hightouch Fivetran

A Warehouse-Centric Approach to Data Integration

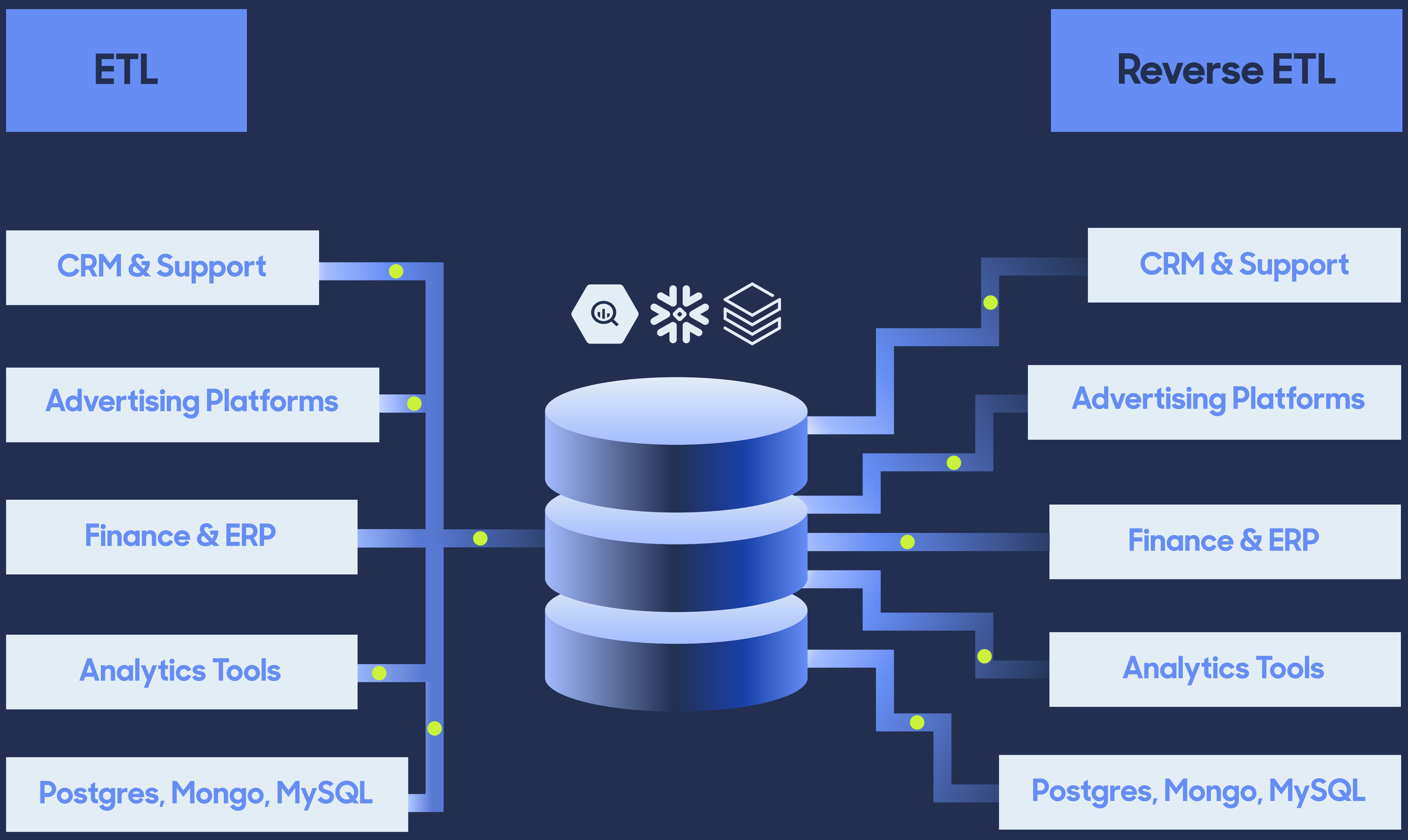


Table of Contents

Preface

The data ecosystem has changed drastically over the last ten years, but data integration has largely remained the same. Despite a groundswell of industry innovation, the challenge of moving data to and from systems, reading and writing to various APIs, and maintaining custom pipelines hasn't gotten any easier. It's only become more complicated as the number of data sources has steadily increased.

Today, companies have hundreds, if not thousands, of data sources across their internal applications, databases, and SaaS platforms. For data teams that are responsible for delivering analytics and actionable data to business teams, there are many factors to consider when choosing an integration approach.

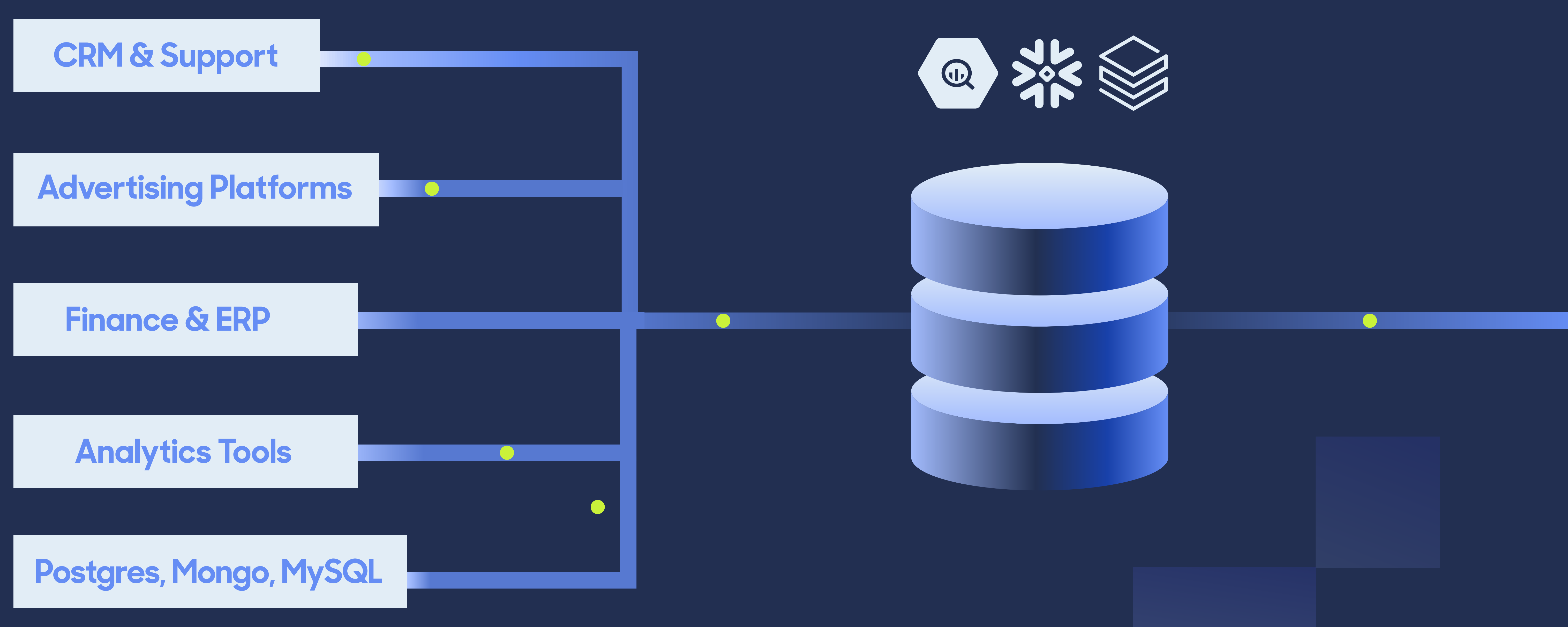
Building point-to-point integrations and pipelines is expensive, time-consuming, and challenging. As data sources and consumers of that data continue to explode, it's becoming increasingly obvious that the warehouse should be the organizational center of gravity. Additionally, it's becoming increasingly difficult to make a case for maintaining custom ETL pipelines and doing data integration in-house.

Data Integration Approaches

For data engineers, data integration has long been synonymous with ETL (Extract, Transform, and Load), the process of extracting data from its source, transforming it into a usable format, and loading it into a destination (typically a data warehouse or database). Conventional ETL has been around since the 1970s, and for the most part, the process of extracting, transforming, and loading data has remained essentially unchanged. There are three core use cases for data integration: ETL, Reverse ETL, and point-to-point integrations.

ETL

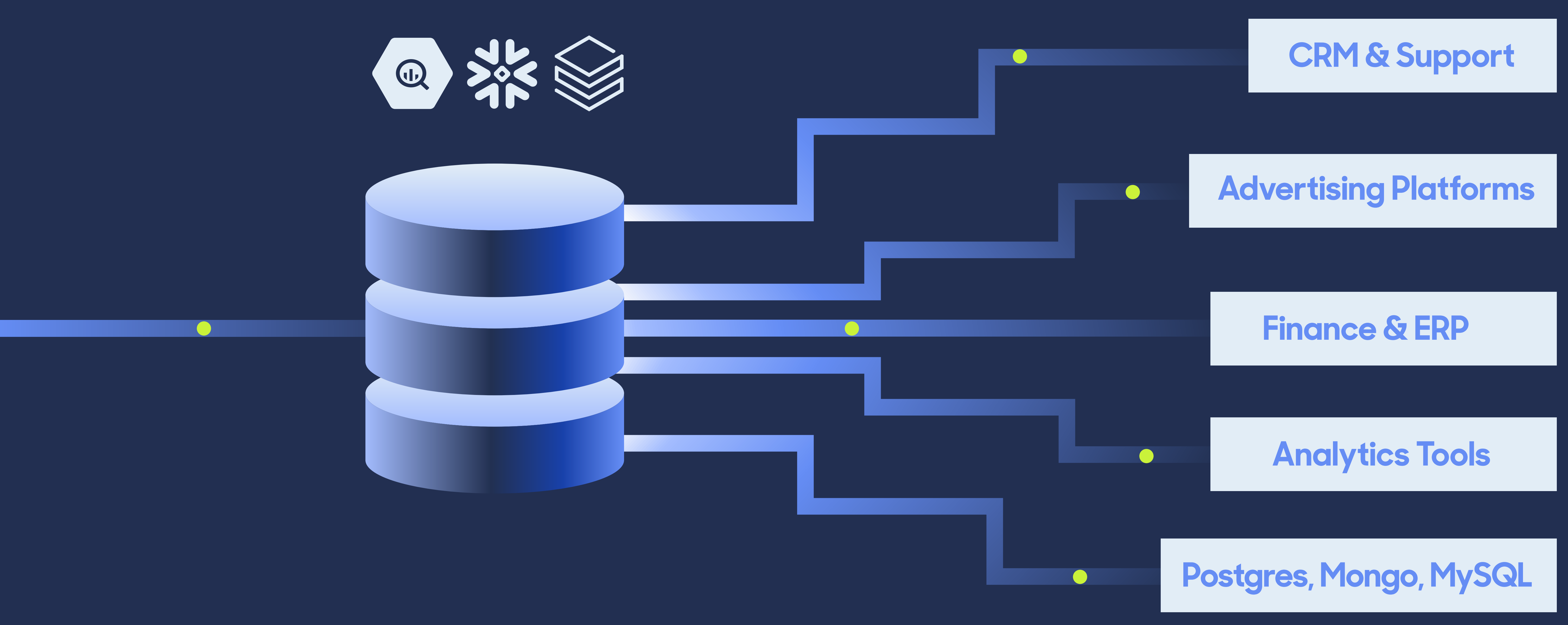
ETL is focused on ingesting data into a centralized platform or data warehouse. This means reading/writing data from a data source to a warehouse (e.g., Snowflake or BigQuery) so data teams can perform analytics and deliver data to internal stakeholders, typically in the form of an operational dashboard or report. Most of the time, this data is consumed through a Business Intelligence (BI) tool like Looker, so business teams can have clear visibility into critical financial data, KPIs, and specific north star metrics.



Reverse ETL

In the past few years, a new pattern has emerged. While ETL brings data into the warehouse, Reverse ETL is focused on moving transformed data out of the warehouse and syncing it not just to a static BI report or dashboard, but back into the operational tools business users spend their time in (e.g., Salesforce, Google Ads, Marketo, etc.).

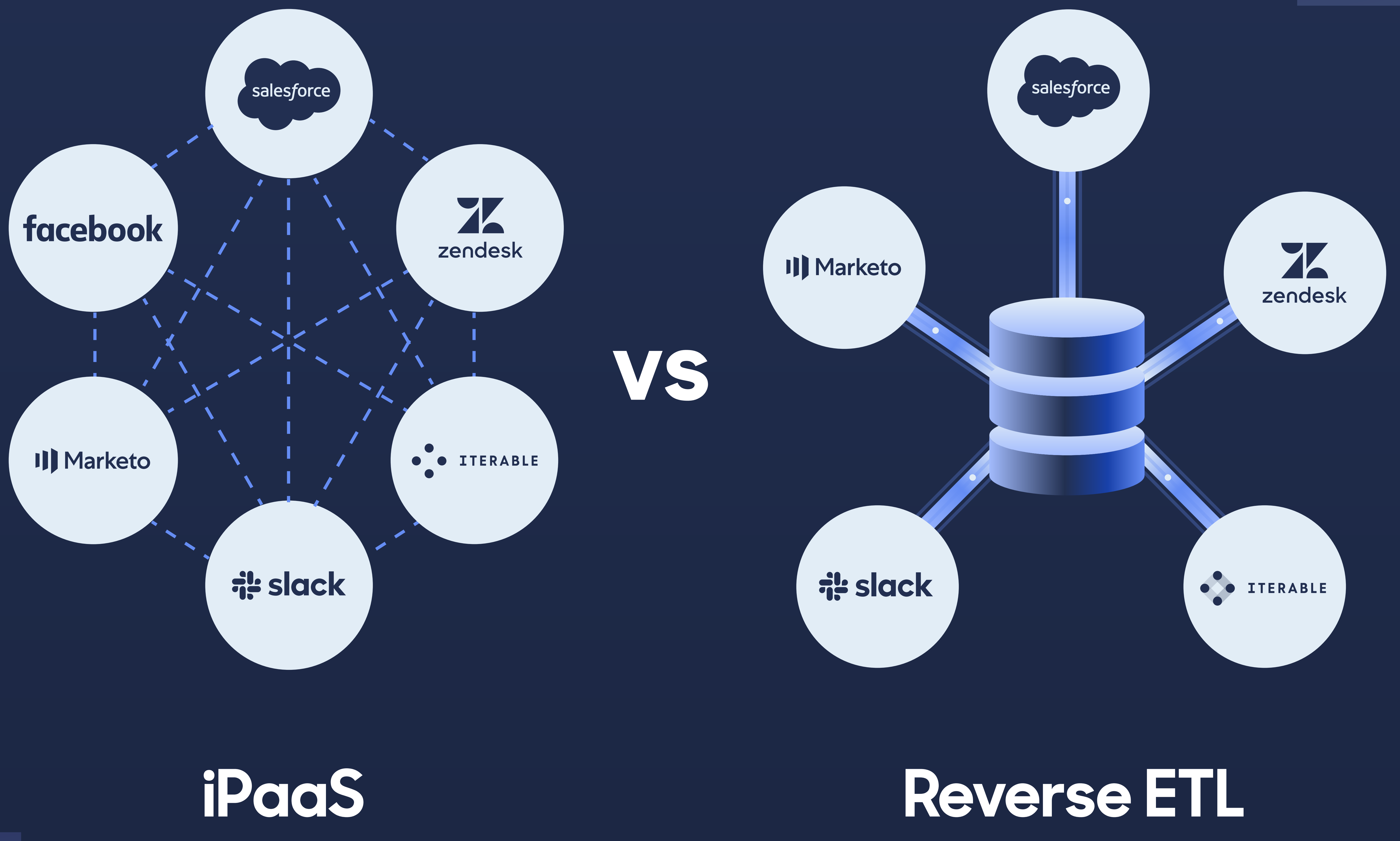
When the category emerged a few years ago, practitioners gravitated towards calling it Reverse ETL because of the fact that it had inverse sources and destinations compared to traditional ETL. Reverse ETL is an architectural approach that enables Data Activation by making actionable data readily available to business users across sales, marketing, support, and more.



Point-to-Point Integrations

The alternative to warehouse-centric pipelines is having each app share data directly with each other. This integration approach is known as "point-to-point" and has been popularized by Integration Platforms as a Service (iPaaS) vendors over the past decade. iPaaS solutions ignore the warehouse, or any organizational source of truth, and instead stand up bespoke pipelines between specific applications.

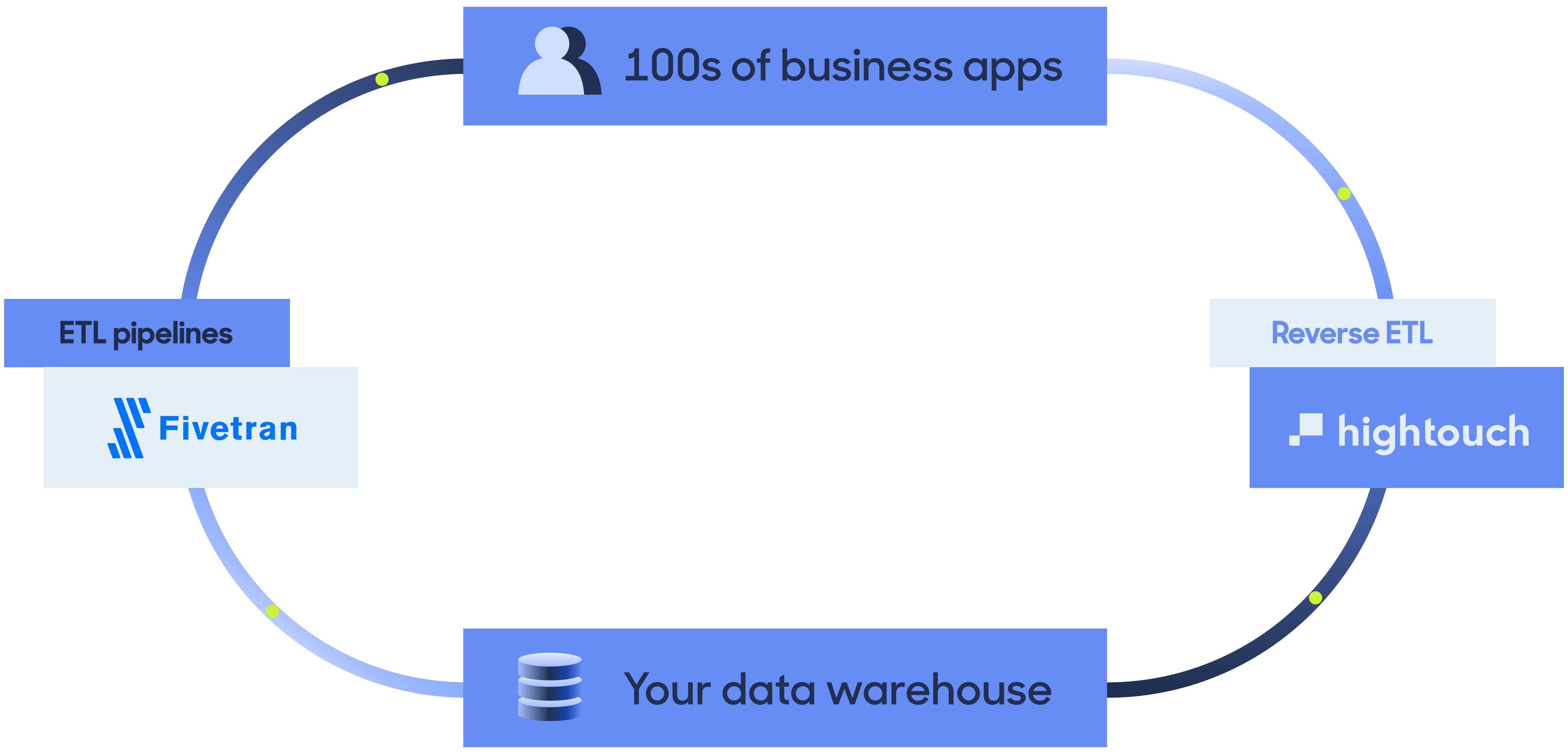
At scale, this becomes an exponential spiderweb of connections for teams to build and maintain with various definitions of "what's true" about the customer contained in each source and destination. You can read more about the tradeoffs between iPaaS and warehouse-centric pipelines in this thorough writeup from Hightouch here.



The Future is Warehouse-Centric

With the rise of the cloud and the native separation of storage and compute in platforms like Snowflake and BigQuery, there has been a steady shift away from on-premises hardware towards "as a service" offerings. This transition, coupled with the flexibility and scalability of these platforms, has given rise to a slightly altered paradigm of ETL known as ELT because it's now possible, more cost-effective, and more efficient for data teams to port as much relevant data as possible into the warehouse and then transform it, all transform data directly in the warehouse without destroying their entire production environment.

With the efficiency and scale of the cloud data warehouse, it no longer makes sense for companies to build and manage one-off point-to-point integrations and the complexity that comes with managing a spider web of data pipelines. The combination of ELT and Reverse ETL introduces a hub-and-spoke approach to data integration where the data warehouse sits at the center of every pipeline as the consistent single source of truth.



Why Data Teams Shouldn't Build and Maintain Their Own ETL Pipelines

The value of having up-to-date, transformed data readily available to business teams is obvious. With real-time and consistent customer data at their fingertips, sales teams can close deals faster, success teams can reduce churn, and marketing teams can drive more personalized campaigns. Modern data teams are forgoing tedious and brittle point-to-point integrations and instead leveraging the warehouse—and a hub-and-spoke architecture—to optimize how the data flows throughout the business.

But then comes the next decision point: whether to stand up custom/in-house ETL and Reverse ETL pipelines or to outsource to a best-in-breed vendor like Fivetran and Hightouch. The are many technical intricacies of building scalable and resilient ETL and Reverse ETL pipelines, and the choice to go at it alone comes with considerable challenges that become untenable in modern data environments.

The Technical Challenges of ETL

As organizations increasingly rely on data-driven decision-making, the ability to efficiently and effectively move and transform data from multiple sources to a destination becomes critical. This process of ETL can be complex and involve many technical challenges. Here are several key challenges that organizations may encounter when implementing ETL:

Building and Maintaining Pipelines

Building and maintaining data pipelines involves designing and implementing the processes and infrastructure needed to extract data from various sources, load it into the destination system, and possibly transform it along the way. This can be a complex process, especially if the data sources are diverse or the data needs to be transformed in some way before being loaded into the destination. It may be necessary to use tools such as ETL software, data integration platforms, or custom code to build the data pipelines. Ensuring that the data pipelines are efficient, scalable, and flexible enough to handle changes to the sources and destinations over time is a key technical challenge.

Schema changes

A schema is the structure of a database or data model, and changes to the schema can have a significant impact on the data pipelines. If the schema of a source or destination changes, the data pipelines may need to be updated in order to continue functioning properly. Modifying data transformation logic, or updating the integration between data pipelines, sources, and destinations often requires data teams to write custom code, modify SQL queries, or even update configuration files and settings.

Data models

Designing and maintaining data models and entity relationship diagrams (ERDs) can be a technical challenge, especially if the data is complex or changes frequently. Data models help to define the structure and relationships of the data, and a well-designed data model can make it easier to work with and analyze the data. However, designing and maintaining data models can be time-consuming and requires a deep understanding of the data and the business requirements, as well as knowledge of data modeling concepts and techniques.

Data normalization

Data normalization is the process of transforming the data into a consistent format that is easy to work with and analyze. This can be a technical challenge, as it often requires writing complex transformation logic using tools such as SQL, Python, or Java and ensuring that the data is consistent and accurate. This may involve using techniques such as data cleansing, data transformation, or data aggregation.

Re-syncs

Re-syncing, or the process of updating the data in the destination to match the sources, can also pose difficulties. Data teams often need to write complex queries or data transformation logic to update the data in the destination. This can be time-consuming and resource-intensive, especially if the data is large or changes frequently.

Data Replication

Data replication involves moving and processing data in specific ways, such as replicating only certain data types or filtering out certain data elements.This requires a deep understanding of the data and business requirements, as well as the ability to write custom transformation logic using SQL, Python, or Java.

Batching

Batching, or the process of grouping data together for efficient processing, requires careful planning and execution in order to ensure that the data is processed efficiently and accurately. It may also be necessary to design and implement mechanisms to handle errors or failed batches, which can further increase the complexity.

Overall, the technical challenges of ELT can be complex and multifaceted, and organizations need to carefully plan and implement their data pipelines in order to ensure that they are efficient, effective, and able to handle changes over time.

The Technical Challenges of Reverse ETL

Much like conventional ETL, building Reverse ETL also comes with a number of technical challenges that should be thoroughly evaluated by any team considering building these pipelines in-house.

Multiple Writers

Writing to a third-party API is vastly different than writing to a data warehouse because every SaaS application and end tool has a unique API. This means there are countless (and growing) end-point APIs to build and maintain. In addition, SaaS tools like Salesforce and Iterable often have multiple writers changing and updating fields/objects (e.g., sales teams, marketing teams, support teams, etc.) With ETL, there is only one writer writing to the data warehouse (e.g., the warehouse admin.)

One crucial factor to note is that most operational tools don't have a time-travel feature, meaning that if the wrong data is accidentally synced to the destination, it's easy to overwrite existing fields, and this can be disastrous (e.g., sending an outbound email campaign with personalized information to the wrong account!) With ETL data, teams can simply just clear the database in the warehouse and re-ingest the data. Unfortunately, this isn't possible with most SaaS tools.

Change Data Capture and De-duplication

To maximize efficiency, speed, and cost, data teams need to use Change Data Capture (CDC) processes and de-duplicate data before syncing. With ETL, it's easy to merge data using the source's most recent "updated_at" field. This is impossible with Reverse ETL. The only way to de-duplicate data is to compare the values in the warehouse against what's previously been synced.

Under-the-Hood Transformations

Unlike the data warehouse, where it's easy to configure every aspect of the environment, third-party tools and SaaS applications all come with their own unique data models and schemas. When standing up in-house Reverse ETL pipelines, this means engineering teams need to perform under-the-hood transformations on the data before it can be synced to the end destination (and to do this for every unique end destination).

Rate Limits and Batching

Every SaaS tool has multiple different APIs. However, they often impose rate limits to restrict the number of calls that can be made to a specific endpoint. Rate limits can bottleneck syncs quickly, so batch APIs are usually the best way to avoid rate limits.

Observability and Error Handling

When dealing with large batches of data, changing schemas, and multiple different endpoints, it's not a matter of if an error will occur but when it will occur. Errors can happen for many different reasons, whether it's for an entire request or at the row level, and this means engineering teams need a way to monitor sync performance. This adds another layer of complexity as data teams are forced to write sync logs back to the warehouse if they want to analyze metadata and build a live debugger to monitor API requests/responses.

User Experience and Maintenance

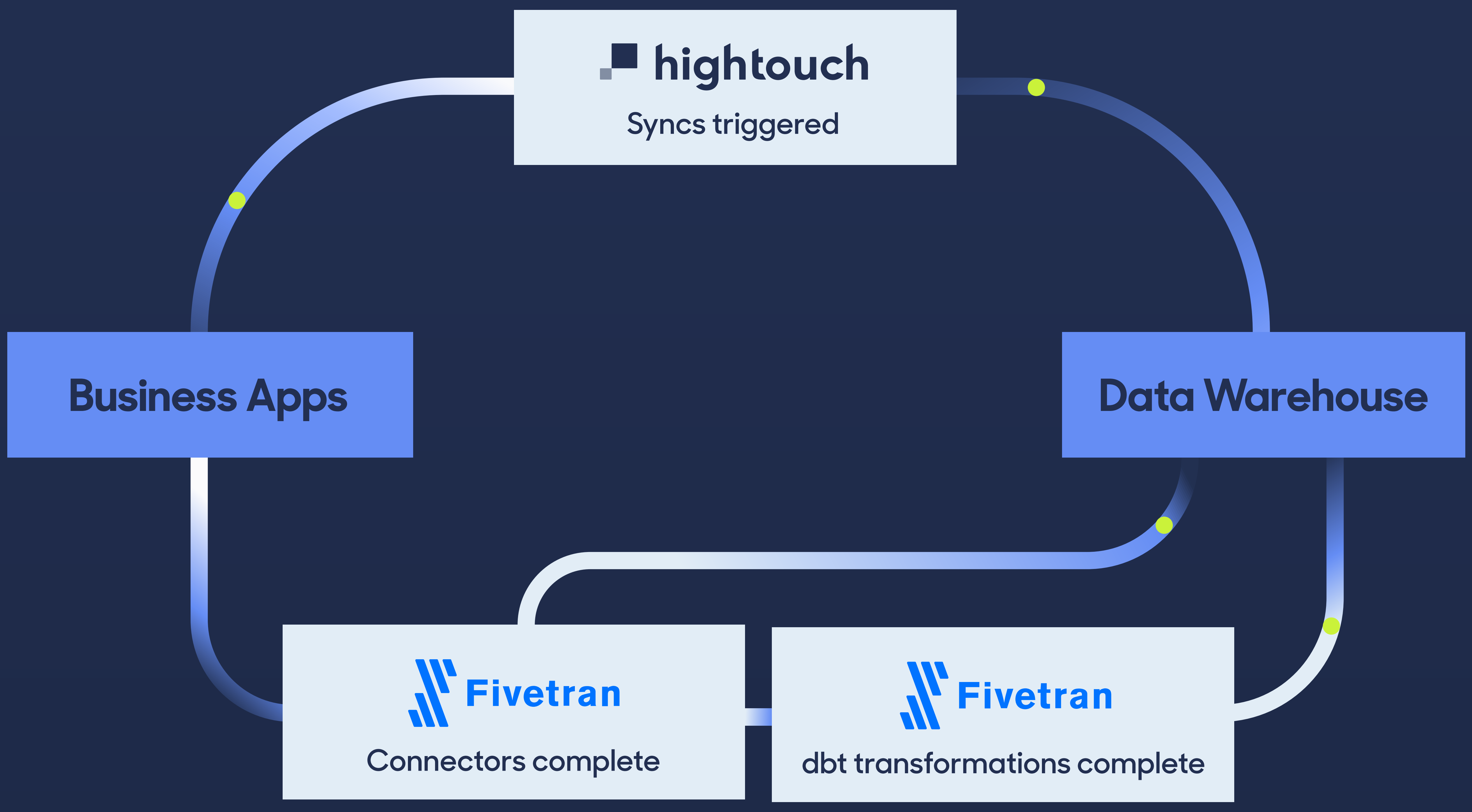
Each of the technical aspects above presents its own complexities, and data teams have to account for each one every time they want to sync data from the warehouse to a downstream destination, and that means they need to ask some important questions:

- *How will you send data (e.g., objects, events, etc.) to your destination?*
- *Which object should you sync data to?*
- *What primary key should you use to map your fields?*
- *How should the fields in your data model be mapped to the fields in your end destination?*
- *How should your data update in your end destination?*
- *When/how frequently should your sync run?*
- *How should errors be handled?*
- *What's the desired rate limit?*

All of this requires a complex, robust, and reliable infrastructure, and that's not even to mention that third-party APIs change all the time. Having a user interface that is powered by well-defined abstractions around reusable components will eliminate custom scripts.

Getting Started with Fivetran and Hightouch

Together, Fivetran and Hightouch eliminate the need for manual and brittle point-to-point integrations, create a continuous flow of data from acquisition to ingestion, and close the loop between business intelligence and Data Activation. With Fivetran's out-of-the-box connectors, data is immediately available in the data warehouse. From there, data teams can use Hightouch to easily sync that transformed data into over 125 downstream tools.



Setting Up Fivetran

Fivetran can help organizations take the first step toward building a modern data stack. We support a wide range of common destinations, hundreds of common data sources, and transformations.

Organizations have used Fivetran to jumpstart their way to the modern data stack in a number of ways as Fivetran enables data teams to quickly and easily set up multiple data connectors so that they can focus on putting that data to good use, rather than building and maintaining complex, costly data pipelines.

To get started with Fivetran, you can sign up for a 14 day free trial account. Then follow these steps to get your first data connector up and running:

1. **Select your data source.** Choose from over 250 different data sources including Salesforce, Google Ads, Shopify, AWS S3, and more. View a full list of our supported connectors. The page that appears has instructions on setting up your specific data source so that Fivetran can interact with it.

2. **Select your data destination.** Fivetran currently supports 11 destinations, including Google BigQuery, Databricks, PostgreSQL, and more. View the list of available destinations. When you land on the destinations page, you'll see specific instructions on how to set up your specific destination, including how to define the roles and permissions that Fivetran needs to connect successfully.

3. **Complete your initial data sync.** Finally, once your data source and destination are set up, the last thing to do is to begin replicating your data. Fivetran will test that your source and destination are properly set up, then begin your initial data sync. If you have a high volume of data, this initial sync can take a little longer, but we'll email you when it is complete. From the sync page, you can choose how frequently you want your data to sync, view any exceptions, and see the current status of your data sync.

4. **Query your data.** Fivetran offers many turnkey data models that normalize your data so that it is instantly queryable with high performance. View and discover the available data models.

For more detailed information, check out Fivetran's Getting Started page.

Setting Up Hightouch

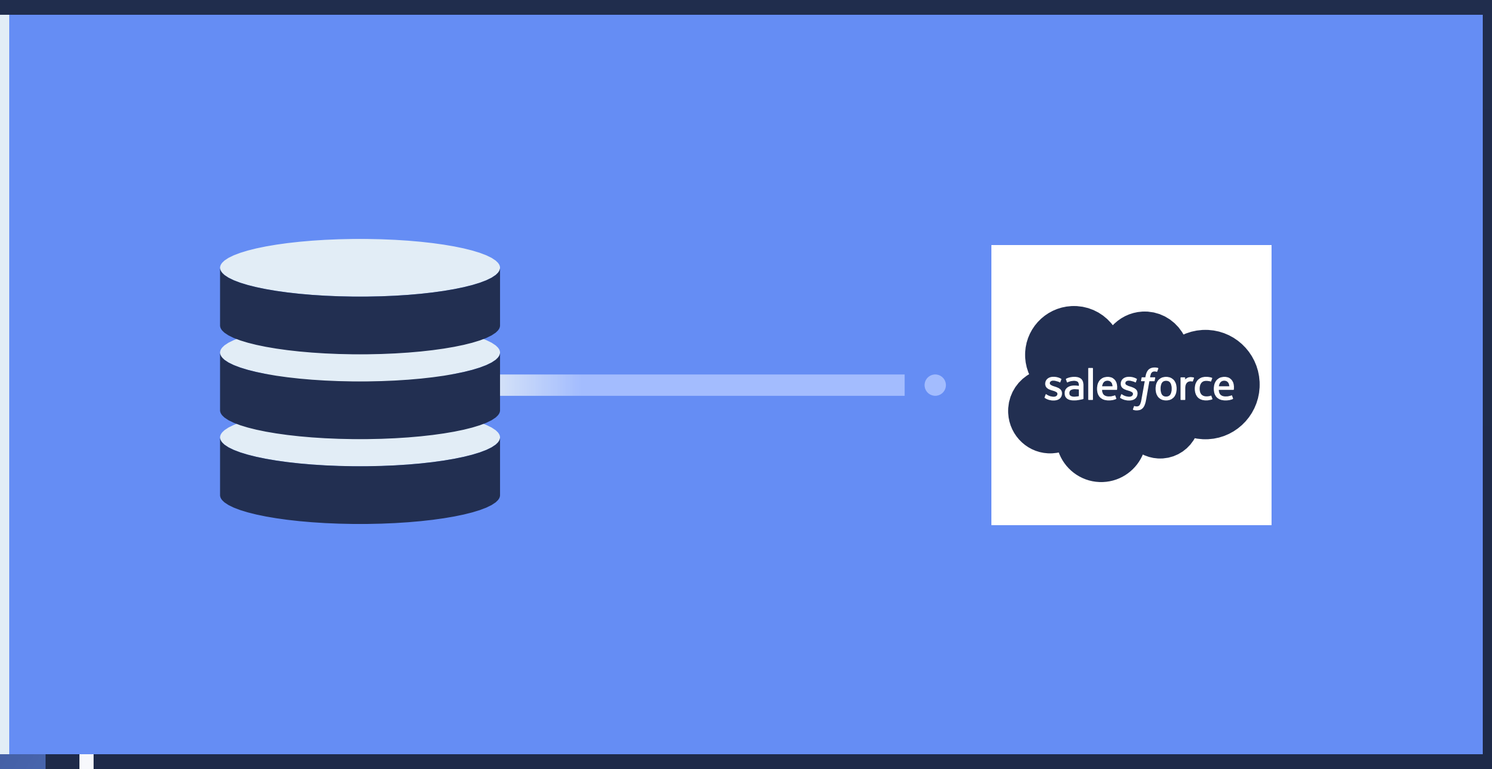
Hightouch is a fully managed Reverse ETL platform that makes it easy for data teams to query the data warehouse and sync data to over 125 destinations. Businesses can define their data using standard SQL or even leverage their existing warehouse tables or dbt models. There's even a no-code visual audience builder for non-technical users to create their own segments, as well as robust observability and debugging capabilities to help data teams identify when, where, and why syncs fail.

Hightouch also integrates directly with Git, which means engineering teams have bi-directional version control of their syncs. Hightouch eliminates all of the technical challenges of standing up bespoke Reverse ETL pipelines, so organizations can focus on activating data in their operational tools to drive business outcomes.

You can setup Hightouch in three simple steps

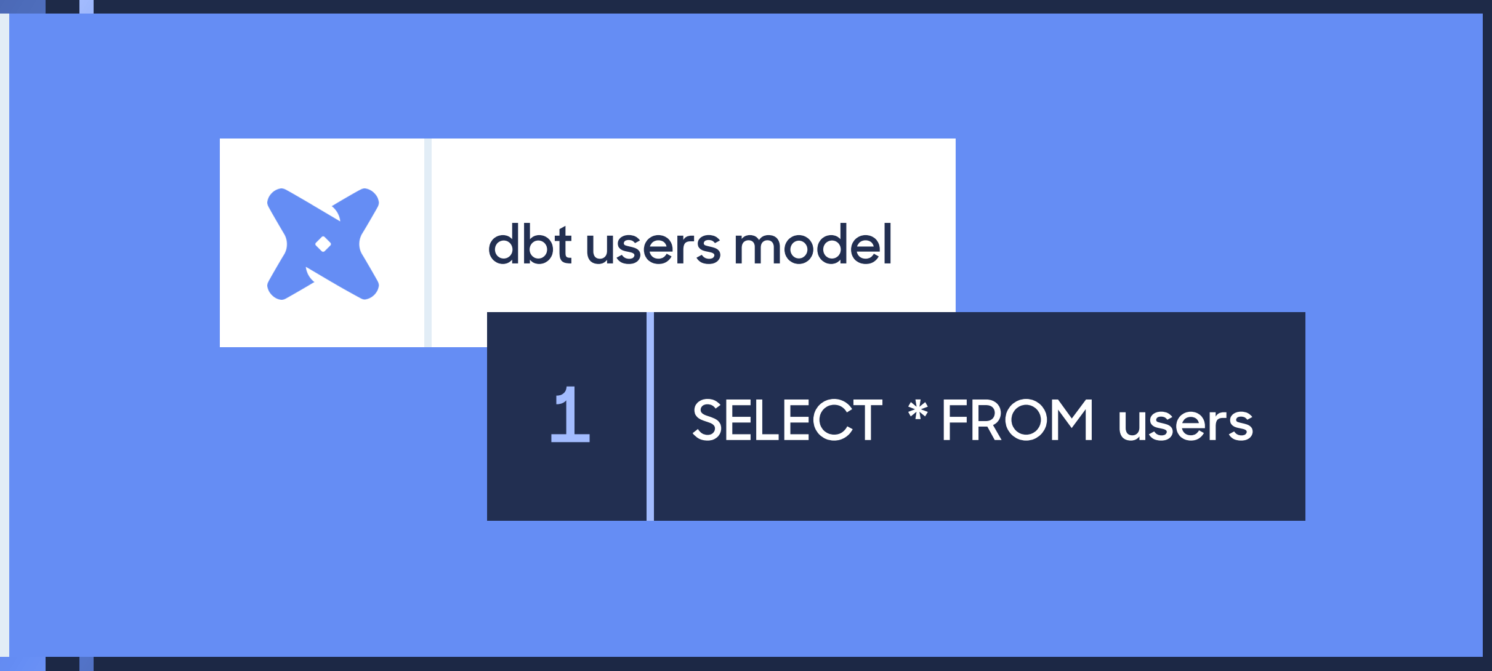
1 Add your source and destination

Connect data from 15+ sources, like Snowflake, to 125+ destinations, like Salesforce.



2 Define your model

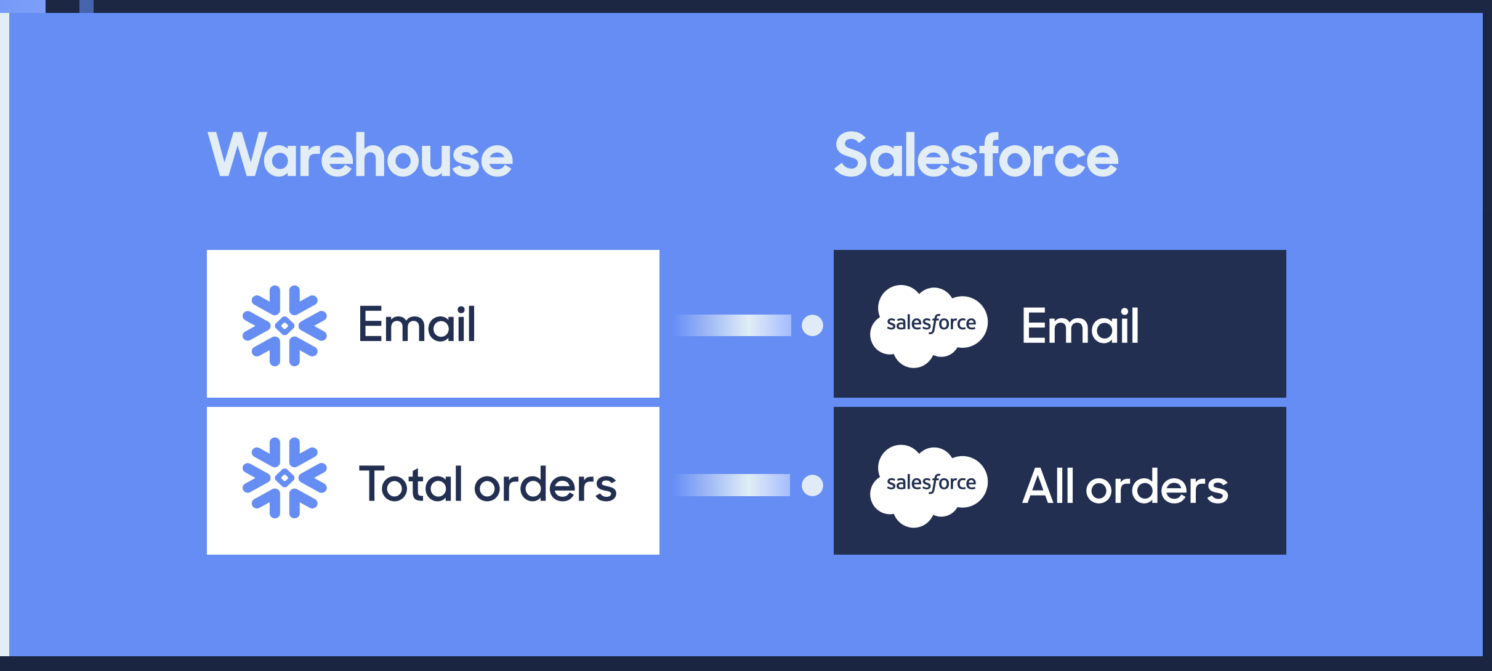
Use SQL or select an existing dbt or Looker model.



```
SELECT * FROM users
```

3 Activate your data

Define how fields from your model map to your destination.



Start Automating Your Data Integration Pipelines Today

Fivetran offers a 14-day free trial, and the first integration with Hightouch is completely free! Hightouch also offers a native integration with Fivetran, so you can trigger your Hightouch syncs to run immediately after a Fivetran job completes. With the freshest data possible flowing through your end-to-end system, there's absolutely no reason not to start automating your data pipelines today.