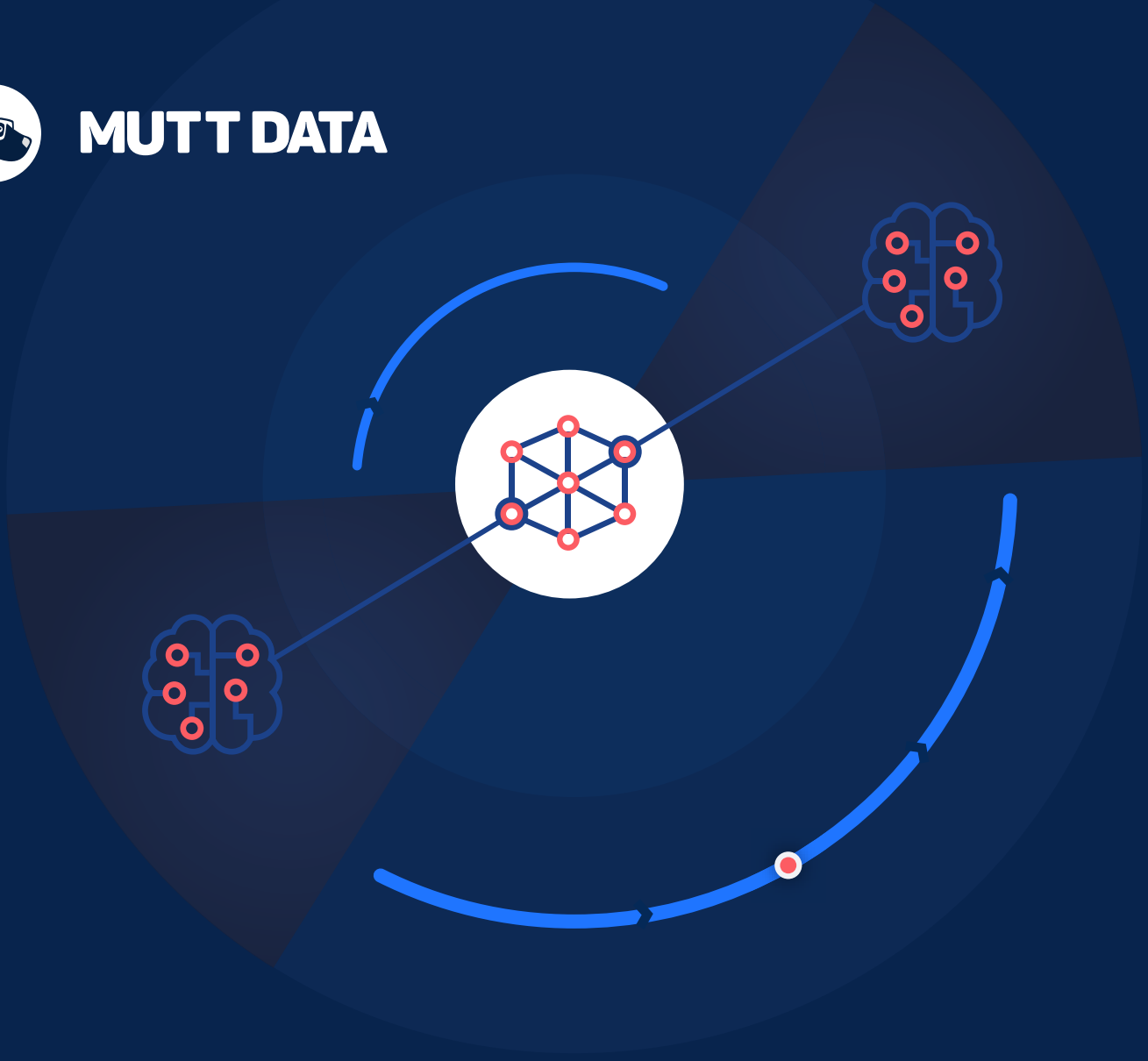
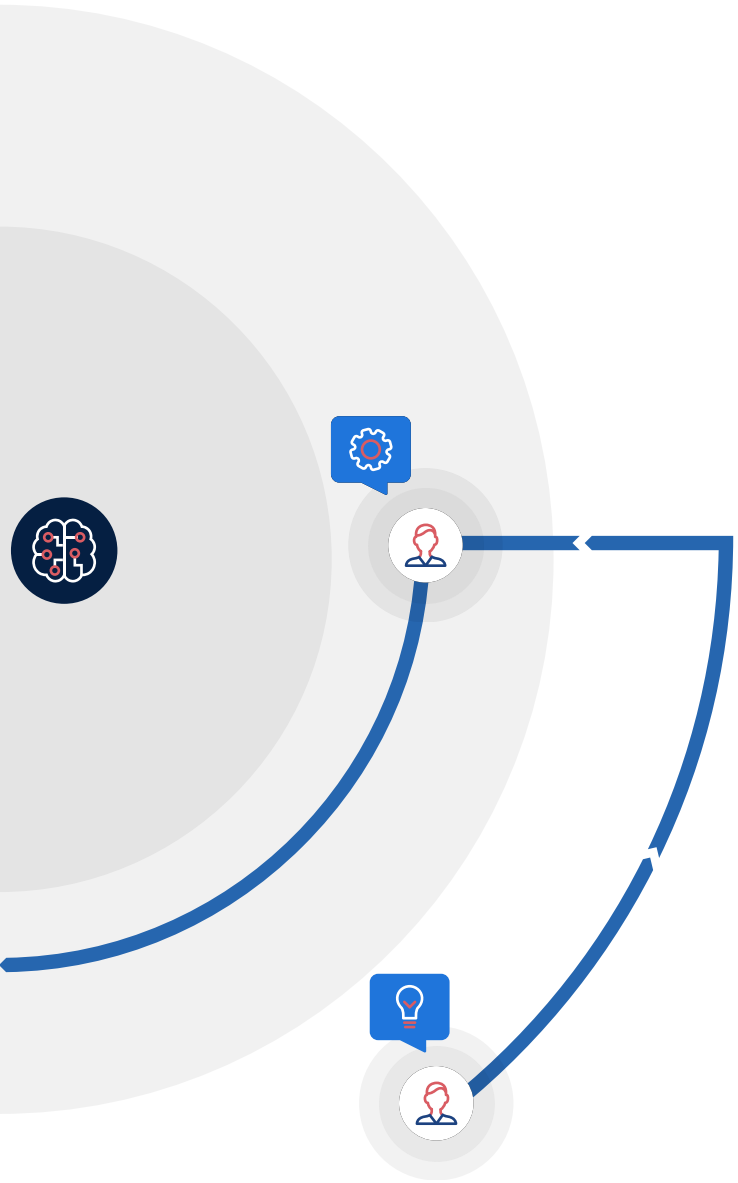
MUTT DATA



The Modern Data Stack:

A Technical Roadmap

Contents



Before We Dive In...

Skipped a beat? Missing some context? Don't worry. We've got you covered. If you're only taking your first steps in the world of Modern Data Stacks, make sure to check out our first whitepaper: Meet The Modern Data Stack, where we cover the following topics:

- What Is A Modern Data Stack?
- Why Implement A Modern Data Stack?
- Game Changing Benefits Of A Modern Data Stack
- Stages Of Modern Data Stack Evolution & Common Challenges Of Each Stage.
- Where To Begin: First Steps To Modern Data Stack Implementation
- What Mutt Data Can Do For Your Data Stack

Great! So you've covered the basics, you understand what a Modern Data Stack (MDS) is and what it can do for your business. Maybe you've even identified how evolved your current data stack is and now you are looking to make improvements to specific components. Before you get started, there are some challenges everyone should look out for when implementing or evolving their data stack.

Although you should strive to achieve a Modern Data Stack, we have yet to witness a fully formed Data Platform spontaneously appear. In reality, the road to a MDS is long and meandering, organically growing guided by the needs of the business.

The key to building a successful Data Platform lies in knowing when to expand, in which way and what challenges are going to appear along the way.

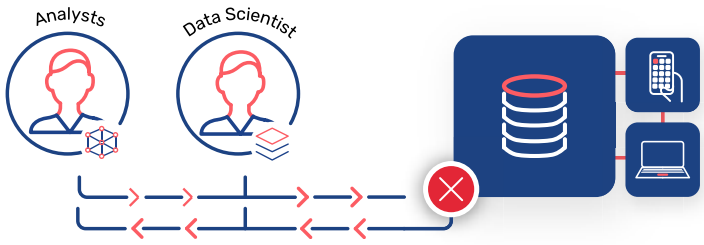
Use Case Fintech App

Let's picture a typical Fintech app: Fantastic Co. In the early stages, when iterations are fast and data volumes are low, a traditional relational relation database like PostgreSQL (our preferred choice) is the ideal option for handling the initial user load and persist state related to business events like: sign-ups, logins, searches, payments, users, products, purchases, etc.

Challenge #1: Scaling

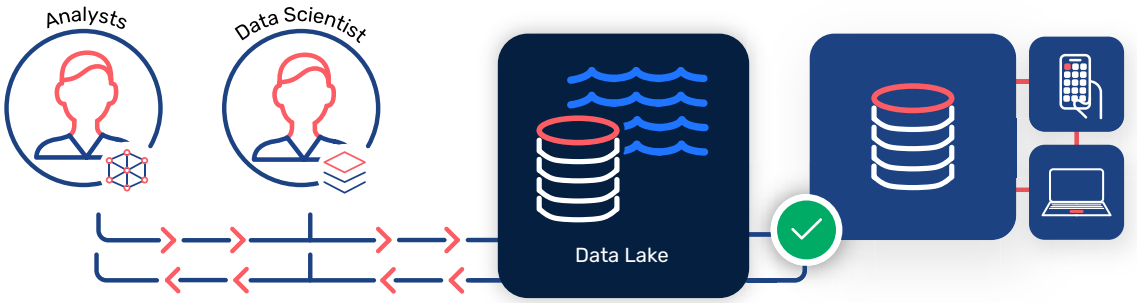
After a successful release, Fantastic Co's app starts growing rapidly in users and transactions, and so does the workload on your relational database. Furthermore, product managers and analysts will want to analyze the data persisted in the database to understand how users are using the app and each of its features. Either with SQL queries or through BI dashboards, running this type of analysis over all users or events usually means aggregating hundreds and thousands of rows in a single query to answer a particular business question. Analytical queries usually take a heavy tall on a relational database's performance since they involve operating over big amounts of rows with little use of indexes rather than operating over a small number which is what relational databases were designed for in the first place.

Initially, Fantastic Co. can vertically scale the database, by adding more resources. This will buy you some time, but there's a limit on how much a transactional database can realistically scale up.



Solution: Building a Data Lake

The next step for Fantastic Co. is to separate analytic workloads from online / transactional workloads **by building a Data Lake**. This will allow their transactional Databases (DB) to take care of the transactional business needs, while analytics can be performed by horizontally scalable distributed storage and **Massively Parallel Processing (MPP)** query engines.



Keep in mind: unlocking the power of distributed storage and separating storage from compute gives Fantastic Co. the potential to gather and exploit vast amounts of data, but it's not inherently efficient. If executed incorrectly, they can end up transferring huge volumes of data unnecessarily over the network which will drive up processing times and costs.

Designing useful data models and partitions (which requires a deep understanding of the data) will have a huge impact on Fantastic Co. ability to scale their operation[1].

1. Partition design can be a tricky thing to do: if you don't partition enough, you end up querying too much data; but if you partition too much, you end up with the Small Files Problem.

Challenge #2: Data Silos and Hard to Integrate Data Sources

As Fantastic Co. grows, new data sources start to pop up (Event Tracking, Product Analytics, Marketing Campaigns, CRM, etc…), and data becomes spread out. With time, as different teams begin to manage their data sources, data becomes siloed.

Fantastic Co. analytics teams will inevitably need to study this data, but they get bogged down by difficult-to-integrate data sources and poorly designed data interfaces. The MDS proposes to solve this problem by consolidating all these data sources into their Data Lake, with different ingestion tactics, such as ELT (Extract, Load, Transform), Change Data Capture (CDC) and using the Lake as a sink for streaming data.

Solution: Data Consolidation

Data Consolidation allows Fantastic Co. teams to query, consume and analyze data through a single data interface. It also helps to make their processes more efficient by materializing frequently used transformations into new tables and separating them into three distinct layers:

Raw / Bronze

This is the point of arrival of every data source. It's the least structured layer and typically only used as a source for the following layers, not for analytics as data arriving at this stage should be almost an exact copy of the source data. The tables within this layer are characterized for being partitioned by processing time, in order of arrival.

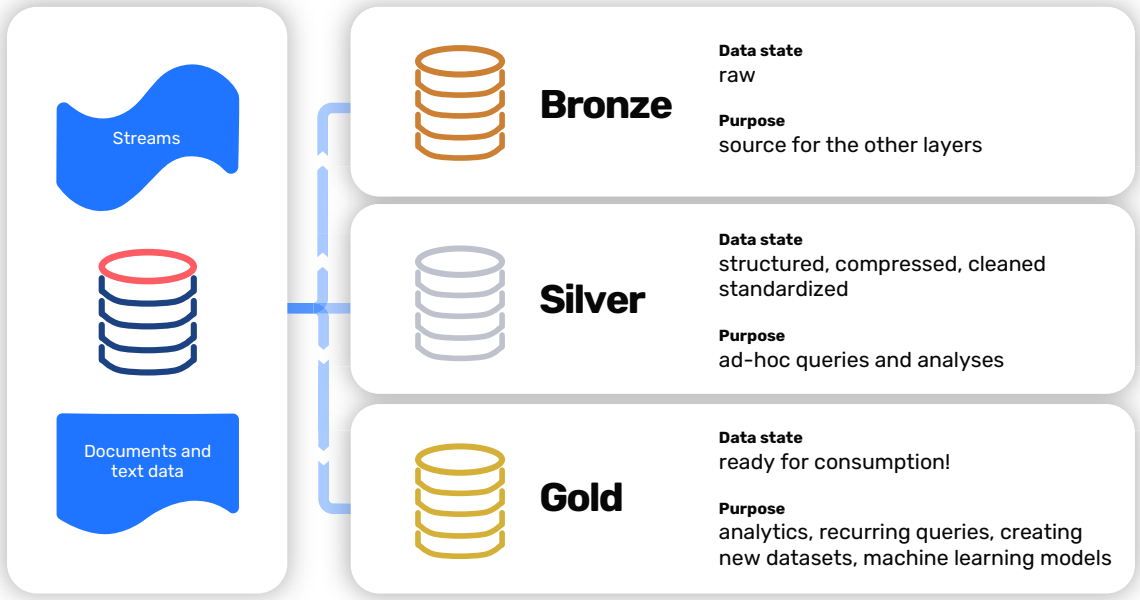
Staging / Silver

Within this layer data is structured, compressed, cleaned, standardized, and stored in a unified format (like Parquet). Typically, it also switches to event-time partitioning (from processing-time in the previous layer[2]).

This layer may be accessed by end users to perform ad-hoc analyses or experiments, but once a query to this layer becomes recurrent, it should end up as an ELT feeding a new table in the following layer.

Delivery / Gold

This is the final user-facing layer, where data is denormalized and joined according to the business needs and use cases (e.g.: analytics, dataset building, Machine Learning ABTs, etc).



2. This switch usually implies a one-to-many repartition, which can make the ELT that performs it non-idempotent, if not using tools like Delta Lake.

Why is this important? Having a consolidated and organized Data Lake will allow Fantastic Co. teams to easily and efficiently access the data, and build an ecosystem on top of it. This evolution of the Data Lake (usually coupled with tools such as Delta Lake or Apache Iceberg) is commonly known as a **Data Lakehouse**.

Challenge #3: Data Consistency

Now that all of Fantastic Co's. data is easily available, they can start to invest time in its quality. To ensure that, having processes that validate and monitor the quality of your data is key, but they also need to keep system failure points that can impact quality in mind.

One such point of failure is **dirty writes** (processes incorrectly overwriting data) and side-effects of processes that need to be run more than once (retries are inevitable in distributed systems so you should be prepared for them). If unchecked, this will contaminate your data sources with duplicate data and will be difficult to detect and deal with down the line.

Solution: Consistent Processes

A crucial concept to consider to avoid issues when re-processing data or retrying tasks is Idempotency.

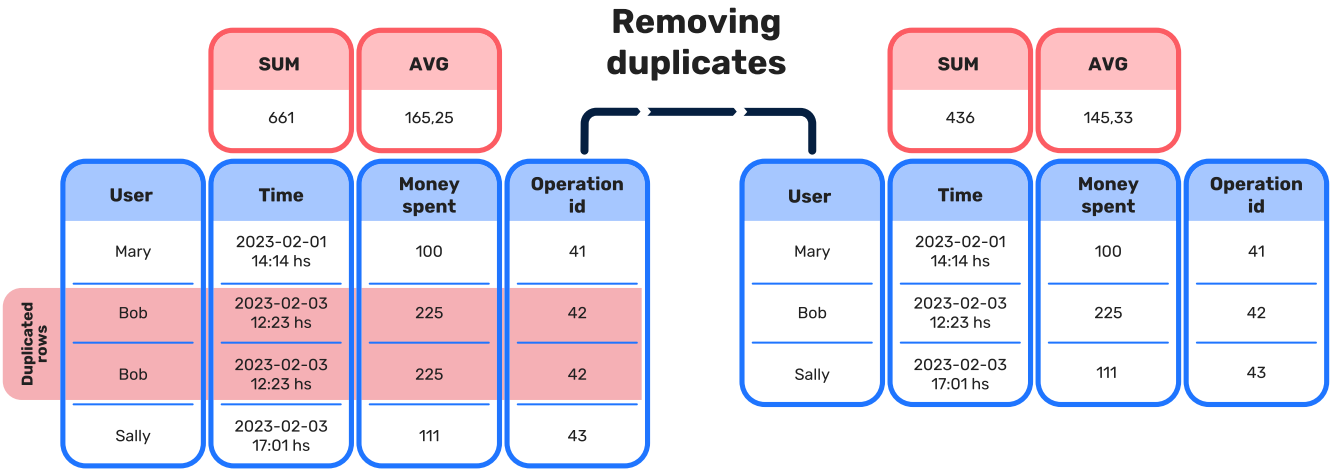
Idempotency is the ability to run a task, function, or operation multiple times without its outcome. **TLDR: If an operation produces different results each time it's run, your data stack becomes error-prone**.

Why is it important? It avoids inconsistencies, duplicated data, and poor-quality data in downstream metrics, which in turn, circumvents costly and time-consuming issues related to having to redesign processes or make decisions based on faulty data.

Ensuring process idempotence and atomicity will go a long way to avoid data duplication.

However, as Fantastic Co. considers data correctness and operation idempotency, they will also want to consider potential causes for duplicate data and how to avoid them. **Duplicate data** can arise for a variety of reasons such as scheduling errors, unclear definitions of source data boundaries, or retries when processing data.

Why is this important? Eliminating duplicates from data downstream can be a costly and resource-intensive process, as it requires additional storage, network bandwidth, and computational power. Additionally, the presence of duplicates can distort metrics and predictions until they are detected and corrected, which can have negative consequences for data analysis, decision-making, and machine learning models.

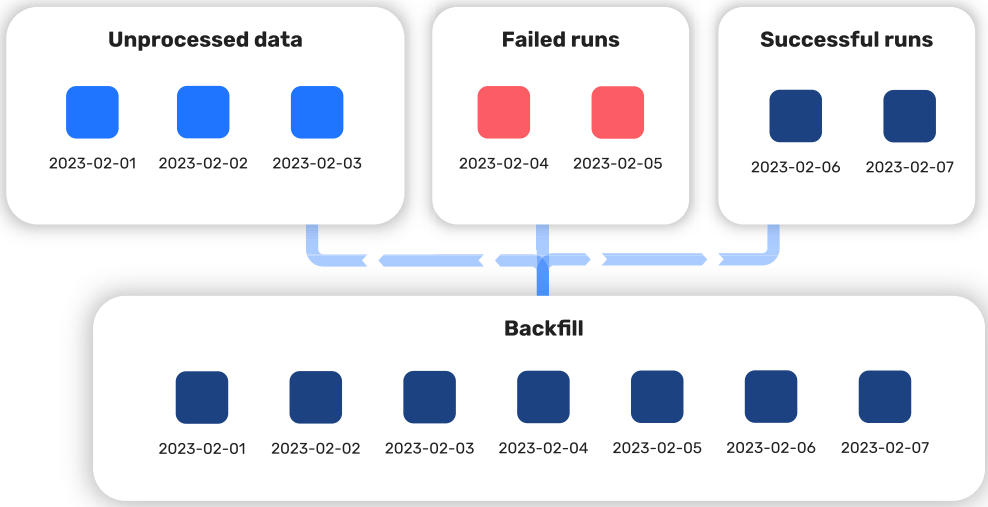


Duplication can be tricky to avoid in traditional Data Lakes (which don't support row-level UPDATE and DELETE operations), especially when one uses writing patterns that impede you from simply overwriting whole partitions. Luckily, Data LakeHouses powered by Delta Lake or Apache Iceberg, enable MERGE and other operations that prevent duplicates from being created, using a primary key.

When considering data correctness Fantastic Co. will also have to consider what "correct" means. As businesses grow, they must adapt to the ever-changing environment that surrounds them. What happens when the leadership at Fantastic Co. decides a KPI definition has to be changed or measured in a different way? Do past KPI values need to be updated? How does one do so? This is where the **back-fill ability** of your pipelines comes in.

What are backfills? The reprocessing of historical data using the same data pipelines due to errors, the evolution of business definitions, the need to retroactively run "older" data, or bugs.

Why is it important? Backfills will be needed eventually. Designing pipelines with this in mind reduces associated downtimes and costs of engineering hours and/or business decisions based on faulty, inconsistent, or incomplete data.



So far, so good for Fantastic Co., but consistent processes will only get them so far: once your system grows, they also need to build tools to monitor the health of your data processes and the correctness of their data using Data Monitoring tools. More info on our Components section!

Challenge #4: Building Data Products

All Data Products and Machine Learning Algorithms are subject to the Garbage In, Garbage Out law: one cannot have functioning ML systems with poor-quality input data.

It's a common pitfall to proceed to develop ML systems before having Data quality and Consistency assurances[3]. One cannot trust the output of an algorithm if one cannot trust its source. Furthermore, one cannot learn anything from said output: if it's not performing as expected, is it because of a problem with the algorithm or a problem with its input?

Once Fantastic Co. invests in ensuring the quality of its datasets, you can start to develop your Data Products, but the challenge doesn't end there.

Data Products usually start as proof of concepts and grow organically, adding experimental features and ad-hoc fixes. Before Fantastic Co. realizes, these PoCs, if successful, end up in production and start taking in more and more volume of data. That is, until they buckle under the pressure. Once they reach this point, they can end up having to invest a lot of time maintaining the system, layering fixes on top of each other, in a complex vicious cycle.

3. Rules of Machine Learning: Best Practices for ML Engineering de Google

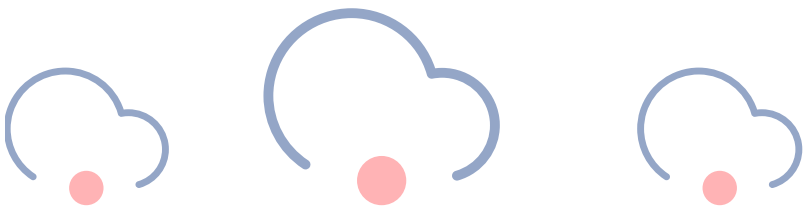
Solution: Proper System Design and MLOps

To fix this, Fantastic Co. has to turn the problem on its head: while the PoC was most concerned with exploring an algorithm's accuracy or feasibility, the Data Product needs to focus on system design, where the algorithm is a tiny (but core) part at the center of it. A redesign is needed to address the existing problems and focus on MLOps.

Interface design and modularity are two immensely important concepts to apply to a Data Product. An effective interface design can simplify the complexity of each module's inner workings in order to focus on the way they can be interacted with. In addition, they facilitate a better comprehension of the data flow through the pipeline, making optimizations self-evident.

Why are cross-system interfaces important? They sustain scalability, comprehensibility, and maintainability. Defining clear interactions and data exchange makes modifying or expanding the system easier.

The MDS allows systems to easily become data consumers and producers, allowing projects to go from simple products to **data products**. Clarifying the responsibilities of each component and its impact on the larger data stack improves the scalability and comprehensibility of the pipelines. This in turn makes it easier to understand and manage various components and their interactions.



Fantastic Co. After Implementing a Modern Data Stack

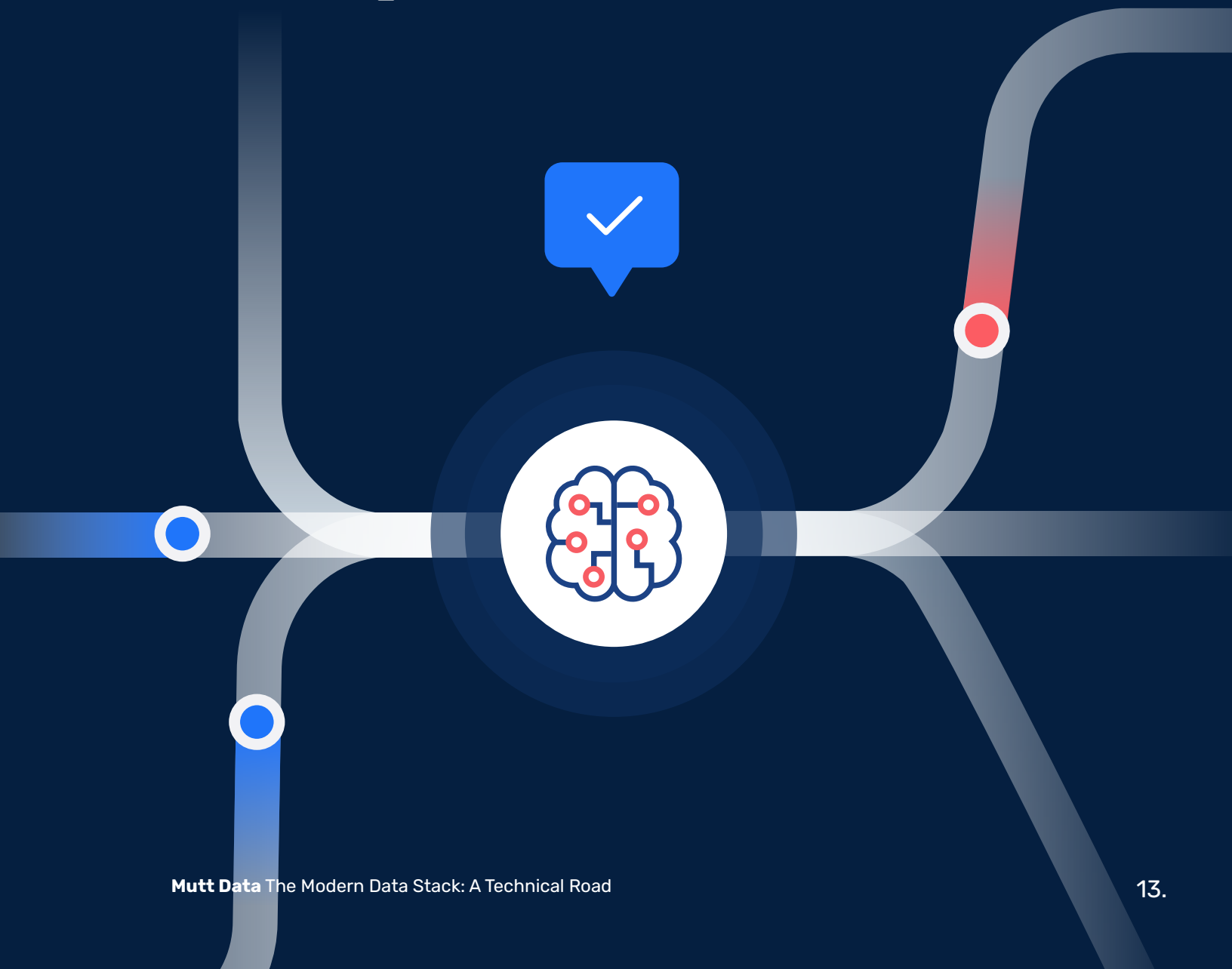
When Fantastic Co. began their data journey, they had only one relational database that couldn't cope with their analytical needs as they grew. However, by overcoming the challenges we mentioned and applying these solutions, they achieved:

- A Data Lake that accommodates both their transactional and analytical workloads, enabling data analysts and scientists to perform their tasks by querying multiple data sources simultaneously.
- Data consolidated into three different stages (Raw/Bronze, Staging/Silver, and Delivery/Gold), Fantastic Co. successfully prevented data silos and enabled their teams to query, consume, and analyze data through a single data interface.
- Consistent idempotent processes ensure the consistency of metrics by removing duplicates. They can also correct errors by backfilling data when necessary.
- Proper infrastructure and interface in place, which with monitoring facilitates the building of data products for the whole organization.

This is all great, but you may be asking "How do I build all of this?". In the next section, we explain all the components of the Modern Data Stack and provide recommendations.

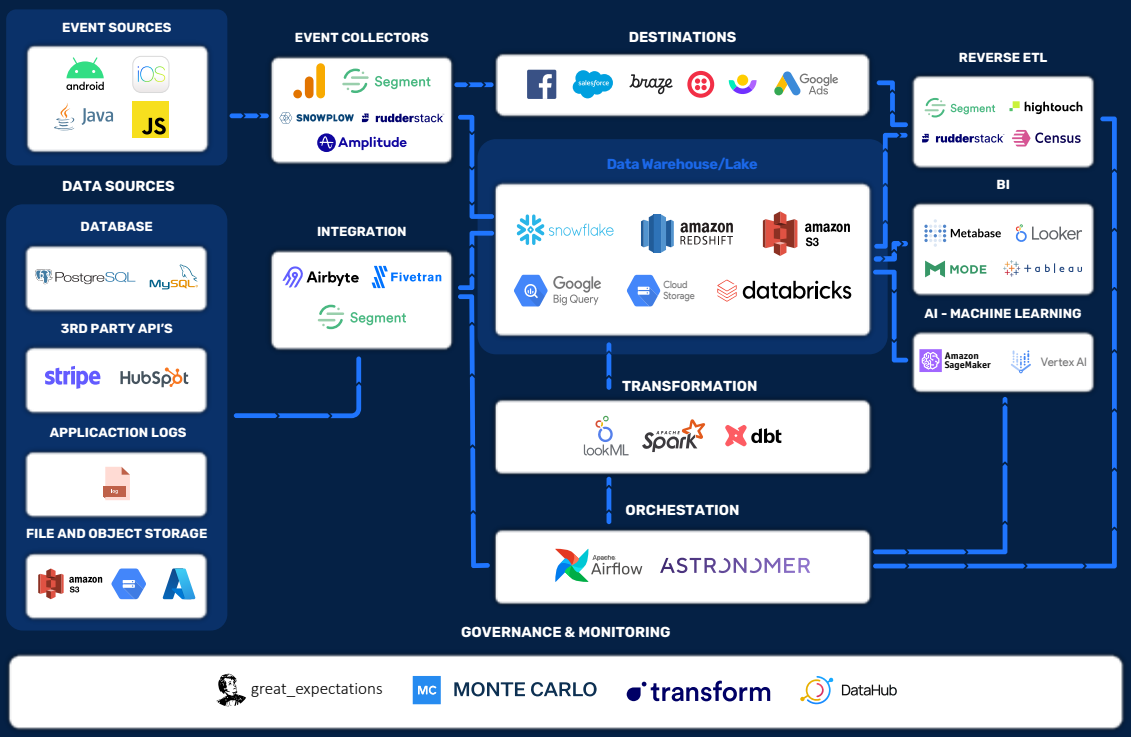
MUTT DATA
Machine Learning for Business

Breaking Down The Modern Data Stack: **Components**



The Modern Data Stack

Is made up of multiple components. Much like a band or orchestra, a single instrument can carry a tune, but it's not until you layer in all the other instruments (or components) that you access the full experience.



Having a proper data storage and engine solution is the core foundation for any data system. The choice of additional components depends on the specific needs and challenges of the organization, as well as its level of data maturity.

One of the benefits of a Modern Data Stack is that new **components can be easily integrated into the existing data systems as necessary.**

Data Source Collection

The Modern Data Stack runs on data, and the first step is to start collecting it. External tools such as Amplitude, Google Analytics, or Salesforce facilitate data extraction as well as the act of keeping it updated.

Recommended Tool: it depends!

If you're starting from the ground up, tools like Amplitude or Google Analytics might help you hit the ground running with collecting behavioral data.

If your business already has systems, transactional databases, services and applications in place, then you probably have more data than you realize: a relational database storing transactions is an excellent source of data.

Data Ingestion

Once you've located raw data in different sources, it needs to be moved and centralized somewhere it can be stored and used. Generally, the data you collect will go to Data Warehouses, Lakes, or Lake Houses. In the latter cases, this can function as both a source and a destination.

There are different types of data ingestion available:

- **Batch ingestion:** Data is collected and transferred to the desired destination when triggered by a predefined event or at a scheduled time.
- **Streaming ingestion:** Data is collected and transferred to the desired destination in real-time. In this case, the benefit lies in the continuity and the speed of data flow.
- **Hybrid ingestion:** Data is collected and transferred using both previously mentioned methods.

The type of ingestion used will depend on the data sources and data latency needs of each business. **Different Data systems and platforms can have varying formats, response times, and limitations.** This component allows users to effectively adapt to these differences to incorporate data and sources without wasting time and resources designing custom ingestion processes.

Recommended Tool for Batch Ingestion

Airbyte covers all critical functionalities:

- **Standardized ingestion layer** where origins and destinations are independent.
- **Possibility of scaling current connectors** or creating new ones by taking advantage of Airbyte's Connector Development Kit.
- Native **Slack alerts.**
- **Possibility of establishing different loading profiles for tables** (full refresh, incremental overwrite, incremental append).
- Easily **integrated** with industry-standard data tools such as DBT and Airflow.
- Native **DBT Transformations** such as normalization, deduplication, and the ability to define custom DBT transformations post-extraction and load (ELT).
- Airbyte's CLI, **Octavia, can be used** to integrate CI/CD processes.
- Strong and growing **open-source community.**

Recommended Tool for Streaming Ingestion

Even though Batch Ingestion is enough for most use cases when you **need to stream your data** you can't go wrong with Apache Kafka and its functionalities:

- **High Throughput:** with latencies as low as 2ms.
- **Easily scalable:** scale clusters to handle trillions of messages and petabytes of data.
- **Connect to anything:** with its connectors from and to Postgres, S3, and Elastic search.
- **Open source:** Apache Kafka is by far the most popular and used data streaming tool, with a very active development community around it.

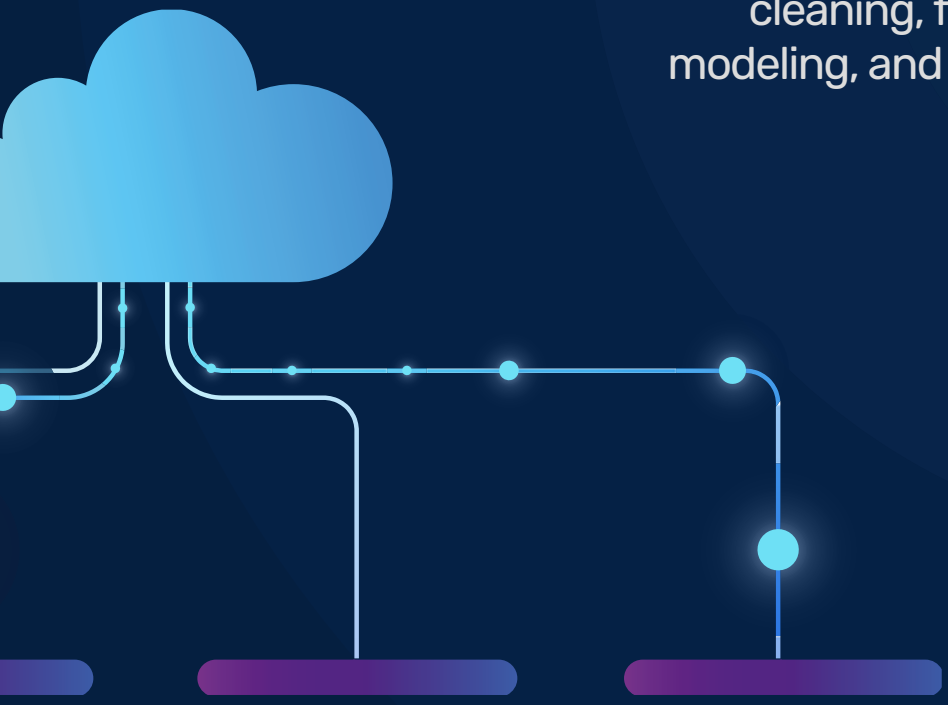
BONUS: most Cloud providers have their own Data Streaming Ingestion Tools, and for this, we strongly recommend Amazon Kinesis. This cloud hosted alternatives are fully managed, making it even easier to get started with Data Streaming.

Data Transformation

Raw Data isn't functional: needs treatment for analytics, machine learning predictions, data products, or system automation. This phase is where data goes from Bronze to Gold stage. Data is processed, transformed, and converted into useful formats to become compatible with your business processes.

The goal is to make customer and business information **useful to the organization.**

This phase will most likely include cleaning, filtering, normalizing, modeling, and summarizing raw or staging data.



Recommended Tool for Data Transformation

The benefits of DBT for Transformation are:

- **Large open source community:** frequently updated, and with an active community constantly sharing solutions and recommendations.
- **Low entry barrier:** (almost) entirely based on SQL, a common language for data teams, analysts, and software engineers. Includes more advanced functionalities with pythonic transformations.
- **Easily integrated** with metadata catalogs, orchestration tools, databases, data marts, and object storage solutions.
- **Data Documentation out of the box** enables navigation through models, definitions, dependencies, column descriptions, functions, and lineage graphs with a user-friendly website.
- **Software Engineering best practices:** teams can implement data quality in their pipelines and execute unit tests for their models on CI/CD pipelines.

Cloud DBT includes the following functionalities: Browser IDE, Job scheduling, Logging & alerting, automatically administered stored documentation, GitHub & GitLab integrations, source freshness reporting, API Accesses, single sign-on (SSO), Custom SLAs, professional services, Role-based ACLs, Fine-grained Git permissions, etc.

DBT cloud offers a SaaS version

Data Orchestration

Orchestration is how data pipelines are organized, automated, and coordinated across different systems and applications to make data available for end-users in the required format at the right time. Before orchestration tools, simple single-machine schedulers like CRON jobs were used to execute data processes but, with time, proved to be too limited. Data pipelines have become complex, replacing simple queries with numerous dependencies and steps. Simple join and insert operations have transformed into complex workflows across various sources and systems.

In recent years, modern orchestration tools have emerged to address legacy orchestration challenges such as:

- **Automatic retries** with configurable wait times.
- **Job success/failure** determination.
- **Parallel execution** of non-dependent jobs for success/failure determination.
- **Dependency checks** before workflow execution.
- **Monitoring with metrics** on job execution and workflow status across pipelines and steps.
- **Re-running parts** of the workflow and backfilling old data.
- Version control for **workflow definitions.**

Tools like **Airflow, Dagster, and Prefect** can eliminate manual work and reduce the risk of human error. Airflow was the first project of its kind and the most popular. It has the largest community and is sponsored by Astronomer, which means faster response to bugs, new features, and regular releases. It is also easier to find experienced or interested professionals.

Recommended Tool for Data Orchestration

Our tech partner, Astronomer, is a cloud-native orchestration platform powered by Apache Airflow. It offers the benefits of Airflow plus seamless deployment, scalability, enhanced visibility, reduced operational times, and expert support. They also provide a myriad of resources and expert certifications.

The benefits of Astronomer are:

- **Based on the Industry- open-source tool: Airflow** is widely used, regularly updated, and continuously improved by a diverse group of top-tier organizations as seen here.
- Contributed by thousands of members, **kept up-to-date and error-free.**
- **Low entry barrier & easy set-up:** Uses the most popular programming language for data teams: Python and it's adaptable to both Cloud and on-premise.
- **Friendly UI:** Anyone with access can re-run jobs, check logs, troubleshoot for issues, and access metrics such as performance, execution times, process status, and execution tries.
- **Easy to integrate:** allows interaction with SaaS platforms, messaging brokers, databases, Data warehouses, Data Lake houses, object storage, serverless compute services, and container orchestration tools, among others. Astronomer also facilitates the execution of tasks in commonly used providers such as Google Cloud, Amazon AWS, and Microsoft Azure.
- **Scaling:** Systems can scale horizontally as tasks or processes are added to the workflow. Also, DAGs can be dynamically generated, and adding a new DAG for your latest product requires no coding, allowing anyone to schedule ETL jobs.

Data Quality

How can you be sure you are not working with faulty, incomplete, or duplicated data? The more people start interacting with and adding new data to a Data Stack, the harder it gets to ensure its quality. As a component, Data Quality involves the automatic processes and tools to ensure that data is fit for its intended purpose: **from ensuring its validity and completeness to ensuring its consistency.**

Poor quality data can lead to flawed decisions or broken data pipelines, ultimately resulting in increased costs. Analysts have to spend hours figuring out why their reports are incomplete or inconsistent, and then engineers have to spend even more hours figuring out what was causing this faulty data and fixing it.

If something is off or out of place, timely identification can save the organization time and money.

The Modern Data Stack makes this easy with **integrations to Data Quality tools.**

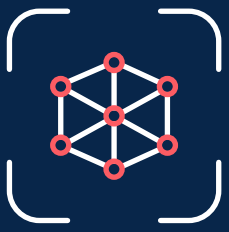
Recommended Tool

Soda is an open-source tool that enables everyone in an organization to improve data quality. Its web interface makes it easy to create and reuse data quality rules to ensure data consistency. **It also includes built-in metrics to test and identify issues.**

Data Governance, Discoverability and Observability

This component is concerned with enabling teams to easily find, understand, and access the data they need for analysis.

The problems this component solves fall into three categories:



- **Data Governance:** Who has access to what data? Who is responsible for keeping it healthy? What are the access and compliance policies?
- **Data Observability:** Who creates data? Who consumes it? Who uses it? How does it impact different users and areas? How is this flow visualized?
- **Data Discovery:** Where can the data be accessed? How can different teams use it?

> **There is a way to answer all of these questions:** metadata, data about data. The way to organize and share this metadata is through Data Catalogs.

Recommended Tool for Data Cataloging

The benefits of DataHub are:

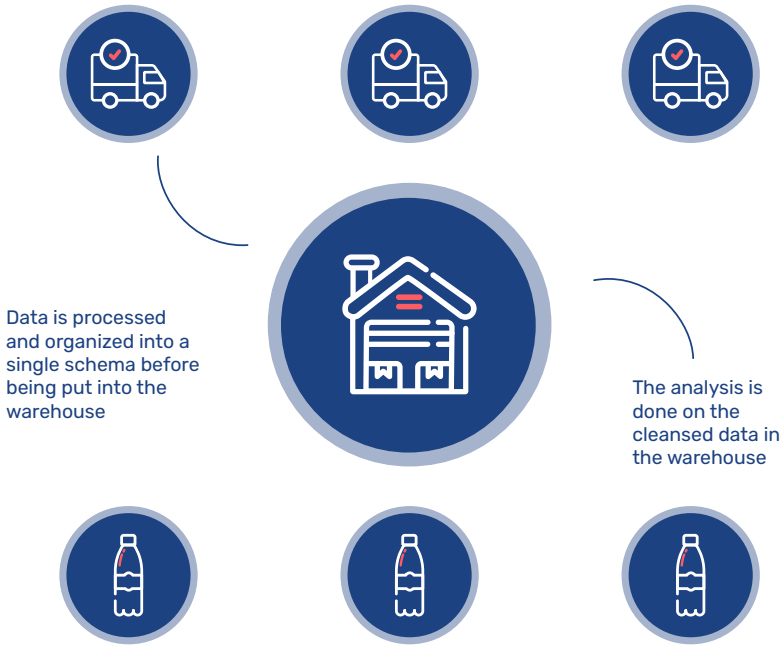
- **Data Governance:** DataHub versions metadata by **default**. When changes are made, the old version of that metadata is stored. It's possible to audit a system: if a user got their permissions changed, even for a moment, it's detectable. Thanks to metadata tags and its glossary, one can assign meaning to data, define DRIs, and set access processes.

- **Data Observability:** DataHub offers *lineage visualization*. Many data products consume data from different sources, and in turn, these sources consume from even more data sources. Visualizing the data lineage allows teams to analyze the impact of potential changes in data, backfills and broken pipelines. Furthermore, when debugging, a team can check their lineage, from end to end, to find the cause of the bug.

- **Data Discovery:** Metadata allows users to add tags about datasets and comments on fields. DataHub enables users to search for users, people, and other assets.

Data Warehouse or Lake

The Data Warehouse, Data Lake, or Data Lake House, whichever a business may select for its storage component, is a **data repository, and it sits at the center of every Modern Data Platform.**



When it comes to this component, there are several leading, industry-standard, tools that are widely respected for their scalability and versatility. These include Redshift, Big Query, Snowflake, and Databricks. However, there are also other contenders to consider, such as Firebolt and modern OLAP [1,2] systems like Pinot and ClickHouse.

For a deeper comparison check out our blog post describing Data Warehouses, Data Lakes, and Delta Lake.

Metrics

The goal of the metrics layer is to standardize how metrics and KPIs are created, calculated, and used to achieve consistency across the organization.

Think of a single source of truth that guarantees that different teams, using different tools, will be applying the same business metric logic for their analyses.

Recommended Tool:
MetriQL & Cube.dev

The field of "headless BI" and metric storage is still quite green, but Open Source tools like MetriQL and MetricFlow are under very active development.

They have a good amount of integrations to the most commonly used tools from the Modern Data Stack like DBT or BI Tools like Tableau.

If you're looking for customer support and shorter development time, **cube.dev is the most promising paid tool** and has very accessible pricing.

Business Intelligence

This phase consists of **business tools that enable analytics, reporting, and dashboards for end users to support decision-making without the need for technical know-how.** Almost all BI tools can be integrated within a Modern Data Stack, but to choose the right one it's important to understand the audience.

Users may want to have drag-and-drop components instead of having to write SQL Queries, or maybe their focus is on efficiency, speed, and being able to connect to other tools and publish new datasets.

With that in mind, it's important to consider the following tradeoffs:

- **Raw vs Aggregated Data:** raw data offers the most flexibility, but is also costly. On the other hand, pre-aggregated data offers improved performance but limits flexibility to the aggregations computed and the metrics that can be analyzed.
- **Serverless vs Fixed Instance Count:** serverless systems provide "unlimited" scalability supporting any amount of queries and users, making it the right fit for teams with irregular access to data. On the other hand, having a fixed amount of instances won't allow your system to scale infinitely, but it's quite cost-effective, which makes it the right fit if there is a low-medium variability in the number of users and data being queried.
- **External vs Internal usage:** external dashboards that share key metrics and reports to clients need to ensure security, data privacy, quick load times, and up-to-date data. This requires building a more robust and scalable system around the dashboard.

Recommended Open Source Tool

Both Metabase and Superset are Business **Intelligence tools with user-friendly UIs**, making it easy for experts and newcomers to create simple visualizations. This simplicity and ease of use come with the cost of not being the appropriate tools for more advanced and custom dashboards.

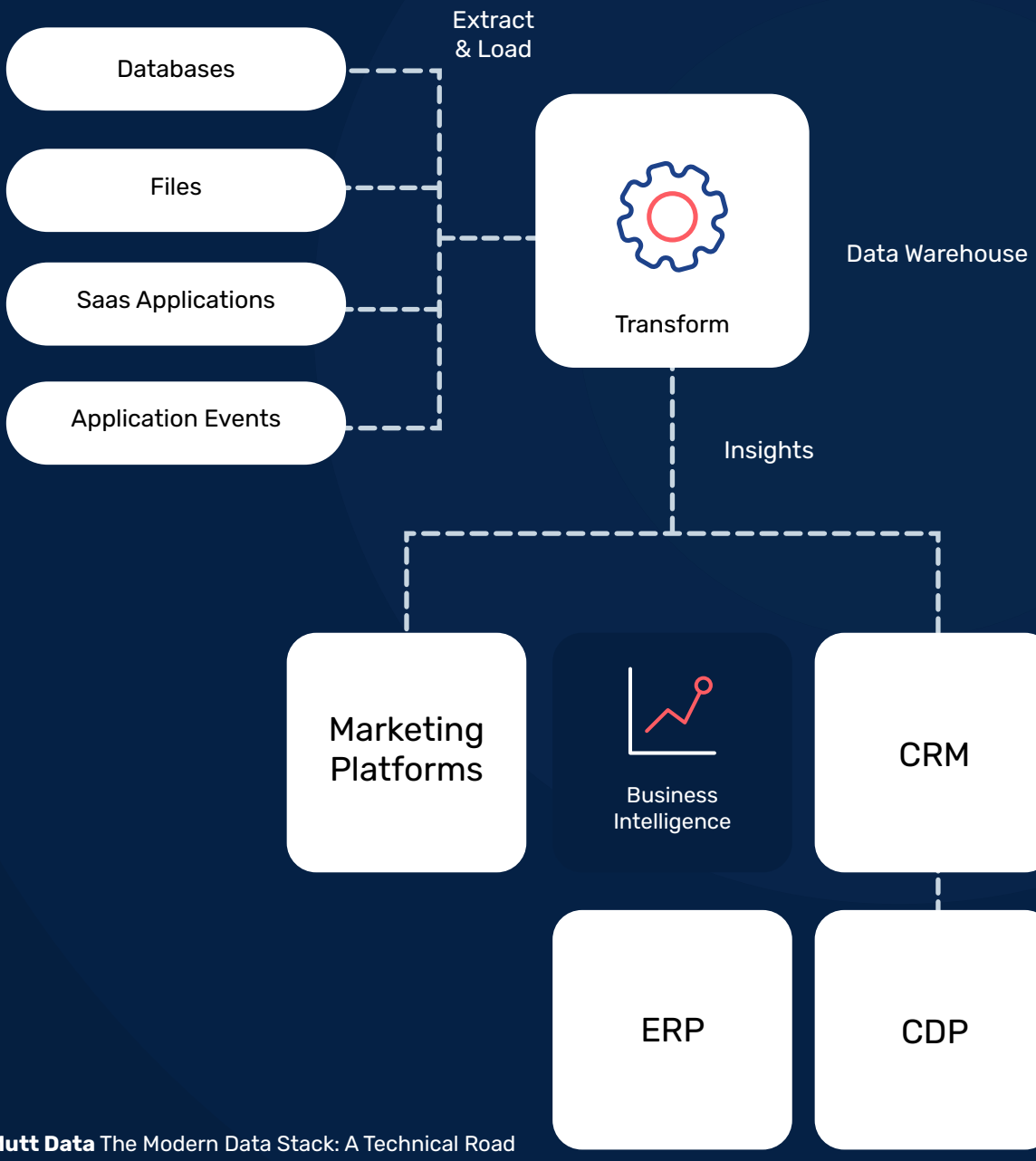
Recommended Cloud Tool

When working with cloud providers like GCP and AWS, it might make sense to use their own Business Intelligence tools. Both Looker (GCP) and QuickSight (AWS) **have many advanced capabilities and can scale up to any amount of users and data.**

Reverse ETL

In reverse ETL, data is extracted from the data warehouse or Data Lake, transformed inside that same warehouse (basically meaning it's changed into a Gold state, reaching the format needed by the third-party systems used in the company), and then loaded into that third-party system for insights, action, etc.

Find more information on this component in our FAQ.



Recommended Tools

Grouparoo **is an open-source reverse ETL tool recently acquired by Airbyte** (the tool we suggested for ingestion). It's the most active open-source Reverse ETL tool today and has great integration capacity.

If you're looking for a mature option to get you up and running in no time, Census and Hightouch are solid paid options. Both offer free trials with limited functionalities to help users decide.

What Can We Do For Your Stack

AI & Machine Learning implementations can be quite challenging and failure-prone. Companies spend significant amounts of time and money implementing these solutions. Industry statistics show AI implementations have been an issue since 2019, and they're still one of the most common pain points in business today.

Modern Data Stacks serve as a robust base upon which we can leverage the power of machine learning and AI to advance our capabilities.

We've climbed the Modern Data Stack mountain on many occasions. We are experts at planning, organizing, developing, and nurturing the necessary teams, capabilities, tooling, frameworks, and best practices for the climb.

Years of experience have allowed us to develop a sixth sense. We know where to look. We predict unforeseen consequences and prepare your company for what you will need in years to come.

> **Most artificial intelligence (AI) projects fail. About 85% never reach deployment.**
>
> Gartner

Strategic Consulting

We don't believe in one-size fits all solutions: Every business is unique. Working with Mutt means working with an expert team of in-house data engineers, scientists, mathematicians, and business #DataNerds that take the time to understand your specific needs, goals, and constraints.

Our team will analyze where you stand today, what your options are, and the best way to reach your goals, making the best possible use of your resources.

Accelerated Time to Value

Our experience building and implementing Modern Data Stacks for different companies in different industries have led us to a collection of tried and best practices and baseline structures that allow us to leapfrog your developments shortening time to value.

Knowledge Transfer & Best Practices

We focus on delivering a robust modern data stack with tech capabilities that last. We don't simply deliver a product and leave our clients to their own devices. We transfer our knowledge on best practices and processes, upskilling the client's team so they can sustain the solution in time and adapt or scale the system if necessary.

Reduce Data Processing Pipeline Implementation from Weeks to Days.

Get Started Today

Is your organization struggling with your current data stack? Are nice-to-have components becoming a necessity? As your business grows, so will the number of sources, incoming volumes of data, and the complexity of its use cases.

Developing and implementing a tailor made modern data platform can be the gamechanger you need to boost your business. Why? The foundation of any data-driven organization is to have a healthy flow of, and easy access to trusted, integrated, and consistent data. The Modern Data Stack enables companies to manage their data from ingestion to visualization saving time, money and effort.

Diagnose the state of your stack:

- Do you trust your data? Are errors hard to identify and reprocess?
- What's the state of your data infrastructure, is it easy to monitor? And how fast can the data pipelines recover from errors?
- Are data engineers over burdened with maintenance issues and business queries and demands?
- Does your business have a centralized self-service source of insights that business users can easily access to make decisions?
- Are machine learning teams sharing a centralized repository of user data or does each project need to generate user features from scratch?
- Are KPIs across teams and departments easy to answer, is the information readily available? Are basic insights easy to come by?

Find out more about what a robust, tailor made Modern Data Stack can do for your specific use cases and goals.

GET IN TOUCH WITH AN EXPERT TODAY

Success Case: Classdojo

ClassDojo is a global community of 50M+ teachers and families who come together to share kids' most important learning moments in school and at home—through photos, videos, messages & more.

Challenge

Classdojo needed a self-serve Modern Data Platform to empower engineers and analysts to make changes efficiently. The company faced technical debt, ETL issues, duplicated extraction processes, troubling data ingestion, and a general lack of testability with no monitoring in place in input and output tables.

The goal was to give every vertically-integrated team ownership over its metrics, tables and data.

Solution

Mutt Data deployed an expert, hands-on team of High-Performance In-House Data Engineering Mutters' to design and implement a Modern Data Stack, its tooling, pipelines, components, and integrations tailor-made to their needs, that could serve the whole company.

EXPLORE THE FULL CASE

ClassDojo

"Mutt's expertise in building Data Ops platforms, Apache Airflow, data warehouses, and best practices for developing data pipelines led us to a cost-effective, top-of-the-line data architecture."

Dominick Bellizzi
Chief Technology Officer

30 %
reduction in data pipeline processing and data delivery time

10 %
increase in Data Warehouse efficiency

Success Case: Ank

Ank is an Argentine fintech that developed a product for money transfers between bank and virtual accounts. It allows you to manage several accounts from different banks from a single app and facilitates quick transfers between them.

Challenge

Ank's team needed to fulfill three main objectives: create an automated flow of reconciliation transactions to reduce manual work whilst improving the accuracy and trustworthiness of the process. Second, increase the frequency of the reconciliation process to obtain updated data. Finally, implement a batch data processing architecture based on best practices and technologies that could serve as a model for future pipelines. Together, these needs could be solved by the implementation of a Modern Data Stack.

Solution

Mutt Data developed a system for automatic reconciliation on an hourly, daily and weekly basis. The system extracts the information from the different sources, centralizes it in a Data Lake in Amazon Web Services, performs the relevant reconciliation checks and, if it detects inconsistencies, corrects them automatically or creates tickets for cases that require manual attention.

EXPLORE THE FULL CASE

Ank

"In Mutt Data we found a partner with extensive experience building data systems from design to production. They worked seamlessly with both our technical and business teams.

Their know-how on building modern data architectures on Amazon Web Services (AWS) was fundamental in developing a reconciliation system which has proven essential to efficiently scaling our operations.."

Gustavo Arjones
Chief Technology Officer

The client saw manual conciliation processes reduced to under 3% of their total monetary transactions.