and risk data are simply different. Financial reporting prioritizes the financial accuracy of transactions and balances, but is less concerned about the accuracy of reference data (i.e. who is making the transaction and who holds the balances does not affect financial reporting). On the other hand, the accuracy of reference data is essential to risk management (i.e. to which entities, individuals, and countries is the bank exposed to), since without accurate reference data the financial, transactions, and balance lack context.

Financial reporting may only really need structured internal data, but may also want it to be "complete" (i.e. all the data in the enterprise needed to close the books). Risk reporting may want not just structured, but unstructured data, not just from internal, but external sources, to enable it to conduct "stress tests" using external, unstructured economic data for scenario modeling. Risk reporting may be happy with "incomplete" data that provides clues to emerging risk issues (e.g. for early warning). As an analogy, risk managers may want the kind of advance, incomplete information weather forecasters use to spot a "hurricane" that is developing in the Caribbean. Financial reporting people may be more like insurance companies who total up the actual damage done by a hurricane in order to pay off insurance claims.

Assuming the decision is made to create a dedicated Risk Data Asset, the next decision is what data to load into it.
The starting point is to inventory all the source data, reference data, and metadata already being used by the existing applications (and databases) for risk aggregation and risk data management, and to load all of that data into the asset. The other related need is to decide how much data history to put into the asset. In the past, banks had stored one to three years of raw data, but the time span usually varies among sources. This will imply building a cluster with at least 1 or 2 petabyte of capacity for a large bank. The good news is that it is easy to scale the cluster rapidly if you need greater capacity.
Building and tuning such a Hadoop cluster will take a few months. The two companies who do most of this work are Hortonworks and Cloudera. Once built, you can then actually load the data from the various source, reference, and metadata systems. Over time, the Risk Data Asset owner is likely to want to add other data from other sources (i.e. structured and unstructured, internal and external).

STEP 3:
INSTALL & READY THE DATA

Step 3 is to install the software needed to ready the data. Unlike the open source software used to run Hadoop, the software to "ready" data is proprietary and needs to be licensed. At the moment, the only company currently offering such software that is able to "ready" the data as is described in this white paper is the sponsor of the report, Tresata. As is standard industry practice, however, many other companies will claim that they can do so. We would urge you to test their claims.

Once you license the software from Tresata, have installed it in your Hadoop cluster, and have determined how you want the asset governed, you are in a position to start building a "ready" Risk Data Asset. Tresata is usually first involved during the process of configuring and integrating the data with its software.

It is admittedly a major undertaking if you want to create a "ready" Risk Data Asset for a G-SIB given the vast quantity and variety of the underlying raw and reference data. More specifically, the data that would be configured and integrated includes all the raw data from every deposit, loan, treasury, and exposure system for all cash and derivative instruments. It also involves configuring and integrating all the reference data models to give its raw data context.

In other words, literally trillions of rows (and columns) of data must be configured and integrated to build a "ready" Risk Data Asset. This is where Tresata's software comes into play to help profile the data (i.e. identifying missing data, duplicates, and bad data), to determine the appropriate remediation approaches, and to then configure, process and enrich the data to make it "ready."

In our experience, most of the thorniest data issues involve curing reference data and related metadata issues (particularly entity resolution).

The entire purpose is to create new, accurate reference data. To provide context, Tresata routinely creates unique customer IDs that relate all the raw data from source systems properly to all the various legal entities in corporate family structures or real family structures (i.e. all the people being served through a "family office" in private banking.) Some institutions may want to create customized reference data to identify, for example, the nature of country exposures using different definitions of country exposure; or they may want to create oil price risk reference data, using different definitions of oil price exposure; or institutions may want to be able to create reference data to be used to evaluate counterparty risk exposure or credit risk exposures inherent in working with other companies in a supply chain.

Because of the scale of the data being "readied," our recommendation is to undertake this effort in "bite sized" portions rather than configuring and integrating all the data set at one time. For a large G-SIB, this may mean starting with the financial data and reference data for all of the large corporates and individual customers before taking on similar efforts for the Treasury, Private Banking, Credit Card, Branch Based Banking, etc.

Once the data has been configured and integrated, it is "ready" for use.

The final part of this process is to install the software needed to load the ready data into legacy applications and databases. Tresata also offeres software to automate this part of the process.

STEP 4:
LOAD READY DATA INTO LEGACY PLATFORM

Step 4 is to make the institution's existing data aggregation and data management processes Basel 239 compliant.

This step involves loading the "ready" data from the Risk Data Asset into all the legacy reporting systems and into all the existing staging systems used by relevant databases and data warehouses. This also involves the need to correct the relevant reference and master data used by these systems (i.e. all the data in the master file that provides a common point of reference for all data in a data warehouse or reporting system).

Once this is complete, you will have taken major strides in becoming compliant with Basel 239, as you will have:

- Greatly improved data quality
- Created traceability and data lineage
- Improved your ability to analyze

Having said that, at this point you will still have many of the issues inherent in using data warehouses—including the cost.

Thus, you will also need to go through all of the various reporting systems and databases to see which ones can be made redundant now that you are able to easily aggregate "ready" data drawn from a common risk data source. In the process, you can also eliminate or reduce how much data is staged, as well as eliminating or reducing the extraction, duplication, and loading of data through ETL processes. Additionally, you can also look for opportunities to reduce data stored expensively in the data warehouse rather than more cheaply in the Risk Data Asset itself.

Once this effort is complete, you can then do a dramatic rethink of the entire data architecture for your institution to figure out how to make fundamental improvements in its cost effectiveness and capability improvements. For example, you should be able to rethink if you need enterprise data warehouses at all for risk management purposes. You might consider eliminating all the needs for staging and use of relational databases. Rather, you might go to an approach where hundreds of risk applications being used by different, discrete user populations are each loaded directly from the Risk Data Asset.

STEP 5:
IMPROVE ONGOING RISK DATA AGGREGATION & MANAGEMENT

The last step is to use the Risk Data Asset to improve how the institution manages risk. This is the most exciting step.

Building such an asset creates an opportunity to rethink fundamentally how you manage risk.

The first opportunity is to rethink how you can build applications within the Risk Data Asset itself. Rather than extract the data from the Ready Data Asset and move it to applications and databases outside the new data infrastructure (Hadoop cluster, Tresata software), you instead write (and run) new applications within this new infrastructure itself. In other words, you eliminate the need for any ETL. You can now fully automate your risk data processes, for example, this creates the opportunity to fully automate the CCAR "stress testing" process to eliminate semi-manual use of spreadsheets to aggregate the data. In addition, you can greatly improve the ability to do automated "what ifs" on historic data and to use the insights gained to create "what if" scenarios for the future. Such scenario testing can be done for the entire data population. There would be no need to do modeling on small samples or use estimation techniques to extrapolate those insights to the enterprise as a whole. Rather, you can model the whole institution to discover and do "what ifs" directly.

Over time, it is possible to think of entirely new risk processes that could only be done once you have a Risk Data Asset. You can build software to help gain the insights needed to understand "contagion effects," "canary in the coal mine" effects, and other related approaches to create early warning systems that identify future risk issues in plenty of time to mitigate or minimize losses.

You will also be able to finally see the development of much more powerful, more predictive applications that continually analyze your exposures, given real time changes in the global economy, and in socio-political dynamics on your exposures to consumers, companies, financial institutions, and governments. You can envision getting predictions and probability assessments of probable losses at a very granular level (i.e. at the level of individual customers,) using Bayesian statistical modeling or other approaches.

You can pretest alternative business strategies against multiple scenarios before you even decide to adopt a course of action. Once the course is set, you could monitor continuously whether the assumptions you had made were accurate or at variance with the emerging reality.

In other words, the potential to use a Risk Data Asset in the future management of your entire institution, and build competitive advantage, is the ultimate end goal...and that is incredibly exciting!

NOTES

DATA DEFINITIONS

Raw data – data as it comes from source systems

Metadata – data that describes the enterprise information architecture (e.g. definitions of tables and columns) in the system catalog of a database

Reference data – data that is used solely to categorize other data found in a database or for relating data in a database to information that gives that data context (e.g. name, address, gender, race, etc. of an individual)

Clean data – accurate data

Dirty data – data, that if used, leads to data errors (e.g. dirty data can include missing data, wrong data, duplicated data, inconsistent data, etc.)

Consistent data – data that is easy to reconcile

Inconsistent data – data that is hard to reconcile or that requires subjective judgment (i.e. guesstimates) to reconcile

Data integrity – measure of the accuracy or how easily the data can be reconciled

Cleaning data – act of making data more accurate and more consistent

Ready data – clean, consistent data that is ready to be loaded into application systems

Unstructured data – data that has not been put into a data schema that requires use of NoSQL programming

Structured data – data that has been put into a data schema for a database that requires use of SQL programming

Database – mainframe based approach for aggregating data from multiple source systems, with the data organized by consistent schema, in order to provide consistent data to different user populations

Relational data warehouse – data warehouse that combines data across multiple bases or data warehouses using consistent schema

Enterprise data warehouse – a very big relational data warehouse (25 plus terabytes)

Dumping data – data that is discarded irretrievably either by updating or deliberately discarding it

Clipping data – techniques to reduce data volume in a data warehouse without loss of functionality

JARGON

Mainframe based computing – industrial strength machines each of which are made up of a central processing unit (CPU), disc storage of data, and random access memory with processing through a central file systems one transaction at a time (very quickly)

Cluster based computing – multiple, low cost computers (i.e. nodes) that each consist of a CPU, disc storage, and RAM that work together so that data can be processed in parallel through use of a distributed file system

Hadoop ecosystem – an open source system developed by the Apache Software Foundation, to enable cluster computing approaches fit for enterprises (e.g. security features such as "firewalls", logical access control of data, etc.) – includes vendors selling proprietary software

Kilobyte – a thousand bytes

Megabyte – a million bytes

Gigabyte – a billion bytes

Terabyte – a trillion bytes

Petabyte – a quadrillion bytes

Transformation – applying algorithms to the data to make it ready to deliver insight

Sampling – applying algorithms to a "statistically valid" sample to estimate data relationships

Machine learning – transformation of an entire population of data (i.e. no sampling) to understand relationships between the data (including hidden relationships) usually through use of self-learning (i.e. feedback loops)

Predictive – ability to understand likely outcomes by processing all available data despite missing data due to unavailability or uncertainty (e.g. can take incomplete data, such as a single fingerprint and predict person it belongs to)

De-identify – removing private reference data about a person from that person's data

Identity resolution – predicting the identity of a person without any accurate reference data that links the data to the person (used to aggregate data from different sources to relate it to the right person, despite missing or inaccurate reference data)

SQL – a programming language used to manage data through structuring the data into schema

NoSQL – all programming languages used for unstructured data

Incremental – data in a field that is updated (old data is dumped)

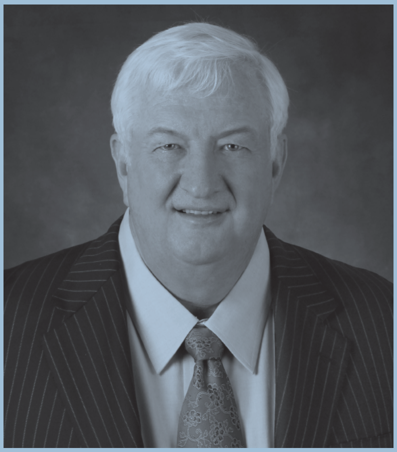
Persistent – data that is always retained

Database schema – a set of formulas, called integrity constraints, within the database, that ensure compatibility of the data in the database

DBMS – DataBase Management System that organizes data into rows and columns in a single database through use of database schema

RDBMS – Relational DataBase Management System that organizes database tables so that data defined in multiple databases can be analyzed together through consistent schema

ABOUT **LOWELL BRYAN**



Lowell Bryan has served as an adviser to the top management of major financial institutions and corporations for forty years. He is a Director Emeritus of McKinsey & Co. where he helped found, and lead, its Global Financial Institution Practice and its Strategy Practice. Since he left McKinsey in 2012, he has continued to provide counseling services to clients through his company, LL Bryan Advisory, Inc. He is currently undertaking a research project focused on understanding how financial institutions should aggregate and manage data in the Digital Age.

Mr. Bryan is a Strategic Advisor, Investor, and Director of Tresata. He also serves as a lead director of DST, a public company and is a director and investor of HwC, a private, online, food company. He has authored six books on a variety of subjects (i.e. banking, capital markets, strategy, regulation) and has written extensively for a variety of publications (including dozens of articles and editorials for the McKinsey Quarterly, Wall Street Journal, and Harvard Business Review).

ABOUT **ABHISHEK MEHTA**

Abhishek Mehta is the CEO & Co-founder of Tresata, a predictive intelligence software company that in a short span of 4 years, he has built into one of the most innovative big data companies in the world.

Abhishek is recognized as one of the most influential thinkers, visionaries, and practitioners in the world of Big Data. His history is a rich combination of stints as a radical technology expert and a practical, in-the-trenches business leader. His experience includes Executive in Residence at MIT Media Lab, Managing Director at Bank of America, and various leadership positions at Cognizant Technology Solutions and Arthur Andersen.

A passionate supporter of entrepreneurship in the Southeast, Abhishek has been included in numerous lists of the top innovators, leaders, and disruptors of our generation. He is a highly sought after speaker on the topics of big data analytics, emerging business models, and all customary intersections of the two.

ABOUT **TRESATA**

Tresata builds next-generation predictive analytics software that enables businesses to **monetize big data™.** Tresata's Customer Intelligence Management software automates complex human processes in the areas of Identity Intelligence, Marketing Intelligence and Risk Intelligence. These automated applications allow businesses to understand customer behavior, and deliver products and services personalized for a segment of one.

www.tresata.com

FOR MORE INFORMATION VISIT **TRESATA.COM/FS** OR CONTACT **CURIOUS@TRESATA.COM**

MONETIZE BIG DATA™