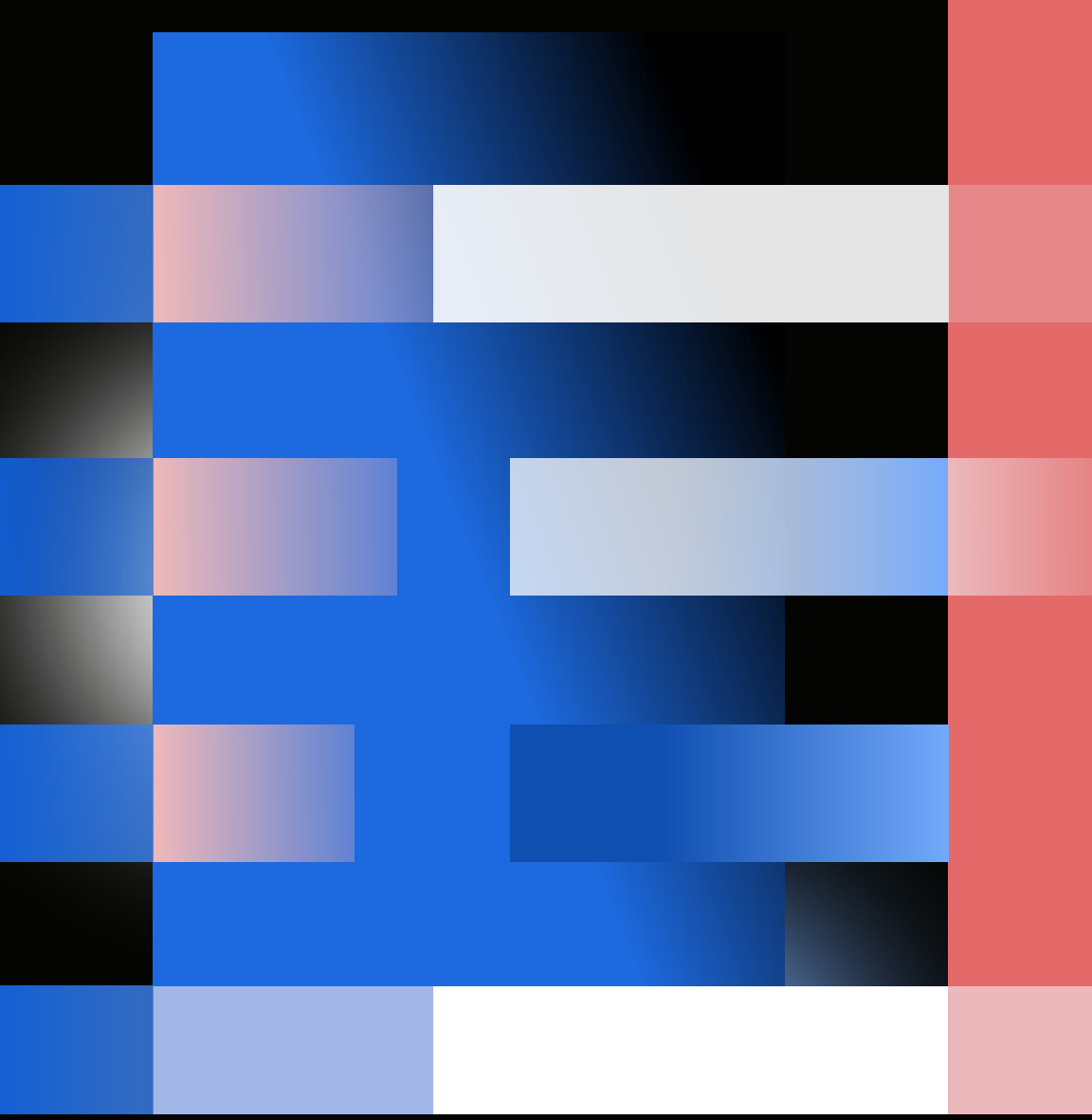


rudderstack

The Identity Resolution Playbook

Soumyadeb Mitra and Eric Dodds, RudderStack

Foreword by David Wells, Snowflake

The Identity Resolution Playbook

Soumyadeb Mitra
and Eric Dodds

Foreword by David Wells

Contributors: Eric Omwega
and Brooks Patterson

First Printing, 2023

Edited by Brooks Patterson

Cover design by Stephen Perkins

Layout and illustration by Suhani Harish

Typeset in Inter

ISBN 9798394821592

RudderStack

548 Market St #48141

San Francisco, CA 94104

www.rudderstack.com

Table of contents

Foreword

Identity resolution on the modern data cloud

David Wells

David Wells
Industry Principal, Media, Entertainment
& Advertising at Snowflake

In the decades since Tom Siebel coined the term "Customer 360" in the 1990s, companies have spent billions of dollars on efforts to build a complete customer view. In more recent years, tremendous advancements in data infrastructure have made this tantalizing prospect of a golden customer record more achievable than ever. Cloud computing made it easy to access and scale computing services as needed. The decoupling of storage and compute resources made storing large amounts of data and running queries against it significantly more cost-effective. Moreover, the advent of the Snowflake Data Cloud brought these benefits to market as a service with no overhead and minimal upfront commitment. With this in mind, this resource on Identity Resolution, a foundational concept in realizing a complete customer view, is required reading for today's data leader.

What exactly is Customer 360, and why is it so vital that companies are willing to allocate tens of millions of dollars to build it? Put simply, Customer 360, or the golden record, is the unification of all customer data spanning all customer touchpoints across the organization. This data could include behavioral data from your website or app, customer records from your CRM system, transaction data from your payments system, customer support data from call centers, and data from external sources.

Companies are willing to allocate large budgets to build this golden record because of the transformative potential of realizing a deeper understanding of their customers and providing tailored experiences to achieve a desired business outcome such as higher customer lifetime value or reduced customer churn. As highlighted in compelling research by McKinsey & Company, personalization can drive a 10-15% revenue lift (gains could be

as high as 25% depending on the sector and ability to execute). More importantly, customers value personalization because it enriches their experience and engenders a greater sense of control, while concurrently reducing information overload. The combined impact of these factors results in customers feeling more valued, further stimulating brand loyalty.

The effort to create this golden record has been stymied by an eclectic variety of factors across the entire lifecycle of data – data collection, data unification, and data activation.

Collection of first-party data is a solved but non-trivial challenge due to the myriad of data sources and associated tooling involved in ingesting streaming and batch data. Products such as RudderStack provide out-of-the-box features to enable companies to collect data from their owned websites and apps as well as third-party SaaS tools.

Assuming you have collected high-quality first-party data you can trust, the next step is the unification of the data around a unique identifier – this is where Identity Resolution comes in. Identity Resolution involves stitching together the entire customer journey across streaming and batch data sources in order to obtain a complete customer view. It is worth noting that in many cases, companies go beyond first-party, deterministic Identity Resolution and use third-party data for targeted paid media advertising.

Finally, creating value from the unified customer record requires activation. Activation use cases could include sending a personalized email or featuring a product that your customer is likely to purchase based on browsing behavior and prior transactions. These use cases, built on high-quality data unified around the customer, have the potential to drive substantial revenue growth.

We built the Data Cloud to provide a step-change improvement in enabling governed access to near-infinite amounts of data and cutting-edge tools, applications, and services. These capabilities make it unique in unlocking the ability for companies to build a complete customer view. More specifically, Snowflake's

ability to ingest all types of data (structured, semi-structured, and unstructured) breaks down data silos. Its ability to elastically scale compute resources ensures identity resolution is achieved comprehensively and cost-efficiently. Ultimately, the native and connected apps built on top of Snowflake, such as RudderStack, enable companies to finally achieve the coveted 360-degree view of the customer to deliver a better customer experience, increased revenue, and higher customer satisfaction.

David Wells

Introduction

Data leaders live in a world flush with both pressure and opportunity. In the current macroeconomic environment, the mandate for data teams to deliver business results is at an all time high.

At the same time, advanced tooling has created new possibilities for data teams. Collecting, storing, processing, and modeling incredible amounts of data has never been easier or cheaper.

Some companies are riding this wave of technology to drive revenue, but many data leaders still struggle to deliver value quickly. This makes justifying the cost of purchased technology more difficult, and it often means data leaders have engineering resources stuck doing low-value plumbing work in their effort to integrate an ever growing array of data and tools.

Nowhere is this struggle more prevalent than in the effort of data teams to build the coveted "complete customer view." It's not uncommon to hear about these projects taking multiple quarters – even years – and costing tens of millions of dollars. Here are examples from real data teams:

- A large North American retailer has budgeted over $40 million annually to solve identity resolution
- A global telecom company is spending $30 million to build a customer journey map
- A large marketplace spent three years building a complete, usable view of their customer

Why is this?

The answer is found in the story of customer data over the last decade, from the rise (and fall) of marketing technology to the fragmentation and consolidation of modern data infrastructure.

This playbook is written as a guide for data leaders and their teams. We start by providing historical context for the problem. Then we dive into practical guidance on how to collect first-party customer data, unify it to create a full view of the customer, and activate it to deliver bottom-line impact.

Chapter 1

A brief history of identity resolution

Companies have used customer data since the beginning of commerce. With the introduction of relational databases in the 1970s and 80s, companies gained the ability to store and query relatively large amounts of data. In the on-prem era, businesses digitized transaction ledgers and used the data for use cases strikingly similar to those we associate with modern digital marketing:

- Customer loyalty programs
- Personalized direct-mail marketing campaigns
- Product placement optimization

Systems from Oracle, IBM, and HP, paired with enterprise resource planning (ERP) products like SAP, enabled companies to build significant competitive advantages with their customer data. Market leaders used data to solve hard problems like tying marketing campaign data to purchases and understanding why customers churned.

In this age of early enterprise infrastructure, just about everything related to data was under the IT team's purview. They were the heroes who built and maintained the systems that enabled companies to realize the power of putting customer data into action. As such, businesses rightly trusted IT with complete power and control over the technology and, in most cases, the data itself.

But data needs and expectations were different then. It was enough for teams to access a limited set of data through the ERP or an early BI tool like Business Objects. Data was the sole purview of the IT team. Access to the raw data outside of IT was novel. In addition, IT was treated as a service center and wasn't forced to collaborate with other business units to take advantage of front-line opportunities. Because of the comparatively simple data landscape, data silos were not an issue because most companies purchased a database, then applications to run on top of that database. Customer data lived in the database, IT

could provide teams with the data they needed, and everyone was happy.

Identity resolution was not a major concern in this era because the data sources were limited primarily to transaction data. It was before the arrival of the digital customer journey as we know it today, and IT teams controlled customer data collection. But this all changed when the internet achieved critical mass distribution.

The web goes worldwide

At the turn of the century, businesses were pushing into a new, online digital frontier. Once again, IT teams led the effort. But things were different this time.

Businesses had tasted the potential of data in the on-prem era, and they were beginning to understand its power. When technological innovations pushed commerce online, data appetites increased exponentially.

Meanwhile, the advent of a rich online customer experience during Web 2.0 pushed a new type of data to center stage: behavioral data. With digital customer interactions, it was now possible to stitch together a customer journey by tracking all sorts of granular data points from product searches to clicks to browsing behavior. Companies quickly recognized behavioral data as the key to understanding and reaching customers in this new era. But comprehensive tracking was expensive, complex to implement, and engineering teams lacked modern tooling to make implementation easy, so everything had to be custom-built.

The smartphone revolution, catalyzed by the launch of the iPhone in 2007, made the tracking landscape even more complex. The same user could now access a company's website on a desktop, laptop, or mobile device. Moreover, they might access both the company's website and mobile application, further complicating the picture.

Identity resolution timeline

2004

Web 2.0

Internet adoption accelerates, websites become more interactive

2007

iPhone hits the market

Smartphone and mobile revolution begins

2014

SaaS explosion accelerates

Widespread adoption of SaaS tools including marketing CDPs

2016

Widespread warehouse adoption

Cloud data warehouses become a key piece of the data stack

2019

Warehouse as the source of truth

Companies focus on centralizing customer data in the warehouse

2021

Warehouse Native CDP

A new architecture emerges to enable the collection, unification, and activation of customer data

A wealth of opportunity lay on the other side of this complexity – companies that could manage to collect online and offline customer data and unify records for individuals in a single profile would have a major competitive advantage over those that could not. But the teams best suited to solve this identity resolution challenge, the IT teams, were consumed with other massive

> **64% of teams today cite data centralization and unification as a top priority**
>
> 2023 State of Data Engineering Survey

SaaS takes over

With so much opportunity on the table, standing still wasn't an option. Companies couldn't just wait around for IT to help them harness data in this new era. In a global upswing of innovation, the choice was either to make a way or get left behind. The software industry recognized this dilemma as an opportunity.

If IT teams couldn't keep up, agile startups could. Fueled by venture capital and advancements in cloud technology, SaaS solutions proliferated to fill the gap between what was possible and what was attainable.

For companies under pressure to keep up with the rapid advancements in the new digital customer journey, SaaS was a golden ticket to operating on the cutting edge. No longer did business unit leaders have to wait for IT to design, provision, and implement solutions, they could simply pick a SaaS tool and get up and running in a matter of weeks.

The benefits of this SaaS-ification were undeniable. SaaS products allowed teams to maintain, and often accelerate, progress. Plus they were cheaper than developing in-house, and they were typically more stable. But as with all major advances in technology, the SaaS-ification of customer data tooling involved trade offs.

Damn the data silos, full speed ahead

In the pre-SaaS days, just about all of a company's customer data lived in its internal database. Developing analytics and applications using multiple sets of data wasn't a problem then because the data was all inside of the same system.

This simple landscape with centralized data is what businesses traded for velocity when they embraced SaaS. The SaaS explosion created a serious data problem that every company struggles with to this day: data silos.

> **74% of teams say data silos impact their ability to ship key projects**
>
> 2023 State of Data Engineering Survey

Every individual SaaS tool locked data in its own ecosystem creating a data silo. Moving data into or out of those systems was difficult at best because they were built to solve for specific business use cases, not data integration. This data silo problem further complicated the identity resolution equation. With valuable customer data scattered across various SaaS products, it was impossible to build anything close to a comprehensive customer profile. The cure was worse than the original problem.

This new, fragmented landscape created pain for teams company wide, but none felt it more acutely than marketing. They needed all of the data and a complete view of the customer, but they were using multiple tools that didn't talk to each other. The market responded, and yet another SaaS category emerged promising to solve all of their problems: the Customer Data Platform (CDP).

The CDP promised to automatically bring in data from every SaaS tool, unify it in a single location, and create a comprehensive customer profile. It would also provide tooling on top for the tactical activation of data across various channels. But this solution ignored a critical fact: IT still controlled key business

systems like web and mobile experiences, eCommerce applications, backend systems, and the databases that powered them.

Plus, these legacy CDPs were no different than other SaaS tools – even though they collected data from websites, apps, and other systems, they still created another data silo. There was no easy way for teams outside of marketing to access the valuable data stored inside the SaaS CDP's black box.

The pain of a fragmented system persisted. But pain bears innovation. IT teams were beginning to adopt cloud data warehouses which were quickly overcoming the technical limitations of databases from decades past.

Warehouse native customer data architecture

Warehouse technology advancements meant the limitations of traditional CDPs could be overcome, but doing so would require a new approach: building the entire customer data stack around the warehouse instead of relying on a marketing CDP. Ironically enough, this approach required a return to reliance upon technical experts.

Teams were learning, through dealing with the consequences of SaaS-ification, that going around IT wasn't a good idea. IT teams hadn't been idle during the SaaS boom either. They had been busy re-inventing every piece of tech in the organization, leveraging both SaaS and the cloud. IT teams themselves had evolved. Many companies, recognizing the importance of data infrastructure, formalized engineering roles for the people who built and managed that infrastructure, creating data teams and the role of data engineering.

Armed with new technologies and the necessary skill set, data engineers could provision infinitely scalable infrastructure from cloud providers and work with developers to centralize both relational and behavioral customer data in a modern data platform like Snowflake.

In theory, this new architecture solved the problem of identity resolution that became acute in the age of data silos. If the data warehouse serves as a central repository for every customer data point, behavioral data from every device, website, and application can live alongside relational data from every SaaS tool, and the data can be unified to create truly comprehensive customer profiles.

But as we've learned, things are never as simple as they seem. There are several unique and difficult technical challenges to building a complete, usable view of the customer, even with modern technology.

Chapter 2

Why it's hard to build a complete view of your customer

Solving for identity resolution requires two primary ingredients: data collection and data unification.

Any attempt to construct a truly complete view of your customer requires using all of the available data about that customer. For data teams, this means ingesting data from a variety of sources to capture every customer behavior and attribute.

After collection, data must be unified to produce a clean, usable view of the customer. This requires cleaning, joining, and modeling on multiple levels – first building a basic identity graph, then computing user attributes based on behavior.

When legacy CDPs failed to deliver on their promise of a complete customer view, companies were reminded that building a complete view of the customer is a fundamentally technical problem, and responsibility for these projects began to shift back to data and engineering teams, a trend we are currently seeing play out across the market.

Data teams and modern data infrastructure are perfectly suited to solve this problem, but collection and unification still pose significant challenges for data leaders. Most teams are unable to focus on making their customer data useful because they're mired in the complexity of building pipelines and endless data modeling.

I (Soumyadeb) faced this problem as a data leader at 8×8, a provider of Voice over IP products. My team was tasked with building a lead scoring model, but we spent so much time collecting and cleaning data that we weren't able to ship a project that truly impacted the bottom line.

Unfortunately, my experience is all too common among data leaders.

> **Data leaders cite "collection & centralization" as the number one roadblock to solving identity resolution.**
>
> 2023 State of Data Engineering Survey

The data collection challenge

Customer data is unique and presents data teams with multiple challenges when it comes to building complete customer profiles:

- Customer data is naturally siloed
- Customer data is big and messy
- Customer data requires complex infrastructure

Customer data is naturally siloed

Front line teams use customer data every day, and they typically store the subset of data required for their use cases in their own tools, creating data silos. This problem is further exacerbated for businesses that produce both online and offline data. For example, retailers with eCommerce sites plus brick and mortar locations face the challenge of collecting and integrating point-of-sale data.

These natural silos create a significant collection problem for data teams who often have to deal with hundreds of different data sources. They come from marketing tools, CRMs, customer success platforms, payment systems, internal databases, and other systems. When you add first party, event-based behavioral data from multiple apps and websites to the picture, it gets complicated quickly.

> **Only 26% of data teams say that they have solved the data silo problem.**
>
> 2023 State of Data Engineering Survey

Customer data is big and messy

Customer data comes in big quantities. Modern eCommerce and marketplaces, for example, are driven by online interactions and generate a massive amount of data every day. Moving data at these volumes has implications for cost, scale, and scheduling.

This complexity and cost only increases when you add more SaaS platforms to the equation.

Customer data also changes often. Not only do user behaviors and attributes change, but new features in apps and websites create new kinds of behavior to track. Even data points that seem simple, like purchase price, can become complex when you layer in coupons, discounts, and margins.

Customer data also comes in many different formats, from structured (a CRM record) to completely unstructured (a chat transcript).

The inherent messiness of customer data means that every sequential step of the data lifecycle, starting with collection and storage, is more complicated than the last.

Customer data requires complex infrastructure

At scale, even basic pipelines can require orchestration across multiple tools like dbt, Airflow, and ETL jobs. This necessitates a monitoring layer to ensure continued operation through the lifecycle of the data flow and compute process.

Running such infrastructure at scale requires full time data engineering, and it quickly becomes a low-ROI use of resources because a majority of time is spent on plumbing not shipping projects.

The data unification challenge

Collecting every customer touchpoint is a significant undertaking, but it turns out that solving the initial problem of centralization is only half the battle. Modeling the data into complete, usable profiles presents multiple challenges:

- User identities
- Event semantics
- Metadata

User identities

The typical user journey begins with anonymous activity on a website or app before the user eventually signs up, logs in, or makes a purchase. This means each user generates data when they are anonymous, and then generates additional data after becoming known. In order to accurately construct a user journey, all of these activities must be logically combined into a single timeline of events.

Semantic user features like “number of products viewed before first purchase” must be computed on the logically combined user journey, not the raw events.

Managing these anonymous and known identifiers, then combining the raw data, is a massive undertaking.

Things get even more complex in multi-device scenarios where the same end user can be associated with multiple anonymous identifiers such as a cookie ID on a browser and a device ID on a mobile device. Multi-user household scenarios complicate the picture further when multiple users with different identifiers and devices are all associated with the same buying group.

Worse still, associations between unique identifiers are often discovered over time, meaning that stitching user identities requires maintaining an identity graph and computing transitive closure — a non-trivial task in SQL. Add in non-event data, and the modeling becomes even more complex... all for a basic deterministic identity graph.

Event semantics

Data points with timestamps (events) are fundamental to building a complete customer view, but user traits and features in the customer profiles table rarely map 1-1 to events.

For example, a feature like “user lifetime revenue” would require summing transactions from the website and mobile app, as well as any subscription revenue, and reconciling with financial transactions from the payment system. Even this simple use case

requires working with multiple events from four different data sources and performing multiple mathematical operations.

Semantic features and events also need to take into account the output of the identity stitching step mentioned above: the raw events need to be associated with the anonymous and known identifiers, while features need to be computed over all events across all IDs belonging to a user.

Accomplishing this requires a significant amount of complex, repetitive SQL joins and unions across multiple events and identities.

Metadata

Once you compute semantic features, you need to keep track of important metadata related to those features. At a high level, you need a description of the metric (what it means), time of last update, provenance (e.g., who defined and built the metric), and any access/ownership requirements.

Tracking historic versions of the metrics is also important, especially for ML algorithms. For example, a churn algorithm may model off of features like revenue and website activity in a 7-day period prior to the churn date. If the definition of revenue changes, the historic version of the metric must be marked as deprecated and the new metric recomputed.

Building out the models and pipelines required to manage event semantics and metadata based on an identity graph (that you are also managing) is an extremely complex undertaking. It's no wonder that data teams struggle mightily to scale the wall of identity resolution when they set out to build complete profiles.

Where there are challenges, there is opportunity

The key to any scalable solution is first understanding the underlying problems. While the challenges mentioned above often plague modern data teams, overcoming them can be a career-making achievement for data leaders.

In the next chapter, we walk through the playbook that modern data leaders and their teams use to ship complete customer profiles quickly.

Chapter 3

The identity resolution playbook

The promised land

Data teams are uniquely positioned to build moats for their companies because their work can help drive competitive advantage across every business function. Still, while the potential is greater than ever, most teams are stuck doing low-level work in support of closing the next ticket in an endless data breadline. Far from delivering transformative change, these teams struggle to believe their companies have a positive ROI in data.

> **52% of data teams feel their ROI is either negative or neutral**
> 2023 State of Data Engineering Survey

The prospect of collecting comprehensive data and unifying it into complete customer profiles that the whole company uses to drive results seems like a pipedream for these teams.

This isn't surprising, given the magnitude and complexity of the challenge. However, there is proof that not only is the promised land within reach, but making the jump from focusing on low-level problems to confidently driving value doesn't have to be such a long-winded, herculean effort.

The vision

For a data team to drive, impact, they must be free from tedius, low value work in order to tackle projects that directly impact the business.

A thriving data team efficiently delivers accurate, reliable reporting, consistently shares novel insights, and regularly generates valuable data products and predictions to push their companies forward. They work with teams across the company to fuel innovative capabilities that drive transformative impact.

Instead of constantly fighting entropy and struggling to maintain brittle data stacks, they utilize a practical and effective toolset

that feels enabling, not limiting. Because they've leaned on technology to automate, or at least streamline, low-level work, they can dedicate most of their energy to work on interesting projects that move the needle.

Plus, because they've established a tight feedback loop between data and impact, the rest of the business no longer looks at them as order takers – they're strategic partners who can ideate, plan, and provide a valuable perspective.

Making it a reality

The data team at Joybird, a La-Z-Boy eCommerce company, used to spend all of their time building brittle custom integrations and managing data pipelines. It took them weeks to respond to requests from their marketing team because they had to pass requirements back and forth multiple times and figure out how to triage a gigantic backlog of other stakeholder requests.

> Adding a new dimension in our email platform used to take two to three weeks. With RudderStack, we have shortened that to an hour.
>
> Brett Trani, Director of Analytics at Joybird

While they had made efforts to centralize data in their data warehouse, the data was unreliable, so they could not confidently use it for activation. Cleaning data made turnaround times even longer. Ultimately, the team was stuck doing mostly menial work and producing uninspiring results. Luckily, this is a far cry from where they are today.

Joybird's team now uses RudderStack to fuel an event-driven architecture that gives them full control of their customer data and unlocks the power of downstream business tools in their stack.

Moving beyond simply storing customer data, they transformed their Snowflake Data Cloud into a single source of truth with a modeling layer that drives consistent definitions throughout

their entire stack. Because RudderStack now handles the data layer for them, their team can focus on driving impact instead of managing brittle integrations.

Today, the data team is able to capture every customer touchpoint, centralize, model, and enrich the data in Snowflake, then activate it across their entire stack. Engineers spend less time doing low-level work, and downstream teams can ship innovative use cases faster.

> We didn't realize the power of RudderStack either in terms of saving engineering time or allowing our downstream teams to work and iterate faster. Our CTO is very excited about using RudderStack to realize our event-driven architecture vision.
>
> Brett Trani, Director of Analytics at Joybird

The very first step in Joybird's transformation, though, was to get complete, trustworthy data into Snowflake.

Step 1: Collecting the data

Collecting every customer touchpoint in order to build a complete customer profile makes sense, but in addition to being complete, that data also needs to be trustworthy. The old adage "garbage in, garbage out" is painfully true for data teams.

Here are best practices on how to collect clean, usable data without the headache of low-level engineering work.

Buy, don't build

Data leaders today should prioritize buying pipelines for data collection instead of building them in-house. Building pipelines is time consuming and resource intensive. It also inevitably forces teams to focus on maintenance and scale, not data quality, which compromises the entire effort of building complete customer profiles.

Equally as important, though, is avoiding the hidden cost of building pipeline and collection features that vendors offer out

of the box. A prime example is schemas. Schema management, especially when it comes to handling new or changing data, is complex. Tools like RudderStack automate schema creation and management. Automation guarantees graceful handling of updates because it syncs changes with the schema in the data warehouse using the same rules every time.

> You have to have your team focus on what is truly a value add for your customer or your business, and doing that requires not building something that you can buy off-the-shelf that gives you 95% fit in terms of what you want.
>
> Nari Sitaraman, CTO, BARK

Pipeline playbook: Everything you need for modern data collection

Modern data teams have the benefit of modern tooling, especially when it comes to data pipelines. The needs of every business vary, especially by industry, but the following set of data sources is typical of an enterprise organization collecting customer data.

First-party user behavior data and pipelines

User behavior data includes all of the actions taken by users on all platforms. Behavioral data is commonly referred to as "event data" or "clickstream data."

To collect this data, you should select a pipeline vendor with out-of-the-box SDKs that you can install in each of your digital properties and that can send user events to integrations across your stack, including your data warehouse, data lake, and business tools. The SDKs should also handle anonymous user tracking, cookies, consent, and schema standardization. It's also important to note any cloud tools, like email marketing platforms, that generate customer data in the form of events.

First-party user behavior data sources

- Website events — events generated by users accessing your site on a web browser
- Mobile events — events generated by users accessing your mobile application(s)
- Server-side events — events generated by your server-side code that represent user actions (e.g., user invited)
- Events from business tools — events generated by users as they interact with a business tool (e.g., email open and click events, SMS)

You can also collect events in the form of database logs (often called CDC, or change data capture). Logs are useful, but using them as a proxy for user events typically requires additional data cleanup and schema management work — hours that could be spent on higher value projects.

First-party batch data and pipelines

Batch pipelines for customer data are often seen as traditional ETL: extracting structured data from a cloud tool, then loading it into the centralized data store. These business tools store customer attributes unique to their particular business function (i.e. marketing and customer success), which are required in order to build a complete customer view. When selecting a vendor for batch data pipelines, ensure that they have robust scheduling and transformation features (if you need them), as well as strong scheduling capabilities.

First-party batch data sources

- CRM & sales — the primary tool used by operational teams to access and act on customer records

- Marketing automation, email, & messaging — the tools used by marketing teams to build customer experiences and send messages
- Customer success & support — the tools used by customer-facing teams for communication, ticketing, and collaboration
- Payments and transactions — events, entries, and subscription data from third-party and internal payments and financial systems
- Offline & point-of-sale data — data generated by users when they interact with your business offline (e.g., in-store purchases or returns)
- ERP & inventory – SKU, stock and fulfillment, and financial data

Second and third-party data sources

First-party data is the most critical data component when it comes to customer profiles, but building a complete picture often requires collecting data from second and third-party sources. These sources can provide helpful data on interactions that your customer has but that aren't directly observable by you. Second-party data often includes platforms that you have access to, like ad networks, while third-party data is generally accessed from data vendors. Data teams often refer to these data sets as "enrichment" data that augments first-party data.

Second-party data sources

- Ad platforms — any ad platform where users see or click on your ads (this data can be extremely helpful in building attribution models)
- Shipping data — data related to the shipping and delivery of purchases made by your customers

Third-party data sources

- Browsing and interest data — data from media and content networks on what kinds of content and topics your customers view
- Intent data — aggregated behavioral, search, browsing, and purchase data that represent user intent to purchase
- Purchased demographic, firmographic, and technographic data — user and company attributes purchased from a data vendor

"Other" data sources... handling the inevitable outliers

Despite the availability of modern, robust pipelines for event data, batch data, and other data sources, tech stacks often include legacy components. These are usually core to operations but cumbersome to integrate with modern infrastructure. The best practice for this kind of data collection is to buy event and batch pipeline solutions with robust APIs or protocols that can receive data from sources not provided out of the box. Your team will still have to write custom jobs to get the data from the legacy source to the vendor endpoint, but this is a lower lift than running an

entire end-to-end pipeline. More importantly, you can align the schema of the source data with the existing pipeline, saving you the work of cleaning and modeling downstream.

Step 2: Modeling the data into complete profiles

Migrating pipelines from in-house builds or legacy tech to modern vendors frees resources and enables you to focus on solving the critical next step of modeling your customer data and producing the coveted complete customer view.

Building complete profiles is complex, but the process requires four steps that build on each other sequentially. When implemented correctly, the completeness and accuracy of each step makes the next step significantly easier. But data teams often create unnecessary complexity by executing multiple steps at the same time, often due to time pressure from the business.

In this section, we will walk through the key components of complete customer profiles:

- Profile schema design
- Identity resolution
- User features
- Methodologies: Config-based vs query-based

We'll also include a breakdown of methodologies that will help you minimize heavy modeling work and ship your identity resolution project faster.

Profile schema design

Before we break down the details, let's define what a complete customer profile actually looks like in your data warehouse. On a basic level, your customer 360 exists as a table with one row per

user and a set of columns that represent everything you know about that user. Those columns are often called user traits or attributes and represent the schema of your customer profile table. There are four key types of traits in this schema:

- **Unique identifiers —** Unique identifiers are all of the unique IDs you have for your user from every data source. As we detail below, stitching all unique identifiers into an identity graph is a critical first step in building customer profiles.

- **Known user attributes —** Known user attributes are all of the additional, non-ID data points about a customer that live in various tools and data sources. These are often **demographic**, **behavioral**, and **stage/state** related.

- **Computed traits (user features) —** Computed traits are calculated by combining data sets that contain information related to the user. These are also called user features. **User features** are often related to key business metrics and are used to drive all kinds of insights, optimizations, and customer experiences.

- **Predictive traits —** Predictive traits, like customer lifetime value, are computed on top of the three types of traits above using predictive machine learning algorithms.

All of these features change over time, and sometimes in real-time, so the schema must be updated frequently to ensure all customer profiles are up to date with the latest data.

Making your schema scalable

As you construct your customer profile schema, there are two key considerations to keep in mind: efficiency and point-in-time features.

Efficiency

Computing features in a scalable way requires balancing the frequency of updates with the cost of running the queries on your warehouse.

Remember that your customer profile schema relies on the schemas of the customer data you have collected. If the schemas of your data sources create unnecessary complexity, you will incur significant costs.

> "It is critical to remember that your customer profile schema relies on the schemas of the customer data you have collected"

As an example, think about a computed trait like user_total_revenue. If all of your user events are in one gigantic table, your query will have to scan a huge amount of data every time you compute the user feature. If, on the other hand, your user events live in dedicated tables (e.g. a "purchases" table), the query will be much more efficient. When you add up those efficiencies, the total cost savings can mean the difference between a good and bad meeting with your CFO.

Point-in-time features

Many times businesses need to look at a feature that represents a point-in-time in history (as opposed to the current state). Not only are point-in-time features helpful for building analytics, but they are also necessary for training predictive algorithms like churn models.

Creating point-in-time features requires storing historical data, not just the current state. These too can have an impact on efficiency. If you have clear definitions that are agreed upon, you should be able to compute point-in-time features infrequently (and sometimes only once). If definitions change, though, you must recompute that feature across all historical data for all users, which can quickly become expensive.

Identity resolution

Solving identity resolution requires stitching unique identifiers, establishing a canonical identifier, and ultimately building an identity graph that includes every unique identifier associated with a user.

Identity stitching

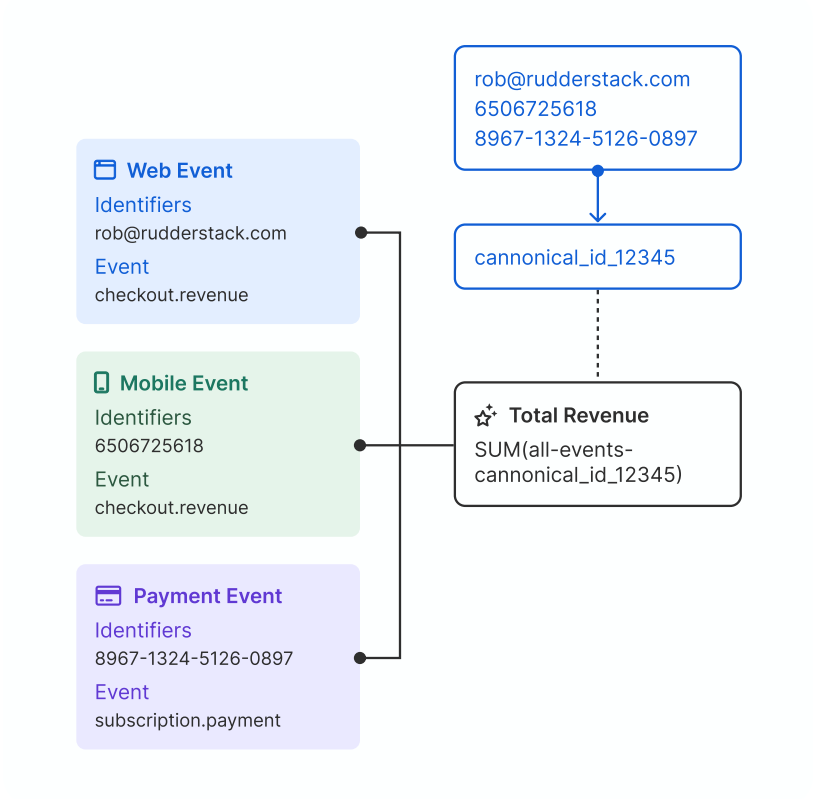
The first step in building a complete customer profile is ensuring you have a comprehensive set of unique identifiers that represent the customer. This is called **identity stitching**.

Even simple user journeys can make identity stitching complex. For example, a user might make a purchase on your website with their email, rob@rudderstack.com, as their unique identifier. Later, the user downloads your mobile app and makes a purchase using a mobile number, 650-672-5618. Finally, after trying the products, the user buys your subscription service. This transaction is tracked through a payments system with an opaque UUID, 8967-1324-5126-0897 (let's call it payments_id).

Without stitching each unique identifier to the user, systems and teams will wrongly assume that these three transactions are from three distinct users. Worse, you are unable to calculate — or inaccurately calculate — important computed traits like user_total_revenue because of the fragmented user identities.

Establishing a canonical identifier

In order to stitch this user's identity, you must map the three identifiers, email (rob@rudderstack.com), mobile (650-672-5618), and payments_id (8967-1324-5126-0897) into a **single canonical identifier** (let's call it canonical_id). This step is called identity mapping, and it's foundational to computing traits like user_total_revenue. In this case, you would sum all purchase events associated with the user's canonical_id.



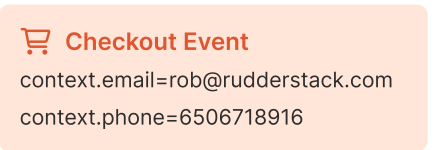
The output of associating these IDs to a single canonical ID should be an identity graph. The work of building an identity graph is a complicated, time-consuming endeavor for data teams. It typically involves a sequence of complex joins in SQL that are often painful to troubleshoot and update as new data sources get introduced — but it doesn't have to be that way.

Building an identity graph

With the right methodology and tooling, building an identity graph doesn't have to be so difficult. To associate different identities (email, phone, payments_id) to the same user, we need to leverage some common information across the identities that allows us to join them all together. The result is a first-party, deterministic identity graph.

Let's say the user in the example above provided their phone number along with their email when they checked out on web (remember email was the identifier for the purchase event). Similarly, when the user bought the subscription service, they entered their email to get notified of shipments (payments_id was the primary identifier for this event). In the tables representing payments data, the payment object itself often contains the payments_id, while a customer object contains specific user information like email.

Formally this data can be modeled as a graph where nodes are unique identifiers like email, phone, and payments_id. Edges are introduced to the graph when we know two identifiers represent the same user because they occur in the same data object. For example, the email (rob@rudderstack.com) and phone (650-671-8916) were both present in the web purchase event, while the email and payments_id (8967-1324-5126-0897) are both present in the subscription event.



Note that the edges are from different sources. The web purchase event binds the email and phone together, while the subscription event binds the email and payments_id together. By combining these two edges, we know that all these identifiers belong to the same end user and can be mapped to the same canonical_id. In graph terminology, every node that is part of the connected component belongs to the same user.

Identity resolution best practices

Start simple

Remember, the process of building an identity graph is just that — a process. Start with a set of stable identifiers that you

have confidence in and that can provide value in a specific downstream use case like providing more visibility to your customer support team, then augment the graph over time.

Mind your query complexity

Inevitably, you will face the challenge of growing query complexity as queries increasingly represent the actual identity graph – especially when dealing with hundreds of identifiers across many data sources. In the methodologies section below, we outline a configuration-based approach (as opposed to a raw query-based approach) that can help you automate and scale your identity graph without enormous amounts of SQL.

Building user features

User features might be a term more common to data science, but features are a key part of any warehouse-based analytics project. Many analytics teams simply call them metrics because they are generally grouped with other KPIs that are used to run the business.

Features fall into the trait types covered above:

- Known user attributes
- Computed traits
- Predictive traits

The methodology of user feature development with SQL queries is a fairly standard practice for business intelligence. The most common approach is to define each feature (and other closely related features) as a model. Using this approach the same model generates features like total_revenue_last_7_days and total_revenue_last_14_days, while a separate model generates features focusing on recent activity.

Though the individual pieces of the process are straightforward, and data teams are well-versed in how to derive individual

features, scaling the system in the context of complete customer profiles remains a significant challenge as complexity increases. To overcome these challenges, data leaders must both anticipate the issues that arise at scale and revisit their underlying methodology.

> **Data leaders cite "cost of time and modeling (and maintenance of models)" as the second most common roadblock to solving identity resolution.**
>
> 2023 State of Data Engineering Survey

Feature development best practices

Don't build user features without an identity graph

You need to build your identity graph before you start building user features. Many data teams run into trouble because they attempt to build their identity graph while simultaneously building features. When a certain feature is needed, the team will solve for the nodes and edges related to that feature within the feature model itself.

Solving identity resolution as you go like this creates significant problems at scale. This is because the queries for individual features become increasingly complex as data changes and new data sources are added. Further, the nature of the features themselves obfuscates a complete view of the customer instead of providing more clarity.

A comprehensive, globally referenceable identity graph separates concerns, making it easier to maintain complete customer profiles and develop user features.

Beware of repetitive and complex identity mapping within features

Even with an identity graph, there are pitfalls of complexity and management to avoid. Ironically, these challenges relate to the

identity graph, but the graph itself isn't the problem. The challenges arise from SQL and the underlying query-based "DIY" methodology for developing features (we'll outline a configuration-based approach that solves these issues in the next section).

To understand the problem, we have to start at the beginning. Features for a customer profile table are computed at the user level. Because multiple identities can map to the same end user, computing features requires you to aggregate across all of the user's different identities. In practice, this means nested joins and queries get increasingly complex at each step and need to reference the identity graph.

Let's return to our example user who has three unique identifiers, email, phone, and payments_id. Your marketing team wants to build segments based on total revenue per user, so you need to compute a total_revenue feature.

This feature needs to combine data from the original data points that contain the unique identifiers. That data includes revenue from two checkout events (web and mobile) plus revenue from the subscription entry (subscription). Here's what the SQL queries might look like:

```
revenue_from_web_checkout: sum(checkout_event_web.revenue)
from checkout_event_web

revenue_from_mobile_checkout:
sum(checkout_event_mobile.revenue) from checkout_event_mobile

revenue_from_subscription: sum(stripe_charges.amount) where
stripe_charges.status is not in (failure, refunded) and
subscription is not null
```

This code seems simple until you realize that your query also needs to GROUP_BY a user identifier to aggregate revenue on a user level. The identity graph makes the final grouping easier by giving us the canonical_id, but we still need to pull the actual revenue data from multiple tables, and there isn't a unique identifier that crosses all three data points.

In our earlier example, the web event contains email and phone, and the mobile event contains phone, but the subscription event contains only payments_id and email.

To make things more challenging, the subscription data lives in multiple tables from that source: the payments_id lives in a payments table, while the email lives in a customers table.

Even with the identity graph, you have to create the following series of queries:

- Join the payments and customers tables on payments_id to pull in email and get the revenue value from the subscription event
- Join web and mobile events tables on phone to sum revenue across events for the user
- Join the subscription, web, and mobile tables on email to sum total revenue
- Join the subscription, web, and mobile tables to identity graph to GROUP_BY canonical_id

For highly normalized tables, it's not uncommon to have three to four layers of joins for simple user features — and that's assuming you have an identity graph established. At scale, the result of a DIY, query-based approach results in hundreds of models and an excessive amount of SQL.

Beware of pushing SQL beyond its limits with funnels

Another problem data teams encounter when building user features has to do with using SQL to construct funnels. Funnels are common for user journey analysis and marketing retargeting.

A funnel captures whether a user has done (or not done) a certain sequence of events with some constraints. For example, abandoned_cart identifies if a user has visited the checkout page but hasn't purchased anything within a certain time window. This feature can be used to identify users who are good targets for

ads or promotions recommending the same or related products. But funnel features are difficult to write in SQL, and the queries are notoriously brittle.

Managing complex SQL is only half of the equation… you are building a feature store

Developing features in SQL (or Python, for that matter) is the first part of the equation. Ultimately, a complete customer profile table is a user feature store. Within that feature store, traits and features change, meaning you need to manage additional metadata about those features to give teams context and understand how features change over time. Metadata should include:

- A user-friendly description of the trait so that downstream users know what the trait means instead of having to guess from the name.
- The point in time at which the trait or feature was computed. As outlined above, activation use cases require current value and the value at a point in time in history. This requires maintaining historical snapshots that can be looked up without having to rerun full computations.
- Definition signatures for validity. For each computed trait, you need to store some signature for the trait definition (the query code that produces that trait). If the definition is updated, then all the time-based computed values of the trait should be invalidated even if the trait name stays the same.
- Time-based validity to inform compute scheduling. Traits often have a validity time within which they do not need to be recomputed. For example, total_revenue_last_30_days probably doesn't have to be computed every 30 mins while total_login_last_1_hour should be updated as frequently as possible.

Methodologies: Query-based vs. configuration-based

Considering identity in the models that build up to features isn't rocket science, but maintaining the logic becomes repetitive and error-prone. Beyond the layered joins, you have to take into account edge cases like NULL and INVALID values and often apply transformations like lower-casing email addresses, removing special characters from phone numbers, and standardizing address formats. There is no easy way to abstract these requirements out of the model.

Building funnels involves the same challenge – it's possible, but doing surgery on a massive funnel query is laborious.

It makes sense that many data teams find themselves dealing with these problems because the initial use cases that require features are simple and the modeling is reasonable. The pain doesn't come until you need to layer in dozens of data sources and hundreds of features. It becomes very hard to maintain the system – especially without an identity graph. All too often a few people with tribal knowledge of the query and model set become the only people who can make updates without anything breaking.

These challenges often push data leaders to realize they need to solve for complete customer profiles at the source, but migrating such a large and complex system takes years.

Thankfully, there are new methodologies for building and managing identity graphs and user features that overcome the pitfalls of the query-based DIY approach.

Benefits of the configuration-based approach

Modern data leaders are adopting a configuration-based methodology that abstracts away the complexity of intricate SQL. Tools that generate SQL based on maps and definitions make this possible.

Understanding the map of tables and relationships is one of the key drivers of unmanageable complexity. If you provide that index up front, however, modern tools can use the map to generate the required joins and queries to unify data.

The process becomes even easier when all of the schemas are known because it's possible to infer the map based on collected data. Tools like RudderStack's Profiles leverage this concept to automatically generate identity graphs and user features. RudderStack's pipelines leverage standardized schemas that include established user identifiers, so the baseline map of nodes and edges is already known, and all of the SQL can be inferred.

Augmenting the graph with new data sources and identifiers is as simple as adding input components and column names to the existing config file. Feature development works the same way.

This approach delivers many benefits:

Benefits to the business

- Data leaders can **reduce the cost of building and maintaining an identity graph** with mountains of SQL.
- More importantly, they can **drive revenue faster by building on an existing graph and feature set.**

Benefits to the data team

- Developing features in a config file is much easier, but the generated SQL is still auditable.
- The config-based approach enables teams to use one feature to define other features, which significantly speeds up development.
- A metadata registry can be generated based on the run-time metadata when the customer profile table is generated.

SQL isn't going anywhere, but with a configuration-based approach, data leaders no longer need to waste valuable engineering time on low-level work that doesn't move the needle.

Step 3: Delivering business value with customer data

Once you've done the work of collecting data from every touchpoint and unifying it in your warehouse, it's time to make it useful. Your customer profiles are a significant competitive advantage, but how much impact they drive depends on how you use them. When every team has access to profiles in the tools they use every day, every team can move the needle.

Analytics

Analytics lay the foundation for powerful activations across the data stack, but quality analytics cannot be produced without good data. Richer, more accurate inputs lead to better analysis and more powerful activation.

With all of your company's data, including traffic data from your websites and apps, centralized and unified in a single location, you can build efficient analytics on large, diverse, high-quality datasets to answer questions that siloed analytics tools cannot answer on their own.

> There's this huge movement going on, that's been enabled by cloud data warehouses, to invest in centralizing, validating, cleaning, and joining customer data. As an analysis tool, data is our input, our grist for the mill, so the better, richer, and more trusted the data we have for input, the more value we can create.
>
> Neil Rahilly, VP of Product and Design at Mixpanel

With a complete customer view, not only can you create trustworthy business reporting to answer top-level questions relating to what happened, you can drill down into every next question to discover how and why things happened. Combine this richer

understanding with ML capabilities, and you can move from descriptive analytics to predictive analytics, unlocking a new world of innovative capabilities.

Customer profiles enable you to build accurate and robust user journeys that paint the picture in full color, allowing you to answer complex questions like:

- What behavior in a user's first day indicates they're likely to sign up for a paid plan?
- How does lifetime value vary by paid advertising channel?
- What patterns lead to an increase in purchases of our products?
- What products should we recommend to specific users to increase revenue?
- How should we price our product based on the cost of acquisition or cost of resources per customer?

Business tools

Perhaps the most exciting thing about modern infrastructure is that it's never been easier to get data to frontline teams. With complete customer profiles in the warehouse and the power of Reverse ETL, you can easily share enriched, post-analysis data from the warehouse with every team and every tool. Not only does this give frontline teams a competitive edge, it's also a concrete way to position the data team as a partner within the company, not just a service organization.

Marketing

Marketing is perhaps the most data hungry business unit of all, and for good reason. With incomplete or bad data, marketing budget gets wasted on inefficient campaigns at best and totally off target campaigns at worst.

But when marketing is armed with a complete customer view, they can truly understand and respond to the customer. They can optimize campaigns across every channel and enable previously out of reach use cases:

- Improved ad targeting
- Advanced personalization
- Better segmentation for more sophisticated nurturing
- More effective messaging

Sales

A sales team with access to rich customer profiles can close more deals faster. First, customer profiles enable detailed lead scoring to help sales understand which accounts are ready to buy. This allows them to prioritize efforts for maximum productivity. The granularity of the information available also allows them to go beyond the account level and gain insight into which individuals at those accounts they should target and when.

After identifying leads for outreach, customer profiles enable salespeople to communicate more effectively with every lead. With every bit of relevant customer information from the warehouse readily available in the CRM, sellers can have more productive conversations and close deals faster.

Detailed information from profiles also allows sales teams to identify opportunities for upselling and cross-selling that might otherwise be left on the table.

Customer success

Customer Success teams can leverage the data from customer profiles to increase retention. Customers don't always share every issue with their vendors. Sometimes churn can seem to come out of nowhere because of hidden dissatisfaction. CS

teams can combat this by identifying signals that might indicate churn risk, allowing them to proactively engage their accounts to address underlying issues. CS teams might also utilize data from customer profiles to identify accounts that are primed for expansion.

Machine Learning

Data from complete customer profiles gives data science teams the ability to build new innovative capabilities. The sophisticated and creative nature of these applications means they have the power to drive transformative change and establish wide moats against the competition. Data from customer profiles can fuel machine learning models for:

- Recommendation engines
- Churn prediction algorithms
- Fraud detection
- Advanced lead scoring

Chapter 4

The stack: How to build a customer data platform

Different customer data platform approaches

There are many approaches to delivering the promise of a complete customer view. But until now, every solution has come with significant drawbacks. The constraints of legacy SaaS CDPs led many companies to try to build out their own capabilities in-house, but most were slowed down by the engineering capital required to build and maintain these systems. More recently, the Composable CDP emerged as a sort of happy medium between these options, and while it's a step in the right direction, it doesn't address the full picture.

The legacy SaaS CDP

The legacy CDP was born in response to the SaaS boom as a way to aggregate data from data silos into one place. But these systems were black boxes that ultimately created another data silo. Early products did succeed in building a more comprehensive customer view, but they still provided incomplete data and were largely useful only to marketing teams for specific use cases.

Today, the Customer Data Platform Institute defines a CDP as "packaged software that creates a persistent, unified customer database that is accessible to other systems." Today's CDPs are more flexible than their predecessors. However, packaged SaaS CDPs are fundamentally limited by their architecture, and they are still made primarily for marketing users.

Because legacy CDPs are geared towards marketing, they don't expose customer data in a manner conducive to building applications for more sophisticated use cases like user journeys, attribution, ML models for churn prediction, and product recommendations.

The in-house build

With inflexible and limiting off-the-shelf solutions, a tempting option for companies with engineering resources is to build CDP capabilities in-house. This option may seem like a good one, but

in reality, most companies get overwhelmed by the magnitude of the project and its ongoing maintenance. An MVP solution may be easy enough to hammer out, but these projects seldom scale, and if they do, they become a significant resource drain. As growth accelerates, data volume grows, integration requirements expand, privacy regulations become harder to meet, and error handling gets increasingly complex. Building an internal system at scale can take years, and the maintenance overhead is enormous. Because of this, building in-house is not a viable option for most companies.

The composable CDP

The Composable CDP emerged in 2022 as an alternative to inflexible, packaged systems and cumbersome in-house builds. While not a foolproof solution, it does get one foundational element right – it places the data warehouse at the center of the customer data stack. But the premise of the composable CDP is essentially a wholesale unbundling, and it goes too far. The composable CDP separates and isolates each major component of a CDP:

- Streaming (and real-time transformations)
- ETL
- Warehouse transformations
- UI/Segmentation
- Activation/rETL
- Storage

This delivers on flexibility, but it does come with a few critical drawbacks. The fragmented nature of the system means you still have issues with incomplete and incompatible data. Plus, managing data quality across a significant number of separate vendors can become problematic. More importantly, the composable system requires the data warehouse as an intermediary, so it cannot support real-time use cases.

The Warehouse Native CDP

The Warehouse Native CDP is a packaged platform that runs directly on the data warehouse and helps data teams deliver value at every stage of the data activation lifecycle: collection, unification, and activation. Like the composable approach, The Warehouse Native CDP solves the data silo problem by building around the warehouse, but it deploys the integration, real-time transformation, and activation layers as a connected, governable, and observable end-to-end system.

Leveraging the data warehouse as the central, transparent source of complete customer profiles eliminates data silos and allows marketing (and every other team) to use their tools of choice. More importantly, downstream teams can use these data activation tools to their full potential because they have access to complete, enriched customer profiles.

Because the Warehouse Native CDP is an end-to-end system, data leaders don't have to invest time and money building infrastructure or bridging the gaps created by siloed legacy CDPs. Moreover, they still have full control over both pipelines and the modeling of customer profiles in their own warehouse.

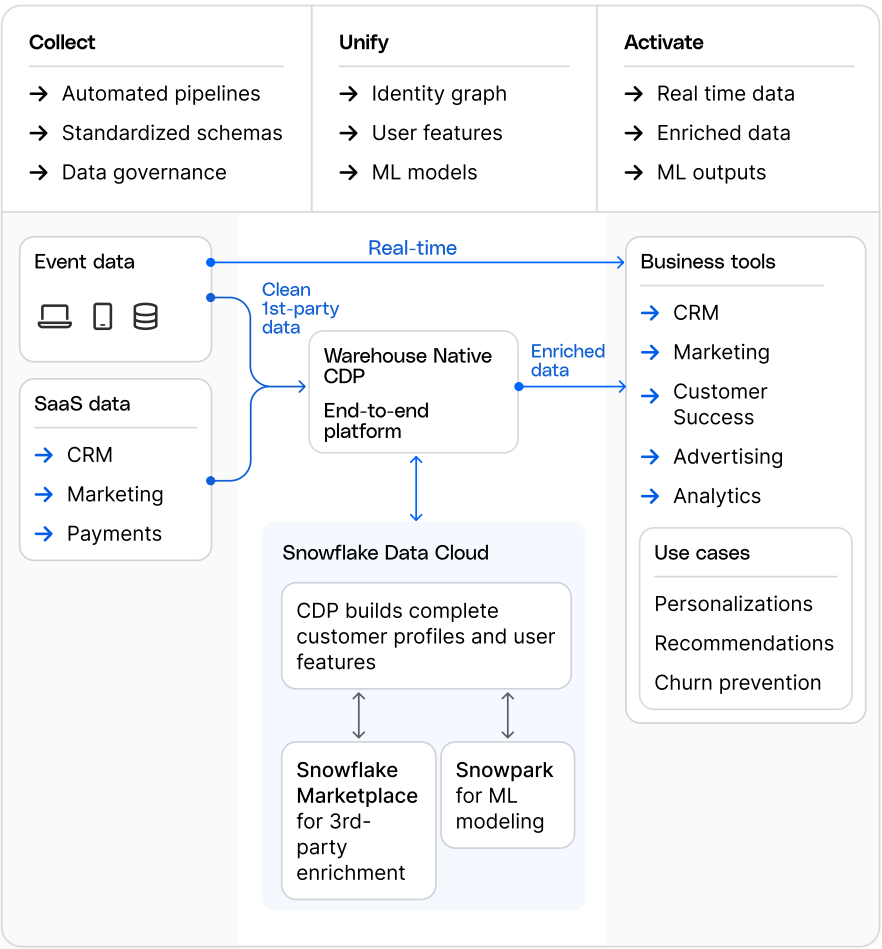
The Warehouse Native CDP provides flexibility without compromise and delivers:

- Seamless integration with every tool in the stack
- Support for real-time and batch
- Automated identity stitching and customer 360
- Single observability plane

Architecture guide: The Warehouse Native CDP

The Warehouse Native CDP provides data teams end-to-end tooling that helps them drive value at every point in the data activation lifecycle:

- Collection
- Unification
- Activation



Collection

- Real-time event streaming pipelines
- Batch ETL pipelines
- Data governance

The Warehouse Native CDP delivers value beyond the simple utility of data pipelines. Transformation and data governance features allow you to ensure data quality at the source. It also ensures that data collection follows standardized schemas designed to populate the identity graph.

Unification

- Identity stitching
- User features
- Machine learning models

As an end-to-end solution, the Warehouse Native CDP can leverage the power of known schemas to automate complex modeling for building identity resolution and user features.

Move seamlessly from identity resolution to machine learning on the data cloud

Building your CDP on the modern data cloud allows you to bridge the gap between traditional analytics workflows and data science workflows. A platform like Snowflake makes it easy to push the identity graph in your warehouse directly to machine learning infrastructure like Snowpark to significantly speed up the development of powerful use cases like lead scoring and churn prediction.

Activation

- Real-time integrations
- Reverse-ETL
- ML outputs

Not only does a Warehouse Native approach enable data leaders to create value from data faster in their data store, activation pipelines also make it easy to push that value to every team and tool across their organization to drive bottom-line impact.

Built to help you deliver on your data strategy

Modern data leaders are rapidly adopting warehouse native architecture because it leverages the best ideas from both legacy CDPs and in-house builds for identity resolution to deliver a combination of benefits that no other approach can.

Complete, trustworthy data

With automated pipelines and the warehouse as the central, transparent data store, data leaders can eliminate silos and the low-value engineering work of building custom infrastructure.

Flexibility and control

Solving identity resolution with dedicated tooling on the data warehouse makes it easy for data teams to update their identity graph and user features to keep pace with the changing needs of the business while maintaining full control and visibility.

Privacy and security

It has never been more critical to protect sensitive customer data, and your identity graph will be full of it. Building on the modern data cloud allows you to leverage your own data store and all of its world-class security features.

Machine learning ready

Having a ready-made identity graph and rich set of user features in your data warehouse is a force multiplier for AI and ML teams, especially when they can immediately operationalize those in ML tools that are directly integrated in the same data cloud environment.

Conclusion

While there have been thousands of articles that have explored the what, the why, and the how of building a complete customer view, we put this guide together because the timing is right to revisit this topic. Recent advancements pioneered by Snowflake's Data Cloud along with tech partners like RudderStack have made this holy grail attainable, on fast timelines, without the expenditure of significant engineering resources.

Beyond tech developments in data infrastructure, prevailing macroeconomic conditions after one of the longest bull runs in history have put pressure on companies to generate more revenue and reduce costs — two areas that identity resolution can help data teams unlock.

To conclude with practical guidance for data leaders, we cover three powerful use cases that our customers have built with identity resolution to better understand their businesses, increase revenue, and mitigate customer churn.

Use case 1: Marketing attribution

"50% of my marketing expense is wasted, I just don't know which 50%" – John Wanamaker

Sales & Marketing cost is a major portion of a company's operating expenses and any opportunity to optimize the spend can have a big impact on a company's bottom line. Being able to identify which campaigns actually drive revenue can help businesses make cost-effective investment decisions. This problem is known as marketing attribution and is usually done in tools like Google Analytics, which provide a limited view of the customer journey.

For marketing attribution to be effective, you need to associate marketing campaign spend to actual customers and revenue, not just top-of-funnel metrics like signups or conversions. Let's take an example of a retailer offering a $50 gift card online to drive customer acquisition. If the gift card recipient decides to make an offline purchase instead, it becomes difficult to attribute the purchase behavior to the gift card offer. This makes it difficult to ascertain lift from the campaign as conversions outside the

website are typically only available in the CRM, app database, or point-of-sale system where they are not associated with the data in Google Analytics or other online tracking systems. Due to this incomplete view, many marketing teams struggle to understand the efficacy of their budget and campaigns.

The Data Cloud is the right platform to bring this data together. Warehouse Native CDPs such as RudderStack can collect ad spend data from all marketing channels, website data to track visits and conversions, and pull in complete revenue data from CRMs or point-of-sale systems. With a clean identity graph linking every touchpoint — online and offline — data teams can help marketing connect the dots with visibility into the complete customer journey, a foundation for multi-touch attribution modeling, and, most importantly, hard proof of what's working and what's not.

Use Case 2: Personalized Recommendations

"Despite making up just 15% of online shoppers, repeat customers account for one-third of all online shopping revenue and spend three times more, on average, than one-time shoppers." – *Yotpo Research*

People who drop off during the purchase process are a significant revenue opportunity for brands. A recent study showed that 23% of people who drop off after adding a product to their cart can be influenced to complete the purchase by relevant product recommendations or discounts.

A recommendation system is only as good as the data it is built on. While first-party user activity data like product clicks and items added to carts are very useful, so is other data like total historic transaction volume, interactions with customer support, or even third-party data like net worth. Most black box recommendation systems have integrations to get first-party data, but not data from transactional, SaaS, or third-party sources.

The second aspect of recommendation systems is the algorithm itself. There are multiple kinds of recommendation engines from simple, market-based analysis to systems built on deep learning.

What works best in a specific case depends on many factors from the quantity and quality of training data to the domain itself. For example, deep learning based systems work best when a significant amount of training data is available. Traditional approaches like next-best-action require high-quality data but can work off of lower volumes.

A CDP running on a Data Cloud environment is the right platform for building robust recommendation systems. First, it gives data science teams access to every customer touchpoint, from first-party behavioral data to in-store data. Using tools like Snowflake marketplace, those datasets can be augmented with third-party data to build a more comprehensive picture of the customer. Lastly, with ML tooling available in the Data Cloud itself, teams can move fast without having to set up additional infrastructure to build performant recommendation models.

Use case 3: Churn prediction

"Customer churn costs US companies over $168 billion per year."
– CallMiner Research

Churn is the biggest revenue sink for any subscription business from mobile providers to SaaS platforms. For those companies, the sales and marketing cost of acquiring the customer is usually much higher than the ongoing cost to serve the customer. If customers are at risk of churning, it is far more cost-efficient to retain the customer through some kind of incentive (discounts, additional services) than it is to lose them. At the same time, you need to be selective in distributing discounts because of the costs involved. Finding the balance between retention and cost requires highly accurate predictions of churn likelihood.

Fortunately, customers often provide multiple signals before churning. For example, their usage of the product declines, they open support tickets, or they complain on the company's review site or social media. Building a machine learning model to take this data as input and predict a customer's likelihood to churn is fairly rudimentary, but most data teams get caught in a

never-ending cycle of collecting and joining the data needed to build the model in the first place.

With every touchpoint and a comprehensive identity graph in the data warehouse, data science teams can ship churn models quickly, and, using the CDP, push those data points directly to the teams and tools running retention campaigns.

Bringing it all together

In a challenging economic climate, understanding customer data is one of the most important initiatives for any business. Marketing attribution and product recommendations are key sources of insight that can help guide strategy for cost savings and growing existing customer relationships. Additionally, better churn prediction ensures that businesses have a clearer understanding of when customers may leave so they can devise strategies for retention. Accurate analysis and modeling of customer data gives businesses an edge over their competitors — particularly in a downturn.

Snowflake's Data Cloud and RudderStack's CDP are purpose-built to address the data infrastructure challenges that arise from attempting to build a full view of the customer. RudderStack enables companies to collect clean data they can trust, Snowflake Data Cloud offers the compute and storage infrastructure to unify this data around an identity graph, and the two tools work in concert to activate it with best-in-class third-party tools that drive exceptional customer experiences.

When data leaders enable their business teams to unlock these use cases built upon identity resolution, their companies are well on their way to delighting their customers and driving outstanding results.

[1] https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/the-value-of-getting-personalization-right-or-wrong-is-multiplying

[2] https://www.yotpo.com/blog/return-customers-infographic/

[3] https://learning.callminer.com/c/whitepaper-churn-index-2020?x=CFl8z6&lx=amFx-JO&xs=225891#page=1

rudderstack

Building a complete view of the customer has the potential to unlock incredible value for businesses. Whether you are looking to build better marketing attribution models, personalized recommendations, or churn prediction models, identity resolution is essential. It is not surprising that companies are willing to allocate large budgets to build what is often referred to as the "golden customer record." If you are a data executive looking to ship projects that move the needle, this guide is for you.

"This resource on Identity Resolution, a foundational concept in realizing a full view of the customer, is required reading for today's data leader."

— **David Wells**
Industry Principal, Media, Entertainment & Advertising at Snowflake