

Data Integration: A Build vs. Buy Analysis

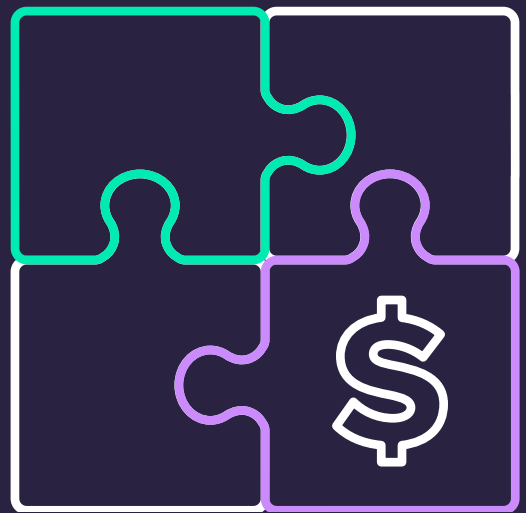


Table Of Contents



INTRODUCTION	1
A BUILD VS. BUY DATA INTEGRATION SCENARIO	2
EVALUATING BUILD VS. BUY OPTIONS FOR DATA PIPELINES	3
CONSIDERATIONS FOR BUILDING DATA INTEGRATION	4
CONSIDERATIONS FOR BUYING DATA INTEGRATION	5
DATA INTEGRATION BUILD VS. BUY ANALYSIS	6
CONCLUSION	7

01

INTRODUCTION

Data is the lifeblood of the modern enterprise, and many organizations are using data to transform their business.

Everyone is under competitive pressure to derive the most value in the fastest way possible from their data. The need to make data-driven decisions based on the most recent information has never been more critical. Yet, most enterprises struggle because they are bogged down by slow, costly, and outdated data integration methods and technologies.

Enterprises have relied on the same basic architecture and processes for data integration for years. Legacy environments built on complex and costly static pipelines are being replaced by a new generation of cloud-based tools and services. The goal is to improve the effectiveness and timeliness of reporting and analytics while eliminating these backward-looking and expensive data integration processes. The result is

something every enterprise will benefit from - analyses that once required weeks of manual [data engineering](#) and integration effort can now be performed in minutes while opening the door to self-service access to data.

It's time for enterprises to critically evaluate the data integration tools and processes that they depend on. Increasing reliance on external service providers turns the data integration process upside down. Individuals and departments are no longer beholden to central IT departments. Approaching today's data environment with yesterday's mindset will result in lots of unnecessary pain. Companies may be running hundreds



of cloud applications and services, each contracted for by a team that doesn't communicate with other teams or with central IT. Data integration is a massive problem today and it is rapidly getting worse.

This paper presents both sides of the classic “build vs. buy” debate when it comes to data integration and data pipelines. Enterprises face two basic, yet nuanced, choices:



1 BUILD

Many companies have already built some level of data integration so they approach an overall data integration program with the mindset that they can build a library of code that connects different systems and stitches their siloed data together. When it works, the building can provide a heavily customized and performance-tuned data pipeline. When it doesn't, enterprises can end up with a morass of code, built by different people for different reasons, with every change and addition running the risk of breaking the entire pipeline.

2 BUY

Enterprises that leverage the cloud for data services can draw on the same principles to buy a data integration service. The best data integration services are fast and easy to use, completely customizable, seamlessly scalable, and, perhaps most importantly, empower business teams to focus on meaningful analyses rather than on technical challenges. Buying a data integration solution frequently results in quicker time to market, greater flexibility, and greater ability to address moves and adds. Drawbacks to buying include variations in the degree and ease of customization that may result in decreased functionality and efficiency. Not all integration services have the same functionality and making the wrong choice will cost you time and money.

02

A BUILD VS. BUY DATA INTEGRATION SCENARIO

Imagine the case of an online retail company that has been relying on services such as NetSuite, Salesforce, Marketo, HubSpot, and an in-house database (Microsoft SQL Server) for operations.

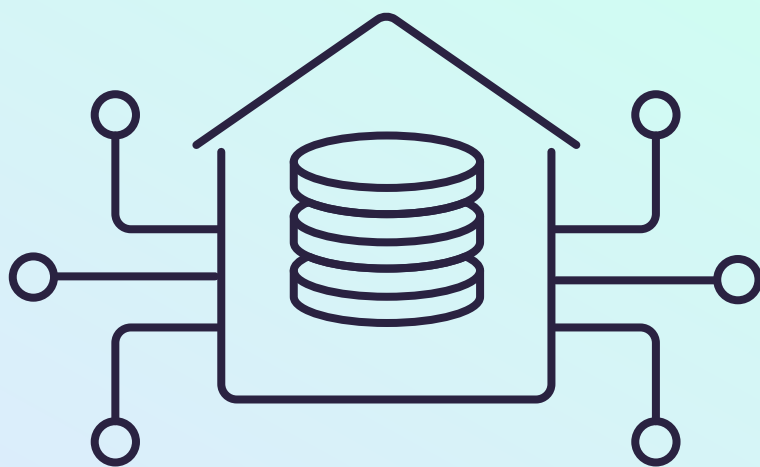
Each tool has its own reporting mechanisms, although complex reports that span multiple data sources require considerable engineering effort when business managers request custom ad-hoc reports. Departments that prioritize operations over insight are pleased – the attitude is that everything is running smoothly so don't risk throwing a wrench in the works just to get better reporting. However, executive leadership is deeply concerned that the company is falling behind competitors due to the inability to glean insight from siloed raw data. Everyone agrees that no one can see the big picture, and while operations are running smoothly, forecasting, inventory management, and reporting suffer.

The CTO has a plan to build custom tooling using open source frameworks to ingest all company data into a [data lake](#), then build ML and analytics workflows on top of it. He feels strongly that he can build a team that will meet the needs of the business by hand-coding a series of data integration pipelines.



The CMO wants to buy (or subscribe to) best-of-breed data integration tools instead of building custom tooling. He feels this will be the most time-effective and cost-effective approach to achieve a scalable, flexible, and feature-rich data integration pipeline.

What should the company do? Focusing on the custom ETL and data integration steps, let's dig deeper into each option to see what's entailed. Building a data integration solution in-house requires engineering bandwidth and technology infrastructure. On the other hand, will a data integration service be able to meet the exact needs of all of your use cases?



03

EVALUATING BUILD VS. BUY OPTIONS FOR DATA PIPELINES

It's critical to understand that data integration technologies and processes collectively termed the data pipeline is undergoing rapid change.

The traditional way to approach the gathering and pre-processing of data is ETL, but this is giving way to an API-driven methodology. ETL was a great answer for many years when enterprises dealt with a limited number of fixed data sources, but, given the multitude and variation of data feeds today, ETL is a bulky and error-prone combination of manual and automated processes.

ETL and batch operations can't fulfill an enterprise's data integration needs in the streaming data age. Even if they could, a hand-tooled data pipeline can quickly become remarkably difficult, if not impossible, to maintain. Data sources and structures change almost as frequently as the data itself. Every single change is

capable of breaking the entire ETL process, and such changes occur often enough that data developers will spend too much time maintaining the data pipeline when they could be coding something of value instead.

The modern way to gather data for analysis is to leverage each service's API to query for the necessary data. Each service and each software has its own API. Consuming data from various sources means learning each service's API. It's a given that APIs change over time, so building connectors to tie together data from every aspect of the business will require ongoing effort. Finally, remember that overall complexity multiplies with each additional data source added to the data pipeline, making constant growth and change a serious issue.

04

CONSIDERATIONS FOR BUILDING DATA INTEGRATION

It's also important to look beyond the framing discussion of ETL vs. API driven data pipelines and help our CTO and CMO decide whether to build or buy the company's data integration solution.

The traditional approach would be to build custom applications and tooling in-house that would then be assembled into an ETL workflow or data pipeline. This is not an impossible feat, yet it is time and resource consuming and therefore prevents developers from working on code that provides value. Of course, everyone wants to say “I built that”, but the decision to build or buy data integration must be based on the value that is achieved when an organization has all of its data on tap for analysis. Today's enterprise IT teams are struggling to stay afloat learning about new technologies and introducing them, patching software and upgrading existing systems. not on a desire for ownership. How

can an in-house team, already fully tasked, successfully develop, deploy, maintain and extend a data integration pipeline, especially when compared to a SaaS offering that is already built and maintained by a team of experienced data integration engineers?

Instead of burdening existing teams, you might decide to build a new team of data integration specialists. The cost of establishing a data engineering team may make it difficult to realize the intended value of building a solution in-house for all but the largest global organizations. Building integration software will require a significant



amount of development resources. Before starting, it is important to evaluate whether you do, in fact, have the capabilities required to start and successfully complete the data integration project. Estimates are that more than half of integration projects are obsolete before completion. In today's competitive environment, no business can afford to squander precious developer and data engineering time and budget and also take on the risk that the project will fail.

It's not just about hiring and developing a solution, it's also about maintaining that solution over time. Teams will change and individuals will move to other groups or even other companies. Over time, tracking and maintaining dependencies will require comprehensive documentation and collaboration. Without that, the data engineering team can be overwhelmed with maintaining the pipeline, and eventually, the data engineering team itself becomes the bottleneck on data projects.

Understanding the true investment and return of building enterprise-grade data integration software in-house requires more than a surface analysis. The major requirement is also the major stumbling

block – hiring and building a team. That will take time and money, probably more than anticipated because good data engineers are hard to find and in high demand. It's important to understand the total cost of ownership (TCO) will include significant maintenance, security patching, and updates.

Employee retention is an important factor when building a data integration team and staffing it for the long haul. It shouldn't surprise IT leadership that no developer wants to toil in drudgery, manually maintaining a system that someone else designed and built. In a typical enterprise, an attempt is made to solve this problem by assigning junior engineers to maintenance projects, but they frequently require oversight and guidance from senior developers, resulting in an entire disgruntled team working to maintain systems.

Talented engineers are quick to leave roles that include what they see as too much maintenance work because they would rather put their effort into building new, fun projects where they can hone their skills.

The decision you make today about building or buying data integration could result in overburdening the team with the technical

debt of maintenance, and that will, in turn, lead to poor engineering retention. Poor engineering retention leads to poorly written and maintained code. It just gets worse and worse.

Today's business runs on data that comes from many sources. The days of simple data engineering pipelines to manually ETL data from a handful of purely in-house transactional systems are over. Businesses consume data from many sources, as in hundreds or thousands, programmatically through APIs. Data engineering teams must maintain an ever-changing catalog of data sources, including associated API definitions and schema. They must plan and build a workflow that makes necessary API calls to each service or software, including tests and verification steps.

The best practice when investing in a data integration project is to focus on the long term. This means that the solution, regardless of whether it is built or bought, must scale to handle growing numbers of data sources and APIs while not burdening the team with cumbersome (manual) processes that can't scale. Data engineering teams must be fast moving – adding and keeping up with changing feeds requires that they deploy, test, and update multiple moving parts of the workflow and code it contains on a daily basis.



05

CONSIDERATIONS FOR BUYING DATA INTEGRATION

The first question to ask when evaluating data integration services is whether or not your business requirements will be met.

Like all software development projects, data integration must also start with a list of requirements. The last thing you want to do is subscribe to a service that will only satisfy a part of your data integration needs and ultimately lacks the flexibility to be adapted to your use cases – today and tomorrow.

In the best case scenario, you engage a data integration service that has more and deeper functionality than you require at the beginning. Over time the data engineering team uncovers capabilities that they didn't even know they needed, and they bring them to bear within your organization.

On the other hand, buying a data integration solution simplifies the required technologies and processes greatly, outsourcing the “plumbing” or “guts” of the data engineering

pipeline so your organization can focus on the data and its insights. The fact that ready-made solutions are available tells you that you are not the only one facing these challenges. Data integration services have already spent thousands of developer hours to build and test their services, so little to no additional burden will be placed on your developers.

A data integration solution should be both powerful and flexible. Typically, these tools are developed by an experienced team that has “been there and done that”. They are able to ensure that there is no data loss or data errors and provide real-time availability of data, no matter its scale or location. Data integration services are typically more robust and observable than in-house solutions.

They include dashboards for monitoring and issue alerts when tasks fail. Customer support teams are available around the clock to help when errors occur and jobs fail.

Data integration services dramatically streamline the process of building a data integration pipeline. Creating an automated workflow doesn't take that much more effort beyond declaring credentials and understanding schema. The architecturally complex multi-service workflow is all behind the scenes, where your team drives it all using a straightforward GUI. Finally, in-house developer cycles do not go towards maintaining the service, minimizing maintenance costs as the service maintains the code and workflow.

Placing the burden of maintaining and supporting the system frees internal developer resources to work on other projects. Self-service data integration platforms are usually fully documented and provide context-sensitive help while working with them in a browser. When there is a problem, dedicated technical support is instantly available. There's also a team already at work building new functionalities and adding more data APIs.

When selecting a data integration service, it is critical to evaluate the vendor as well. Building and maintaining a data integration pipeline is no small feat. You will run the risk of wasting this effort should you decide to leave the provider. Choose carefully to minimize this risk. Another key factor to assess is product functionality and the feature roadmap so you can be confident that the provider will meet your needs in the future as well as today. Finally, evaluate the vendor's support policy and quality because you will be dependent on the vendor to support your team and fix issues.

A final note about security and confidentiality. A data integration platform is fully managed, including security configurations and patching. Connections are made using HTTPS or a VPN whenever possible, encrypting sensitive data as it traverses the internet.



06

DATA INTEGRATION BUILD VS. BUY ANALYSIS

Reverse ETL lets your teams access up-to-date information in the business applications they already use.

Planning data integration services is a critical step in building and maintaining the data pipelines that feed business operations, reporting, and analytics. Each side of the build vs. buy equation has its own pros and cons. Yet, the most important aspect of the decision is which will meet the unique requirements of your business.

The table below summarizes the various aspects of the build vs. buy decision for data integration platforms.

CONSIDERATIONS	BUILD	BUY
FOCUS	Consider building if data engineering is a core competency	Consider buying if there are no in-house data engineering resources and you require a streamlined solution
REQUIRED RESOURCES	Front-end designers, engineers and developers specializing in data engineering to build, test, support and maintain the pipeline	A solid business understanding of the company's data and its sources
TOTAL COST OF OWNERSHIP	Hiring dedicated staff	Licensing and training costs
EASE OF USE	Typically not user-friendly, not easily extensible	Data integration services are built for business users with a focus on user-friendly interface
PERFORMANCE & SCALABILITY	Requires considerable expertise to build efficient, performant, and scalable data pipelines that run on-premise and at multiple cloud providers.	Data integration services are designed to process large data sets quickly
UPDATES & COMPLEXITY	Better for a fixed number of data sources with static schemas	Better for changing numbers and schema of data sources

07

CONCLUSION

The age-old question of build vs. buy now extends beyond systems to include data workflows, processes, and technologies.

No system is an island anymore, no enterprise data decision is made in a void. In order to stay competitive, organizations must make greater use of their data as a whole.

This paper explored both sides of the build vs. buy decision for data integration and concluded that the clear winner is buying because of advantages in time-to-market, cost, and ongoing developer effort. The result is that business teams don't require IT support for every data operation or report. Instead, they have a GUI with support merely a click away.

Enterprises that choose to build an in-house solution run the very real risk of failing due to the complexity of maintaining a data integration pipeline. APIs are constantly evolving and changing. Hand coding fixes to keep up with such changes is time-consuming and error-prone. Even the data

engineering team that gets everything right from the start could still end up with messy and unmaintainable code. If data is the most valuable asset an enterprise has, would you really want to put its integrity at risk every time an API changes?

There is an order of magnitude of value created by buying. Instead of devoting developer resources to maintain and support the data engineering code, they can focus on higher-value projects like refining ML models, building dashboards and reports. Developing new applications can even become faster when using an integration service because most new applications will draw on data that's stored in existing systems and processes that team members are already engaged in. Businesses benefit when development and data teams are more closely aligned with the business, not heads-down focused on every line of integration code.

We hope that you have enjoyed reading this whitepaper.

If so, be sure to [check out our other Whitepapers](#) related to "[Data Integration: A Build vs. Buy Analysis](#)".

Interested in learning more about Integrate.io?

Chat with an Expert!

