Datavalue.ai

Evolution into latest Data Architecture model for intelligent decision-making processes and lower operational costs

www.datavalue.ai

Evolution into latest Data Architecture model for intelligent decision-making processes and lower operational costs

By all means, information is a company's best asset. Nowadays, there is no doubt that data is key to making thoughtful, detailed, and democratic decisions. Companies want to embrace actions that have been thoughtfully designed to achieve the results that they envisioned.

In fact, for many companies, it is of the utmost importance to achieve a data-driven strategy as they are well aware of the potential that benefits the intelligently empowered. It helps them provide the best possible customer support experience that's based on reliable data that promotes a certain hyper-personalization, bringing operational costs down as well as the amount of time that's being devoted.

Additionally, empowering employees by giving them the tools of trend analysis and business intelligence is indispensable. This has been driving considerable investment in building enablers that include intelligence and data platforms. However, the results have not been at the speed we´d expect.

It should come as no surprise that there is a trend for organizations to meet multifaceted obstacles in their efforts to become more data driven. Feeding into this complexity is a number of things such as a distinct resistance to accept data as an asset and, of course, changing over-arching business priorities.

What has been lacking is a certain architectural perspective that is at the heart of the failure that many data platform initiatives share.

Are data platforms favoring data management?

The team structures that stand in most of today's platforms leaves a lot to be desired. When it comes to advantageous opportunities for decision-making, they are quite often left in the dark as there isn't a directionality regarding where data is drawn from, and how it is ultimately used.

Because it's not just a matter of having the information but most importantly ***how that information can be made reliable, easily accessible, operational at every moment, complete, and adequately organized.***

You and everybody in your company should be able to manage it at all times and from wherever you are. You need to know what you have. And how you can better make use of it. Data is the best tool to make your processes more efficient and increase your earnings.

However, this has not been the case in most companies which have favored a **monolithic data architecture** that prevents efficient information management among the different team players.

Within a company, information usually takes a turbulent ride from its sources to the decision makers. This can not only increase costs but, most importantly, disappoint the clients as decisions take more time than expected.

When you look at the differences among the data platforms based on data lake architecture, you'll find that they share failure modes which result in much less than ideal unfulfilled promises at scale. In order to address these recurring failure modes, we need to rework the centralized paradigm, or preceding data warehouse, of a lake that persists so much. **The answer lies in a much more sensible paradigm that's based on a more modern distributed architecture.**

In this article we´ll present you with the most efficient approach to data management in 2021. Firstly we´ll go over the successive data solutions to then describe how your company can focus their efforts to organize and manage data.

From Gen 0 to Gen 4

The first solution, or **Generation 0** can be viewed as the pre-history of data management. It relied on one sole server that featured a limited capacity and required engineers for constant maintenance.

Data was manually accessed. The system was prone to errors, and it was difficult to trace the route that the downloaded information followed. In short, how information got to the hands of the decision makers remained a mystery.

The second solution, or **Generation 1** was the era of the big servers. You probably remember what a blast was counting on powerful servers that could process tons of data, and how much money companies invested in securing these servers.

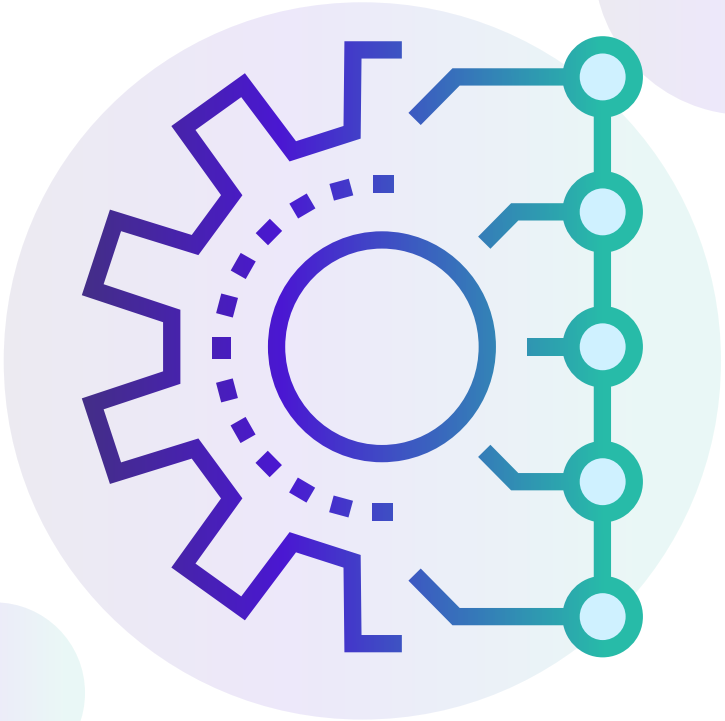
By having two servers, companies assigned one for storage and the other one for data processing. Thanks to this division of tasks, the system decreased its risk of errors as the access of information was automatized. Information could be traced more easily even when its speed was limited but optimal.

Furthermore, this system provided Business Intelligence solutions to companies by allowing a convenient visualization of the information paving the path for quicker and better informed decisions.

At this point, you might probably believe that data management couldn't go any better. But it did!

Generation 2, or a distributed computing system, multiplied the number of servers allocated to data processing. There is one bigger server assigned to data storage while a cluster of servers works as a Data Warehouse that is connected to a Business Intelligence platform.

This complex big data ecosystem is operated by a centralized team. This team is composed of very specialized data engineers that make their mission to create what are referred to as data lake monsters that hopes (quite wishfully) to actualize R&D analytics.



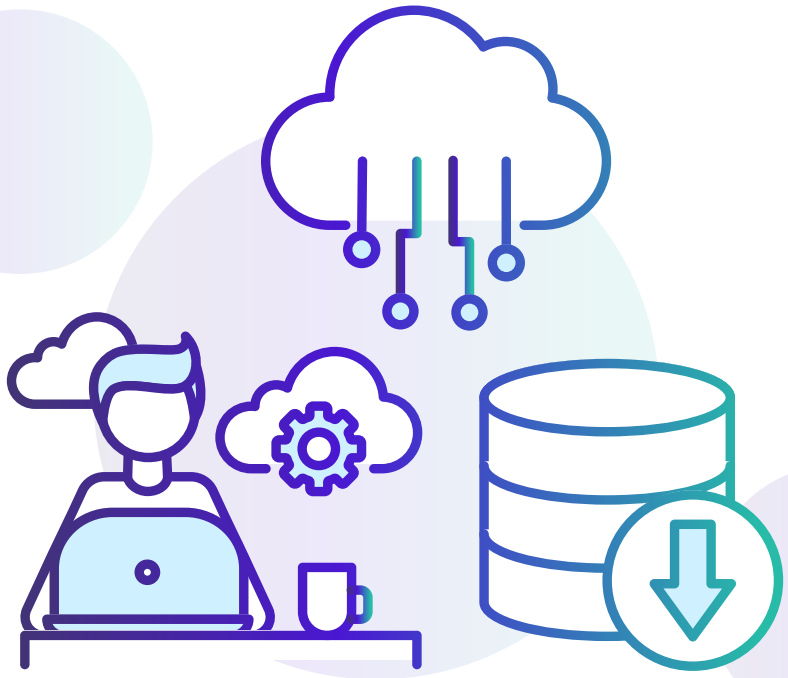
The improvements were immense. By counting on a system that is able to efficiently distribute the data in the cluster of servers, its processing capacity grew exponentially even though its speed was still limited. Thanks to the automatized access of information, the risk of errors was minimal and, as it is the case in generation 1, information could be easily traced.

The thing is that information and companies using it as their most valuable asset grew even more than the previous system could afford. For this reason, Cloud Computing made its way in the data management landscape. It is **Generation 3.**

Now, companies buy space in the cloud, replacing the physical servers.

This system incorporates an emphasis on streaming for real-time data availability with architecture as is the case with Kappa, unify the stream and batch processing with respect to data transformation through frameworks such as Apache Beam, and rely on machine learning platforms and data pipeline execution.

Cloud computing made data management not only more effective but also cost efficient. Less engineers are needed and systems are put into place to distribute the data through the cluster with no electric energy. Its capacity and speed is unlimited, leaving no room for errors. **This last generation platform improved the efficiency of real-time data analytics, and brought the costs of managing big data infrastructure down.**



Failures of a centralized data management model.

The bottom line is that the present data architecture is centralized, domain agnostic and monolithic, all at the same time.

When the architecture is centralized, data from all parts of the enterprise are drawn from. This includes all the operational systems transactional and domains that allow for the business to function.

The centralized model isn't too adaptable and, in the end, falters under the weight that comes with a large number of sources, especially when they deal with a wide array of consumers.

Where the real faults in the centralized data platform lie more often than not include:

A Too much data in one place: The pervasive availability becomes a liability as it becomes too complicated a process to review and package in a way that it can be controlled readily.

Think about it this way: the more sources you have. the less efficient a centralized platform can respond. It may seem like a sensible strategy but it doesn't answer the concerns that come along within the context of a sizable enterprise that comprises numerous domains that are expected to receive new sources on a regular basis.

B This leads to the big deal for organizations: the need to experiment as they receive all manner of novel data. In order for data to become useful, it needs to be transformed into something that's workable.

Being able to simplify the architectural quanta should be of paramount interest in order for a data platform to grow. In order to accomplish this, an architect must decompose said platform to a pipeline of data processing stages.

Data Mesh Platform: The New Generation

So, what should a data platform architecture look like?

The key is inclusivity through a well distributed data mesh wherein a self-serve approach is implemented so that product thinking may be truly achieved. It may sound like a lot, but it really eliminates many more problems than it creates.

In his book Domain-Driven Design, Eric Evans proposes decomposing systems into distributed services to establish a micro services architecture that serves business domain capabilities. This **Generation 4** or data mesh which is based on the individual management of micro-services which no longer require engineers but data architects.

This means that the underlying storage processes should be set up in a way that's separate from existing operational databases as they need to be able to fill the requirements of big data. Thus, that data-driven and intelligence services are free to refer to the business facts and put together new projections or aggregations.

This is how a distributed data platform should come into its own, namely by changing the outlook. No longer will you be talking about a single platform owning and extracting the data for its use, but rather seeing individual domains as being capable of providing their data as a product that's open for discovery.

Cleansing data is of paramount importance. As a result, centralized data pipelines should devote much of their attention to this area. It makes sense as you ideally would like **data that reflects the real situation that your business is in.**

This is where a service level objective should come in to assess how close to reality the truthfulness of collected insights actually are. This may be achieved by ensuring that there is adequate automated data integrity testing as well as an apt data cleaning in place.

The quality thus requires a value which should be measured with a data integrity indicator that compares the different domain data products.

Self-serve data assets are a major goal that can only be accomplished when quality products are able to be understood, consumed and easily discovered in an independent fashion.

When you distribute a domain data architecture, you face the obstacle of correlating data across domains. In order to work this must be done in productive and sensible ways through the rightful use of aggregates and filters. You need to have the necessary foundational standards for interoperability between datasets that pertain to polyglot domains.

In this new scenario, data engineers should direct their attention towards matters of vision and form a roadmap for data products to follow to please the expectations of consumers as well as make sure to provide them with an overall quality. They're also charged with knowing when they must change, revise and withdraw schemas and data, and all of this is tied in with reconciling the needs that domain data consumers present. It is in this manner that they are able to define a rationale for what makes for a success and what the business-aligned Key Performance Indicators should ultimately be.

Thus, data engineers should avoid an isolationist dynamic and become free to adjust and grow in a cross-functional way wherein software engineers and data engineers can learn from one another to improve automated testing and ensure a steady flow of deliverables.

Conclusion

here is a trend for modern enterprises to direct their investment in their next generation data lake so that they may democratize their data architecture. Thus, they can find business insights that may eventually become automated intelligent decisions.

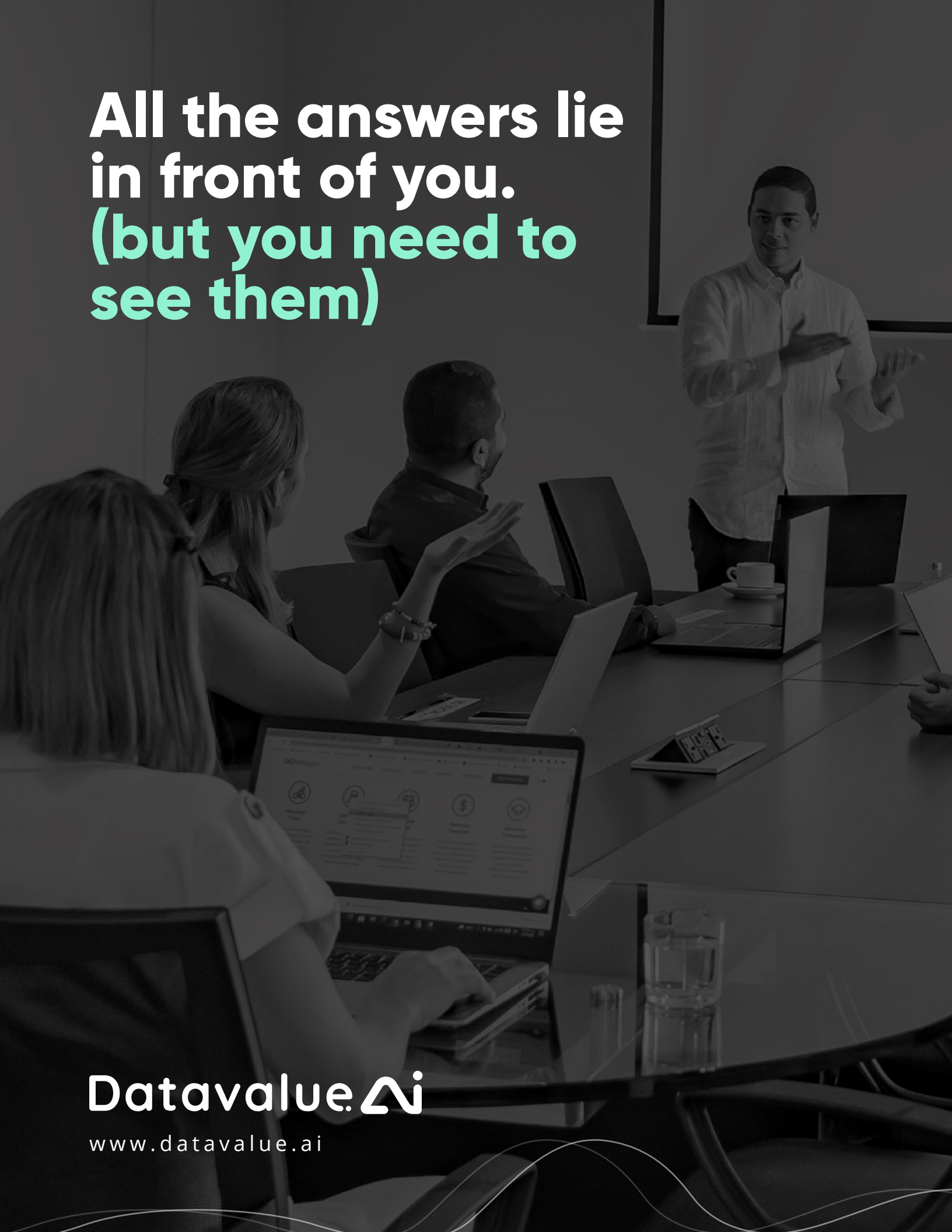
Decentralizing the monolithic data platform feels like quite the feat. More than anything, we just need to look at data in a new light, specifically with respect to its ownership and locality. The idea should be for domains to serve their domain datasets as hosts. The challenge, however, is to accomplish this in a manner that's consumable.

If a software ecosystem is to truly thrive, it shouldn't unnecessarily duplicate the resources put towards building storage, streaming infrastructure and pipeline engines. By counting on a competent data infrastructure team they are free to own as well as provide the vital technology that the data products of domains require to be served, stored and processed. All that is really needed in order to address this is to construct a common infrastructure through tooling and proper technique applications, which tend to come about quite naturally as a data ecosystem becomes more and more established.

You look at how much the lead time is used to create a new data product. This will set a parameter wherein a smooth system of automation can be developed with favorable capabilities such as data product auto-registering, automated data ingestion in the form of configurations and data product scripts that can go ahead and put a framework.

This kind of cloud infrastructure will ultimately bring operational costs down as well as the amount of effort that's invested to establish the data infrastructure's on-demand access.

I encourage you to be open to the prospect of moving past the monolithic and centralized data lakes towards a deliberately distributed data mesh architecture.


All the answers lie in front of you.
(but you need to see them)
Datavalue.ai
www.datavalue.ai