REACTOR

Future-Proof your Cloud Data Infrastructure

A Reference Framework

Jared Stiff, CTO, Co-Founder
Rachel Workman, VP Value Engineering
Eric Best, CEO, Co-Founder

CONTENTS

Purpose of this eBook

This Reactor eBook is intended to provide a roadmap to plan and design your future-proof cloud data infrastructure, whether you build it from scratch or leverage emerging platforms like Reactor.

Our hope is that this framework **helps you design and implement useful data capability fast while building long-term relevancy for engineers and non-engineers alike.**

Message from Our CEO

Building upon the foundation of elastic cloud or utility computing, **the modern data stack has revolutionized the way organizations and data practitioners approach data infrastructure.** Modern cloud data warehouses like Snowflake and Google Cloud BigQuery provide the core of a flexible, or "composable," data stack – while innovative software providers have surrounded these data warehouses with useful engineering tools to facilitate critical functions like data ingestion, data modeling, generative AI algorithms, business intelligence reporting and visualization, and data activation. The era of cloud data engineering has arrived.

The next era will be defined by the democratization and consumerization of the modern data stack. Confining data work to engineers constrains the value and utility of data. On the other hand, those organizations that enable every stakeholder – not just data engineers – to build and consume data assets hold a distinct competitive advantage in the marketplace.

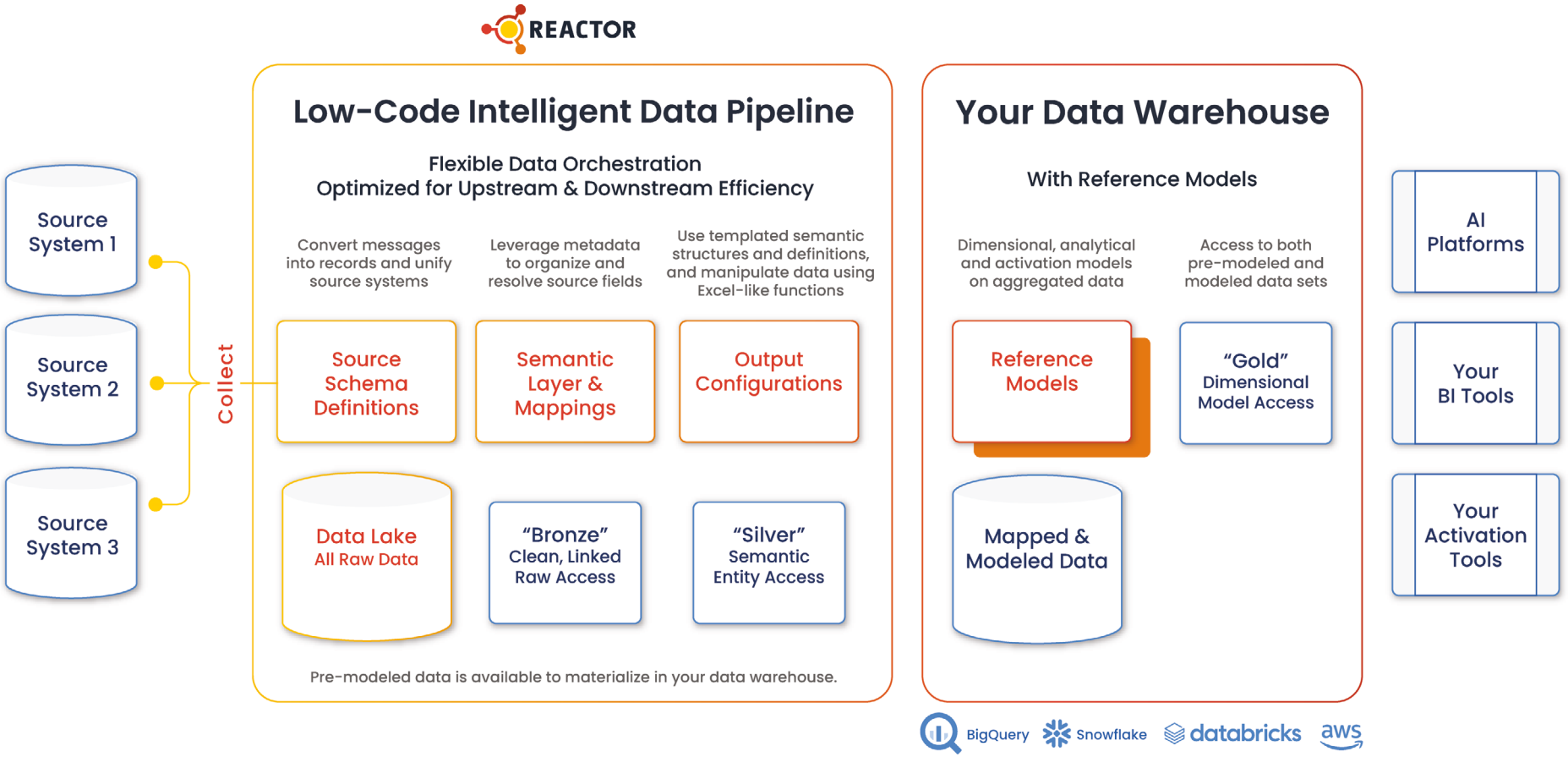
As organizations democratize data, it proliferates. Compute and storage costs rise. Shared understanding of data and metrics, data security and data governance become exponentially more complex. Successful data infrastructure, built to accommodate the needs of both engineering and business stakeholders, requires an intentional architectural approach, distinct capabilities, and unique features – to contain cost and complexity as data and use cases scale.

Eric Best
SoundCommerce Founder and CEO

Introduction: The cloud changes everything

Cloud data warehouses can transform the way you run your business, revealing the drivers and detractors of profitable growth. But cloud data warehouses can also become expensive dumping grounds for unusable data.

A useful and cost effective data infrastructure requires more than just a data warehouse filled with raw data, dependent upon brute-force data engineering to map and model data into useful business output.



In this framework we present nine important characteristics of a functional cloud data architecture, with specific features and capabilities designed to:

- Increase time-to-value on data infrastructure projects by reducing time, cost and risk to get to business-ready data, and increase the impact of useful data on business outcomes
- Reduce the cost of software licenses, storage and compute – especially incurred by the data warehouse – by optimizing both upstream and downstream data flows
- Shift the focus of data engineering and analysis teams from plumbing to driving high-impact business outcomes

The nine characteristics of a future-proof data infrastructure include:

1. The local, immutable logging of raw data;
2. The ability to reinterpret data in the future to support unexpected and unplanned use cases;
3. Shared understanding of data across your organization and partnerships;
4. The early and complete semantic labeling of data for shared understanding in the data warehouse and across stakeholders and use cases;
5. The structuring of data sets at the (atomic) field rather than message level;
6. The democratization of data pipelines, flows and models using no code/low code interfaces;
7. Reusable investment in data flows and models through shared code libraries;
8. Streaming data flows for real-time applications and use cases;
9. Rapid value creation through the use of industry-specific, pre-built but customizable data models.

1 Make it Permanent: Log your Data for Immutable, Local Reference

In arguably the most iconic scene from **Bladerunner**, replicant Roy Batty describes his personal memories as **"lost in time, like tears in rain."** Until immortality is invented, we'll have to settle for solving the same problem in data enablement.

Actionable data lost to time. How are we still talking about this? With incredible advances in data storage and processing, cloud-native solutions supporting streaming ingestion, and convergence of the data warehouse and data lake into the big-data-analytics-ready lakehouse, how can this possibly still be a challenge?

One way to look at it is that we are going against nature itself. Our very brains rarely try to store information for which they haven't already identified some specific purpose, and when they do, our ability to use information in a meaningful way degrades, so perhaps we're asking to eat our cake and have it too?

For actionable data not to be lost to time, we need reasonable ways to ingest and store data as it is generated, regardless of whether or not we've defined a specific purpose for it yet, and we need to store it in such a way that:

- Allows for fault-tolerant ingestion
- Stores all the data (full history)
- Leaves the data highly representative of its source
- Keeps the data immutable and auditable
- Makes the data accessible in the context of when it was ingested

Technology advances are making all of the above possible, yet going against nature doesn't come cheap or easy. There are a few ways to go about getting this done, but one way is rising to the top as the most cost-efficient, flexible, and forward-thinking solution.

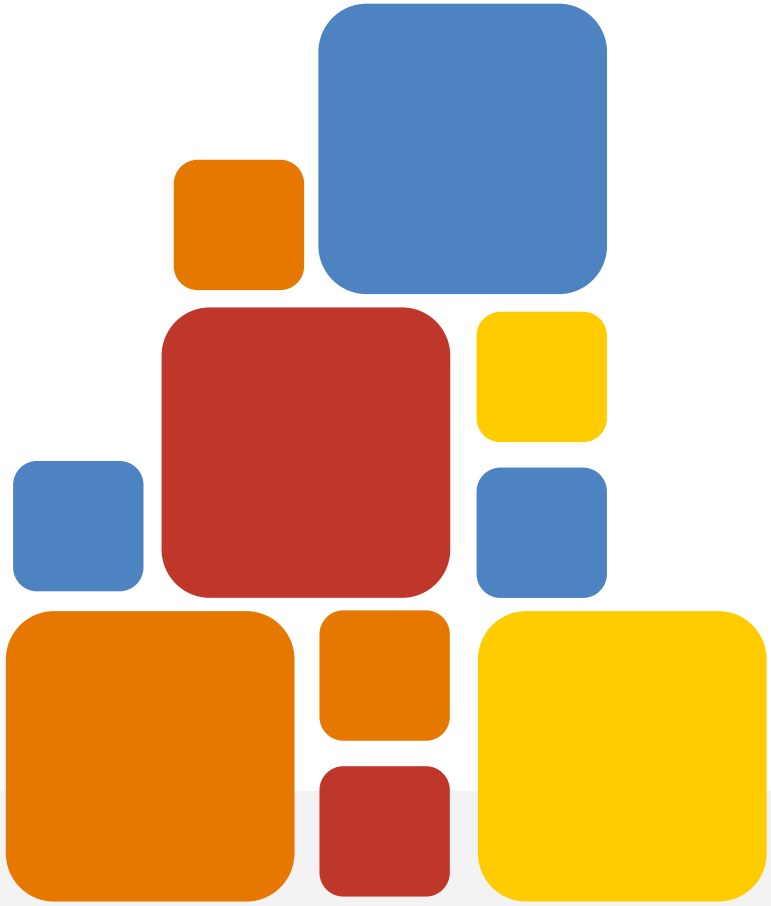
Challenges Posed by ETL

The various macro building blocks for this solution are not new. A data warehouse, with its structured, business-consumable data marts, is excellent for supporting business intelligence and reporting needs. A data lake, with its ability to ingest and store vast amounts of raw data, is cost-effective for storing data ingested from a source that hasn't yet been earmarked for a business purpose as well as for storing data in raw form for experts to access for AI/ML and big data analytics. And those two constructs, as noted earlier, have already in many cases merged into a single solution: the data lakehouse. Most cloud-native data warehouses support big data, and most cloud-native data lakes support business intelligence.

Yet we are still left with two key challenges:

Storing data in a way that makes it useable now or in the future is difficult and costly

Storing data only at rest restricts future flexibility to support real-time use cases



The first challenge goes back to our need to defy nature itself. **To make data usable for business consumers, it needs to be structured**, which is why the data warehouse with its business-specific data marts persists in its key business role. To structure data, we must make decisions about purpose. Data without purpose doesn't have a place in structure.

Prior to ELT becoming a viable alternative to ETL, the decision of what data had purpose occurred when retrieving data from the source, as only data with purpose was retrieved, transformed, and stored. However, in many cases, ETL-based data pipelines were built with at best little thought to future reusability and at worst ad hoc design to satisfy isolated functional needs. Notably fragile in many cases due to cascading effects of batch processing, it was not uncommon for some data pipelines within one organization to be wholly or partly redundant, apply varying transformations for derived values, or have pipes that had been abandoned without any official decommission as business needs continued to evolve. Data not retrieved from source was lost in time as sources changed or purged data of a certain age.

The rise of the data lake and its ability to accept data without structure permitted us to switch around the order from ETL to ELT and load directly into the data lake prior to any transformation. Data at rest in the data lake could then be transformed at will as business needs arose. As all available data could be pulled from the source, in theory, no data was lost. Unfortunately, as often happens, theory translated into practice gets messy, and the explosion of data landing in the data lake in raw form left many organizations with data that was not able to be audited for compliance and also not able to be accessed in the context of its source and origination time when a future need for it arose. This left many organizations with what came to be known as virtually unusable data swamps.

To make that data usable, many organizations evolved by implementing the aforementioned lakehouse and their various modeling overlays, such as Delta Lake or Data Vault, applying light but necessary structure, at least enough to ensure auditable compliance requirements (e.g., ACID) and an ability to access data in the context of its source and origination time when a future need for it arose. Additional modeling, however, even when applied to raw data, comes with overhead and cost. Expert resources are needed to carefully construct these complex models, and the larger teams, including business subject matter experts, need to be trained in complex concepts since even data ingested into a data vault still requires additional modeling to make it business consumable, and centralized data teams cannot be expected to understand all domain data to the point needed to make it usable.

So, with a data vault (or similarly) modeled data lake (house), we are at least at a point where we don't lose any data to time, though with a tradeoff of greater cost. We still have our second challenge, however, and that is data stored at rest in a lakehouse can't enable real-time use cases.

Is there a way to resolve both the greater cost and inability to flex to real-time usage by taking one more step forward in modern data architecture?

Dump the Monolithic Architectures, Adopt ETL

That answer, of course, is yes. Just as continually changing business needs required us to find a way to make sure all data lake data was auditable and usable no matter when accessed, the continually increasing need for real-time data demands we move away from monolithic architectures that don't support real-time use cases.

Wait! Move away? Many organizations haven't even been able to establish lakehouses yet, let alone move on to the next thing. The good news is that, in this case, moving away from a monolithic architecture is additive. The data warehouse/data lake/data lakehouse is still very relevant because data at rest is needed to enable reporting, BI, AI/ML, and big data analytics. However, making an architectural change to add the ability to also process data in motion (not just use streaming for ingestion), can:

- Reduce overall storage costs
- Reduce/eliminate the need for modeling overlays for raw data
- Enable real-time use cases
- Maintain high data fidelity and audit-ability
- Enable full data replay

How does this happen?

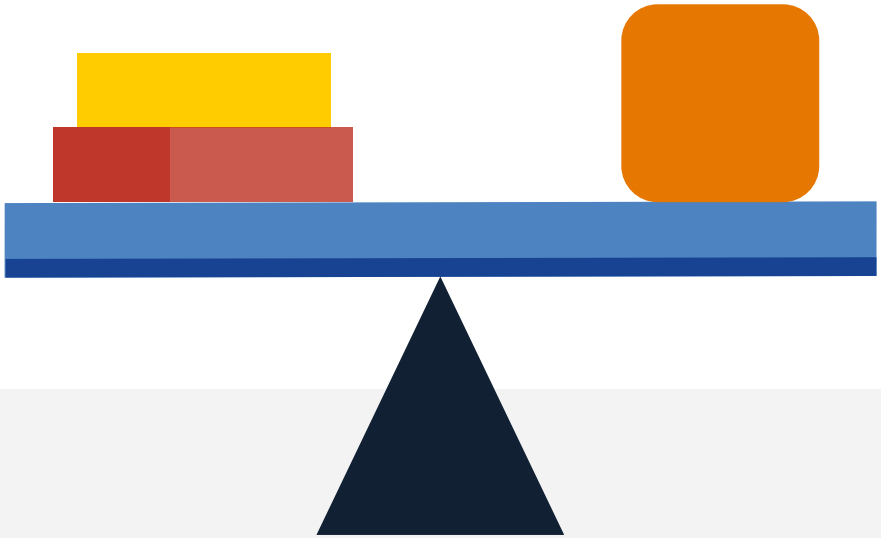
Well, adding a technology like Apache Kafka as a core component to your modern data architecture gives you more flexibility in how you "land" data. Data no longer has to be landed in totality into raw data zones with modeling overlays applied in order to store "all the data" in a usable and governable format. Kafka logs are natively immutable and shippable. All logged data persists with high fidelity in cost-efficient, replayable storage. Because processing data in motion is its core capability, only data with an identified purpose needs to be placed in semi-structured/governed storage for further processing, reducing overall complexity along with the storage cost. Having the data accessed and processed in motion opens up the new capability to interact with real-time systems and use streaming ETL, even while not restricting existing interactions with batch-based systems and storage.

Further, for those organizations that haven't yet made the move to implement lakehouses or other raw-data modeling, making a move to this type of architecture could alleviate the need for this heavy lift while remaining overall cheaper and more in line with evolving architectural concepts like domain-driven design and data mesh. For those who are even earlier in their journeys, taking this direction can avoid the pain experienced by other organizations but that resulted in the key learnings that led us to this point.

Make it Immutable

Organizations can avoid losing important data to history and entropy by leveraging modern cloud infrastructure and the right enterprise architectures.

Until humans are directly linked to, or even part of, the cloud – perhaps by 2049? – we'll have to settle for the business benefits that come with modern data architecture.



2 Let's See that Replay: Stage your Data for Future Remodeling

In one of the most underrated and better action movies of the last decade, **Edge of Tomorrow,** Bill Cage (played by Tom Cruise, of course) is forced to replay a segment of his life over and over and over. Though this replay isn't voluntary, he quickly realizes he can do things differently on each replay, using trial and error to meet his objectives, ultimately saving the human race from being conquered by an alien civilization.

Imagine if we could do that for data enablement.

Though Reactor isn't taking on the preservation of the human race quite yet,
we are solving for outcome-changing replay in data enablement.

As we discussed previously, rules applied to a data pipeline are often lossy and destructive (important data are filtered out due to cost or complexity) and brittle in their implementation (rules are hard-coded in ways that make future changes to the pipeline and its outputs complicated, risky and expensive).

Looking Back to See Ahead

If the world were a static or at least predictable place, this approach wouldn't create problems for data practitioners and consumers.

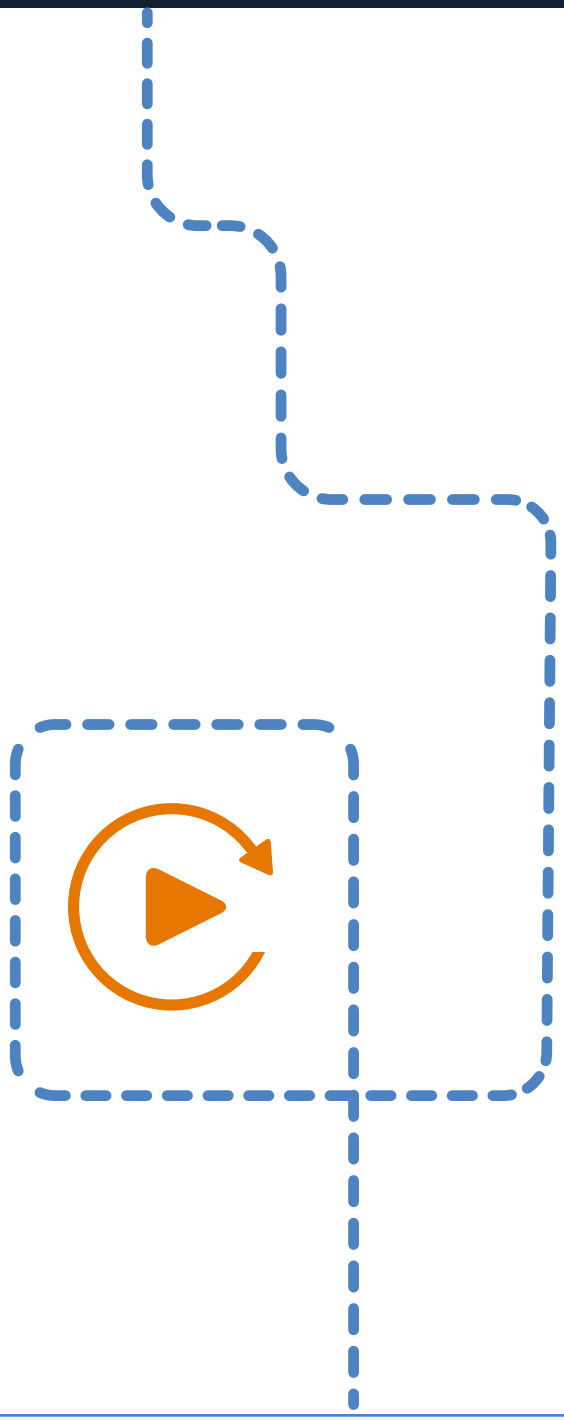
But as we've all seen and know, the world contains constantly changing data systems, schemas, and flows – and decision makers present constantly evolving needs for new outputs, reports, visualizations and analysis. In general, growth in data volume, data set complexity and systems footprints compounds complexity over time.

Since data analysts and engineers can't know in advance what business decision makers will need in terms of use cases and related data models, it makes sense (cloud storage costs notwithstanding) to collect and store as much raw event data as possible – and make it as easy as possible to remodel and reinterpret the data as necessary over time.

This has of course been a key driver in the rise of the modern data lake. But more data without context and especially governance creates new problems. More data can be helpful, but more data exacerbates the complexity problem rather than solving it.

What if we had a means of reinterpreting the past (especially when new facts or data sets become available, ideally landed in our immutable event log) in ways that enable new use cases and new analysis, without costly data engineering refactoring?

What if you could "future-proof" your data flows and transformations — reserving the right to reinterpret history, or "replay the tape" – when stakeholders have new use cases and need new insights?



Alternative 1: Early Data Pipelines

ETL-based data flows were the status quo for decades. Take only what you need, move that data to databases and cubes, leave the rest behind. Need something new? Data engineers revisit the entire pipeline from source to output in the form of a major IT project.

Transformations to a normalized model are usually "one-way" - the data has been transformed, the other data you now need (but didn't know back then you would need) is lost due to the transformations. **Some organizations believe they can predict and accommodate the future – and discover later that this is harder than it first appears.**

Alternative 2: The Current Approach

With the rise of elastic computing and massively scalable storage, the latest generation of pipelines swaps ETL for ELT, loading as much data as possible up front and putting destructive transformations last in the data flow. Per our last blog entry, this is a major advancement toward a future-proof data architecture.

The catch is that modern tooling stores and processes everything through complex logic IN the cloud data warehouse, commonly Snowflake or Google BigQuery, sometimes AWS RedShift or other flavor of cloud warehouse. The result – experienced and written about elsewhere – is that while storage is certainly cheaper in the cloud, ELT promotes spiraling costs in both data engineering and cloud computing. These costs grow exponentially with data volume and complexity.

The goal from here is to preserve the flexibility and other benefits of modern ELT, while simplifying data engineering, governance and processing expense - and to be able to specialize data (and the pipelines that move and model it) without regret.

Why doesn't ELT in the abstract solve this problem? **The dirty secret of ELT-based architectures is that Transformation (with a capital T!) at the end of modern data pipelines is still the bottleneck to buildout, maintenance and scaling.** Just like software development, data transformation projects grow moss. Refactoring transformations are a huge cost burden to data teams. We might call this phenomenon data "modeling-debt" or "transform-debt." We need a way to cost-effectively transform raw data into usable models on an ongoing basis.

ELT also requires reimporting all raw data from source systems every time data consumers present a new requirement – in order to "replay" the data. There is real computing cost and analytical load on source systems to re-run data ingestions for reinterpretation.

Alternative 3: Our Recommended Approach
Common Semantic Modeling upon Ingest

Consider this – What if there were a hybrid model that combines the best of both ETL and ELT principles. ETLT?

First, we apply a **non-destructive transformation at ingest** to semantically label and ascribe context and meaning to the data on the way in. If we can organize and label data ongoing as we capture it – effectively capturing meaning and context as we go – then all of the data (with its metadata) becomes flexibly usable for any purpose downstream, now and in the future.

Data is transformed – but only for meaning and context – on the way in, with semantic metadata and therefore governable understanding of the data generated as early as possible in the data flow. Common semantic models are rendered in-stream, in-memory as soon as the data is ingested, but these models are only current until new data comes along – refreshed or replaced.

Transformation for semantic labeling and structure only upon ingest results in governable semantic models.

This in no way limits the system's ability to apply transformations to outbound data, to accommodate new data models or specific orchestration end points. These outbound (ELT) transformations should be simpler and more manageable as the semantic understanding was already established far upstream at ingestion.

Outbound transformations inherit and uphold the common semantic layer, while offering flexibility for more complex modeling logic and downstream orchestrations.

At Reactor, we've been building and experimenting with this hybrid architecture with good results. We keep source data intact in its original and lossless form, stored locally in the cloud rather than captive in source systems.

Streaming transformations run constantly between raw ingest logging and semantically labeled models; and those transformations are expected, and in fact designed to change over time. As the raw data is well-organized and semantically labeled, this enables downstream transformation flexibility to address future use cases.

We are able to accommodate changes to source data schemas, transformations and modeling, and new stakeholder use cases without losing control.

"Metadata, Not Code"

As we designed our initial architecture for what has essentially become multi-stage or continuous transformation, it became apparent that there are opportunities to optimize for efficiency in the transformation code itself.

In most data pipelines today, transformations are executed through some combination of SQL, Python or UX-driven tools like Zapier. The most popular integration and modeling tools today are squarely aimed at data engineers writing code. Assigning and/or defining semantic metadata labels to data sets at ingest, upstream of the data warehouse, allows us to move much of the transformation logic from custom code to simpler metadata manipulations.

The more accurate and complete the initial semantic labeling, structure and definitions are at ingest, the more responsive downstream transformations can be. A library of semantic handler services can address this need. These services can be applied via fixed schema database or custom code. Fully productionizing these components moves **join- and stitch-logic upstream within the data pipeline and moves mapping logic from compiled code to composable data models.** The end result is a metadata-driven process that vastly increases our reusability and maintainability, with the eventual goal being governed, end-user control (and community sharing of common definitions).

With this approach, schema descriptions and transformations can be updated & modified within our application with no software code recompilation and deployment necessary. This metadata-driven approach supports standardized lifecycles for SCM and release management. And it allows reuse of transformation logic across and between egress points to allow customizations and extensions without breaking the ability to make common upgrades through a multi-tenant product versioning lifecycle. Proprietary extensions to our standard models are easily added to support new BI systems, data warehouses and operational systems.

Instant Replay Architectures

One of the most important features of this reference architecture is a concept we've dubbed "replay." Once you've properly organized and labeled raw data at ingest (immutably stored locally), we can change the definitions captured in semantic labels and "replay," or reprocess the data to accommodate new downstream models, orchestration schemas and stakeholder use cases and related outputs. The architecture enables the ability to efficiently change the models with little or no code, and few or no breaking changes. This is profoundly different from and superior to prior approaches that require data engineering for any change at any stage.

The result is the ability to truly democratize and future-proof data flows and models. By gathering all of the data up front, you don't need to know now all the ways you might need to use the data in the future, and can instead focus on short-term usability, relevancy and ROI.

Realized Benefits

How does continuous transformation in Reactor compare to the status-quo approach characterized by SQL or Python modeling in or on a cloud data warehouse?

Here are a few of the differences and benefits:

Specialize and future-proof your data models and flows without regret, as all raw, lossless data is retained for future use cases and reinterpretation

Maximize complex data modeling flexibility without compromising data governance, by separating, defining and maintaining the semantic layer upstream of modeling logic

Increase your time-to-insights and lower your cloud compute costs with in-stream processing rather than processing materialized data sets

Allow all data stakeholders (especially non-engineers) to refine and reprocess ever-changing and imperfect data models without breaking changes, using continuous transformation and "replay."

Simplify modeling and reduce cloud data warehouse processing expense by addressing data prep including semantic labeling during data onboarding

Limit data reprocessing costs and latency by applying new or modified transformations to affected data fields only

Remodel as Needed

With data replay, data practitioners and the business stakeholders that rely on their work have the tools necessary to replay targeted segments of their pipelines at any time and as many times as needed, driving outcomes that meet the fluid and ever-changing needs of the business environments they support.

As Bill Cage discovered, the **flexibility to replay, rewrite and reinterpret history can be a superpower.** The same holds true for stakeholders building modern data pipelines and data models.

3 Open for Business: Define your Data with Semantic Labels and Metadata

In the science fiction drama **Arrival**, linguist Dr. Louise Banks discovers that (spoiler alert!) the language shared by visiting aliens is a rosetta stone that unites Earth's global powers by **enabling shared understanding across countries and cultures**. This shared language provides a common "semantic" layer for the people of Earth to communicate and understand each other.

When we use the word semantic to describe a data model, we are basically talking about a conceptual model of the data, including objects or entities, how they are classified, and their relationships. A semantic model might include metadata (descriptive data describing the original data) as labels to help capture the essence and meaning of the information. In retail, examples of entities are customers, orders, shipments, and digital advertisements. Entities might also include stores and warehouses, units of inventory, and parcel post shipments. Metadata labels might describe how COGS is assigned to order items, or how net revenue is calculated for a certain order given promotional discounts and markdowns.

What's the value of this semantic understanding, and how does it help businesses thrive using data as a competitive advantage?

In general terms, defining data with detailed, industry-specific semantic labels and metadata can provide several benefits to business end-users, including:

Improved Data Relevance:

Semantic labels and metadata can help end-users quickly identify and locate the data they need for their specific business needs. By using industry-specific labels and metadata, data can be categorized and organized in a way that is familiar and relevant to the end-user's business domain, making it easier to find and use the data.

Increased Data Understanding:

Semantic labels and metadata can help end-users better understand the data they are working with by providing a standardized and consistent way of describing the data. By using industry-specific labels and metadata, end-users can better understand the context and meaning of the data, making it easier to interpret and analyze.

More Accurate Analysis:

Semantic labels and metadata can help end-users perform more accurate and precise data analysis by providing a clear and consistent way of describing the data. By using industry-specific labels and metadata, end-users can reduce the risk of errors and inaccuracies in their analysis, resulting in more reliable and trustworthy results.

Improved Collaboration:

Semantic labels and metadata can help end-users collaborate more effectively by providing a common language and framework for discussing and sharing data. By using industry-specific labels and metadata, end-users can better communicate and collaborate with colleagues, resulting in more effective teamwork and better decision-making.

Overall, defining data with detailed, industry-specific semantic labels and metadata can provide significant benefits to business end-users in the field of data analytics, including improved data relevance, increased data understanding, more accurate analysis, improved collaboration, and better data governance and compliance.

In the field of retail analytics, defining data with detailed, **industry-specific semantic labels and metadata** can provide several benefits to business end-users, including:

Improved Sales Insights:

Semantic labels and metadata can help end-users understand sales data better by providing context and detailed information about products, promotions, channels, and customer segments. By using industry-specific labels and metadata, end-users can gain deeper insights into customer behavior, preferences, and trends, resulting in more effective sales strategies.

Better Inventory Management:

Semantic labels and metadata can help end-users manage inventory more effectively by providing information about product attributes, such as size, color, and style. By using industry-specific labels and metadata, end-users can better track and manage inventory levels, reducing stockouts and overstocking, resulting in better cost control.

Enhanced Customer Experience:

Semantic labels and metadata can help end-users improve the customer experience by providing information about customer behavior, preferences, and demographics. By using industry-specific labels and metadata, end-users can tailor marketing campaigns, product assortments, and pricing strategies to meet customer needs, resulting in higher customer satisfaction and loyalty.

More Accurate Forecasting:

Semantic labels and metadata can help end-users make more accurate sales and inventory forecasts by providing historical data and trend analysis. By using industry-specific labels and metadata, end-users can identify patterns and anomalies in the data, leading to more accurate forecasting and better resource allocation.

Improved Data Governance and Compliance:

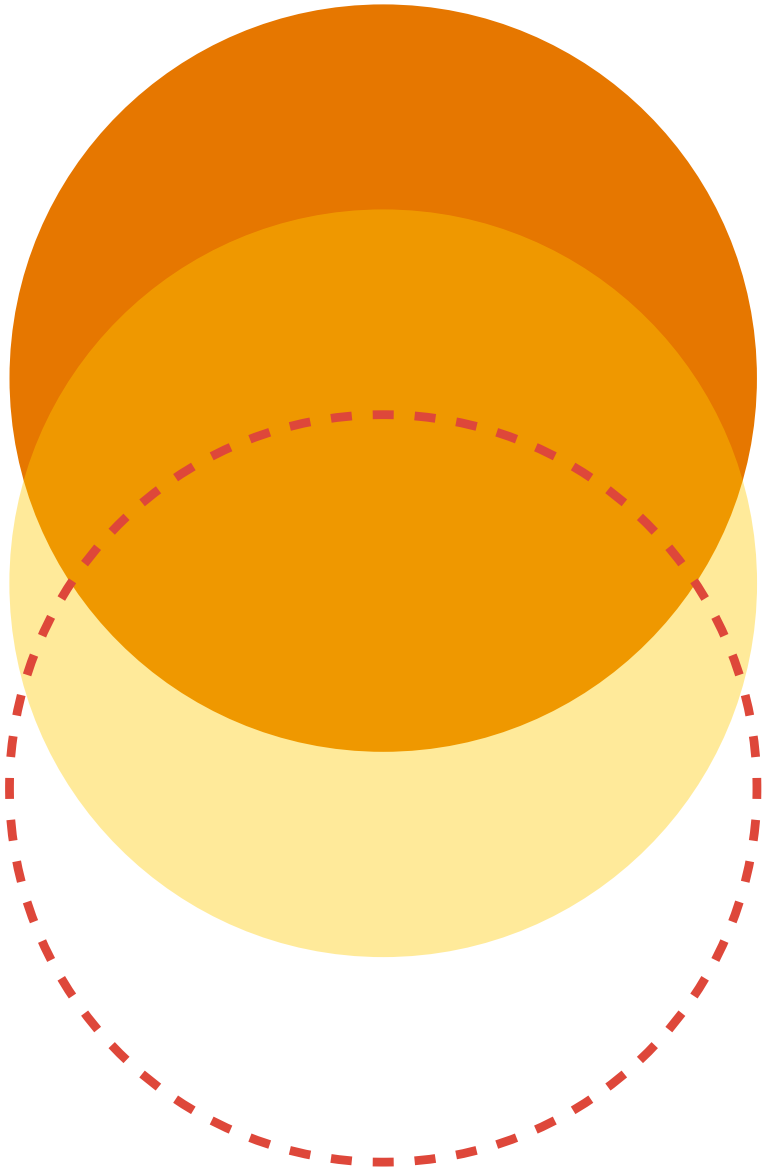
Semantic labels and metadata can help end-users ensure compliance with data regulations and governance policies by providing a standardized and consistent way of categorizing and managing data. By using industry-specific labels and metadata, end-users can reduce the risk of data breaches and ensure that data is being used in a compliant and ethical manner.

Define your Data

Overall, defining data with detailed, industry-specific semantic labels and metadata can provide significant benefits to business end-users in the field of retail analytics, including improved sales insights, better inventory management, enhanced customer experience, more accurate forecasting, and improved data governance and compliance.

Are the departments and teams at your brand operating more like the twelve disconnected global tribes at the beginning of Arrival?

You may need a semantic layer for your data (applied early and holistically) to promote shared understanding of key concepts and calculations, and **unite the factions to achieve global harmony!**

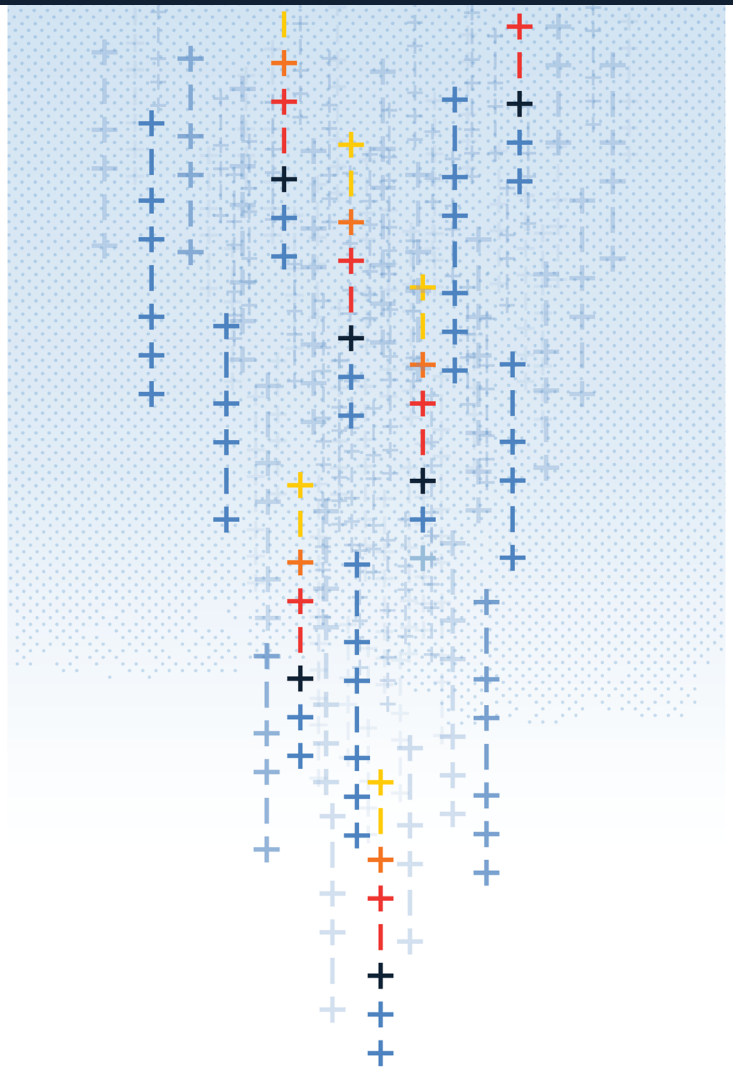


4 Do it Right the First Time: Define your Semantic Layer as Early as Possible

In the iconic 1999 cyberpunk movie **The Matrix**, the human rebel protagonists ascribe meaning to the (virtual) world around them by **reading and understanding real-time streams of data**. The downward flowing green characters are one of the most recognizable visual hooks of the film.

As far as we know, we're not living in The Matrix (or are we?!). Yet modern data engineers and data practitioners now have the ability to decipher data streamed from SaaS APIs and modern cloud data services, interpreting the information in real-time to create "semantic" models of understanding of the data in advance of analytical use cases and downstream data flows.

In industries like retail, data models and outputs (business intelligence reports, reverse ETL orchestrations, etc.) and the teams that use them often have different definitions for the same fields, calculations, and KPIs. These inconsistencies can cause confusion throughout the organization and even risk disseminating incorrect metrics driving false conclusions for key business stakeholders.



Data pipeline owners have tried to address this challenge in many different ways, most of them involving the retroactive creation of data catalogs to define and label data post-mapping and processing.

There is a better way. Defining semantic labels and metadata at ingest – as early as possible in the data pipeline – can provide several key benefits for data analytics practitioners and consumers, including:

Improved Data Understanding:

Semantic labels and metadata provide a standardized and consistent way of describing business data, making it easier for data analysts and data scientists to understand and work with the data. This can improve the accuracy and reliability of data analysis and reduce the time and effort required to explore and clean the data.

Faster Time-to-Insight:

By defining semantic labels and metadata early in the data pipeline, data analysts and data scientists can quickly locate and access the relevant data they need for analysis. This can reduce the time required to process and analyze all of the organization's business data, resulting in faster time-to-insight and faster decision-making across all levels of the organization.

Better Data Governance and Management:

Semantic labels and metadata provide a clear and consistent way of categorizing and managing data, making it easier to enforce data governance policies and ensure compliance with data regulations. This can reduce the risk of data breaches, improve data quality, and increase confidence in the accuracy and reliability of the organization's data.

Improved Data Integration:

Semantic labels and metadata can facilitate the integration of data from different data sources by providing a standardized way of describing the data. This can improve data interoperability, reduce the time and effort required for data integration, and enable more accurate and reliable data analysis.

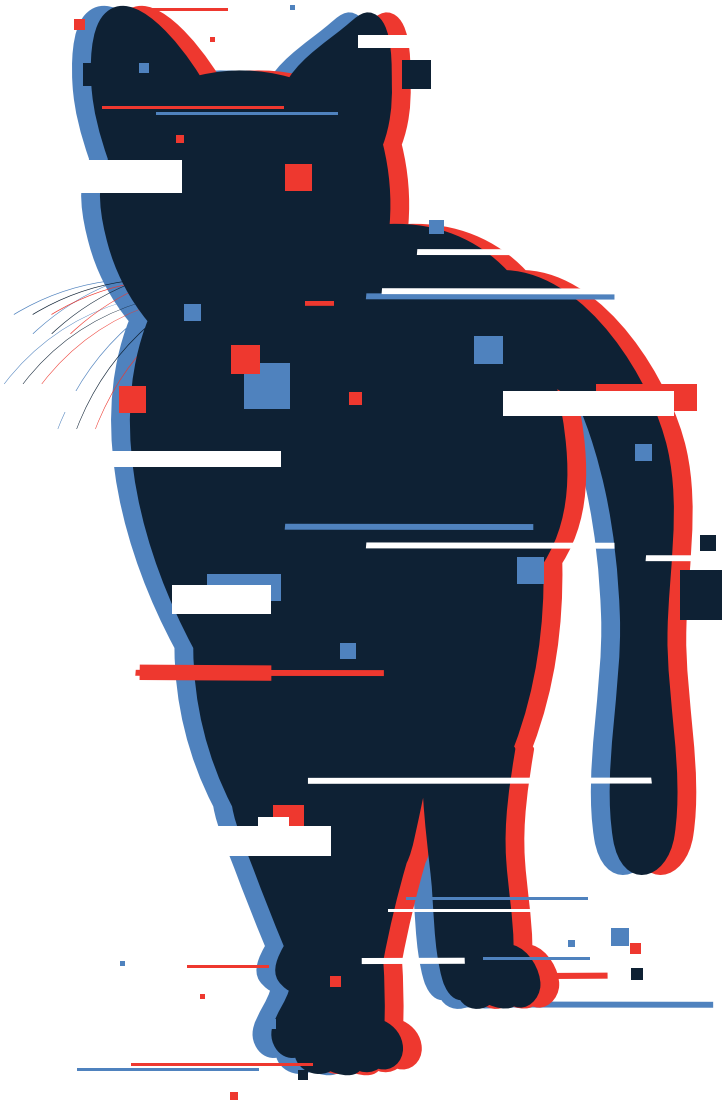
Enhanced Collaboration:

Semantic labels and metadata can accelerate collaboration between different stakeholders involved in the data pipeline, including data analysts, data scientists, and business users. By providing a clear and consistent way of describing the data, semantic labels and metadata can enable more effective communication and collaboration, leading to better data-driven decisions.

Overall, defining semantic labels and metadata at the point of ingestion can provide significant benefits for the downstream systems and use cases of data-driven organizations**,** including improved data understanding, faster time-to-insight, better data governance and management, improved data integration, and enhanced collaboration.

Independent of Reactor, data pipelines generally wait until the last moment to define the things that matter. As such, the data pipelines themselves contribute to confusion and competition across users, teams, departments. **By contrast, Reactor was purposefully designed to define what data means (and how it will be used) as early as possible in the data flow.** With Reactor, all the downstream data mapping and analytics work benefits from continuity, accuracy, and shared understanding across the entire organization.

Whether the steak and red wine are real or not, building a shareable, clear understanding of the data that describe these objects is important. Even more so in business settings where shared understanding is required across disparate teams, departments, geographies and use cases.



Label Early and Often

Semantic models can make us all Neo, able to glean the most important information out of the continuous, abundant stream of data. This allows us to make crisp, critical decisions for the best outcomes. Your next decision may not be a life or death struggle against a matrix agent, but **exact definitions applied early and precisely to your data can make you a data superhero too.**

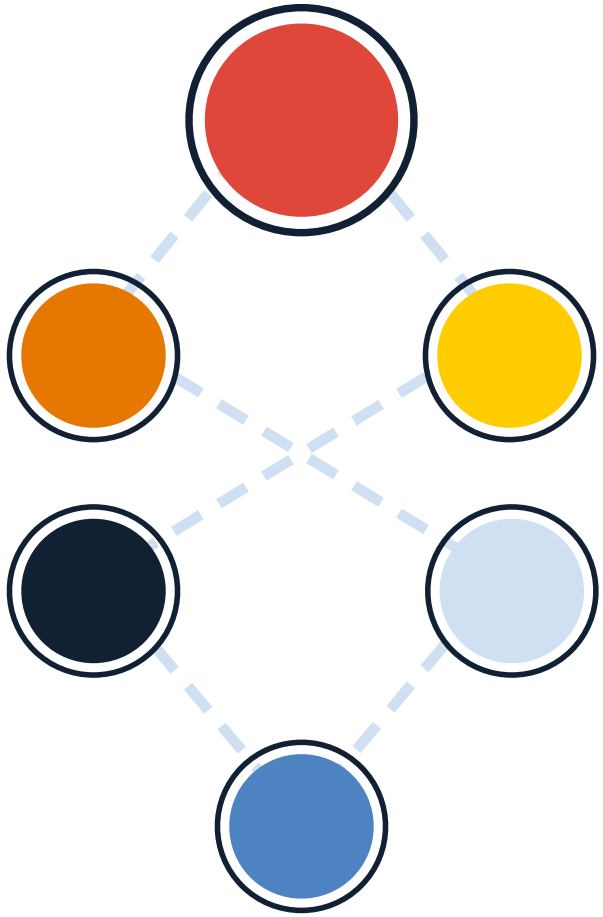
5 Smaller is Better:
Atomize your Data for more Flexible Mapping and Modeling

In the **Marvel Cinematic Universe**, mythical Infinity Stones grant their owners great powers. One stone in particular, the Reality Stone grants its holder the power to manipulate matter and defy the laws of physics.

Back in the real world, **data engineers and analysts and the data consumers they serve face the reality of complex data whizzing from system to system**, and complex data modeling work taking place within modern cloud data warehouses like Snowflake and Google BigQuery.

In reality, point-to-point data integration tools lift-and-shift data bundled into messages from one system to another. But extracted, transformed, and loaded (ETL or ELT) data usually lands in disconnected, unjoined tables. It requires a massive lift from engineering to get to entities and models that are useful and business-ready for analysts and end users. Even with the modern cloud and modern data stacks, we still spend massive amounts of time giving data the right business context so it's useful for consumers and customers of the data.

Wouldn't it be great if data teams had access to a Reality Stone to grant them data manipulation superpowers? There is a **real-world corollary for matter manipulation in the data world**. It's an emerging concept, and it's called **"data atomization."**



Data Atomization in the Real World

Data atomization is the process of breaking down a larger dataset into smaller, more granular pieces or "atoms" of data. Each atom typically contains only a single data point or fact, along with relevant contextual information, such as when and where the data was collected.

Atomic data stores and atomic data warehouses are data management structures designed to store and manage granular, atomic-level data. Atomic data stores are typically used to store individual data points. In contrast, atomic data warehouses are used to store and analyze relationships across and between large collections of atomic data points.

These atomic structures are important in building analytical data pipelines because they enable efficient data processing and analysis. By storing data at the atomic level, analysts can easily filter, sort, and aggregate the data in various ways, depending on their specific research questions and needs. This allows for more targeted and precise analysis and can help uncover insights and trends that might not be apparent when analyzing the data in its original form, ingested from system silos.

For example, ingested messages from Shopify, NetSuite, and ShipStation contain customer billing and shipping address data attached to order origination, order management, and shipping system messages respectively. All of these systems generate address data, but to unify, rationalize and harmonize that address data without atomization, a data engineering team would have to manually parse and isolate the address data – dealing with all of the context and complexity of each system's message format and schema. With atomization, however, data contained in each message is stripped from the message itself (its data "container"), and the resulting, atomized customer address data is much easier to access, label, map and use. With atomization, the customer address information from ANY system can be rendered using common definitions and mappings, regardless of which source system generated that data.

Atomization and associated data abstraction have the potential to create HUGE efficiency gains for data teams and consumers, as data can be organized and defined independent of the scope and purpose of incoming (or outgoing) messages. Teams have a flexible data structure to use within the data warehouse because the source system inputs aren't wrapped up in fixed-schema messages or tables, they've been broken down into the smallest possible components to maximize their flexibility. It's faster for teams to gain insights on atomized data because the data has been cataloged, often in real-time and in a way that was designed to enable retail business use cases.

Atomization also offers improved data pipeline iency because source system changes are handled before they can impact (i.e., break) downstream analytical and behavioral models in the data warehouse.

Another benefit of atomization is the lower operational processing loads – meaning your Snowflake or BigQuery data warehouse bill! – as a result of breaking down and standardizing the data earlier in the process.

With atomization, data teams can stop worrying about the peculiarities of each source system and instead spend their time improving common models that actually deliver value for business customers.

Data Atomization in Retail and Direct-to-Consumer (DTC) Commerce

In the retail industry, atomic data stores and warehouses can be used to manage a wide range of data, including sales, customer, inventory, and marketing data. Here are five examples of how atomic data structures can be applied in the retail industry:

Sales Data:

Retailers can use an atomic data store to manage individual sales transactions. This data can be used to analyze sales trends by product, store location, or customer segment. An atomic data warehouse can be used to store a larger collection of sales data, enabling retailers to analyze sales performance over time and identify patterns and trends.

Customer Data:

Retailers can use an atomic data store to manage individual customer interactions, such as purchases, returns, and website visits. This data can be used to analyze customer behavior and preferences and to develop targeted marketing campaigns. An atomic data warehouse can be used to store a larger collection of customer data, enabling retailers to track customer behavior over time and identify trends in customer loyalty and engagement.

Marketing Data:

Retailers can use an atomic data store to manage individual marketing interactions, such as email opens, website clicks, and social media engagements. This data can be used to analyze marketing performance by channel, audience, and campaign. An atomic data warehouse can be used to store a larger collection of marketing data, enabling retailers to analyze marketing trends over time and optimize their marketing strategies.

Inventory Data:

Retailers can use an atomic data store to manage individual inventory transactions, such as stock levels and replenishment orders. This data can be used to analyze inventory performance by product, store location, or supplier. An atomic data warehouse can be used to store a larger collection of inventory data, enabling retailers to analyze inventory trends over time and optimize their supply chain operations.

Operational Data:

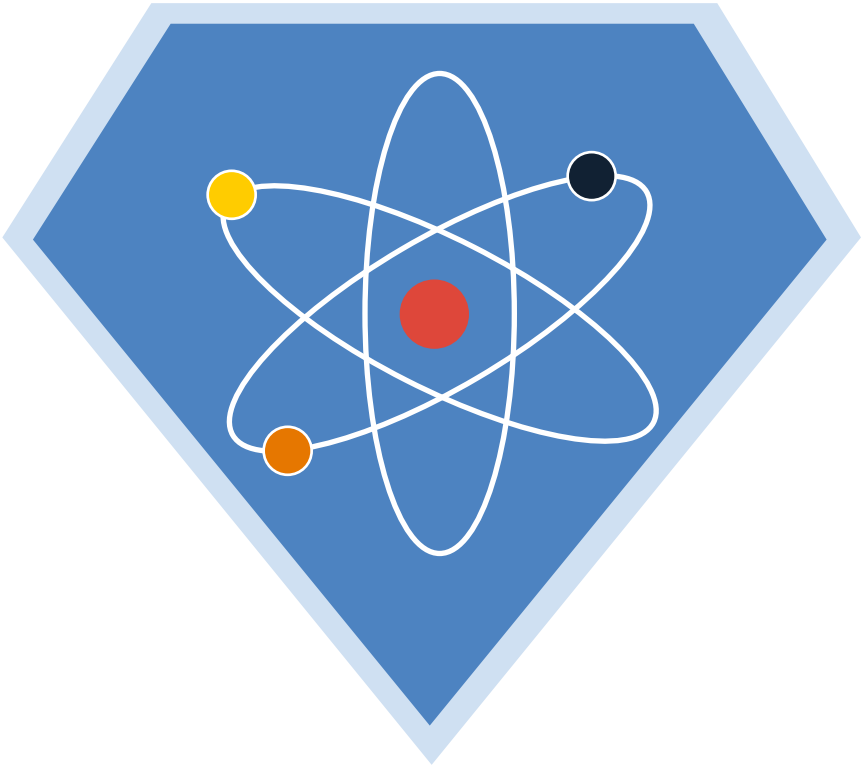
Retailers can use an atomic data store to manage individual operational transactions, such as employee schedules, payroll, and vendor invoices. This data can be used to analyze operational performance by store location, department, or vendor. An atomic data warehouse can be used to store a larger collection of operational data, enabling retailers to analyze operational trends over time and optimize their business operations.

Atomize for Flexibility

With data atomization, retail data practitioners can separate, label, define and use data without worrying about the peculiarities and complexities of system end-points and message structures.

Atomize your data before your data warehouse analytical modeling to build more efficient, observable, flexible, and useful models for your retail business decision-makers.

Even in the real world, and without the need for the Reality Stone, you can manipulate data like an MCU superhero!



6 Data for All:
No Code/Low Code Data Pipelines Expand your Data Team

Like C3PO, the original **Star Wars** protocol droid, **LCNC interfaces allow anyone to do mission-critical data work**, reducing time-to-value for business stakeholders and redirecting data and IT teams to focus on strategic insights and data activation, including AI-driven outcomes...

Few science fiction robots are more iconic than the astromech droid R2D2 and his protocol droid buddy, C3PO. In the Star Wars universe, protocol droids like "3PO" assist sentients with planning, problem solving, and translation – being fluent in "more than 6 million forms of communication!"

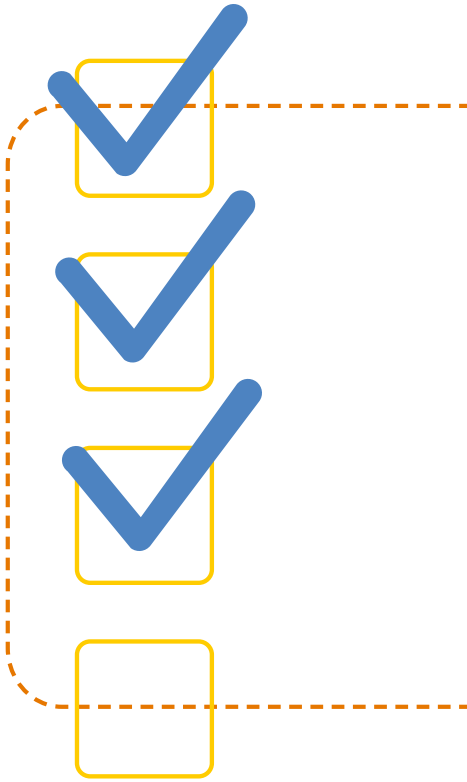
To achieve this vision here on earth, we're innovating in areas like humanoid robotics and generative AI. And while both of these fields hold huge promise to transform the way humans interact with data and technology, there's a simpler revolution underway helping humans tackle protocol, and that's the rise of low code/no code (LCNC) interfaces to cloud services, software applications and data.

IT and data teams everywhere are maxed out, being asked to do more with less everywhere at once. They're often spending too much time on tactical buildout and maintenance instead of helping stakeholders drive business value.

Business users that have the core domain knowledge and context to define requirements and goals don't have technical know-how or access to be self-sufficient – and instead rely heavily on engineering teams to get work done.

The promise of LCNC interfaces is to democratize (for all employees) and consumerize (for non-engineers) the work at hand. This has become especially important as data proliferates and data increasingly offers an "unfair advantage" to companies and brands that can wield it without overwhelming IT.

In data warehouses, the central component of the modern data stack, landing data in tables might solve the first 1% of the useful data problem. Data engineering, responsible for mapping and modeling, is the bottleneck for the other 99% of the work. Tools and services like DBT and Snowflake make things easier for engineers, but the work today is still primarily constrained by technical complexity and engineering capacity.



What if anyone could collect and organize the data they need, in ways that uphold data trust and governance?

What if we could create as many mappings as necessary to meet business requirements without additional engineering tickets?

What if we could create useful transformations and logic that don't require data engineering?

When we solve for these cases, Data and IT teams move from tactical pipeline and workflow plumbing, free to focus on strategic insights and activation of an organization's data.

So how do low code / no code (LCNC) interfaces benefit businesses building and maintaining complex data analytics systems? How does LCNC help make analytics projects faster, easier and cheaper to start, and on an ongoing basis?

Low code no code (LCNC) interfaces benefit businesses building and maintaining complex data analytics systems by reducing the technical skills required to build and maintain these systems. LCNC platforms provide drag-and-drop interfaces and pre-built templates that enable users to create complex data pipelines and analytics applications without having to write code from scratch. This makes analytics projects faster, easier, and cheaper to start, and on an ongoing basis.

With LCNC interfaces, business users can take a more active role in building and maintaining analytics systems, rather than relying on IT teams or external vendors to do the work. This enables business users to have more control over the analytics systems and to respond more quickly to changing business needs.

LCNC platforms also reduce the time and cost of building and maintaining data analytics systems. Because LCNC platforms provide pre-built templates and pre-configured components, users can build analytics applications and pipelines faster and with fewer errors. This reduces the time and cost of development, as well as the ongoing maintenance and support costs.

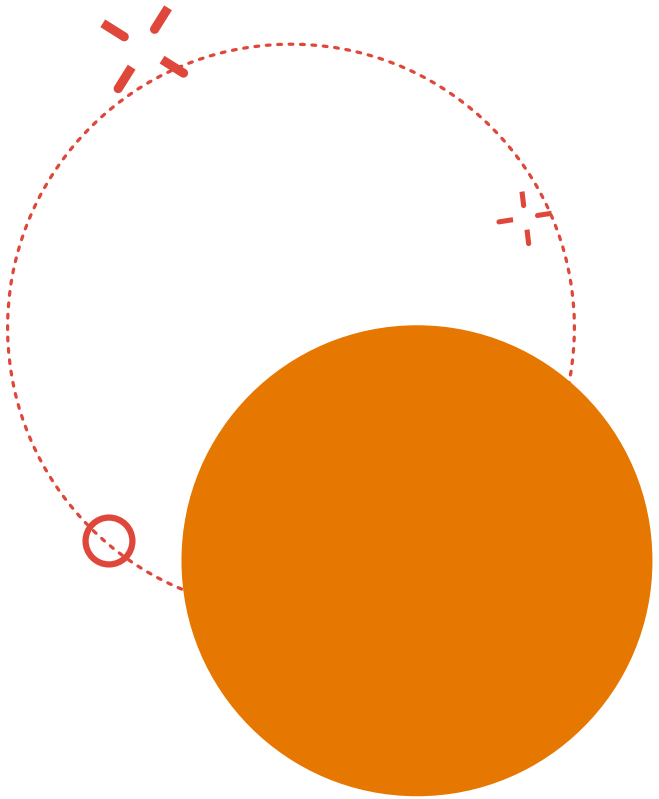
Additionally, LCNC platforms can provide access to a wider range of data sources and analytics tools. This enables businesses to use a variety of data sources to inform their analytics applications, and to integrate different analytics tools and services into their workflows.

Democratize Your Data Flows

C3PO has his character flaws, and low code / no code interfaces aren't perfect either. While LCNC platforms can automate some aspects of performance tuning, it's still important to for IT and cloud data teams to optimize query performance by using appropriate indexing, caching, and partitioning strategies. Humans need to validate data flows and models in the context of their unique business and use cases.

Acknowledging these limitations, **LCNC interfaces can still provide businesses with a faster, easier, and more cost-effective way to build and maintain complex data analytics systems.** By empowering business users with the tools they need to create and maintain analytics systems, businesses can become more agile and responsive to changing business needs, and can more easily capitalize on new opportunities and insights.

Like C3PO did for the Rebel Alliance, low code / no code interfaces can help businesses translate data into useful insights and outputs. **Try putting LCNC tools to work for your team, and see what you can accomplish!**

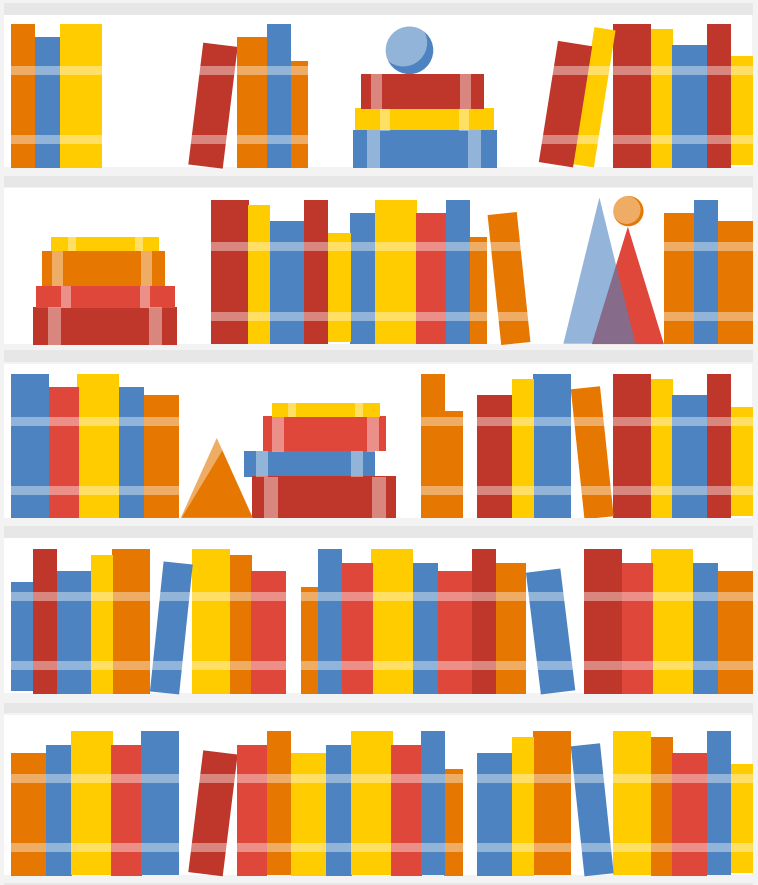


7 Share, Share Alike: Use Mapping and Modeling Libraries to Accelerate your Work

Fans of **Isaac Asimov's Foundation Series** (the books, not AppleTV!) know that The Imperial Library on the planet Trantor was both the future galactic repository for human knowledge, and the place where scientist protagonist Hari Seldon developed his theories of **Psychohistory – the ability to predict the future with advanced probabilistic mathematics.** Asimov was a polymath, and his writings were amazingly prescient regarding artificial intelligence and machine learning today.

Back here on present-day earth, data scientists face a similar cost/benefit conundrum. **How can I develop useful machine learning algorithms on complex data sets and models – without the heavy lift of engineering everything from scratch?** Generative AI represents a step-change increase in the speed of analysis, but the utility of even the best GenAI tools are still constrained by the quality of the data and data models analyzed.

No one wants to build their data infrastructure from scratch. Thankfully open cloud standards (the "modern data stack") and popular programming languages like Python and SQL give data teams a massive head start toward useful, actionable and ML-ready data. A growing number of commercial data integration tools offer users the ability to leverage and expand shared libraries of mapping and modeling logic, presenting the opportunity to greatly accelerate data time to value and analytics time to insights.



As with Asimov's Imperial Library on Trantor, there are major advantages to using commercial data integration tools or software applications that offer open-source or community-maintained libraries of transformation and mapping logic.

First, these tools can help businesses save time and money by providing pre-built components, connectors, and transformations that can be easily integrated into their ETL or ELT workflows. This can reduce the need for custom development and testing, and speed up the overall development process.

Second, these tools can help businesses improve the quality and accuracy of their data integrations by providing a library of pre-built components and transformations that have been tested and validated by the community. This can help reduce errors and improve the reliability of data pipelines.

Third, platforms that allow end-users to use and contribute to ETL or ELT code written by other users can help foster collaboration and innovation within the data integration community. Users can share their own custom components and transformations, as well as learn from others and contribute to the development of the platform.

Overall, using commercial data integration tools or software applications that offer open-source or community-maintained libraries of transformation and mapping logic can help **businesses build more efficient, reliable, and innovative data integrations.**

Here are a few providers active today in the data onboarding ecosystem:

Fivetran is a cloud-based data integration platform that offers pre-built connectors for Snowflake and Google BigQuery, as well as a range of other data sources. FiveTran offers no advanced analytical modeling capability – users are expected to build their own models using tools like DBT or Coalesce in the data warehouse. www.fivetran.com/.

Matillion is a cloud-native data integration platform that offers pre-built connectors for Snowflake and Google BigQuery, as well as a range of other data sources. Analytical modeling is performed downstream of Matillion in the data warehouse. www.matillion.com/.

Reactor is a low-code, intelligent, data pipeline that provides the fastest, most efficient path to useful, business-ready data for generative AI, analytics and activation. www.reactordata.com/.

Talend is an open-source data integration platform that offers connectors for Snowflake and Google BigQuery, as well as a range of other data sources. www.talend.com/.

Stitch is a cloud-based data integration platform that offers pre-built connectors for Snowflake and Google BigQuery, as well as a range of other data sources. www.stitchdata.com/.

Choosing the best data onboarding tool for Snowflake from among FiveTran, Matillion, Reactor, Talend, Stitch depends on a number of factors, including the specific requirements of your business, the complexity of your data integration needs, and your budget.

Here are some key criteria to consider when choosing a data onboarding tool for Snowflake or BigQuery:

Time to insights and activation:

Does the platform provide out-of-the-box data flows and logic to leapfrog the manual efforts of a data engineering team or system integrator? How fast can you have data flowing and rendered into useful data models that support BI analytics and data activation via reverse ETL and data query/segmentation tooling? Are these analytics and activation tools offered natively by your data onboarding partner?

Ease of use:

Consider how user-friendly each platform is, as well as the level of technical expertise required to use it effectively. Look for a platform that offers an intuitive, easy-to-use interface and requires minimal coding or technical knowledge. Does the platform support common languages like Python and SQL? Does it offer simpler "no code" interfaces to get data labeled, mapped and flowing?

Data sources and connectors:

Look for a platform that supports the specific data sources and connectors you need, such as Shopify, NetSuite or Manhattan Active Omni. Consider the number and variety of connectors offered by each platform, as well as how frequently new connectors are added. Consider how the provider maintains compliance with source system APIs and schemas over time.

Data transformation and mapping capabilities:

Consider the range and complexity of data transformation and mapping capabilities offered by each platform, including pre-built transformations and mappings, as well as the ability to create custom transformations and mappings. Does the platform offer pre-built labeling and mapping specific to your vertical industry and use cases?

Performance and scalability:

Look for a platform that can handle the volume and complexity of your data, and can scale up or down as your needs change. Consider factors such as processing speed, data latency, and the ability to handle large volumes of data. Does your provider immutably (permanently) log your raw event data locally for failover and to expedite analytical processing as new use cases arise?

Cost:

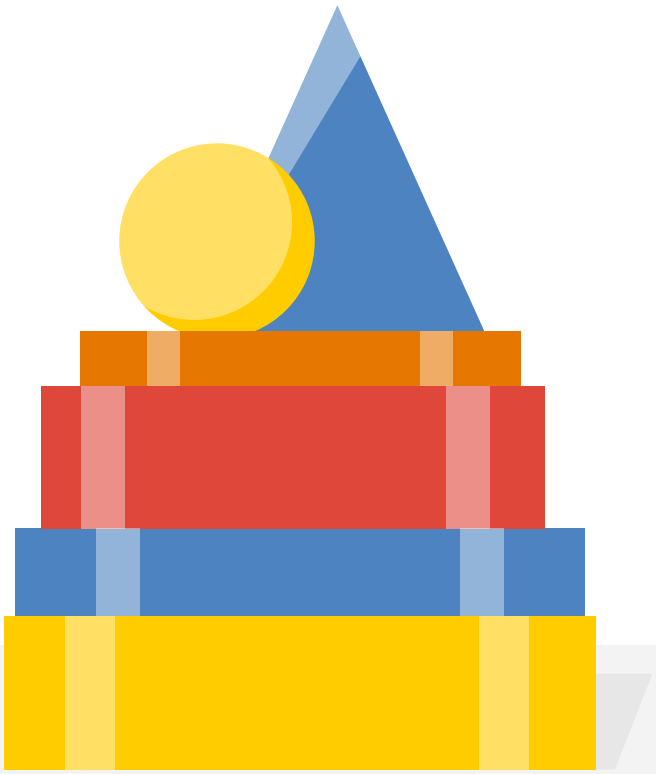
Consider the cost of each platform, including licensing fees, subscription costs, and any additional costs for features such as data transformation and mapping. Look for a platform that offers transparent pricing and a clear pricing model.

Leverage Shared Libraries

Based on these criteria, the best data onboarding tool for Snowflake and BigQuery will depend on the specific needs and priorities of your business. All of the platforms listed above offer a range of features and capabilities, so it's important to evaluate each one in terms of its suitability for your business.

Like the citizens of Asimov's galactic empire, data practitioners can call upon rich libraries of content and code (or no-code logic) to more quickly capitalize on data to drive better outcomes – especially through Generative AI and ML algorithms.

The key to fast time to useful data insights and activation is leveraging industry best practices in the form of shared data labels, mapping and models to leapfrog the most tedious and time consuming data engineering tasks!

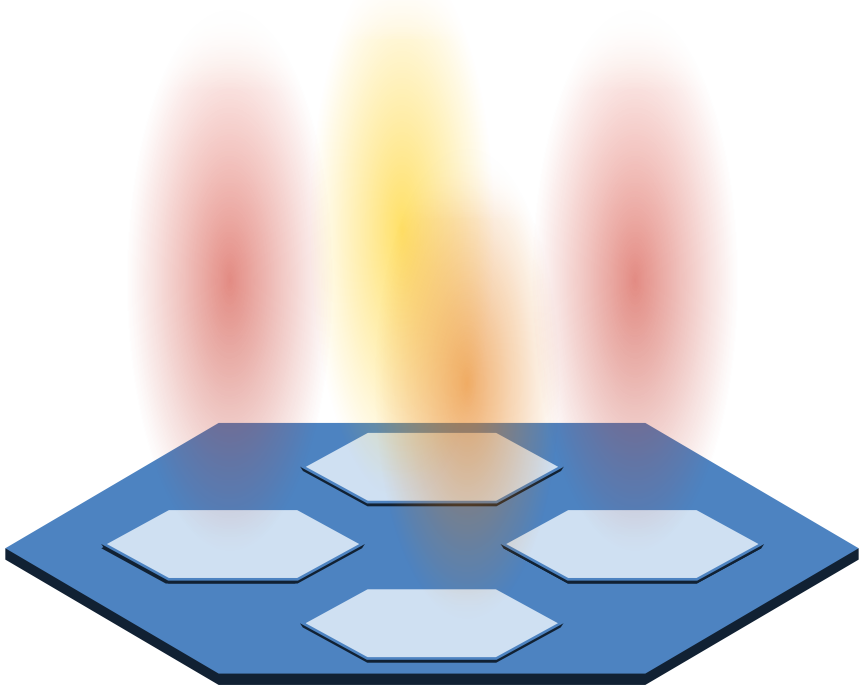


8 Faster is Better:
Use Streaming Data Pipelines for Real-Time Analysis and Applications

Beam Me Up, Scotty

While early science fiction shows like Buck Rogers (1939) and The Fly (1950) depicted teleportation technology, it was **Star Trek**'s transporter room that made real-time living matter transfer a classical sci-fi trope. While we haven't built technology that **enables real-time matter transfer yet**, modern science is pursuing concepts like superposition and quantum teleportation to facilitate information transfer across any distance at speeds faster than light. Thanks, Albert Einstein!

Back in the data world, we have no need to wait for these future technologies to arrive. Data practitioners today are already using real-time data pipelines to enable a broad set of use cases ranging from website optimization to reactive and predictive fulfillment and delivery routing. Modern data flows including iPaaS and ETL services can achieve millisecond latencies, moving useful data into downstream apps almost instantaneously. The advent of generative AI is massively increasing the uses and value of real-time data for predictive software applications and analytics.



From Batch Processing to Streaming

Batch processing of data is the established paradigm – a function of practical limits on storage and processing power dating back to punch card computing. With the advent of cloud computing, moving from batch to real-time or "in-stream" processing has become practical and even affordable. Data streaming is now a driver of new business capabilities and a source of competitive advantage. Real-time data streaming can enable businesses to optimize decisions and actions in seconds rather than minutes, hours, or days.

Shifting from batch to real-time streaming data transfer can serve to unify disparate and potentially redundant data flows that previously served operational (e.g. payment processing) and simple analytical (e.g. BI dashboard) work. In the retail industry, applications for real-time data range from responding instantly to shopper behavior to flagging and resolving operational exceptions as they occur.

Getting to Practical Applications

Here are a few examples of how DTC and omnichannel brands are using real-time data streaming in practice today:

Real-time inventory management:
Retailers can track inventory levels in real time and trigger automated reorders when inventory reaches a certain threshold, helping to avoid stockouts and overstocking.

Real-time order routing:
Brands can use real-time data to track order fulfillment lifecycles, determining where and how product should be picked, packed, routed, and delivered to optimize time to doorstep and fulfillment costs.

Personalized marketing:
By analyzing customer behavior in real time, retailers can create personalized marketing campaigns that are targeted to individual customers.

Fraud and loss detection:
Ecommerce and POS platforms can use real-time data analysis to identify and prevent fraudulent transactions, reducing the risk of financial losses and damage to reputation.

Dynamic pricing:
Retailers can use real-time data to dynamically adjust pricing based on supply and demand, competition, and other market factors, optimizing revenue and profits.

Customer support:
Retailers can use real-time data to provide personalized support to customers, offering relevant recommendations, and answering questions or concerns.

Supply chain optimization:
Retailers can use real-time data to optimize their supply chain operations, improving delivery times and reducing costs.

Social media monitoring:
Retailers can use real-time data to monitor social media channels for mentions of their brand or products, responding quickly to customer feedback and concerns.

Store layout optimization:
Retailers can use real-time data to analyze customer behavior in physical stores, optimizing store layout, product placement, and staffing levels for maximum efficiency and sales.

Predictive maintenance:
Retailers can use real-time data to identify and prevent equipment failures, reducing downtime and maintenance costs.

Key Benefits of Real-Time Data Streaming in Retail

Streaming data to address use cases like those outlined above can deliver key advantages for your retail brand. Real-time data pipelines enable organizations to respond quickly to changing business needs and market conditions, creating a more agile and competitive brand. Streaming data pipelines provide real-time insights, enabling faster and more accurate decision-making. Real-time data processing ensures that data is processed and made available for analysis as soon as it's generated, reducing processing time and latency. Real-time processing enables organizations to streamline and automate data processing workflows, reducing manual effort and improving operational efficiency. And perhaps most importantly, real-time insights into customer behavior and preferences enable organizations to deliver more personalized and relevant experiences, increasing customer satisfaction and loyalty.

For data engineering and analytics teams, real-time data processing can enable faster identification and correction of data errors, ensuring greater accuracy of data used for analysis. Real-time data processing ensures that only clean, validated data is available for analysis.

These advantages apply directly to analytics and business intelligence use cases, as they enable organizations to process and analyze data more quickly and accurately and respond more effectively to changing business needs.

Optimizing for IT Return on Investment

A key risk worth noting for organizations pursuing real-time data capability is cost. Streaming rather than batch processing data may not generate additional storage or compute expense, but the engineering burden to cost-optimize streaming data pipelines and analytical models can be significant. Cloud computing costs can spiral, especially when storage and compute are concentrated within cloud data warehouses like Snowflake or Google BigQuery.

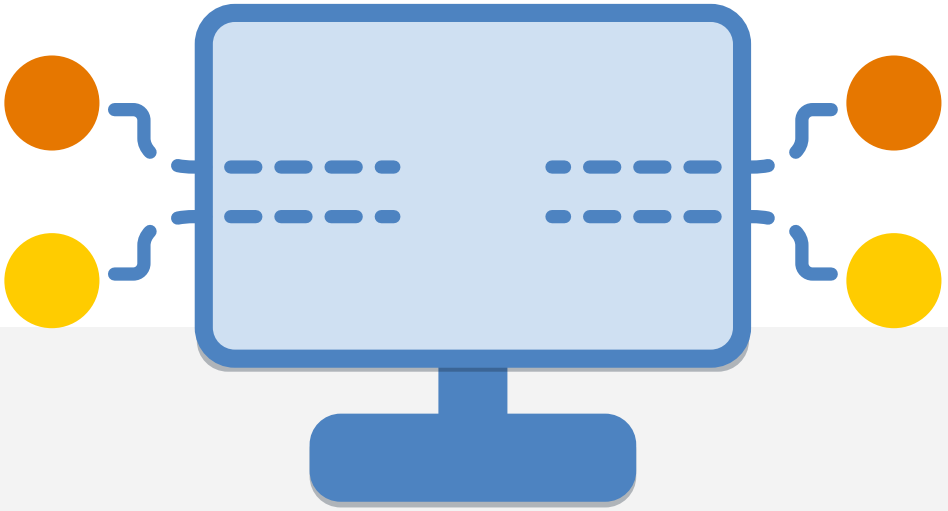
Addressing logging and semantic cataloging and mapping of streaming data early in the data pipeline can help reduce analytical expense downstream when it comes time to materialize, model and activate data.

Get Started with Real-Time Data

How does an organization adopt real-time data streaming? Many modern cloud services and retail data platforms already support streaming data transfer and processing. **You can check with your current software and cloud service providers to confirm they support streaming data transfer.**

Imagine what life will be like when we can instantly move our stuff and ourselves from anywhere to anywhere else instantaneously! We can thank futurists like Gene Roddenberry and scientists like Albert Einstein for advancing our vision of what's possible for humanity.

Until that future stardate — when we finally invent a working transporter, or when quantum computing becomes a commercial reality — **you can apply real-time data streaming to create an unfair advantage for your retail business today.**



9 Industry Context Matters:

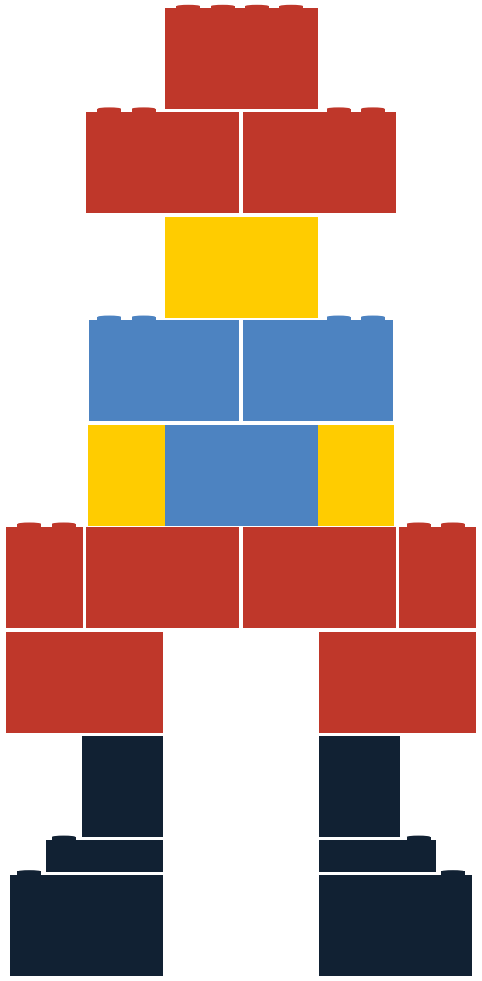
Use Purpose-Built, Industry-Specific Data Models for Time to Value

In 1978, **LEGO** introduced a brand new line of construction sets branded LEGO Space. The sets in the series included parts and features built for science fiction adventure and were among the first to include the now-iconic LEGO minifigure. Winning toy of the year in Germany and the UK in 1979, LEGO Space helped drive a 50% increase in sales for LEGO that year.

LEGO's strategy for LEGO Space – and every other theme that LEGO has launched since that early success – is to engage customers by making it easy to open a box and quickly experience the joys and benefits of a finished product. Through detailed instructions and clear pictorials, the toy company makes it easy for builders to get started quickly without limiting their future ability to create just about anything their imagination allows with the individual bricks included in each set.

This same guided approach to building can be applied to data models, especially useful for achieving fast, low-risk, and low-cost models suited for purpose in specific vertical industries like retail commerce.

In data engineering, one might equate individual data elements along with their semantic meaning, labels, and metadata as the buildable "bricks," with analytical or behavioral models landed in a data warehouse, ready for visualization or orchestration as the finished sets.



For data practitioners across industries, **there are many benefits of having predefined and prebuilt data models for analysis and activation.** The key is implementing a data architecture that supports prebuilt models without restricting the future customization of models and outputs in the process.

Here are a few key benefits of predesigned, prebuilt data models:

Time and Cost Savings:

Employing predefined and prebuilt data models can save time while achieving useful analytical and behavioral models. Analysis-ready models eliminate the need for manual data engineering, science, and analysis, all of which can be time-consuming and prone to error. Analysts save time that would otherwise be spent developing data models from scratch.

Semantic Consistency:

Predefined and prebuilt data models can ensure consistency across the analytical metrics and outputs like dashboards and visualizations. With predefined models, it is easier to ensure that data is organized consistently, which makes it easier to unify and understand data from different sources. Prebuilt models can help ensure that data is structured and presented consistently across all functions and departments, reducing confusion and errors. Different departments can collaborate more easily and work towards a common understanding of the data, promoting better decision-making.

Model Reusability:

Predefined and prebuilt data models can be reused for different analytical and data activation use cases. This can be particularly beneficial for retail businesses that have multiple brands or products, or seek to apply common metrics and models to disparate marketing or sales channels, or stages of an order lifecycle or customer journey.

Better insights:

By providing consistent measurement and reporting over time, predefined and prebuilt data models can help uncover new insights that might not be apparent with ad-hoc models. By starting with a solid foundation, businesses can more easily explore new patterns and relationships in their data.

Model Agility:

Prebuilt data models, when run on the right architecture and data pipelines, can be modified and adjusted to suit changing business needs and emerging trends more quickly than building new data models from scratch.

Data Compliance:

Prebuilt data models can be designed to comply with industry regulations and best practices, ensuring that data is handled securely and appropriately.

Improved Accuracy / De-Risked Data Projects:

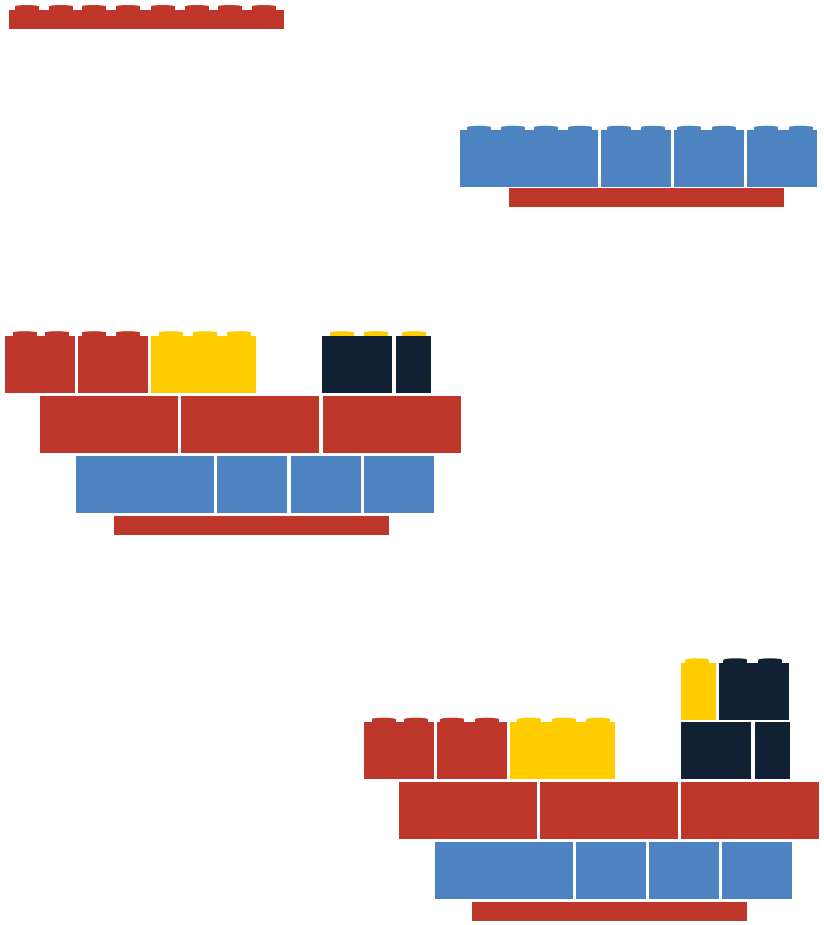
Prebuilt data models can help improve data accuracy by minimizing the risk of errors and inconsistencies that can occur when building data models from scratch.

Specific Purpose, Specific Model

Like LEGO sets that get creative builders started with guided instruction, prebuilt data models can help organizations improve their data management and analysis capabilities, promote collaboration across departments, and ultimately make better decisions based on data-driven insights at lower cost, time, and project risk.

LEGO's creative diversification that began decades ago with LEGO Space has continued with the proliferation of predesigned sets across interests and franchises, from sports and robots to princesses and ninjas.

As with LEGO, specific industries like retail have seen a proliferation of useful, prebuilt data models fit for purpose. Some of these data models still exist in the form of standalone applications like Customer Data Platforms (CDPs) and BI solutions. Increasingly though, these models are built or landed in open data infrastructure components (sometimes called the "Modern Data Stack"), with open data warehouses like Google Bigquery and Snowflake at the center.



What can retail decision-makers accomplish with prebuilt data models running in the Modern Data Stack?

Here are just a few examples:

Marketing Attribution Model:

This model helps in determining which marketing channels and campaigns are driving the most traffic and revenue.

Customer Segmentation Model:

This model groups customers based on similar characteristics such as demographics, behavior, and purchase history to personalize marketing efforts and improve targeting.

Lifetime Value Model:

This model calculates the expected revenue a customer will generate throughout their relationship with the business to prioritize acquisition and retention efforts.

Merchandising Performance Model:

This model helps retailers analyze the performance of their products by tracking metrics such as sales, margins, and inventory turnover. It is useful for identifying top-performing products and optimizing product mix.

Pricing Optimization Model:

This model helps retailers optimize their pricing strategies by analyzing market trends, competitor pricing, and customer behavior. It is useful for maximizing profits and staying competitive.

Product Affinity Model:

This model identifies which products are frequently purchased together to inform cross-selling and upselling efforts.

Inventory Optimization Model:

This model helps retailers optimize their inventory levels by predicting demand, identifying slow-moving products, and minimizing stockouts. It is useful for reducing costs and maximizing profits.

Supply Chain Management Model:

This model helps retailers manage their supply chain by tracking inventory levels, logistics, and supplier performance. It is useful for reducing costs, improving efficiency, and ensuring product availability.

Store Performance Model:

This model helps retailers analyze the performance of their physical stores by tracking metrics such as foot traffic, sales per square foot, and customer behavior. It is useful for optimizing store layout, staffing, and marketing.

Fraud Detection Model:

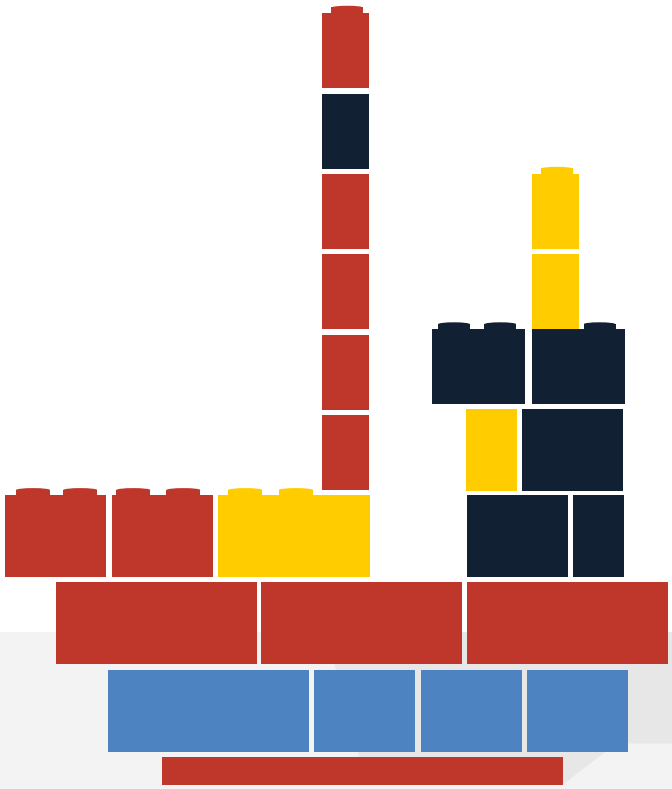
This model helps retailers detect fraudulent transactions by analyzing patterns in transaction data. It is useful for reducing losses and protecting against financial risk.

Build the Right Tool for the Job

Like the LEGO sets available for purchase today, the list of useful data models in an industry like retail commerce is almost endless. (If it's been a while since you last picked up and built a LEGO set, it's never too late to become a LEGO maniac!)

When it comes to data engineering, data science, and data analytics in your industry, now is the time to explore the rich and diverse analytical and behavioral models available. Starting with prebuilt models will reduce your engineering time, expense, and risk.

If you build or land your models in a modern cloud data warehouse like Bigquery or Snowflake, integrated with modern tooling, you'll have all the benefits of immediate time-to-value, and all of the flexibility to extend and customize your future-proof models as end-user needs and use cases dictate – just like a LEGO set.



Considerations and Next Steps

Cloud data warehouses like Google Cloud BigQuery and Snowflake provide the opportunity to do more with data than ever before. Modern tooling makes data work accessible to engineers and non-engineers alike, accelerating insights and actions across the organization. But cost and complexity are real concerns in the cloud data era, and the right approach and architecture can minimize your cost, time and risk to make data a true competitive advantage.

When implementing your cloud data warehouse, look for ways to simplify the data work by tackling key considerations both upstream and downstream of the data store. Immutably logging your data ahead of the data warehouse reduces analytical load on operational systems, and makes reinterpretation of data easier in the future. A robust semantic layer, defined and applied at data ingest simplifies downstream understanding and governance. No code/low code interfaces calling shared libraries of mappings and models improve access to data and accelerate time to value. Streaming data through your data pipelines unlocks real-time use cases ranging from site personalization to triggered customer communications.

In concert, all of these reference features make data more available and useful across the entire enterprise – driving competitive advantage and profitable growth.

We hope you've found this framework useful.

REACTOR

Put Your Data to Work.

CONTACT US

www.reactordata.com

1-888-417-6863

Grow@reactordata.com

Glossary

Atomization
noun

Separating something into fine particles. In a data context, processing data at field- or attribute-level fidelity rather than message-level groups. A simple example is the isolation of address fields from an order message to better handle the address data in isolation from the other order data.

Immutable
adjective

Not changing, or unable to be changed. Event logging tools like Kafka and Kinesis allow for "local" cloud storage of raw data in a way that allows for concurrent reads (from sources) and writes (out to the data warehouse) – a very fast architecture for flexibly and permanently staging raw data upstream of the cloud data warehouse.

Low Code/No Code (LCNC)
adjective

A method of designing and developing applications using intuitive graphical tools and embedded functionalities that reduce traditional engineering skills requirements. Low code/no code tools enable business analysts, office administrators, small-business owners and others who are not software developers to build and test applications. These people can create applications with little to no knowledge of traditional programming languages, machine code or the development work behind the platform's configurable components.

Pipeline Replay
noun/verb

The ability to modify data mapping and modeling rules (ideally with the ability to test and non-destructively rollback changes), applying new rules to an entire data set — both current and historical information. Pipeline Replay works on immutably logged data to affect changes to an entire data history when changes are required.

Semantic Layer
noun

The metadata or labeling that defines and describes a data set. Semantic labeling was traditionally applied to data sets using legacy data cataloging tools like Informatica or Collibra — in arrears, after the data was mapped and modeled. Modern architectures flip this sequence, applying rich semantic understanding to data sets at ingest, so definitions, labels and mappings are available at all points downstream.

Data Streaming
noun

Data that is emitted at high volume in a continuous, incremental manner with the goal of low-latency processing. Streaming data includes location, event, and sensor data that companies use for real-time analytics and visibility into many aspects of their business.

About the Authors

Jared Stiff

CTO, Co-Founder
SoundCommerce

Jared brings more than a decade of software development and analytics experience. Jared was Director of Engineering at CommerceHub, helping to lead the company through its NASDAQ IPO in 2017; and co-founder at MindCorps where he developed early production eCommerce systems for Fortune 500 companies before MindCorps' acquisition by Amazon.com.

jaredstiff

Rachel Workman

VP Value Engineering
SoundCommerce

Rachel brings more than 15 years of management experience in B2B SaaS senior leadership roles. Her career has focused on addressing big data and business analytics including applied artificial intelligence and machine learning challenges for leading corporations on a global scale. Today she leads the data practice at SoundCommerce, helping design and deploy the Reactor intelligent data pipeline.

rachelanneworkman

Eric Best

CEO, Co-founder
SoundCommerce

With more than 25 years of experience as an entrepreneur and executive, Eric is a tenured leader in consumer-direct and retail commerce, SaaS, cloud integration, digital currency and digital advertising. He has a proven track record growing and guiding companies and their teams to shareholder liquidity, including exits to NASDAQ:AMZN and NASDAQ:LVNTA.

ericbest

REACTOR

Reactor provides the fastest, most efficient path to useful, business-ready data for generative AI, analytics and activation. Reactor works with companies of all sizes and from every industry, onboarding and ingesting data from business critical systems and applications, landing clean, well-defined data modeled directly in your data warehouse. Founded by Amazon veterans and backed by leading venture capital investors, Reactor is headquartered in Seattle.

SOUNDCOMMERCE

SoundCommerce, powered by Reactor, is a composable data platform that connects and models marketing, operations, and merchandising data so retailers can optimize order and shopper profitability across all business functions. Built for retailers of any size or complexity, SoundCommerce transforms your unique data infrastructure into an easy-to-use, no-code environment that's accessible to everyone — no engineering degree required. With SoundCommerce, retailers have confidence that every decision and dollar drive profitable growth from first click to doorstep delivery. Founded in 2018, notable customers include: Eberjey, Hearst, PacSun, and GLDN.