

# Data pipelines for your analytic application

How to connect data from your customers to your app



# Introduction

The year is 2022. Access to data is at an all-time high. The cost to store data is cheaper than ever. So is the computing power required to aggregate data and run analytical queries.

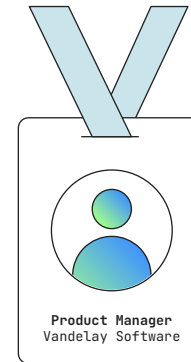
These economies of scale, coupled with time-saving software development frameworks, are giving rise to a new breed of applications that not only help users complete tasks and track progress, but provide predictions and recommendations to help them optimize their activity. These are **analytic applications**. And if you're not building one, you should be.

This book will introduce you to key concepts, challenges and recommendations for developing an analytic application of your own.

Get ready. You're the starring character in our story.

## Imagine this is you

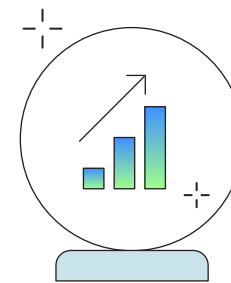
You are the proud product manager at Vandelay Software.



## Your company makes a magical tool that can predict the future.

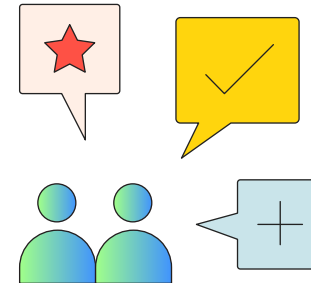
What can it predict?

- The return-on-investment (ROI) for a new advertising campaign
- Which products existing customers will purchase based on their past shopping behavior
- Which prospective customer segments should be targeted based on demographic data



## Everyone wants it

It's no surprise, your magical tool is in high demand. There's an appetite for the insights and predictions that it can provide.

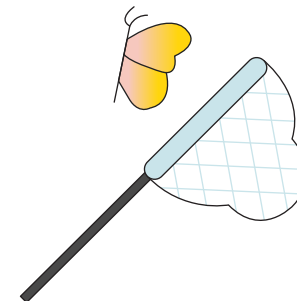


---

## But there's a catch.

As the old saying goes,

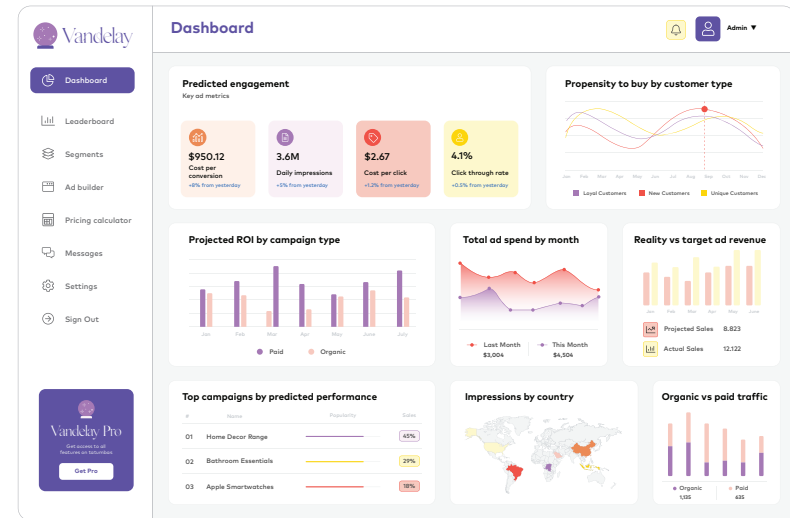
**"If it sounds too good to  
be true, it probably is".**





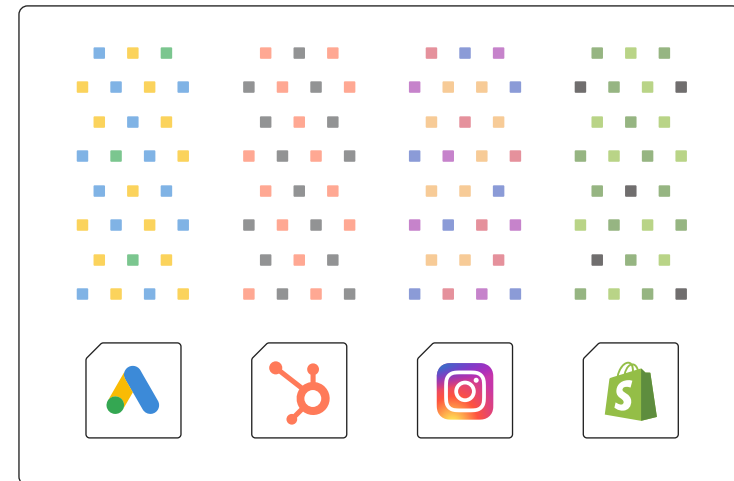
# To predict the future, Vandelay needs data.

Accurate predictions come from somewhere and they're not your horoscope. Data is the fuel to analytics and each bit of it strengthens your ability to predict the future.



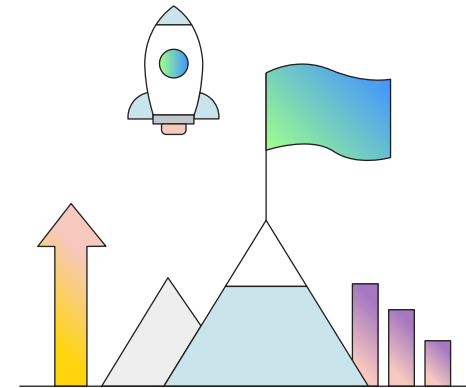
# And the data lives in many different places.

Just as you don't take a single perspective as truth, you shouldn't consider a single data source as truth either because each source inches you closer to a holistic understanding.



# Your team is motivated. You set out to get all the data.

You have a competent team of engineers. They have plenty of experience writing Python scripts to fetch data.

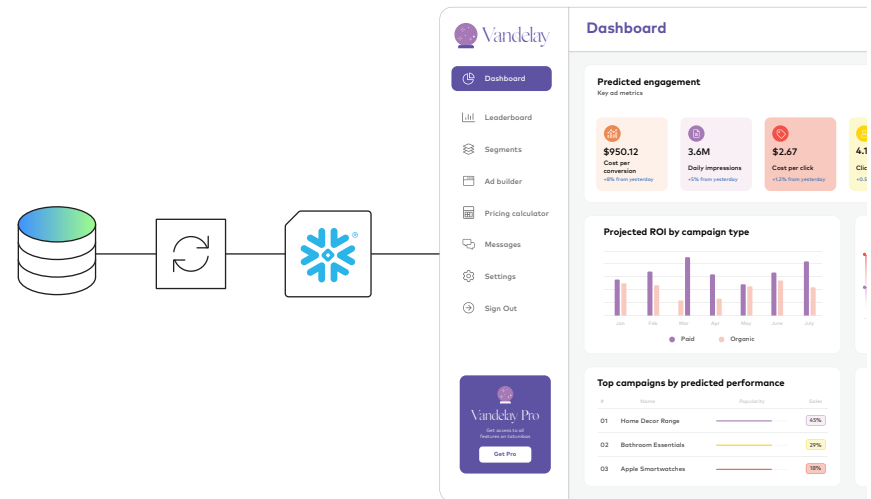


## You find that getting the data is boring work, but not that hard.

- 1. Study the API**  
Evaluate which data the source makes available through its API endpoints.
- 2. Create requirements doc**  
Determine which data is required to support the analytics you plan to provide.
- 3. Create engineering design doc**  
Specify how to extract data, in what frequency, and how errors will be handled.
- 4. Design schema**  
Determine how to format incoming data and the optimal relationship between data sets.
- 5. Do a proof of concept (POC)**  
Run your first full test of the data pipeline and document what happens.
- 6. Conduct bug bashing / QA process**  
Squash inevitable bugs identified during the POC in order to reduce future issues.
- 7. Release to General Availability (GA)**  
Make the new functionality available publicly through your product.

## Data begins flowing into your app, fueling your predictions.

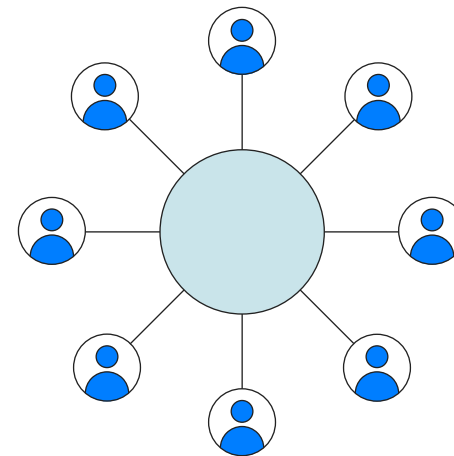
Using a script written by your engineer, data is extracted from the source and loaded into your destination.



## You quickly acquire customers.

New customers appear from out of nowhere begging to buy your product.

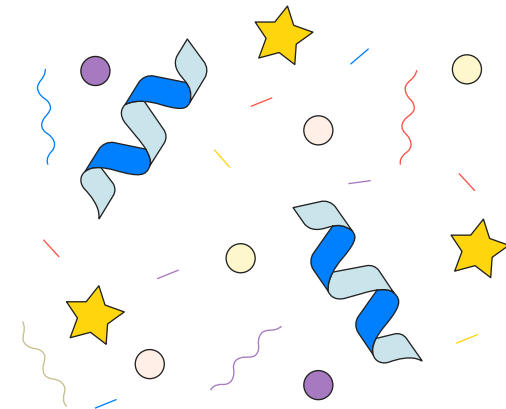
No one can resist magic.





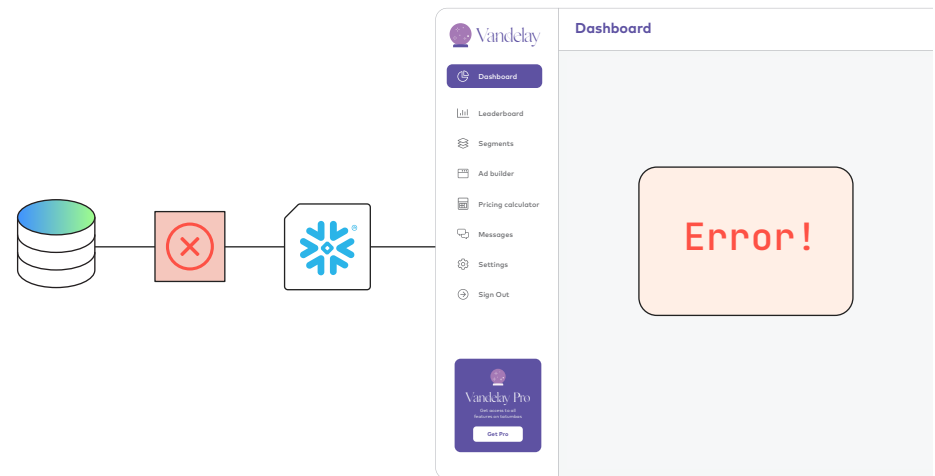
## You and your team are ecstatic.

Naturally, your team is filled with excitement. The time and tears spent laboring over your product were all worth it.



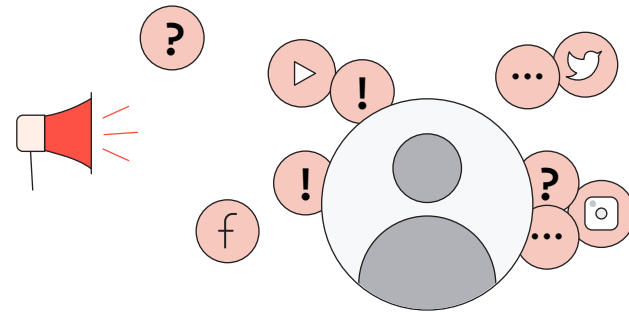
## But then, without notice, your product stops working.

The flow of data from source to your destination stops unexpectedly. Your predictions become stale, and your customers want to know what's going on.



## Your team spirals into a panic.

Finding a fix to the broken pipeline is the only thing that matters. Upset customers take to social media to vent their frustration over the outage. Some of your customers are more understanding than others.



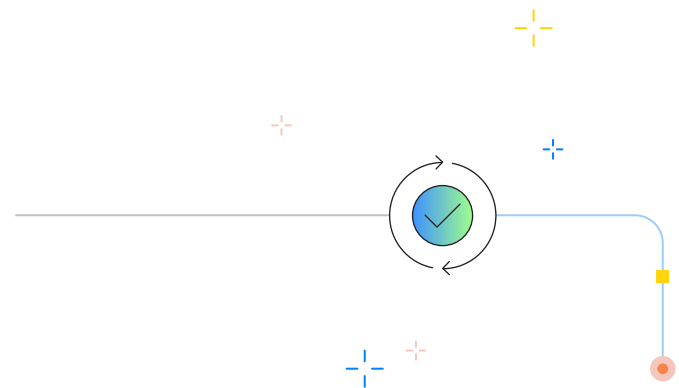
---

## Your team diagnoses the problem and fixes the issue.

It only takes one kink to break the pipeline, but yours was struck by several. Thankfully, because you have a smart team, it doesn't take them long to find out what's wrong.

Fixes are deployed, high fives are given, and business returns to normal.

The day has been saved.





## But then...it happens again.

Your customers unknowingly changed the schema of their data source causing the pipeline to fail once again.

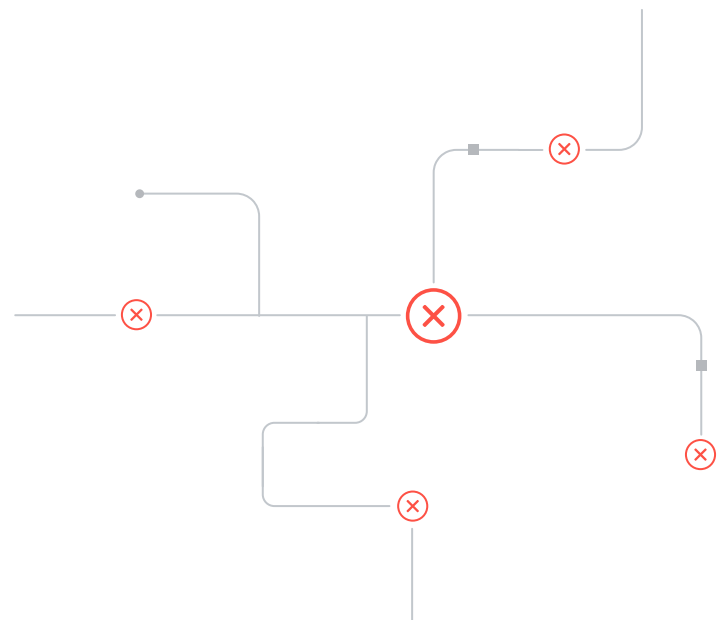
It's not their fault. They didn't know they were doing anything wrong.



---

## And again.

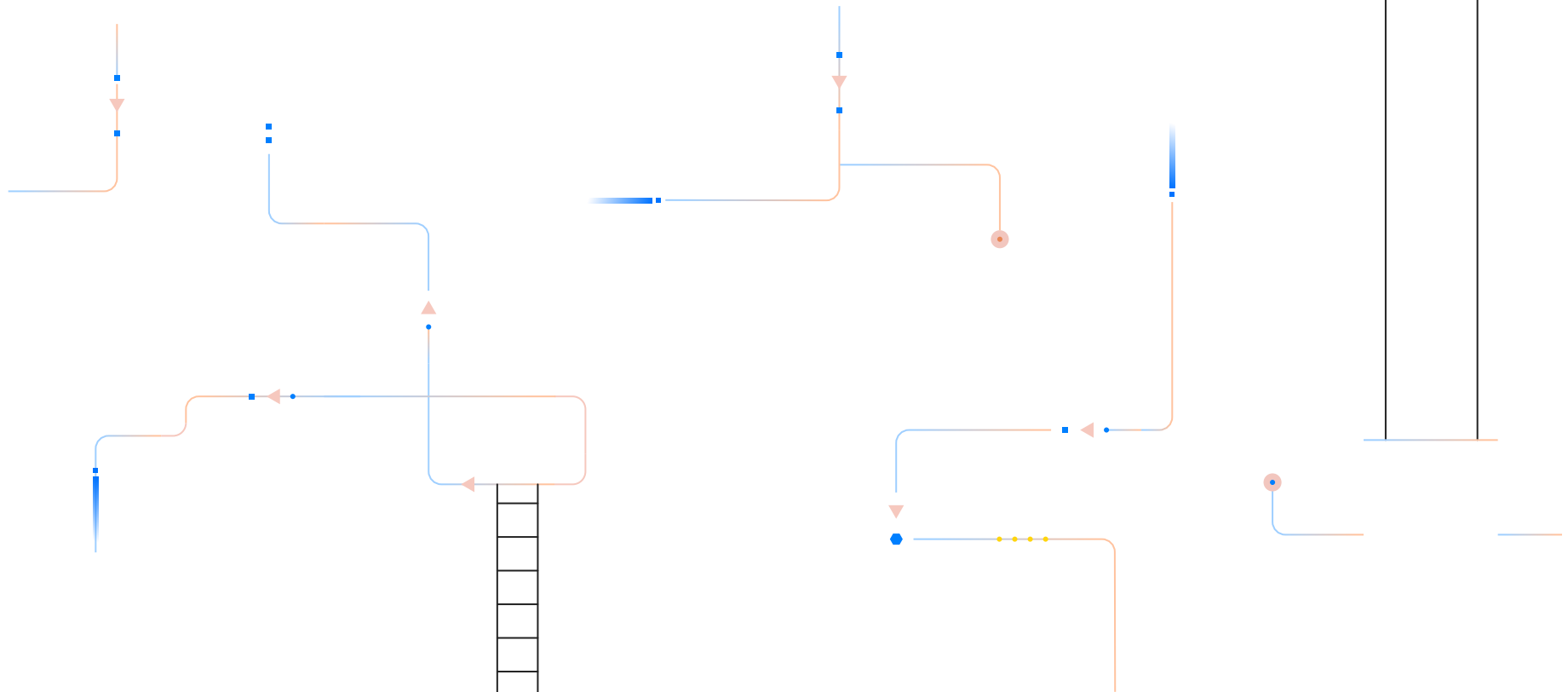
The data source provider added a new parameter to its API. This change hasn't been accounted for in your data extraction script and your sync job fails because of it.





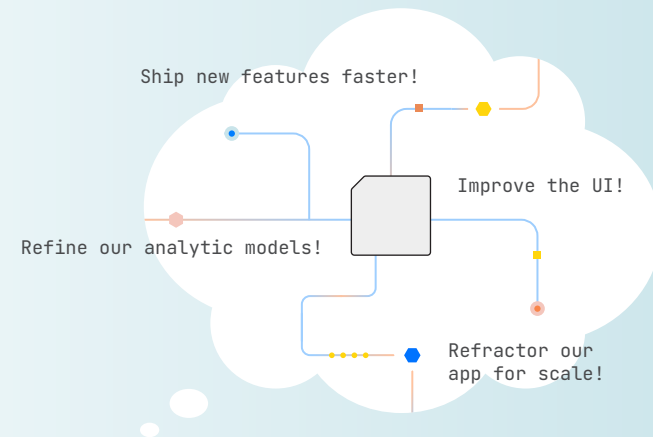
# You realize your team is spending a lot of time on data pipelines and not on features core to Vandelay.

Your application is evolving, your predictive models are becoming more magical, but you need to go faster. Data pipeline work is impacting your time to market. New features and bug fixes that you planned to deliver this quarter are pushed to next.



## You wonder what you could do if you never had to build or fix another data pipeline.

Daydreaming consumes your afternoons. You imagine data flowing into your app perfectly, every time. You finally have the time and focus to give Vandelay the love it needs. After all, you're a magic company, not a data pipeline company.







## For the first time, you consider outsourcing your data pipeline work.

Your team never wants to look at a data pipeline again. The work is impacting your team's morale and your ability to keep Vandelay competitive.

## Your team creates a wish list of required capabilities.

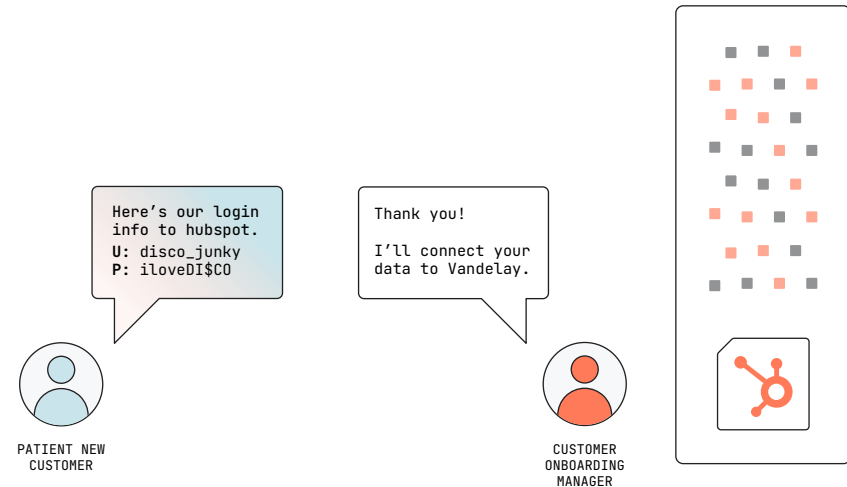
Scars from your recent pipeline work are still fresh. The team is weary, but their minds are sharp. The possibility of offloading work to someone else is a savory thought.

-  Support all our data sources
-  Auto-update to API changes
-  Schema drift handling
-  User self-authentication to sources



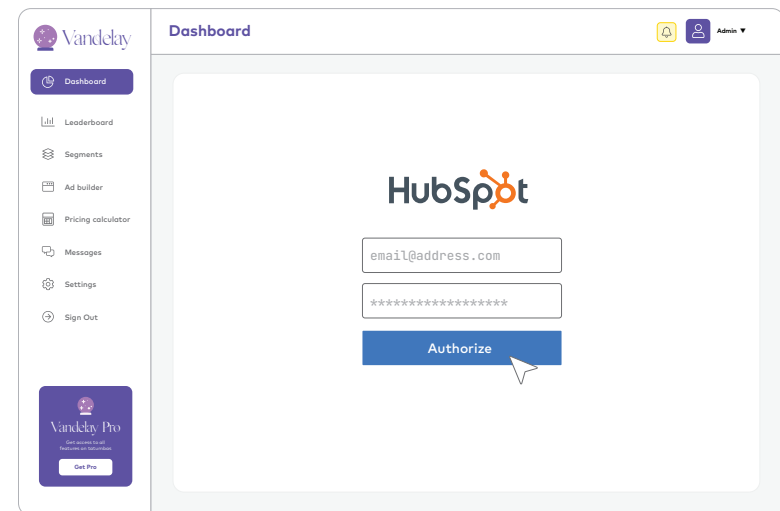
## Notably, you need a better way to access your customers' data.

Authenticating a connection to your customers' data sources has been a thorn in your side. It's a manual process and your customers aren't thrilled to be sharing their sensitive credentials with you either.



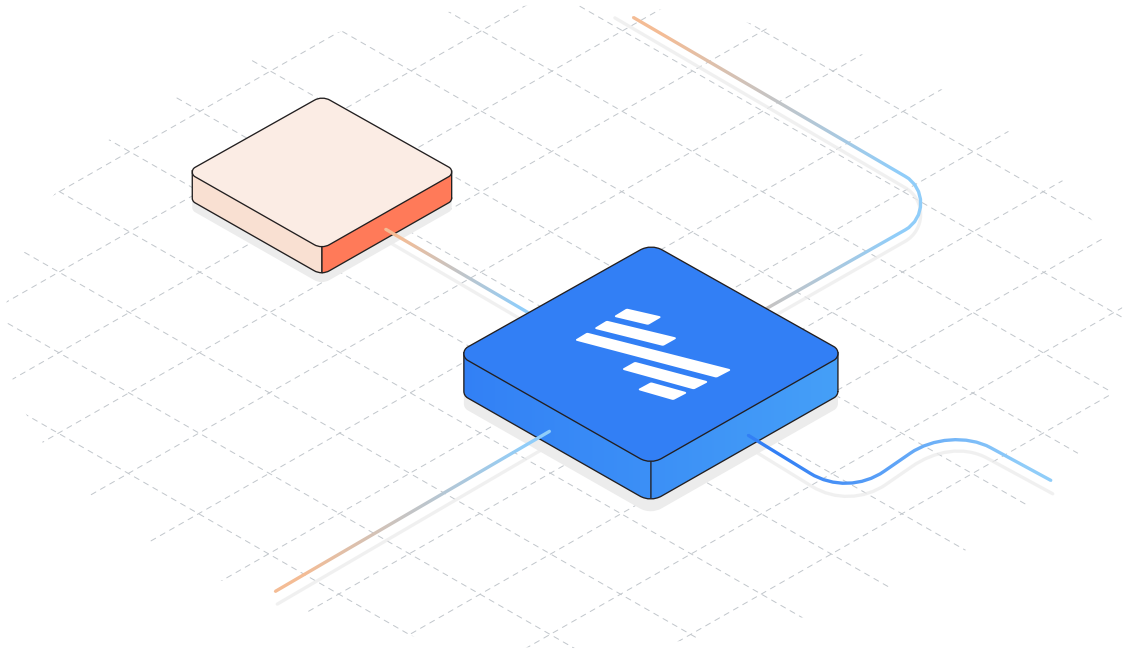
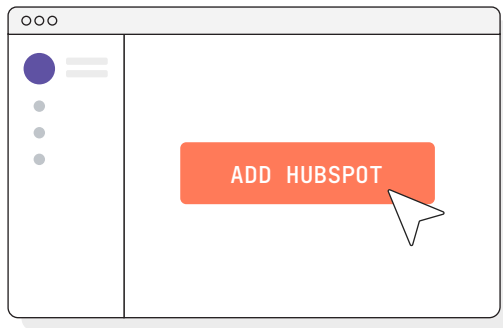
## You want to let customers connect their data to Vandelay themselves.

Customers should be able to authenticate a connection between their data sources and Vandelay without your help. Creating connections on their behalf is a friction point for onboarding new customers and running POCs with prospects.



## After a tumultuous search, you find what you need.

1. Data pipelines that self-heal, automatically updating to schema and API changes.
2. A pre-built authentication UI allowing users to create data pipelines, entirely on their own, from the Vandelay UI.

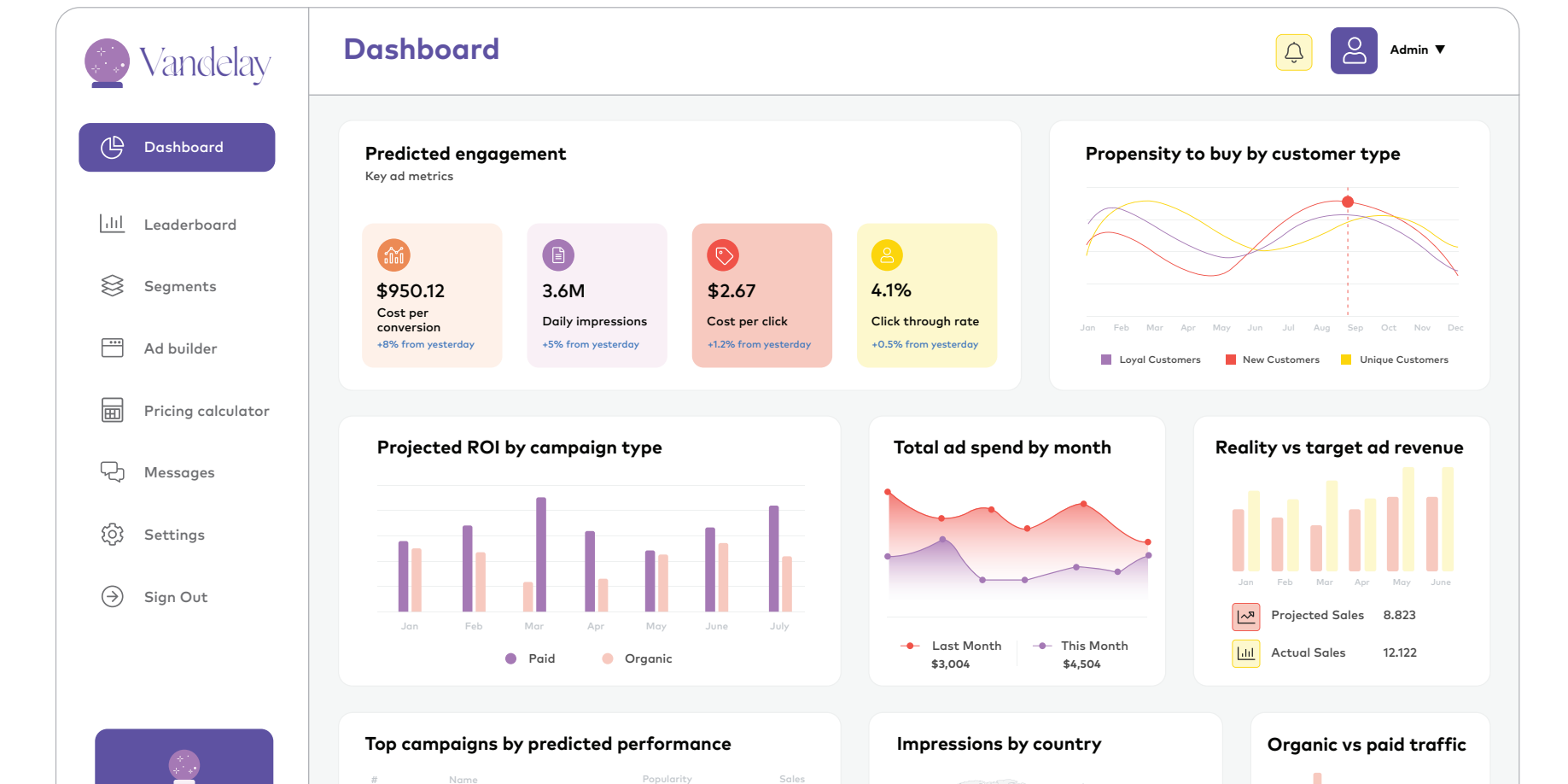




**Your customers are happy.  
Your team is happy.  
You are happy.**

With data pipeline work out of the equation, Vandelay gets the attention it needs. Building a magical tool is still hard work, but now a critical part of your data infrastructure is automated.

**Congratulations!** You have implemented data pipelines for your analytic app.





# Key considerations for choosing a data pipeline provider.

Consider the following evaluation criteria when selecting a data pipeline provider:

## 1. Data source support.

Does the provider offer integrations to the sources you need to extract data from?

## 2. Data connector quality.

Is data from API feeds normalized automatically? Are schemas illustrated through ERDs? Is data replicated incrementally or does it query full data sets every time it syncs?

## 3. Automation.

Do pipelines automatically adapt when changes are made to source schemas such as adding or removing columns, changing a data element's type, or adding new tables?

## 4. Configuration vs Zero-Touch.

Do you want to define and manage the connectors yourself? Or do you want to use standardized connectors managed and updated by the provider?

## 5. API integration.

Can your application interface with the tool programmatically or does it only have a GUI? Does the API support connector creation and authentication so your users can provision pipelines themselves from your app?

## 6. Specialization in data pipelines for SaaS.

Does the provider have a dedicated team that understands your use case? Or do they primarily focus on internal data integration?

## 7. Data transformation support.

Is data aggregated and transformed before or after it is loaded into the destination? Does the tool offer pre-built data models and integrate with tools like dbt?

## 8. Security and regulatory compliance.

Can sensitive data (e.g. PII) be obscured or omitted from every table you sync? Does the provider adhere to major compliance standards?

## 9. Pricing model and costs.

Is pricing based on volume of data or a flat subscription fee? If by volume, does the cost per unit decrease as your volume increases?



Designed to be embedded into web applications and analytics portals, Powered by Fivetran enables product and data teams to stop the low-value activity of building and maintaining data pipelines and focus on their secret sauce: surfacing valuable insights and building great product experiences.