Becoming Data-Driven

In order to compete in the new economy, organizations need to transform their business model and very culture to be data-driven or risk becoming obsolete and perish. Join us as we examine the journey to becoming data-driven from several angles: Business Culture, Technology, and Governance.

About the Authors

Eliud Polanco

Eliud Polanco is a seasoned data executive with extensive experience in leading global enterprise data transformation and management initiatives. Prior to his current role as President of Fluree, a data collaboration and transformation company, Eliud was formerly the Head of Analytics at Scotiabank, Global Head of Analytics and Big Data at HSBC, head of Anti-Financial Crime Technology Architecture for U.S.DeutscheBank, and Head of Data Innovation @ Citi.

Julia Bardmesser

Julia Bardmesser is a technology, architecture and data strategy executive most recently with Voya Financial. Julia has led transformational initiatives in many financial services companies such as Voya Financial, Deutsche Bank Citi, FINRA, Freddie Mac, and others.She is also a founding member of Women Leaders in Data and AI (WLDA). At Voya Financial Julia served as Senior Vice President, Head of Data, Architecture and CRM.

Peter Serenita

Peter Serenita is the Chairman of the Enterprise Data Management Council. One of the first Chief Data Officers (CDOs) in financial services, the 28-year JPMorgan veteran held several key company positions in business and information technology, including Chief Data Officer of the Worldwide Securities division. More recently, Peter was the Enterprise Chief Data Officer for Scotiabank.

Table of Contents

Chapter #1

The Data Driven Organization

This is the first in a series of white papers that explains how innovative and successful organizations transform from data nascent organizations into data-driven organizations, outperforming competitors in the process. When we say 'Data-Driven,' we don't mean companies that simply run management reports, perform analytics or even implement complex statistical or self-calibrating machine learning models. We mean companies that have completely re-designed their business processes around sharing and using data as a fundamental part of their business strategy. They have fundamentally transformed their culture, technology and data governance procedures to leverage as much internal and external data as possible, even the bits they didn't originally think were going to be of value. Data becomes an enterprise asset, the centrifugal force around sustainable competitive advantage. So what does this look like to the company and its customers? Let's explore the customer angle. For customers it will feel like the company knows their needs and provides solutions and products that fit those needs exactly. I am sure, as a customer, you can think of companies that do this better than others. Which companies provide you with selections and choices that closely (or exactly) match your needs? A simple example would be Netflix. Prior to Netflix, you might go to a video store like Blockbuster (now I am dating myself). At the video store, you would search for titles that interested you or you heard about. The store didn't have any suggestions or recommendations based on them 'knowing you' that would guide you to titles you might be interested in. Netflix changed that. With data about you, the data about the titles and the data about other peoples' choices, Netflix is able to provide you with recommendations which you may have never considered. And the success of those recommendations feeds back into the data to continuously improve the product (e.g. the recommendations). This is just one rather simple example. I am sure you can come up with others based on your own experiences of how data driven organizations have changed (and hopefully improved) the customer experience. What do these companies have in common? They have access to a vast and diverse set of quality data and they harness this data as an enterprise (and valued) asset to drive their business (and their competitive advantage).

Over the course of this series of articles, we will explore many of the critical dimensions of the Data-Driven transformation. What does it mean to transform? Why do it? What technologies enable this transformation, how does this affect the governance of data, and where do you even begin?

In this first paper, we will explore the business reasons and implications for a shift to a data-driven organization. Topics will include:

1. Why you need to care about transforming into a data-driven organization.
2. The characteristics of a data-driven organization.
3. The business changes and cultural changes required.
4. Implications for technology.

The next paper in the series will describe the new technologies and data architecture patterns that have emerged to make transformation possible. The third paper will provide a point of view on what Data-Driven transformation means for Data Governance, and how managing data as a collaborative, shared asset will fundamentally change how we think about ownership and control. And finally, the series will end with a discussion around how to begin the transformation—how to sell it within the organization, finance it and take the first step.

Section 1. Why Become a ‘Data-Driven’ Organization?

Business dynamics are rapidly changing. There will be winners and losers. New methods are replacing the traditional ways of interacting with your customers, your vendors, your suppliers and even your own company and products. Data is at the core of this transition.

If you are already convinced that you need to become a data-driven organization, then feel free to move to the next section. If not, let’s take an example from the banking industry, where the dynamics of how we interact with customers have drastically changed. Traditionally, you would know your customer because they would come into the branch. The customer would know the teller and maybe even the branch manager. The teller would know the customer. They would strike up casual conversations, which would yield important information to help the bank service the customer. For instance, the bank might find that the customer has an imbalance between the dates they get paid and their bills. Maybe a credit line would help. They may find that a child will be attending college soon, so maybe a loan may be necessary.

Interactions with customers today are very different. Customers may on occasion still visit a branch, but they also interact with the bank digitally. Banks have better data as it relates to customers’ transactions, accounts and balances, both currently and in the past. Banks can also access additional customer information that they can use to better understand the customer’s needs. These may include products that they have looked at but not transacted, life events, relationships inside and outside the family, and so on. The list is practically endless and ever-expanding. The more data the organization has about the customer, the economic environment such as interest rates, world events, and so on, the better informed the organization is, and the better it can service the customer. In the past, the bank would segment customers and service each segment with certain products, but a data-driven organization no longer needs to lump customers into segments. Instead, it can be customized to each individual customer. This is sometimes called a ‘segment of one.’ In addition to providing the customer with better-suited and potentially more products, such data processes also lead to improved customer retention.

That is just one example of how to know your customers better and therefore understand, and potentially even anticipate their needs. But the value of data doesn’t stop with customers. It actually infiltrates all aspects of the business: products, locations, employees, and so on. The organization can better understand how products are performing and make adjustments to better fit customer needs. The business can not only recommend more suitable products, but potentially customize these products to

better fit each customer's need, assuming operational systems can support these customizations.

The list of benefits of a data-driven organization are nearly countless, so I will just provide one more potentially obvious area. Data-driven organizations are able to bring the whole firm to the customer. What does that mean? In most cases today, organizations tend to be siloed by business line and/or geography. The ability to offer the customer the full set of products and services across the entire enterprise is very difficult, if not impossible. Data-driven organizations can provide their customers the full range of products and services irrespective of business line or geography, as regulatory laws permit. Business and geography silos are eliminated.

Hopefully you will agree that becoming a data-driven organization needs to be a core focus. If you are not yet convinced, I will leave this section with one more thought. Competitive differentiation used to come from being able to execute your business processes better or more cost efficiently, whether through the leverage of globalization, automation or digitization. But that is now basic table stakes. Out-executing your competition can only achieve ever-diminishing marginal returns. To gain a consistent and sustainable competitive advantage, the winners in this new environment will outsmart their competitors through maximum leverage of data. Your competitors will be moving toward this objective. Will you be able to compete in the new economy with a traditional business model when your competitors are leveraging the value of data for their customers and businesses? Not to be overly melodramatic, but it can mean the difference between survival and prospering.

Section 2. What are the Characteristics of a Data Driven Organization?

The case put forward in the previous section must have some merit, because most organizations are already on their journey from a traditional business model to becoming data-driven organizations. Few, however, have fully achieved this transformation. Even the ones who have claimed victory, while further along than their competitors, haven't attained their goal. There is actually a continuum that you can map each organization against. To simplify it, I have divided the continuum into four significant phases as shown in Figure 1.

Data-Aware (Function focused)	Data-Informed	Data-Enabled	Data-Driven
Data is created as a by-product of a process/function.	Data is created as a byproduct of a process/function.	Data is created as a byproduct of a process/function.	Data is the product and business functions organize around the data.
Data is siloed into each function.	Data is siloed into each function but shared on a bespoke basis into business warehouses.	Data is siloed into each function but consolidated into enterprise warehouses and data lakes.	Data is freely shared and available across the organization.
Data is input and used in each function separately.	Data is transformed and moved to each function (warehouse).	Data is transformed and moved to each function (lake).	The function is moved to the data (i.e. one copy of the data which all access).
Data is on a "need to know basis."	Data is on a "need to know basis."	Data is on a "need to use" basis.	Data is defaulted to "always share" basis.
Data primarily used for operational purposes.	Data is used to inform decisions.	Data used to enable decisions.	Data used to drive behavior and products.

Figure 1. The evolution of data adoption and readiness inside institutions.

Most mature organizations were initially focused on business processes (the Data Aware phase). Their main concern was to standardize and automate these processes to improve

control and efficiency. This resulted in the creation of business (process) silos and resulted in data being locked into those silos with significant (bespoke) effort required to provision data across the organization or to other business functions. The move from Data-Aware to Data-Informed started with the introduction of Management Information (MI) and Business Information (BI) tools. Multiple data warehouses were created to support different uses of the data, usually by different groups, such as risk, finance, etc. Data was moved and transformed for each data warehouse or use case.

Then, over the last 10 years, with the advent of Big Data and a heavy emphasis on analytics, many organizations moved forward on the continuum from Data-Informed to Data-Enabled. Global data lakes were created and analytics were used to enable some capabilities for the business and, in some cases, the customer. While this was an improvement over the data warehouses, as described before, it is no longer sufficient as it still requires copies (sometimes multiple copies) of the data with specialized ETL (Extract, Transform, Load). The winners will take the extra step to become Data-Driven.

So what does a data-driven organization look like? Simply put, data-driven organizations have easy access to high quality data across the entire organization and utilize nearly all of it as part of their normal business processes. With that tenet in mind, there are four characteristics that distinguish a data-driven organization. They can:

1. **Find the Data.** The data you are looking for is easy to find, usually through a robust data catalog.
2. **Know what it means.** It is clear what the data represents or means, usually through a robust data dictionary.
3. **Know the quality.** The data is of known quality and can be trusted
4. **Get access.** Access to the data is quick and seamless to any region, product or function. Data is also stored securely and provisioned in line with legal and regulatory restrictions.

Some might think that this sounds relatively straightforward and easy to accomplish, but there are two important transformations required: a business transformation and a technology transformation. We will examine each of them next (and in more detail in subsequent papers). Mature companies in particular need to shift their business, processes, technology and people—their very culture. New companies, on the other hand, can start their business without the baggage of the past. Hence we have seen some really impressive results from some of these smaller, newer companies that large mature organizations are trying to mimic.

Section 3. Business Transformation Required

One of the keys to transformation in a large, diversified organization is a shift in each line of business's model. Every business in an organization needs to run itself, meaning it needs to be accountable for its own strategy, revenues, expenses, and so on. However, it can't run in isolation. It needs to operate in the context of the enterprise. If it does not, it may as well be a separate company, as it is not realizing any benefit of the larger, more diversified organization. Thus the business needs to share both its processes and data with the rest of the organization, as well as consume processes and data from the rest of the organization. We move from a producer or consumer model to a model where everyone is both a producer and consumer. This will stretch the organizational boundaries of trust and control. The companies that succeed will reap the rewards.

The recognition of the whole is greater than the sum of its parts requires that all data naturally be shared, meaning that the default is to openly share data across the enterprise. The only exception would be due to privacy or data protection concerns, but there are solutions for this too that will be discussed in the second paper. Cross-organizational sharing will reduce the friction that currently exists when a data consumer in one part of the enterprise needs to access data produced by another part of the enterprise. Business leaders will have easy access to all data across the enterprise to include in their business plans and customer acquisition and retention strategies. This will allow these business leaders to leverage the strength of the entire enterprise (e.g a macro-view) as they develop and execute their business strategy and not be limited to the micro-view within their business.

That is harder than it sounds. The initial business model of most organizations was set up with a product or location focus. Teams developed processes and systems specific to this focus. As organizations evolved, or matured, they had wider business objectives to solve (e.g. multi-product, multi-location). Prior solutions were not set up for this new, broader objective. For example, in the early days of financial services, a bank's primary objective would have been to optimize individual products. But as time marched on, the emphasis shifted to offer customers additional products or an entire portfolio of products. Data processes and systems were not optimized for this. Because data was not widely shared, cross-product capabilities required significant bespoke effort as the data could not easily be found and utilized. The result is the business and data silos that we see in most organizations today.

The idea of data sharing by default is not only a problem of silos but it is also a cultural problem. Most think that data is not to be shared unless explicitly required. Accordingly, we have built mechanisms to make it difficult to share data. This is because the business

model, and the architecture built around it, identifies the business and/or function as the owner of the data, rather than the enterprise. If the enterprise was the owner of the data, then the default emphasis would be on sharing data enterprise-wide. Mechanisms would make it easy, rather than hard, to share data (and yes, we will have the ability to limit sharing of sensitive data—stay tuned).

Another important thing to discuss is how people become data capable. As we move toward a data-driven organization, data becomes an essential driver of the business. Ultimately, data becomes the business. We need to ensure that our people have data skills. Just as people need business skills to be successful, they must now also be informed, active contributors and consumers of data. They will have new responsibilities like ensuring that their data is well-defined in a data dictionary, is of good quality and is easily accessible. They must be active participants in the exchange of data, with the enterprise as both receiver and provider. I could go deeper into the evolving role of the CDO, their responsibilities in the data-driven organization and what the data organization would look like, but I will leave that for the third paper.

In reading this paper, some may think that with my focus on the enterprise, that I am advocating for a centralized business- and data organization. The contrary is true. I have found that driving from the center, or the enterprise, is very difficult and usually ends in failure. In fact, it needs to be a federated model where every part of the organization does their part for the whole. Think of it as crowdsourcing or a federated organization, where each part of the organization has their areas of expertise (responsible for its processes, data, costs, etc.), is an active member of the enterprise and functions according to rules defined by the enterprise. In this way, each part operates effectively and efficiently as an individual business, but also provides capability and data to the rest of the enterprise. This will ensure that each part of the organization is focused on its own area of expertise, but also provides the collaboration required to share, and not replicate, data across the enterprise. The impact on the Data Governance processes and organization will be explored in the third paper.

One final point that I want to make is that all of this is reliant on the corporations building trust with their customers. Customers need to feel confident that the company secures their data, respects their privacy and is using their data for their benefit. Clearly this requires sound data security capabilities (including numerous technics, e.g. masking, anonymization, etc.), clear ‘contacts’ (terms of business) with the customer on how their data will be used and a data ethics program to ensure that all data is used within the guidelines that the customer would be comfortable with. In short, if a corporation is data-driven, it needs to maintain a good relationship (trust) with the data providers (e.g. customers, et al).

Section 4. Technology Transformation

In this section I wanted to lightly touch on the technology transformation required to become a data-driven organization. We will go into more detail in the second paper, so I will just introduce the topic here. In the new economy, technology is usually a differentiating factor towards becoming a data-driven organization, meaning a technology transformation is required.

Most systems were initially developed with a focus on function, such as a trading application, a confirmation system, or a settlement system. Each of these managed the data necessary to perform its own specific function. When a sequence of functions needed to be strung together to complete a process, data was sent from function to function—system to system—leading to the function-based architectures we see in most financial services firms today. The emphasis was on the function and hence we needed to move the data to the function. Systems became optimized, but data exchanges became duplicated and complex. While some functions have moved to a pub/sub or API-based mechanisms, the focus is still on the function as the center of the architecture. We will talk more about this later.

Next comes the move from a function-first architecture to a data-first architecture. As stated earlier, the priority of our infrastructure has been the function and the data has been a consequence of the function. The function produces and uses the data. If we flip to a data-first architecture, then the data is the primary focus and the functions are processes that act on the data. This may seem somewhat subtle. One way to think about it is that the data is permanent, and the function is a temporal process that has a beginning and an end. Inverting the focus significantly changes the way we build systems and treat data. Instead of functions owning data and moving the data around from system to system, we would leave the data in place and have the functions act on the same data. Another way of saying this is, “Instead of moving the data to the function, move the function to the data”. This would reduce or eliminate the number of copies of data while making data sharing the bedrock of the architecture. We would now optimize around the data, and not the function as was previously done. In the next paper we will look a little deeper into data-centric technology architecture.

Lastly, I wanted to spend a little time talking about the technology toolset, which will be explored in more detail in the second paper. New tools are evolving that will be essential for the move to a seamless data-driven organization. One of the bedrocks of the capabilities is the ability to actively manage metadata and bridge the gap between business metadata, for example through a data dictionary, and technical metadata, for example the data catalog. That is to build a bridge between the way the business knows

and uses data to the way the data exists in the technology. Once we are able to actively and seamlessly manage the metadata, all of the tools will ‘sit on top’ of this foundation. The data across the enterprise will be opened up to the enterprise as a critical component of its strategy, enabling it to finally become a data-driven organization. This data-driven business model will be ‘data hungry’. Those that are successful at this transformation will continually require more and more data (different types of data, from different sources) so the key will be to reduce the amount of time it takes to acquire data and push it through a process to identify it, assess it, catalog it and make it available. The only way to do this is through automation of the full end-to-end data process (i.e. a robust data architecture and toolset).

Summary

As you have read throughout this paper, the move to a data-driven organization requires a shift of business model and a new set of responsibilities. Each business must act as part of a larger organization. In doing so, it also benefits from the other businesses across the enterprise that also play by the same rules. Every business will benefit from enterprise-wide data for its own customer purposes. This will more than outweigh the effort required. While a data-driven organization can be accomplished, and benefits realized, at the business level as opposed to the enterprise level, if accomplished across the enterprise, the whole organization will experience an exponential uplift. Customers will see and feel the difference. Just imagine the customer experience if a business is able to fully understand the entirety of a customer's activities, interactions, experiences and aspirations without the friction of silos and provide the customer seamless access to all of the organization's products across all of its lines of businesses globally. Why does the customer have to bear the friction caused by these artificial barriers? The organization that conquers and breaks down these silos will reap the benefits.

So, where to begin? In our next paper we discuss the enabling technology that makes data-driven transformation possible. While it can get a little technical, it will describe some of the terminology and jargon behind the innovations in data processing, as well as describe what it is about existing data technology that traps data and keeps it from being fully leveraged in the company.

Chapter #2

The Architecture Of a Data-Driven World

In our previous paper, we introduced the concept of **data-driven organizations**. Such firms are capable of outsmarting the competition by leveraging their own internal proprietary and third-party external data to a degree that their peers cannot. They accomplished this by purposefully creating, collecting and using data **at the center** of their business process and technology architecture design. We believe that “data-first” is the new business and technology imperative of our era.

Becoming data-driven requires significant changes to the internal business culture and social contracts between the business functions that produce and consume data. It also requires breaking down foundational barriers inherent in the technology architecture that unexpectedly trap data and keep it from being fully leveraged. As we will see, the technologies that store and process data have not historically made it easy for companies to securely share data both internally and with external partners, suppliers or customers. This is because data sharing requires continuously copying, translating and transforming data across many contexts of use. This makes data overall harder to manage and protect from leakage and cyberthreats. To foster a culture where the default habit is to share data, an entirely new architecture pattern will be needed. This pattern must both be more secure than existing technologies and reduce the need for massive copying.

In this paper, we will:

1. Describe what a data-centric technology architecture looks like.
2. Describe how it is different from, or similar to, existing data processing technologies.

The third and fourth papers in the series will close with a discussion around data governance and how to practically adopt a data-centric architecture, especially in light of the legacy footprint of existing data processing technologies.

Section 1. What does a Data-Centric Technology Architecture Look Like?

Data processing- and storage technologies and architecture patterns have historically evolved to keep pace with changes to business models, as well as cultural attitudes around how data can and should be used. In the beginning, all business logic and data processing could be performed in a single integrated computer, the mainframe. As businesses evolved to become more diversified, componentized and complex, new technologies such as the database and the warehouse were invented. With each continuing major transformative business shift came a new generation of data processing technologies and architectures. These both enabled new business capabilities and fixed the unintended consequences or perceived shortcomings of the previous generation. In the same fashion, the transformation towards becoming data-driven necessitates innovations in data processing technology. We are now witnessing the evolution of data processing technology into its fourth generation since the inception of the modern computer. We summarize the four generations below:

	1st Generation	2nd Generation	3rd Generation	4th Generation
Business Model Innovation	Introduction of business computing	Business function digitization at scale	Big data and analytics to optimize business functions	Data centered business model
Generation Start	1960s	1980s	2000s	2020s
Data Attitude	Data-Aware	Data-Informed	Data-Enabled	Data-Driven
Architecture Pattern	Computer-Centric: “Bring the function to the computer”	Application-Centric: “Bring the data to the function”	Analytic-Centric: “Bring the analytics to the data”	Data-Centric: “Bring the function to the data”
Data Technology Innovation	Mainframe	Relational Database Management System (RDBMS), Business Data Warehouse, Business Data Mart	Enterprise Data Warehouse, Data Lake, NoSQL Database, Graph Database	Collaborative Database

Figure 1. A summary of the four major evolutions of data processing technology over the last 80+ years. With the rise of data-driven organization, and business models centered around data, a fourth and new generation of data patterns and technology has emerged.

What actually *is* the fourth generation that we call data-centric architecture? How is it different from prior generations? What does it do that wasn't possible in generations before? Let's take a quick look back at the history of how computer-, application- and analytic-centric models have built on one another, and use that trajectory to characterize the major trends that will drive innovation in the fourth generation.

1st Generation (1960s): Mainframe

As mentioned above, data arose out of the first commercially viable enterprise computing platform, the mainframe. Businesses brought their functions to the mainframe, which in turn hosted the application logic/programs, compute, storage and data within a vertically integrated and tightly coupled stack. Because mainframes were a very large capital investment, the platform was usually shared across multiple business functions to fully amortize the cost. The sales team could run a mainframe program to list new customer accounts and another program to provide MIS reports around how many new sales were captured over the previous month. Simultaneously, the order fulfillment team could run its own independent programs to process transactions and generate reports within the same machine. Figure 2 displays a simple conceptual model of what the mainframe architecture pattern looked like.

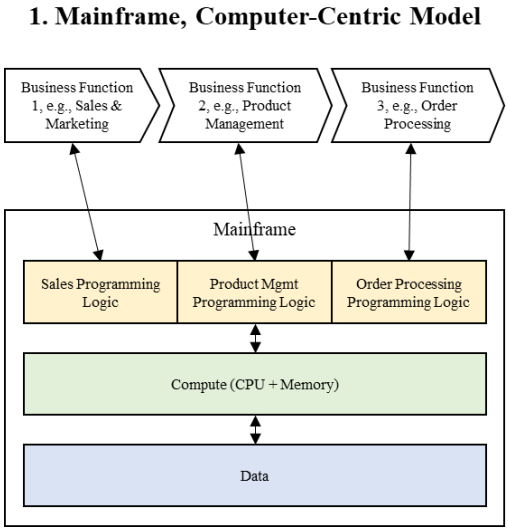

Figure 2. The computer-centric architecture pattern. Functions share one common technology platform, and the programming logic, compute and data are all vertically integrated inside the one machine.

Having the logic, compute and data integrated neatly in one place made mainframes incredibly optimized, secure and blazingly fast. Many organizations still use mainframes for high-speed and complex transactional processing. However, there were two major

limitations to the mainframe:

(1) Mainframes stored and encoded data in a way that was highly optimized for technical programs, but were otherwise inscrutable for everyday users. There was no intuitive model for representing the data. For example, a salesperson could register a new customer account by logging into the program 'W3TC' and entering the code '99' in the field 'AVXN' on screen '03.' While this was a highly efficient way for the mainframe program to store and retrieve data, it also trapped the data into the context of the program, which made it nearly impossible to share with others who weren't familiar with the program.

(2) Mainframes by design required finite technical resources (i.e., compute cycles, memory and storage) to be shared by multiple business functions. Very often, programs would sit in a queue waiting for one function to complete their processes before another could run. This created competition and friction across the functions, which wasn't easily resolvable by simply buying more mainframes. The capital investment required would be cost-prohibitive.

As business executives started to realize how information technology and automation could create value, it became a business model imperative to digitize as many business functions as possible. This would create unmanageable stress on the shared mainframe model. Something new was needed to make business function optimization at enterprise scale a reality.

2nd Generation (1980s): Distributed computing, relational database management systems (RDBMS) and business data warehouses

As businesses analyzed how to increase profitability, they began decomposing big monolithic processes into more discrete business components. Each component could be individually optimized for a peak level of productivity. Specialized enterprise business software and applications emerged in the marketplace to accelerate function-specific digitization and automation. Such business software could run its own programs on its own technology using equipment much cheaper than the mainframe,[1] and without having to share resources with any other applications or functions. In addition, this generation introduced the ability to save data in a model that represented the business function.

[1] And in the future, the business will be able to 'rent' Software-as-a-Service without buying any capital equipment at all!

Instead of locking data in fixed codes tied to the program, it became possible to define business subject areas, attributes and relationships using a vocabulary natural to the function, and then save the data in tables using matching descriptive labels. It was further possible to continuously customize and extend those definitions as needed through simple configuration changes to the underlying data model. The end result was that each business function would define its own custom data model required to operate their processes. The business function would then store its model inside a dedicated relational database management system (RDBMS) linked to an application.

As companies evolved to become more open to using data to inform business decisions (i.e. “data-informed”), the need emerged for the business functions to analyze and introspect their data beyond its original operational context of use. However, because the RDBMS was mostly focused on saving and querying data for operational use only, a separate system would be needed to archive historical data and analyze it. Analytic processing workloads (i.e. OLAP) typically require more memory and CPU in order to introspect larger volumes of data, from longer time horizons, compared to normal operational transaction processing (i.e. OLTP). Learning from the painful experience of the mainframe generation, the business functions did not want to have OLAP queries and workloads compete with the business-critical OLTP workloads for the same fixed compute resources inside of the RDBMS. The concept of dedicated business data warehouses was introduced to avoid such competition. Each data warehouse contained its own CPU, memory, data storage and data model types (e.g., star schemas and dimensional models) packaged in a configuration that was optimally designed for executing the more resource-intensive OLAP queries. This combination became the foundation of the application-centric pattern, which distributed compute, memory, storage and data across the multiple operational and analytic systems inside of each business function.

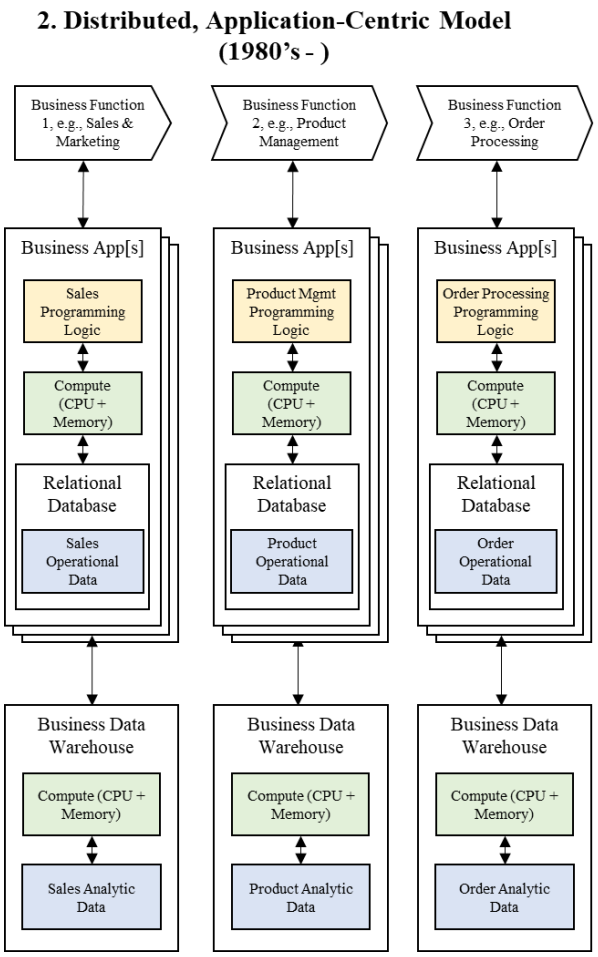

Figure 3. The application-centric architecture pattern. Business functions independently purchase one or more business applications. Each business application comes with its own resources to run programming logic, and saves its operational data in its own data model within a dedicated relational database. To enable data analysis inside a function, data is copied, archived and integrated inside business data warehouses.

This model was initially tremendously successful, and along with further innovations in computing, storage and networking contributed to an era of unprecedented business productivity. However, new companies born in the internet era realized that having a complete view of data across many functions end-to-end was tremendously valuable. They used this intelligence to design innovative products and deliver them via novel sales and service channels, which enabled them to begin capturing and retaining sizable market share over incumbents. Furthermore, savvy bad actors realized that many companies had information blind spots, and that they could exploit intelligence gaps across functions to commit fraud or launder money. The inability for many large companies to respond to these threats exposed weaknesses in the application-centric model:

(1) Offering each business function full control over their own technology made the individual function agile and flexible, but at the cost of creating data towers or data silos

inside companies. Data was created with the sole purpose to operate the function; it was 'owned' by that function's business application, for the primary use by that application, and saved in a data model that was best understood by the application. Business processes that required access to data across functions and systems (such as upsell/cross-sell marketing programs or corporate functions like compliance, risk, finance and human resources) needed to reach out to each application owner individually to extract copies of their data and then stitch them together into their own data models. Only then could data be processed by their applications. In other words, you had to continually copy and bring the data to each function. This made it very difficult and cumbersome to understand what was happening holistically inside a company.

(2) Distributed computing made managing data **really, really complex**. Before, data used to sit in a few mainframe machines in one big room in a data center. Now, data could be in any of hundreds, if not thousands, of servers across multiple data centers. Not only did this make protecting data from unauthorized access and use more difficult, it also meant that data governance was imperative. Investment was required in people, processes and systems to manage data. These included inventory catalogs, metadata management systems for both business metadata (the data dictionaries or glossaries that provide semantic context to what the data means) and technical metadata (the physical information about how the data is being saved in the data store); reference data systems, master data management systems, data quality systems, data lineage management systems … and on and on. Without these capabilities—and unfortunately, even with them—it was nearly impossible to know what data even existed, let alone what it meant, where it came from, what its quality was and whether it could be trusted for use by others. As data was copied and transformed from one function to the next, poor data began spreading virally across enterprises. This, in turn, reduced trust within the organization around using data for competitive differentiation. To stop the spread of bad data, a business had to prioritize which data was most important, also known as finding its critical data elements, and then devote its entire focus to managing and cleaning them. Meanwhile, new entrants were using data to grab market share by the bushel, and somehow didn't seem to be encumbered by the same challenges. Clearly, they were doing something different.

3rd Generation (2000s): Enterprise Analytics: Enterprise Data Warehouses, Big Data, NoSQL and Data Lakes

Whereas the purpose of data in the application-centric model was to optimize the business function, the next stage of evolution for data-enabled companies emphasized the analysis of end-to-end data that came from many business functions. The new generation of data processing technology was largely influenced by the rise of the

internet. Structured data generated as a byproduct of application transaction processing now lived side by side with unstructured data, which included user-generated content—HTML pages, blog posts, image uploads, etc.— as well as other digitized forms of documents, such as PDFs, Word documents and spreadsheets. Digitization created vast volumes of both structured and unstructured data. These technologies needed to store data, bring it together and run analytical models to compute it all. Multiple terms were invented to describe and market these data innovations: enterprise warehouse, big data, Not only SQL (NoSQL) databases, enterprise data lakes, data lakehouses, data oceans, data fabric. The core premise was that you could copy and save data from multiple source databases and warehouses. You could then combine and integrate them, even though they came in different forms and models. You would save them inside of platforms with a configuration of compute, memory and storage that was optimized for even more complex and resource-intensive analytics over larger data volumes than ever before. Then, finally, you would use new programming languages to build natural language processing-, statistical-, and machine-learning models on top of the data to mine it and uncover insights that likely would not have otherwise been found. Those insights would then be brought back to the business functions.

This generation also included new technologies to save semantic and conceptual data relationships so that (mostly) unstructured data could be easily interpreted and shared by both people and machines. Rather than saving data in a relational table structure, which stores data in matrices made up of row-and-column relationships (or two-dimensional tuples), graph databases emerged as a way to save more complex semantic relationships, such as saving subject, predicates and objects in a three-dimensional or triple model. The linking of conceptual and semantic dictionaries to the data gave rise to the concept of knowledge graphs, a new way for businesses to save information about their functions and processes. Business functions now had multiple options for serving data out to consumption and analytics. They could save data in relational tables inside a data lakehouse and/or in knowledge graph triple stores for semantic querying.

The analytic-centric pattern evolved data architectures into something that looks like Figure 4.

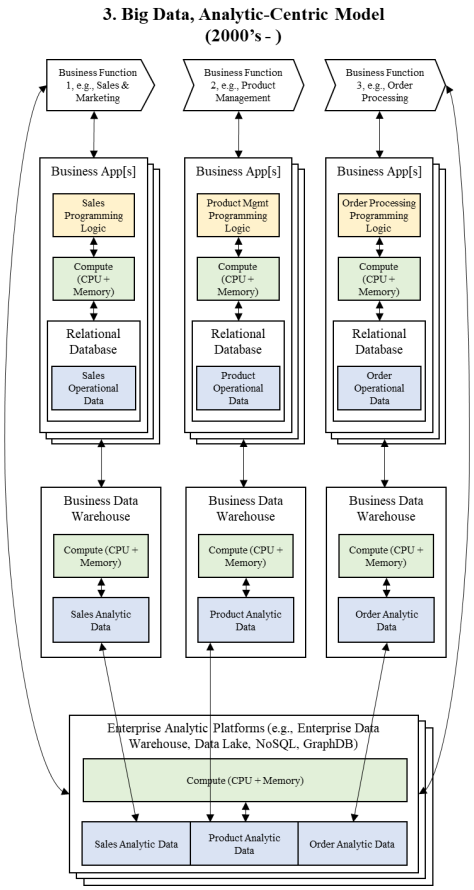

Figure 4. The analytic-centric architecture pattern. Data is copied from source systems and business warehouses into enterprise analytic platforms which are used to mine it for insights to be fed back to the business functions. These new platforms can save the data in relational (tuple) or semantic (triple) stores.

Analytic technologies further evolved with the advent of cloud computing. Cloud economies of scale allowed data platforms to be stored and run more effectively and efficiently compared to a data center. This opened up the opportunity for companies with any size IT budget to invest in enterprise analytic capabilities.

While cloud computing enabled users to more easily integrate and analyze data across business functions, it didn't fundamentally resolve all of the issues inherent in the application-centric model. Worse yet, the application-centric model introduced more issues:

- It relied on copying and transforming data even more than before, exacerbating the complexity of data management and trust issues around data integrity and quality.
- It offered a way to work around, but not fundamentally break down, data silos.
- It weakened data protections even further, because new platforms were designed with centralization and ease of integration in mind, not security.

Because it was easy for businesses to stand up data lakes, many large enterprises ended up with multiple data warehouses and lakes. Some were inside of data centers in private clouds, others on multiple different public clouds. The ultimate result of the transition to the analytic-centric model was a mixed bag. Even though businesses continued to invest heavily in data, it conversely became even harder to find data, understand what it meant, know its quality or enable access to high-quality data. Becoming data-driven would require a whole new technology architecture pattern.

4th Generation (2020s): Collaborative Database

Here, we finally arrive at data-centric architecture, where the primary resource is the data and everything else revolves around it. In other words, we bring the function, which is temporal, to the data, which is permanent. This differs from the previous architectures where the primary resource was the function, and you brought the data to the function—which inevitably led to creating copies and copies of data as more functions required access to the other's data. Looking at how prior generations of data technology saved and processed data informs how 4th generation technology should ideally work. This new generation should be able to:

- Simplify and consolidate the data ecosystem like the mainframe model did, by going back to a single copy of source data in one 'system.'
- Still give a business the flexibility to define the data model so that the data is interpretable by the function outside of the context of the program, like the distributed computing model did.
- Give the business the ability to control its own IT resources (CPU, compute, memory, storage) without having to share.
- Enable functions across the enterprise to access and analyze transactional and semantic data in the same place like big data technology did.
- Also make data more secure and resistant to cyberthreats.

We call the data innovation that meets these principles and enables data-centric architecture **Collaborative Databases**. These data stores comprise a consolidated platform that multiple business applications read and write into, but with security and privacy built in. This same system can also be used to run analytic processes for data at a massive scale without requiring additional copies to be made. Further, this architecture, which puts data in the center, can enable sharing both inside and outside of organizations. The architecture pattern gets simplified to the design in Figure 5.

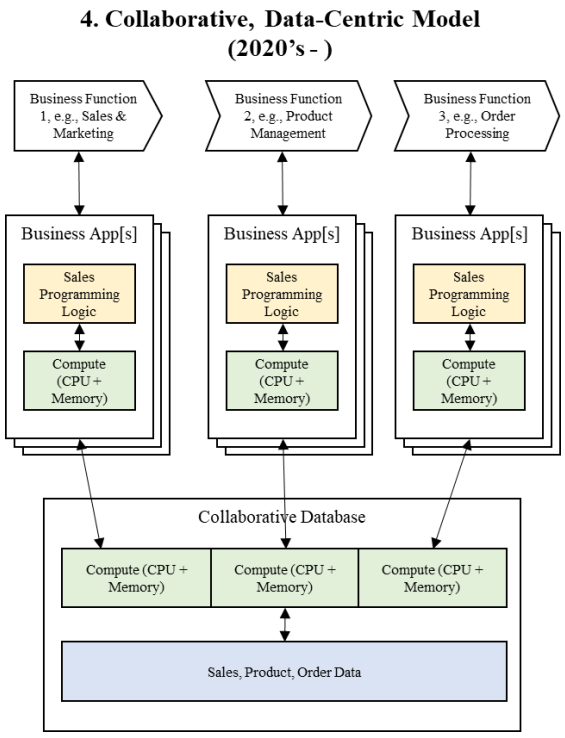

Figure 5. The data-centric architecture pattern, which builds from prior generations. Data is pulled out of the application layer and saved in one place. All business functions can access data for both operational and analytic processing using their defined data models. But, like the distributed computing pattern, functions can still share data without needing to compete for CPU or memory resources. This common collaborative data store can further be extended to functions outside of the company, such as partners, suppliers or customers.

The ultimate business objective behind data-centric architecture is twofold. One, data-centric architecture establishes a business culture in which any business function can safely and securely consume any other internal or external data, whether coming from another function, a supplier or a customer. Secondly, data-centric architecture dramatically simplifies the data environment and tears down any data silos that were the unintended consequence of the prior generation's function-centric focus. Adoption of data-centric architecture must inevitably be followed by the gradual consolidation and retirement of the many redundant copycat data stores implemented from past generations.

Before we get there, we need to explain how this technology actually works. What is a collaborative database, anyway? How will it fundamentally change how data is stored, processed and shared compared to relational databases and analytics platforms? There are a series of technology innovations, especially around Web 3.0 principles of decentralized data storage, interoperability and control, that provide the foundation for this fourth generation of data processing technology. In the next section, we will cover the eight most interesting innovations that differentiate collaborative databases and make them the necessary engine to power organizations looking to migrate from data-informed or data-enabled to data-driven.

The Evolution of Data Technology Architectures

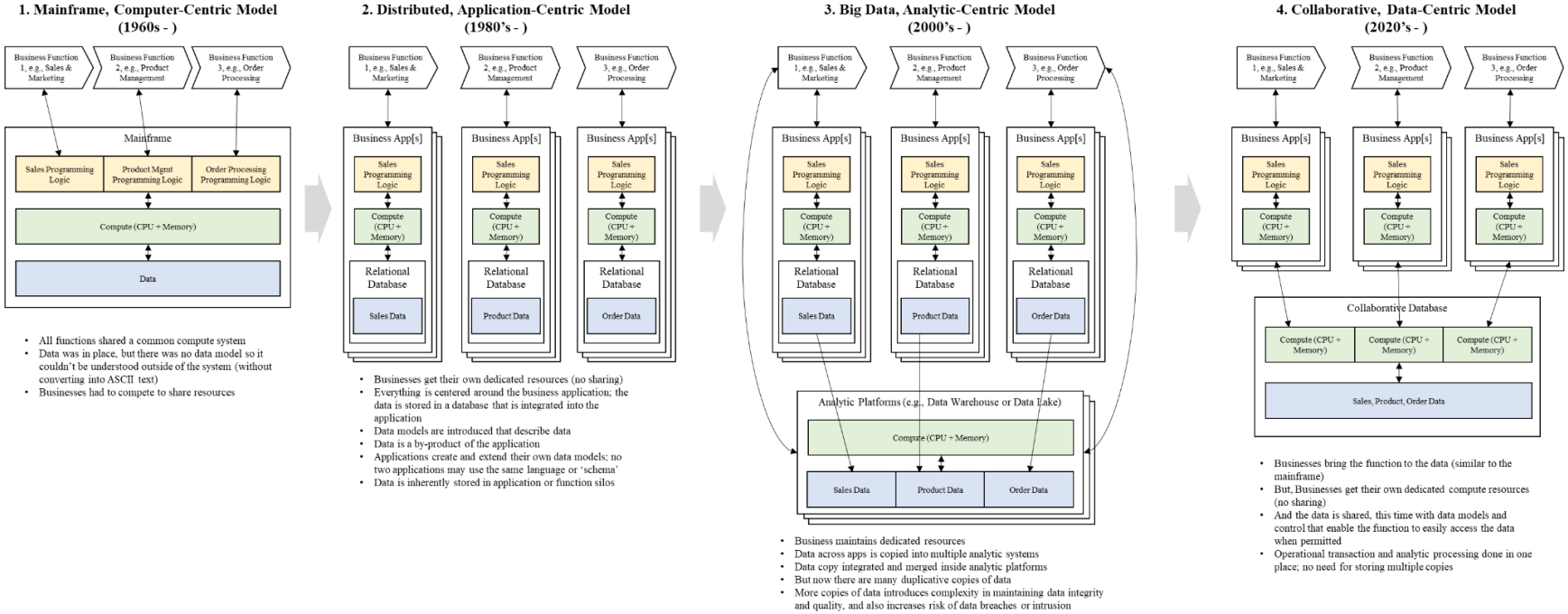

Figure 6. The evolution of data processing technologies and how they've aligned to business model transformations. The fourth generation of data processing technology allows organizations to become data-driven.

Section 2. What Makes Data-Centric Technologies Different From Existing Data Processing Technologies?

As mentioned above, data centricity relies on a new data technology—the collaborative database—which makes it easy to manage data as the product, as opposed to data as a byproduct, of business applications. The technical design of the collaborative database rests on a set of bedrock principles:

- It must be easy to define what data means, in the semantic contexts of both the original producers as well as multiple consumers of data.
- Everyone's contributions to data must be immediately available to everyone else, as long as they have permission to see the data by policy.
- It must be easy to define policies at the level of granularity necessary to control who is allowed to contribute or see what data.
- It must be easy to define who you want to share data with, and it must be easy to add or remove producers and consumers from your trusted network.

In order to meet these principles, collaborative data stores are built by combining elements of eight innovative technologies that address how data can be created, stored, protected, accessed and consumed:

How Data Is Created, Stored and Protected	1. Multiple writers, multiple readers 2. Network-distributed by nature 3. Composable, reusable data 4. Semantically describable data 5. Defensible data
How Data Is Accessed and Consumed	6. Multi-modality 7. Schema-on-query (multilingualism) 8. Self-harmonization

Figure 7. The eight innovations that make up the collaborative data store technology platform.

Let's quickly run through each of them to describe how they all contribute to data-centric design and solve the issues inherent in prior generations of data processing technology.

Part One: How Data Is Created, Stored and Protected

1. Multiple writers, multiple readers:

Traditional RDBMS systems were designed based on a trust model where there is one predominant creator of original data—the business application. By comparison, collaborative databases operate under the presumption that multiple business functions or applications will be sharing the same 'database.' In order to achieve this, they are inherently designed in a 'zero-trust' manner. Theoretically, anyone can write to or read from the same database, provided that the following five conditions are met.

1. Prior to each individual request to read or write data, the individual writer or reader can be verified to be who they say they are by anyone in the network.
2. Each individual write or read request is permissioned by policy before it is executed.
3. Each individual write transaction is tamper proofed, i.e. you can prove mathematically and through logs that what was written has not been altered.
4. Conflicts between write operations can be managed by policy, i.e. you can define the set of rules for how to deal with two writers trying to alter the same piece of data simultaneously.
5. The history of every individual executed request to write or read data is available to everyone publicly and traceable in a tamper-proof method.

2. Network-distributed by nature:

Traditional databases physically exist inside one or more host servers. When a writer records a transaction in their database, information can only be communicated from one application to another if the database broadcasts it to other host servers (e.g., databases or data warehouses) through some integration channel. This can happen by writing to a message queue, publishing an event, exposing data via an API endpoint or extracting a copy and sending it via secure file transfer (SFTP). The problem with sharing data this way is that it encourages the creation of many point-to-point connections between data sources. These connections, in turn, make it very complicated and expensive to make

changes without inevitably affecting some consumer downstream. The proliferation of connections also actively contributes to the continuous redundant proliferation of data copies taking place today, making managing data ever so difficult.

collaborative database are conceptually more of a peer network than a single physical system. They are built on standard protocols that enable multiple computers anywhere in the world to maintain a shared index or ledger of all transactions made by all participants in the network. When any participant in the network, such as the sales business application, records a new or updated transaction into the collaborative database (by writing the change using the standard protocol), the record of the change automatically gets broadcast in real-time to the ledger copies of all permissioned participants in the network. Each consumer can independently validate the integrity and accuracy of the change. This allows each business function to manage their own copy of the index, or ledger, **without physically moving data**. This is critical as it ultimately limits data sprawl, duplication and proliferation.

The boundaries of the collaborative database will not necessarily be a physical host server like an RDBMS, but the network of willing participants. To add business functions to the collaborative database, whether in or outside of an organization's four walls, simply invite them to participate in the network. To remove them, remove them from the network or change their access policies. The underlying method of storage is not relevant; the data can physically be in a private or public cloud, inside a data center or outside, as long as it is reachable by parties via the network.

3. Composable, reusable data:

Business applications have evolved from big, complex programs to a microservice design. Individual, discrete units of functionality and programming logic are now made to be reusable, exposed via application programming interfaces (APIs), and chained and orchestrated together by workflow. collaborative database do the same with data. No longer is data bound by a database host that collects data into a schema made up of tables linked together by a data model for each host, which then requires complex systems integration to merge data from multiple systems together. Instead, data is hosted in discrete, reusable and interconnectable blocks. These are designed by protocol to be chained and orchestrated together at query time, based on policy permissions, by anyone in the network.

Think of how the adoption of microservices fundamentally changed how application development teams wrote source code, managed change and released new business

functionality. Composable data will similarly change how data producers and consumers serve, consume and collaborate on data. Businesses will no longer have to create and maintain multiple consumption views or physicalized extracts of data for each consumer. Instead, consumers will be able to request and retrieve whatever relevant blocks of data they need, at query time, based on policy.

4. Semantically describable data:

In most organizations, the meaning and context of data—its data dictionary, or business metadata—was typically saved separately from the data itself, stashed inside of a metadata management or data governance system. Adding to the complexity, a separate data catalog contained the inventory of physical metadata about the data across its various data stores (such as schemas, tables, column names, data types and relationships to other tables). It required active, consistent effort to link business metadata in data dictionaries with the technical metadata in data catalogs. In the collaborative database, the business and technical metadata, as well as the contextual relationships between them, are stored **directly inside the data**. This means that each atomic block of data, by definition, must contain technical information about itself (its class) as well its business context (the semantic model). This can be thought of as an embedded knowledge graph view of the data for all blocks within the network. This is important as it enables blocks of data to be discovered and reused by different functions. After all, you need to know what something is before you can attempt to use or reuse it.

Furthermore, most organizations don't simply have just one data dictionary or business glossary. As previously described, each business function will likely have its own model to describe its processes. To facilitate interoperability between functions for users, programs or APIs, these dictionary-to-dictionary semantic relationships also need to be preserved inherently inside the data. Because of the extensibility of graphs and triple stores, it is easy to define and add those relationships as extended properties of the data block. The power of enabling semantic describability of data inside the data, using one or more business data dictionaries, will become evident a little later, when we discuss multilingual querying as a unique feature of the collaborative database.

5. Defensible data:

This term describes the novel method for how access controls and data entitlements are managed in a collaborative database. In the beginning, the policies and business logic (rules) for who can read and write data was built into the program hosted by the mainframe or the business application itself. If an organization needed to enforce a data

privacy policy that applied to every enterprise function, each individual program or business application owner would be accountable for making the changes in their software to execute the policy. As systems became more complex, centralized corporate directories and identity and access management systems were introduced. These enabled policies to be controllable from outside of a business application. Enterprise information security management became a little bit better, but then big data came around and made the world more complicated. Identity management systems were built to manage roles-based authorization and access control inside databases. Big data platforms weren't actually databases. Rather, they were file-based systems that strove to act like databases. Integrating big data platforms into existing enterprise access control methods required complex and expensive systems integration. In fact, many organizations today still haven't been able to implement access control effectively in their enterprise data lakes. Because they don't have faith in how access controls were implemented inside the data lake and feared unauthorized access or data leakage, they may avoid copying sensitive sources of data into the lake. collaborative database change the game in four ways:

- The data access control policy is a set of logic or code that is embedded **directly into** a block of data, as opposed to living outside the data in an external system.
- The logic can be based on very sophisticated business rules (such as a smart contracts), can be made context-sensitive by using the semantic describability of the data block, and can take into account far more features than traditional rules-based entitlement systems (like today's role or attribute-based access controls).
- The logic can depend on relationships with other data blocks in the network, so that as data changes in real time, the execution of policies automatically evolves with the data.
- The policy itself *is* just another block of data. It can, in turn, contain its own more granular control policies that govern who can define, view and alter it.

We call the idea of data recursively hosting and enforcing its own access policies within itself "defensible data." Defensible data will revolutionize how access controls are managed in a collaborative database. It can enable entirely new decentralized, but governed, models for managing access controls. For example, in the future, it would be possible for individual end users or customers to be given control, via one policy, over managing the access policies for any data block that contains information about them. They could actively grant consent over what producers or other consumers can see or do with their data. We believe that emerging new data privacy regulations will lean towards these stricter, active consent management regimes versus the existing passive data

policy consent approaches. We'll talk a little bit more about other enterprise data governance implications in the third paper.

Defensible Data In Action

Defensible data is different from traditional roles-based or attribute-based access control. One of the key differentiators is how data access policies are enforced in real-time as data in the network changes. To see what this means, here is an illustration. Let's say that we have a clause in our data privacy policy where only senior customer service representative managers that are associated with a specific customer account are allowed to see certain attributes within that specific customer's record.

For the policy to be executed, the following series of events has to happen:

- The reference data about the customer service reps, and their job titles, is saved in the HR system's database.
- The information about customer accounts, and who manages them, is saved inside the sales application's database.
- HR and sales data may need to be copied into a corporate directory in order to centralize information about users and their identities, as well as applications, their roles and authorizations.
- The corporate directory must then be integrated into a central access management system, which actually enforces the access control for the databases and APIs that subscribe to it.
- When a request is generated by Jane to see "Acme Client Inc." data inside the sales application, it is routed to the access management system, which looks up Jane in the corporate directory, and then applies the access control logic to determine whether to permit the request.

When Jane changes her role and she is no longer either a senior manager or tied to the "Acme Client Inc." account, these changes are independently updated in separate data tables within the HR system and sales application respectively. This data needs to be refreshed and reconciled into the corporate directory. If Jane ever requested access to "Acme Client Inc."

after her job transition, in theory the access management system would look up the directory and deny access.

In practice, however, it is extremely difficult to effectively enforce enterprise-level data privacy policies. This is because of (1) the volumes of changes being processed each day, (2) the multiple physical instances of different corporate directories within most large organizations, and (3) the multiple physical instances of different access management systems implemented independently inside of large organizations.

In a data-centric model, all of the business applications would write into a common collaborative database using a common secure data protocol. HR information about users and job titles, sales information about client accounts and who represents them, and all of the corporate directory information about credentials and permissions would all just be blocks of data within the same 'virtual database,' connected by virtue of the systems being participants in the same peer data network. To enforce the data privacy clause, data access policy and control would be embedded directly inside the blocks of data that host client account information. Business functions would have the ability to define even more restrictive policies to blocks that contain really sensitive data. Because of defensible data, no additional access management system is required. When a request to access data is made by a user or API, each block of data checks its policies. This requires it to look at other blocks of reference data to determine whether the request can go through or not. When data changes in any block, it's immediately available to every other block in the network at query time, which affects policy execution right away.

So, in the scenario where Jane changed her job title, this update is recorded into the collaborative database as a verifiable change to the block of data containing Jane's information written by the HR application. This is immediately broadcast to every ledger in the network and therefore available to any other block of data that is allowed to access that block's information by policy. In the immediate second that Jane attempted to access the "Acme Client Inc." information, the blocks containing "Acme Client Inc." data that were **written into the collaborative database by the sales application** would refer to its internal access policy before executing the request. Inside that policy program are instructions to:

- First look up the credential of the requestor against another block in the data store that contains credentials (which must be provably written by the corporate directory application)
- Then look up the job title from another block of data that contains the job title associated to the holder of the credential (which must be provably written by the HR application)
- Then look for the active customer service representatives associated to “Acme Client Inc.” from another block of data (which must be provably written by the sales application)
- Then apply business logic to determine to allow or deny access if the credential holder is associated with a name associated with the customer account and the title is Senior Manager or above.

The access control policy is acted upon immediately without requiring further synchronization or changes to any source business applications. Furthermore, the access control policy itself is a block of data that can have its own policy on who is allowed to read or update that data. This allows for novel models for data governance and access control.

Part Two: How Data Is Accessed and Consumed

6. Multi-modality:

As discussed before, traditional RDBMS systems were originally designed to handle the transactional operations required to run business functions in real time (such as OLTP database reads and writes). Analytics processes (i.e., OLAP queries), which consume far more compute and memory to execute, could not be performed inside the RDBMS without potentially significantly impairing business operations. To compensate for this, data was typically copied out of databases and transformed significantly to be used inside analytic platforms (data warehouses, big data and NoSQL databases), which were better designed to handle OLAP-type workloads. Unfortunately, in this pattern, the more data that needed to be analyzed, the more copying and transformation was required, exacerbating the complexity of data management.

collaborative database, by contrast, are designed to natively process both OLTP and

OLAP queries. This is accomplished by using modern on-demand distributed computing techniques that spin up or automatically provision resources on a just-in-time basis, with dedicated CPU and memory for each workload type. Transactional read, transactional write or analytic workloads can each spin up their own compute resources when needed, and avoid competing for the same fixed compute resources. This enables many parallel processors to operate on the same physical instance of data without requiring data to be copied.

7. Schema-on-query (multilingualism):

Traditional RDBMS systems are based on the principle of ‘schema-on-write,’ meaning that you must have one data model defined in advance before you can start writing or loading data into the database. Analytic platforms introduced the concept of ‘schema-on-read,’ which decouples the definition of the data model from the loading of the data. This enables the ability to bring together and load copies of data from multiple sources into one target system, namely the analytic platform. You can then build and define one or more consumption data models later by inferring the source data model when reading or introspecting the data.

collaborative database advance this further with ‘schema-on-query.’ Because each block of data saves its semantic context inside itself, and the knowledge graph triple store structure enables one or more data dictionaries to be semantically linked, it is possible to query data from one of many possible schemas, or semantic contexts. At query time, the consumer can choose a specific vocabulary that it understands, define it as part of the query context, and then query data using that model. In response, the data store can traverse the network of semantic relationships and retrieve equivalent data from any relevant block even if the producer originally saved it in a completely different dictionary. This prevents the need to extract copies of data and transform them into multiple consumption schemas so that business functions can share data among one another. A visual example of the multilingual properties of the knowledge graph is provided in Figure 8 below.

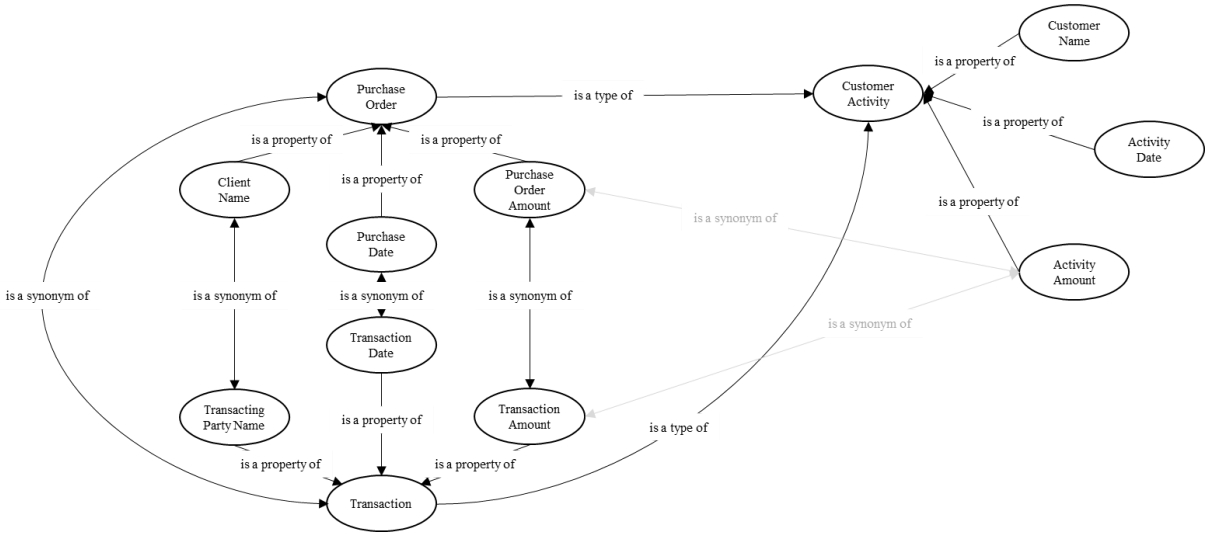

Figure 8. Sample representation of a graph for saving the relationships between different data models or schemas. In the example above, semantic links are maintained between entities called Purchase Order, Transaction and Customer Activity, each coming from different data models but meant to represent synonymous things. An end user can query to find data using multiple paths, such as "Retrieve all Purchase Orders" or "Retrieve all Transactions", and the database can equally retrieve data that was originally saved as Purchase Orders, Transactions or Customer Activities.

8. Self-harmonization:

In traditional RDBMS systems, there was one primary writer (the business application), one data model associated with the data, and one set of data values, which was presumed to be a version of the 'truth.' Once data from multiple business functions and systems began to get integrated into analytic platforms, it became obvious that there was no one 'truth,' for several reasons:

- Because each business application could define its own data model and data dictionary, two applications could use different standard forms to represent the same value. It was really difficult to harmonize the data values at scale across hundreds of applications. A simple example could be that one system describes the country of domicile of a customer by an ISO three-digit country code, e.g. "USA." Another system could save the country via their ISO two-digit country code, such as "US." A third could use its own internally managed valid value list, namely "United States." In order to merge data and generate proper analytic results, such as "return all US-domiciled customers," the values for the same descriptive attribute—in this case, the country of domicile—must be harmonized to the same set of values. Dealing with country codes is a simple scenario, but imagine that this plays itself out across potentially hundreds, if not thousands, of varying business-specific data attributes.

- Because of how data is being copied and transformed by different functions, there are multiple and often conflicting versions of the 'truth,' even if you harmonize the reference data values. Two systems could be describing the same exact entity, but retain very different information about them. For example, one system could say that "Acme Inc." is domiciled in the "US" and another could say that it is domiciled in "UK." This could often have major downstream implications on the functions that depend on accurate data values across processes, such as regulatory compliance.
- The sheer act of identifying that data from two different sources even represents the same business subject areas and could be merged together—much less working through the mechanics of how to harmonize the schemas—requires a lot of work.

The issues around managing truth have traditionally been relegated to reference and master data management programs. Reference data management programs are used to standardize data values for common things that will be referenced by multiple business functions, such as country codes, currency codes, industry- and sector definitions, city names and so on. Teams manage mapping tables that linked variations of possible values to the reference standard. For example, a team might link US as follows: "US" = "USA" = "U.S.A." = "United States," along with the standard that they should all be normalized to "US." This is fine as long as teams can keep their mapping tables up to date as the business applications change, and the number of reference data standards can keep up with the number of distinct types of common referential attributes that can be used across functions.

Master data management (MDM) programs depend on business and data analysts to define business rules that enable MDM systems to link entities across different source systems together and survive the correct information to create "Golden Records of Truth." When the data sources being merged together are internal to an organization and under enterprise control, it is possible to define the business rules with enough specificity to generate trustworthy Golden Records.

By contrast, collaborative database cannot by definition depend on the fact that data will come from one source or writer. They also won't depend on the source(s) always being internal. In fact, the boundaries of internal versus external can be blurred. collaborative database also don't need data harmonization standards to be consistently applied at the source, nor do they require teams of people to be on hand to manage mapping tables, or match rules or survivorship business logic. Instead, they will self-harmonize new data added into the network by using artificial intelligence and machine learning embedded inside the network, which in turn are trained using crowdsourced feedback from network members. In order to maintain the integrity of the data in the network, the data store itself will have the intelligence to detect that new blocks of data are semantically similar to something that already exists in another block in the network (i.e., schema

harmonization). The data store will also detect whether the content inside the block is equivalent to the content values in another block, or whether it needs to be harmonized. The more sophisticated and advanced intelligent agents in the network can not only introspect the contents of data and learn whether there are conflicts, but also learn how to resolve them and create “Golden Sources of Truth” inside the network.

To illustrate, imagine that a new block of data was just written into the collaborative database by a trusted member of the network that contains account information about a customer, “Acme Client Inc.” Based on introspecting the data in the block, the intelligence embedded inside the network can infer and observe that:

- The block of data contains “customer account” data.
- “Customer account” is semantically similar to what some other data blocks call “party” or “client account.”
- The new block added for “Acme Client Inc.” is describing the same entity of information as another block from another verified source called “Acme Incorporated.”
- The value for the country of domicile for “Acme Client Inc.,” which was listed as “United States” inside that block of data, is different from the country of domicile for “Acme Incorporated,” which is listed as “US” in a different block.
- The proper value for the country of domicile for “Acme Client Inc.” is more likely to be “US” and not “United States.”

All of these observations about the data can be programmatically saved inside the knowledge graph. As the members of the network consume data, they can reinforce whether observations and inferences were correct or not. This, in turn, improves the quality of the network’s self-harmonization for the next new data block that gets added, or else tells members to go back and look at similar inferences made to previous blocks. Data governance shifts from a model of centralized management to one where a crowd of producers and consumers participate in the action of describing and fixing data just by actively using it. We’ll talk about this a little bit more in the data governance paper.

Putting It All Together

As should hopefully be evident, the integration and combination of these eight technology innovations into a single platform called the collaborative database will empower businesses to transform towards becoming data-driven organizations. As discussed in the first paper, it will make it easier for organizations to:

Find Data	• By supporting a single common platform to host data, the collaborative database, instead of having data copies proliferating across hundreds of systems.
Know What Data Means	• By using semantic describable data to enable business and technical metadata (the data dictionary and the data catalog), which are saved directly inside each reusable block of data. • By supporting multiple parallel semantic ontologies and using schema-on-query to enable multiple business functions to retrieve any relevant data using their native vocabulary.
Know the Quality	• By using the inherent properties of self-harmonization to autonomously discover and remediate data discrepancies.
Enable Access	• By making data available instantaneously to trusted members in the network as composable, reusable data, but also securing it under highly granular privacy and access control using defensible data.

We hope that by now it is clear what a data-centric architecture is, how a collaborative database enables it, and what makes this different from other existing data technologies. Moreover, we've discussed the business benefits of adopting a data-centric architecture, namely faster access of data to the business, simplified and less complex data architecture, and more secure information security control, among others.

But if you're come this far, reading this may have yielded even more questions. Where do we even begin? How do we even think about adopting yet another technology on top of the hundreds of mainframes, relational databases, data marts, data warehouses and data lakes across on-premise data centers and/or multiple cloud providers? Or even before that, who in the organization actually owns this system and who manages the data, if not the business function owner? What does it mean to manage data in a data-centric architecture?

These are critical and important questions. In our next paper, we will explore the implications of data-centric architectures to data ownership and governance models. And

we will conclude with a final discussion on adoption considerations, including how to finance the business case for transformation, which process to begin with, and how to undertake a transformation effort across the dimensions of people, skills, processes and technologies.

Chapter #3

Data Governance in Data-Driven Organizations

In our previous two papers in this series, we discussed the emerging transformation that innovative businesses are undertaking right now. They are evolving from data-enabled to data-driven. We described what this means (hint: it's more than just hiring more data scientists), as well as the business case and drivers for taking the journey. We further discussed new technologies and architecture patterns that enable the data-driven model.

However, becoming data-driven involves more than just introducing new data policies or technology. Putting data into the center of the business requires rethinking some basic ideas. Who owns data if it's meant to be shared by everyone? Who is responsible for cleaning it? How will you keep data safe and secure? Data-driven organizations optimize the processes, roles, accountabilities and responsibilities for how data is created, stored, protected, shared and maintained. It becomes both safer and easier to share data across the company, as well as with partners, suppliers and customers.

In this paper, we will:

1. Provide an overview and definition of what we mean by data governance.
2. Describe how today's mature, data-enabled organizations typically govern data.
3. Discuss the shortcomings of current approaches.
4. Describe how data governance evolves in a data-driven model.

Section 1. Defining Data Governance

Since this is a paper about data governance written by a data professional, let's start with a definition. What exactly is data governance?

Definition:

Data governance is a business discipline whose goal is to ensure that data across the enterprise is usable for any business purpose, current or future.

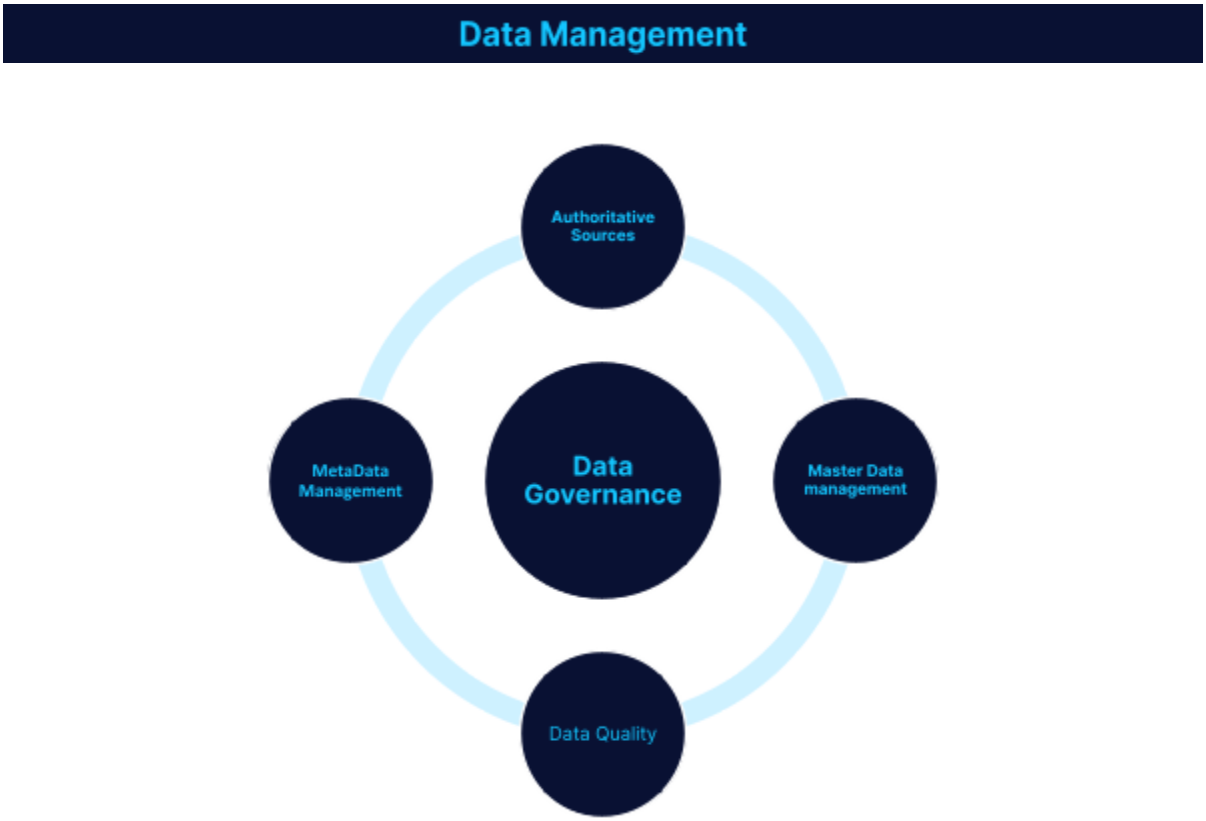

Figure 1.The components of a holistic data governance model.

Data management encompasses the technology platforms, operations processes and business stewardship of the key data disciplines depicted in the illustration above.

Data governance is embedded at the center of the data management discipline. It provides the oversight to ensure implementation and measurable success of data management practices. Data governance's role is to orchestrate the pieces within the data management ecosystem. It provides policies, standards and business processes that tie together data management technology and operations with the business processes required to enable data to become useful.

In our first paper, we identified the four stages of data maturity in organizations, from data-aware to data-driven. We summarize the stages below again in Figure 2.

Data-Aware (Function focused)	Data-Informed	Data-Enabled	Data-Driven
Data is created as a by-product of a process/function.	Data is created as a byproduct of a process/function.	Data is created as a byproduct of a process/function.	Data is the product and business functions organized around data.
Data is siloed into each function.	Data is siloed into each function but shared into business warehouses on a bespoke basis.	Data is siloed into each function but consolidated into enterprise warehouses and data lakes.	Data is freely shared and available across the organization.
Data is inputted and used in each function separately.	Data is transformed and moved to each function (warehouse).	Data is transformed and moved to each function (lake).	The function is moved to the data (i.e. one copy of the data which all users access).
Data is on a “need to know basis.”	Data is on a “need to know basis.”	Data is on a “need to use” basis.	Data is defaulted to an “always share” basis (subject to privacy and security restrictions)
Data is primarily used for operational purposes.	Data is used to inform decisions.	Data used to enable decisions.	Data is used to drive behavior and products.

Figure 2. The four stages of maturity and readiness inside organizations.

As you can imagine, data governance did not exist as a discipline in “data-aware” enterprises. It first appeared as organizations started to progress towards the “data-informed” stage and mostly concentrated on data privacy and access. However, as organizations have moved into a “data-enabled” stage, they have acknowledged that when data is created as a byproduct of a business process, there needs to be a discipline that makes the data usable for other purposes, especially the ones that require cross-silo data. To accomplish that, data governance creates rules and processes to:

1. Know and codify which business processes produce data that can appropriately be used

by other businesses and functions—**authoritative sources of governance.**

2. Recognize the most important business data concepts (e.g. customer, product, account) and steward rules for unique identification and disambiguation of these entities—**master data management.**
3. Understand which level of data quality is fit for different purposes within the enterprise. Create processes to measure and remediate the data that doesn't meet its assigned quality level—**data quality management.**
4. Understand and make available business and technical descriptions of data elements, including ESG, privacy, retention and information security classifications; collect and make available data lineage—**metadata management.**

Section 2. Data Governance in Data-Enabled Enterprises

In most data-enabled organizations today, data is governed via a delicate interplay between three types of actors in a company:

1. **Producers:** Typically, this is the business function and function-aligned information technology (IT) team responsible for developing applications that enable a function to operate and create data. This can also include the team(s) that make copies of data available for consumption outside of the core business application, for example loading data into function-aligned operational data stores, data marts, data warehouses or data lakes. Examples of producers, in, say, banks, would be credit card, mortgage loans, corporate loans and securities trading businesses.

2. **Consumers:** These are usually the business end users and function-aligned IT team that uses someone else's data for decision support, or who integrate someone else's data into their own software (which in turn, creates new data, i.e. most of the consumers of data are also producers of , often derived, data). Examples of the consumer functions in banks are corporate finance, credit risk and anti-money laundering compliance functions. As companies move to embrace data and analytics-driven growth, data science/data analytics teams become important consumers of data, though they are still oft-forgotten in data governance structures.

3. **Governors:** These are generally the individuals responsible for ensuring that data is well-managed from cradle to grave. These can include a Chief Data Officer (CDO), a data governance council or committee comprised of producers and consumers that make or regulate policy, a data governance team that supports policy definition, data stewards representing the business functions who enforce policy, and data quality analysts who work with the governance team and data stewards to fix errors.

This organizational model, while much more effective than the access-focused governance of yesterday's data-informed organizations, is still subject to significant friction between the main actors. There are two main sources of friction. One is the misalignment of priorities between consumers and producers of data. The other is the lack of real enforcement power in the data governance team.

In the data-enabled, application-centric model:

1. Producers create and shape data as needed to support the primary operation of the business. They are measured and compensated by how much they can improve the productivity, revenue or net income of their business function. Business executives, product managers and supporting technology teams are not incentivized to make use of any data generated outside of the immediate concerns of the producer business. There is a high-level acknowledgement of the importance of data for either regulatory or overall growth objectives. A lack of data literacy on the part of data producers, however, prevents full engagement. Producers' short-term needs are prioritized over the long-term needs of the enterprise.
2. Governors, on the other hand, are accountable for how effectively data is leveraged in an organization—how good it is, how available it is, and how well-protected it is. Their success is measured by their ability to effectively leverage data. Governors define the policies to make data useful, but are dependent on producers and their willingness to engage (and often fund) data management efforts. Producers are also the ones who are actually responsible for implementing policies, standards and controls.
3. Consumers are typically held captive to the needs of the producers. Perhaps the producers' business will actually fix and define data and make it available to them, perhaps not. It depends on whether producers have the budget, time, resources and a compelling financial interest or imperative to prioritize making data better and available for others' benefit.

Let's take a look in detail how these three groups interact across all four areas of data governance:

Governance Area	Governor	Producer	Consumer
Authoritative Source	Defines and publishes the criteria for what constitutes the authoritative source; establishes control process for designation and usage of such sources.	Designates which of the multiple business systems is authoritative based on agreed-upon criteria.	Only sources data from the designated authoritative source.
	Reality check:	Reality check:	Reality check:

Governance Area	Governor	Producer	Consumer
	Complexity of producer business processes and systems creates challenges in defining granularity for an authoritative source. For example, attribute-level granularity is impractical, however, subject area-based granularity may be too high-level. Lack of data literacy in producer teams makes authoritative sources designation a guessing game. Governance function seldom has enough teeth to enforce the rules effectively; the controls often devolve into a list of exceptions.	• Source systems often have overlapping data that's out of sync, making the "authoritative" designation complex. The complexity of the number of systems that data is passed through make the identification and management of data difficult. • Source systems data is often locked into legacy monolithic applications and requires significant investment to get out into a usable format.	• Consumers tend to source data from the same place they always did based on familiarity and a reliance on undocumented adjustments, making it hard to reconcile existing reporting with that coming from authoritative sources.
Master Data	Designates subject areas that are subject to mastering, defines stewardship processes, establishes control processes for stewardship.	Uses mastering process when onboarding the entities that are subject to mastering—i.e., new products, clients, vendors, etc.	Consumers use mastered data sets as an authoritative source and help governors drive stewardship and mapping.
	Reality check: • Master data management is a complex and messy business process with still-insufficient technology tools to make it easier to implement.	Reality check It's a pipe dream for most businesses that rely on legacy technologies. Mastering onboarding requires significant—and therefore expensive	

Governance Area	Governor	Producer	Consumer
	• Lack of data literacy among producers makes it difficult to stick to real improvements and to see them through.	and risky—changes to existing business processes and technology systems. Moreover, ROI of such changes is difficult to quantify. The best outcome to hope for is that producers will map their native identifiers to the enterprise master ones and actively participate in data stewardship match and merge process.	
Data Quality	Defines criteria for critical data elements, processes and tooling for data quality measurement and issue management resolution, as well as stewardship practices for creating business data quality rules and prioritization of data quality issues. Manages data quality issue resolution process.	Data stewards in business and technology teams investigate and remediate data quality issues as they are identified and prioritized.	Define critical data elements and data quality rules; identify and document clear and quantifiable business impact.
	Reality check: • Data quality practices and tooling have matured in the past ten years. However, driving significant improvement in data leveraging data quality scorecards requires a much more advanced level of data literacy in the business and technology teams than currently exists.	Reality check: Producers of data business- and operational processes and technology systems have evolved to mostly satisfy internal requirements for data quality. Issues identified by consumers external to the business rarely receive priority to be investigated or	Reality check: Defining data quality rules that catch issues with clear impact requires a much higher level of data literacy. Business impact of data quality issues is hard to quantify for businesses that do not routinely measure and report the amount of manual workarounds and

Governance Area	Governor	Producer	Consumer
	• Most data quality tools use explicit deterministic rules in assessing quality. The real world requires more intelligent, practical assessments of quality (new tools are beginning to provide this).	remediated, unless driven by clear regulatory mandates or sufficient corporate funding.	adjustments resulting from poor data quality.
Metadata	Defines scope of data lineage and metadata collection—e.g. only critical data elements for critical business processes? Something broader? Narrower?	Business- and technology data stewards provide meaningful business definitions and tie them to physical metadata.	Articulates value of collecting business- and technical metadata.
	Reality check: • While tooling for physical metadata collection has matured significantly in the past ten years, the real value of metadata is in tying business metadata to physical metadata. This still remains a mostly manual exercise, which the budget-strapped data governance organization usually deprioritizes.	Reality check: • This is still a mostly manual and labor-intensive exercise. It usually gets short shrift due to a lack of data literacy and dedicated budgets.	Reality check: Business value of meaningful metadata is difficult to articulate and even more difficult to quantify.

Section 3. How is Data Governance Different in Data-Driven Enterprises?

Let's review the definition of the data-driven organization from the first paper in our series, then extrapolate how data centricity would inform changes in data governance, structure and practices.

In a data-driven organization, data:

- Becomes the product. Business functions organize around data.
- Is freely shared and available.
- Defaults to "always share."
- Is used to drive products and behavior.

The adoption of data-centric architectures and new collaborative database technology, which we covered in our second paper, enables a successful transformation to a data-driven model. The collaborative database is a common platform where (1) producers can save data for their own operational use; (2) consumers can find and use data for their own operational or analytic needs; and (3) governance can apply common enterprise standards and controls.

There are two major implications for data governance:

- **Data doesn't move.** Data is created, stored and used in the same place. Many of the challenges that data governance addresses originate from the movement of data between business applications. In a data-driven architecture, however, data doesn't move. Any consumer or producer with access to data can immediately observe any transformation. Data provenance, data lineage and data traceability thus become non-issues. Given how much time and money institutions (especially financial institutions with large regulatory mandates) spend on discovering, documenting, visualizing and updating data lineage, this is a huge benefit of the data-driven architecture in and of itself.
- **Incentives align.** Becoming a data-driven enterprise requires significant changes to internal culture and the social contracts between business functions that produce and consume data. This leads to new incentives for business and technology teams that produce data, in turn improving data literacy (though it might be argued that it's the other way around—changes in data literacy lead the

way for the enterprises to become data-driven). In fact, a misalignment of incentives is a major source behind the previously described “reality checks.”

When data becomes the company’s main asset and product, data quality becomes a priority. The goal of ensuring quality changes how businesses prioritize- and fund data, as well build business- and technology architectures. Data governance evolves from a separate, often audit-enforced process into a pervasive business- and technology practice. Some current, common data governance practices become obsolete.

Let’s see how this shift in understanding, incentives and technology changes the data governance discipline:

1. Producers and consumers utilize data from the same repository as everyone else. They are measured and compensated by the quality and usability of data, which is now one of the enterprise’s main products. Business executives, product managers and supporting technology teams are incentivized to make data useful and usable in ways beyond the immediate application of the producer business. Additionally, for the enterprise that has fully implemented the collaborative database architecture, data quality, data integrity, and data centricity are encoded via self-harmonization properties that autonomously discover and remediate data discrepancies.

   In a data-driven organization, the clear line of demarcation between producers and consumers blurs. All functions produce and consume each other’s data, governed by policies around who is allowed to see and change what. Smart contracts, directly embedded in the data itself, enforce policies to prevent one function from accidentally viewing or overwriting data that is critical to another function. Furthermore, all data changes and lineages are natively preserved in collaborative database. If you need to understand the provenance of where data came from, or revert to an earlier snapshot, this can be more easily accomplished.

2. Governors are still accountable for and measured by how effectively data is leveraged in an organization—how good, available, and well-protected it is. They will continue to coordinate and work with producers and consumers to ensure that data is always available and of high quality. However, in a data-driven organization, they also play a more active and hands-on role for some data management functions, especially around the creation and enforcement of data access and retention policies. In essence, one of the major implications of data-driven transformation is the **direct responsibility of the governors** over some governance functions that used to live exclusively within the domain of the producers.

Let's give an example. Let's assume that a country has just issued a new data privacy directive. No personally identifiable information about citizens can be exposed to any personnel outside of the country without legal and regulatory approval. A company's Regulatory Compliance & Legal team has raised the issue, and the governors within the company (the data governance council) modify the company's data privacy policy. The governors also issue an internal notification socializing the change to the policy, and set an enforcement date. They are accountable for ensuring that the company is compliant with the policy by the enforcement date.

How would this policy be implemented and enforced in a traditional, data-enabled organization? Usually, the task would fall on the producers to execute the change to make the policy real. First, the business leaders in the organization need to know which producers will be impacted by the policy change. Then, the producers will need to request the funding to make changes to their software in how they store or provide access to data about customers in the affected country, so that they can provide the necessary protective controls. They will need to prioritize the effort to implement the change on top of other business priorities. Finally, if originating sources of data about individuals in that country were copied to other operational or analytic data stores, the owners of those systems would also need to invest in changing the access controls in their respective systems to be compliant with the new policy. Those changes will go through the typical software development lifecycle, meaning that producers will modify, test and release code to production systems in line with their respective application lifecycle management processes. Consumers may scan some of their systems or reports to ensure that they flush any affected data based on the policy, but they generally depend on the producers to comply with the policy. Depending on the size and complexity of the organization (and how much personally identifiable information about affected citizens has diffused through the organization), it may take years for the company to be fully compliant.

Let's contrast this with a data-driven organization that has adopted a data-centric architecture (with a collaborative database). Based on the principles of "defensible data," governors can directly create a data access policy that limits access and embed it into blocks of data that contain personally identifiable information about individuals from the affected country. Producers and consumers can continue to use the collaborative database, and their applications will be restricted from accessing the data blocks if they do not meet the policy requirements. Governors not only set the enforcement date of the policy—they can directly set the enforcement of the policy itself!

With this example in mind, let's now take a detailed look at how these three groups interact across all four areas of data governance:

Governance Area	Relevant?	Governors	Producer/Consumer
Authoritative Source	No. Data proliferation is stopped by functions coming to the data.		
Data Quality	Yes. Much more pervasive in the data-driven organization, where all data is "critical." Leverages the self-harmonization capabilities of the collaborative database.	Define stewardship practices for creating business data quality rules and prioritization of quality issues. Manage data quality issue resolution process. Implement scorecards to measure.	Data stewards define data quality rules. Technology teams embed them into the smart contracts. Data stewards and technology teams investigate and remediate data quality issues as they are identified and prioritized.
Master Data	Yes, but significantly slimmed down due to automatic enforcement by self-harmonization capabilities embedded in the collaborative database[2] and pervasive data literacy.	Designate the subject areas that are subject to mastering, define the stewardship processes, establish control processes for stewardship.	Use mastering process and inherent collaborative database capabilities when onboarding the entities that are subject to mastering—new products, new clients, new vendors, etc.
Metadata	Yes. High-quality metadata is paramount to the success of collaborative databases.	Define metadata dimensions to be collected. Establish minimum standards for metadata.	Business definitions and relationships are crowdsourced throughout business and technology teams, who train machine learning algorithms to create and collect meaningful metadata. Teams handle the exceptions produced by the automation tools.

[2] Much of the potential for disruptive transformation in data governance is dependent on the efficiency and effectiveness of machine learning and AI-based learning systems that make self-harmonization possible. Without machine assistance, the implications for governors, producers and consumers are the same, but more manual efforts would be required.

Summary

In new data-centric models, data is treated as the main asset of the enterprise and the driver for its growth and sustainable development. That shift in focus, incentives and corresponding shift in culture the data governance/data management function evolves from being overseer and an enforcer of (often) unpopular rules into a business function directly accountable for the success of the various business lines. The concept of producers and consumers is blended into a model where business functions participate in a community. Everyone equally creates and consumes data in a virtuous loop. Each business function may still ‘own’ its composable blocks of data. Data governance, however, oversees the policies that bind composable blocks and organize them into a cohesive chain where data can be discovered and consumed by other parties within and across functions.

Over the course of this series, we (1) described the business value for the transformation to a data-driven culture and model, (2) described the underlying technology architecture and innovations that make this possible, and (3) have described the governance and management implications of managing data in a data-centric way.

About Fluree

Fluree's mission is to flip the traditional "application-centric" model that has left behind a wasteland of data silos into a "data-centric" model where data is natively interoperable, trusted, secure, and readily consumed by participant networks, AI, and machine services.

Fluree offers 2 integrated product offerings: Fluree Core, a collaborative database, and Fluree Sense, a "golden records" data cleansing pipeline.

1. Fluree Core - Fluree is an immutable semantic graph database packaged with digital trust, data-driven access policy, and linked data standards. Whereas most databases live as static silos behind one application, Fluree's linked graph technology enables data to be shared in-place across many environments while protected by policy, trust and privacy.

   Organizations must share data in order to remain competitive, decrease risk, and unlock efficiencies, but most enterprise data is lost, incorrect, duplicated, and untrusted. Fluree combines open standards, semantics, and distributed ledger technology into a flexible database platform so data can be verified, integrated, and shared seamlessly across analytical and operational environments.

2. Fluree Sense - Fluree Sense is a data cleaning, transformation, and mastering pipeline solution that creates order and structure out of chaotic, disparate data. The product uses machine learning, data subject matter experts, and semantic ontologies to automate the ingestion, classification, and remediation of disparate legacy data. Fluree Sense cuts the time and cost of the most expensive task IT teams are faced with – cleaning and making sense of disparate information.

Learn more

info@flur.ee
www.flur.ee
+1-336-283-7288