



INTEGRATE YOUR BUSINESS WITH

# The Power of $\leftrightarrow$ Sync

Every action your organization takes generates data. You can either spend your time manually leveraging that data through complex automation...

Or harness your data for competitive advantage through the power of sync.



# THE POWER OF SYNC

<b>Introduction</b> .....	3
<b>Part One</b> .....	4
<b>The State of Integration</b>	
<b>Part Two</b> .....	10
<b>2-Way Sync: The Simpler Approach to Integration</b>	
<b>The State of 2-way Sync</b> .....	20
<b>Transform The Way You Work With 2-Way Sync</b> .....	21

# INTRODUCTION

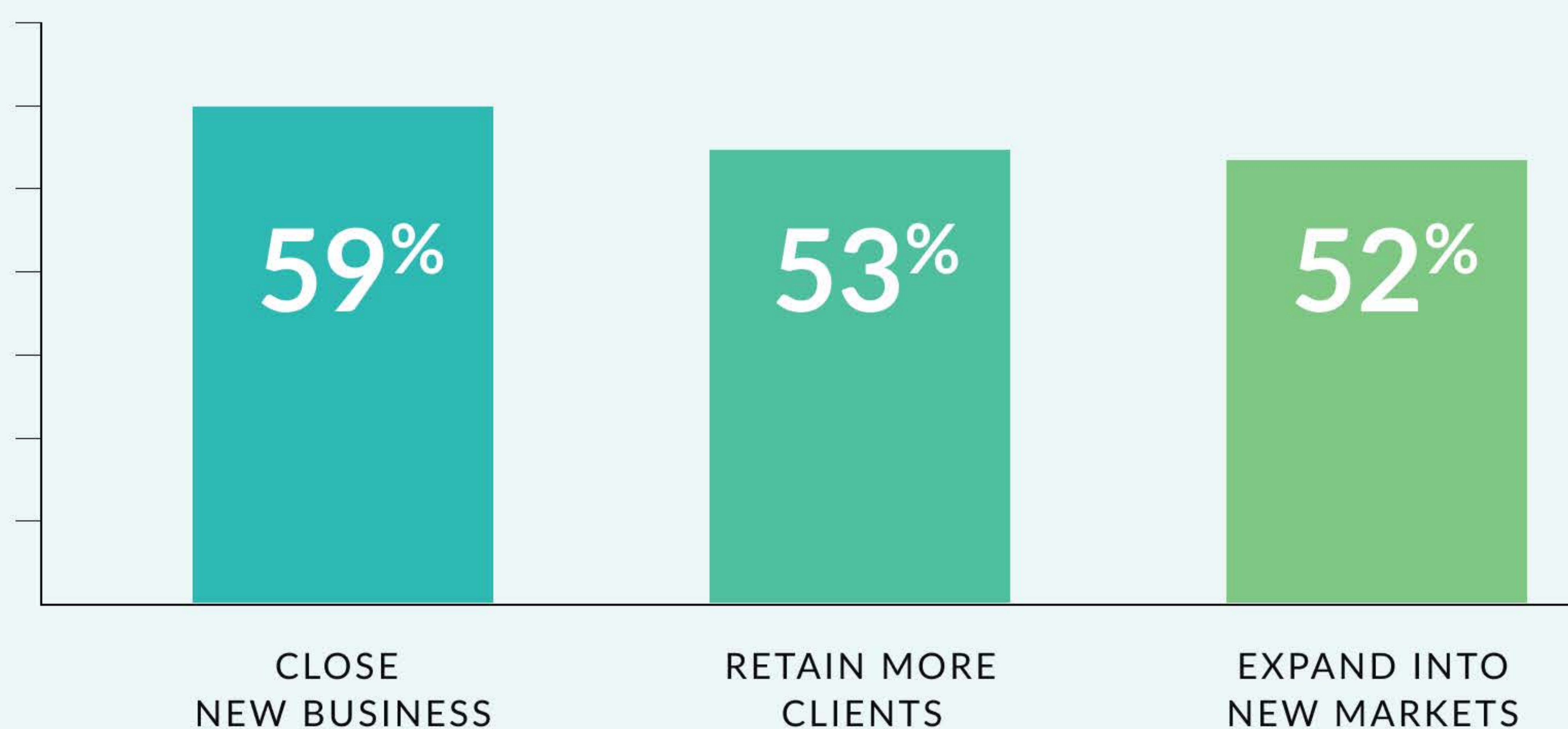
**Questions about an organization's performance are being answered with data where intuition might have once been enough.**

Senior executives today expect to have reports and dashboards delivered on demand so they can quickly make better decisions and scale their business. Large language models (LLMs) have enabled teams to analyze more data faster than ever before.

Dashboards, spreadsheets, work management tools, databases, and CRM tools all factor into this equation. Every action your organization takes generates data. Every new customer generates data. Every tool you use is constantly generating data. You can't make the most of that information without a system to extract, process, and ship it between tools and teams.

Many organizations patch together multiple integration solutions to manage the flow of data between tools, but there's one type of platform that can replace them all.

## Integrations Help B2B SaaS Companies To:



*Study conducted by Merge1*



Learn more at [www.unito.io](http://www.unito.io)

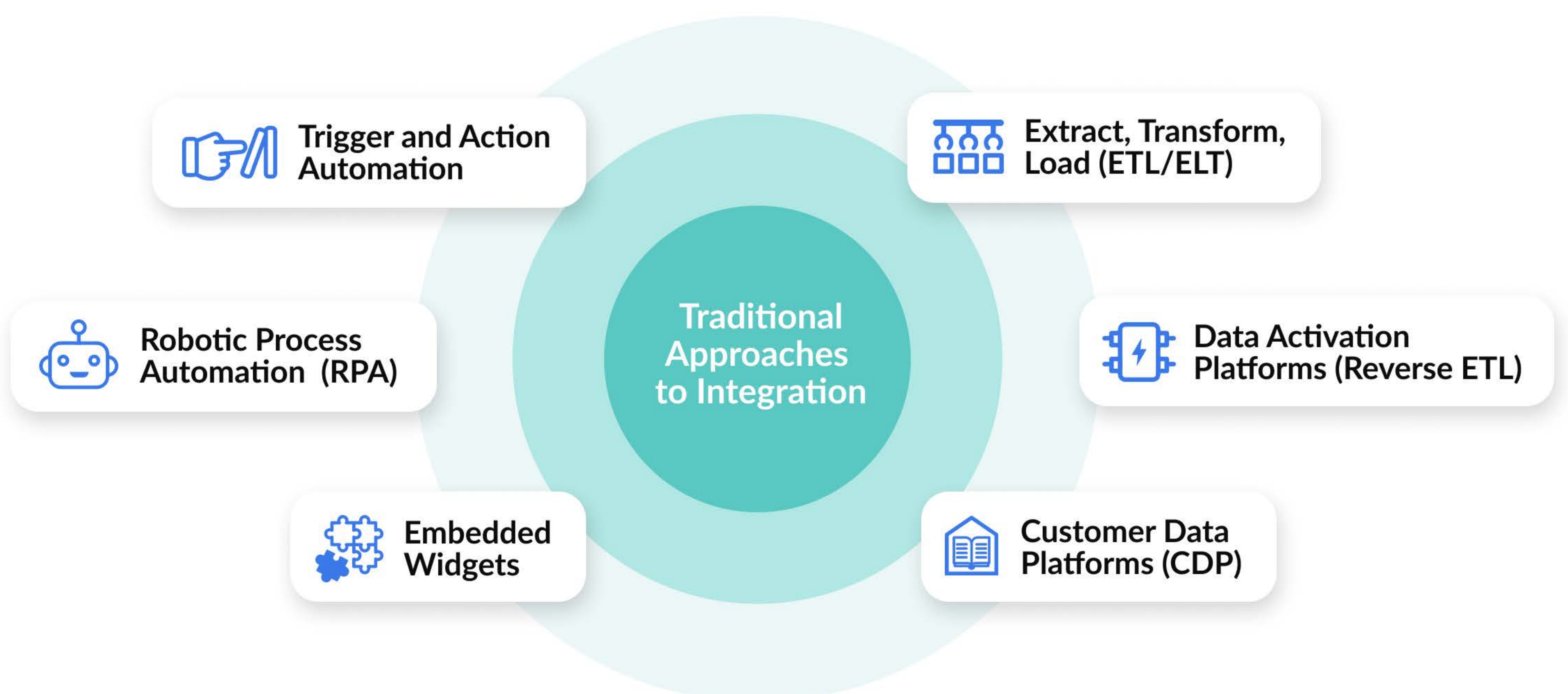
## PART ONE

# The State of Integration

For as long as there's been software, there has been a recognition of the need to move data between otherwise incompatible platforms. This led to the creation of expensive, on-premise solutions for moving that data. These solutions could only be deployed and maintained by specialists with the right technical skills.

## Traditional Approaches to Integration

Most of the integration solutions you'll find on the market now use one or more “*traditional approaches*” for integration. This technology has existed for decades and hasn't changed dramatically, beyond becoming somewhat more accessible for some business users.



Here's a breakdown of options available in the market in recent years.

## Trigger and Action Automation

This approach allows platforms to **handle integrations** for thousands of tools based on if-this-then-that logic. That means every integration built using this approach only has two stages: a triggering event and an action.

- **The trigger** may be a specific kind of data showing up in a spreadsheet, a new task being created in a project management tool, a new entry in a database or any user action in a tool.
- **The action** includes a range of possibilities that are just as wide. Imagine populating rows in a spreadsheet, sending automated emails, booking meetings, etc.



## Embedded Widgets

Many tools provide built-in widgets — known as **native integrations** — designed to support other platforms. Asana and Trello, for example, support other project management tools, messaging apps (e.g., *Slack*), form-building tools (e.g., *Jotform*, *Typeform*, etc.) and more. This gives teams snapshots of data from other tools without requiring them to check another interface.



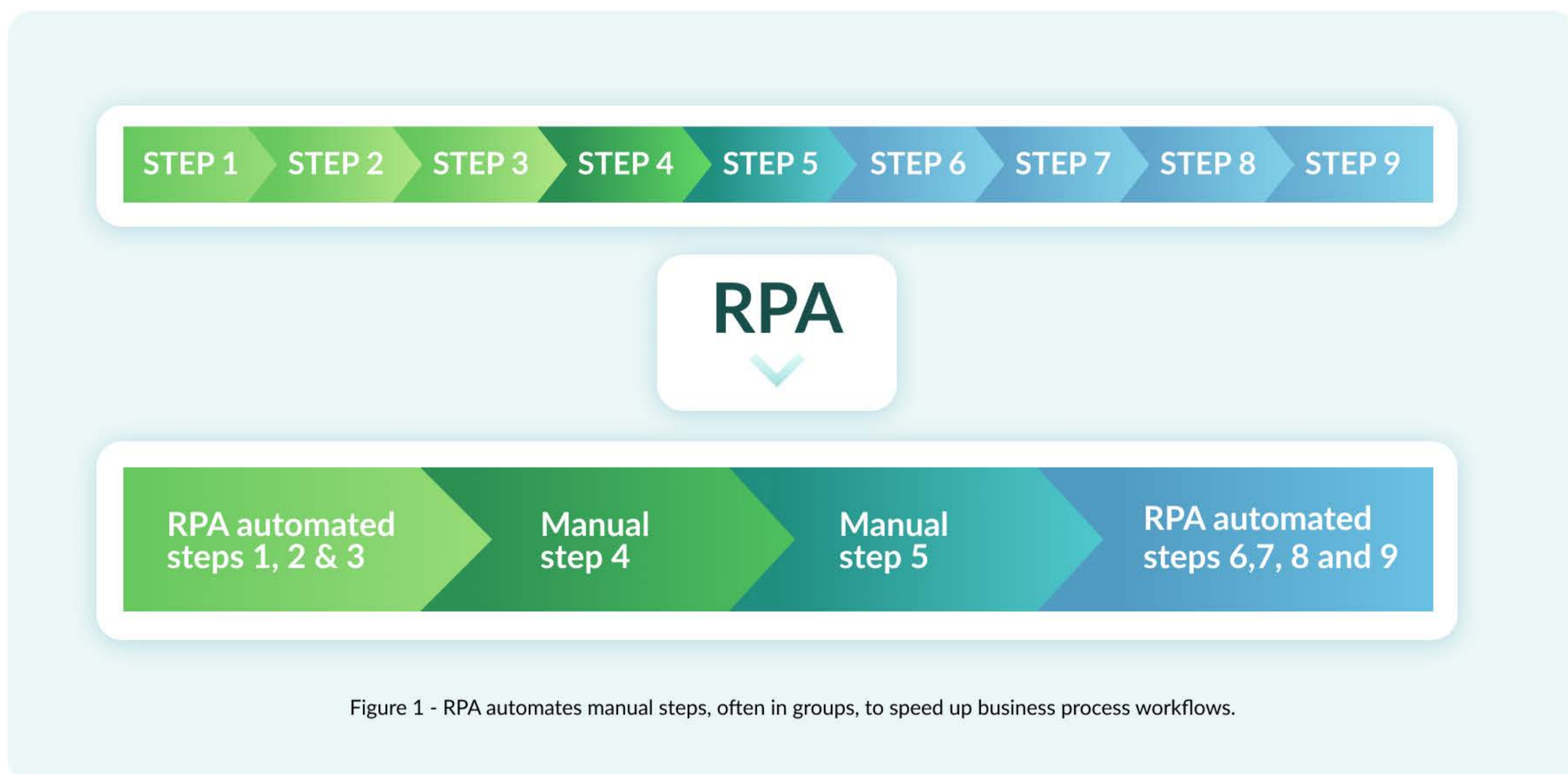
“Native integrations are generally easy to set up, but often just give you a link or preview of the data in the other tool — they don’t let you act on that data or have access to the source of it. For that, you need something more sophisticated that supports 2-way sync.”

– Eryk Warren, Cofounder and CTO @ Unito

Native integrations are popular because they take almost no time to set up and they’re fairly uncomplicated. They’re usually available as an add-on that can be toggled on or off. Despite their limitations, they do allow non-technical users to go from “*no integration at all*” to “*some integration.*”

## Robotic Process Automation (RPA)

RPA allows you to use software bots to **replicate sequences of actions** (clicks, data entry, etc) that would usually be done manually within an application's user interface. Each action is "*captured*," like a recording, and fed to a platform that automates each action in a chain.



With this approach, workflows that would usually be beyond other traditional approaches to integration can be effectively replicated – helping speed along data transfers between tools and automating away hours of manual work.

- **While this approach** can be more flexible than some others, it also has its downsides. Namely, an over-reliance on RPA can quickly lead to a complex, bloated system that requires extensive documentation and maintenance. That makes depending on RPA for your entire integration infrastructure untenable.
- **RPA** is also more brittle than API-based integrations since user interfaces change more often than APIs.

## Extract, Transform, Load (ETL/ELT)

This approach to integration is specifically designed to **supply a data warehouse** with clean, standardized data. Platforms built for this approach pull data from a source tool, clean and process it, and then load it into a data warehouse.

The ETL approach is all about creating a single source of truth from which the organization can pull data and reports, rather than distributing data throughout teams and tools.

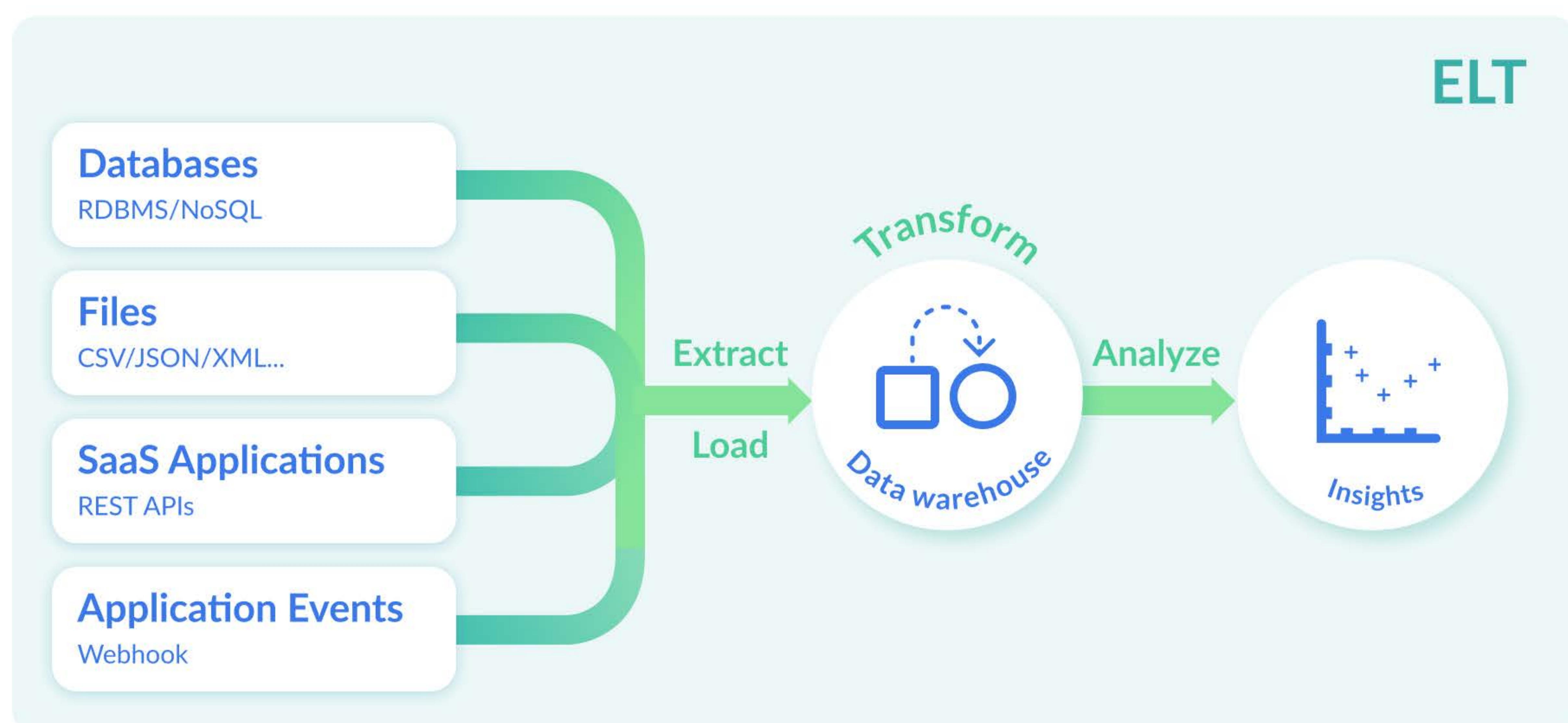
Users can always go to that single source of truth for their data, instead of looking for it across many tools.

This approach pushes data in one direction to a central warehouse and, depending on the tool you use, transforms it. ETL usually deals with **large volumes of records** processed in batches. This process leaves room for error if someone tries to access data between batches, since they would potentially be reviewing outdated information.

## Data Activation Platforms (Reverse ETL)

Reverse ETL accesses an organization's data warehouse to **send information** to other platforms and turn centralized, standardized records into useful inputs right in the apps or tools teams work in each day. Combined with ETL, this approach can essentially create an infrastructure that handles the shipping of data between multiple sources through a central database, on to platforms where it can be acted upon.

Like ETL, reverse ETL infrastructures are highly-technical and complex, depending on a precious, limited resource: data engineers. Since most organizations only have a few of these essential collaborators on staff, this can create a serious bottleneck.



## Customer Data Platforms (CDP)

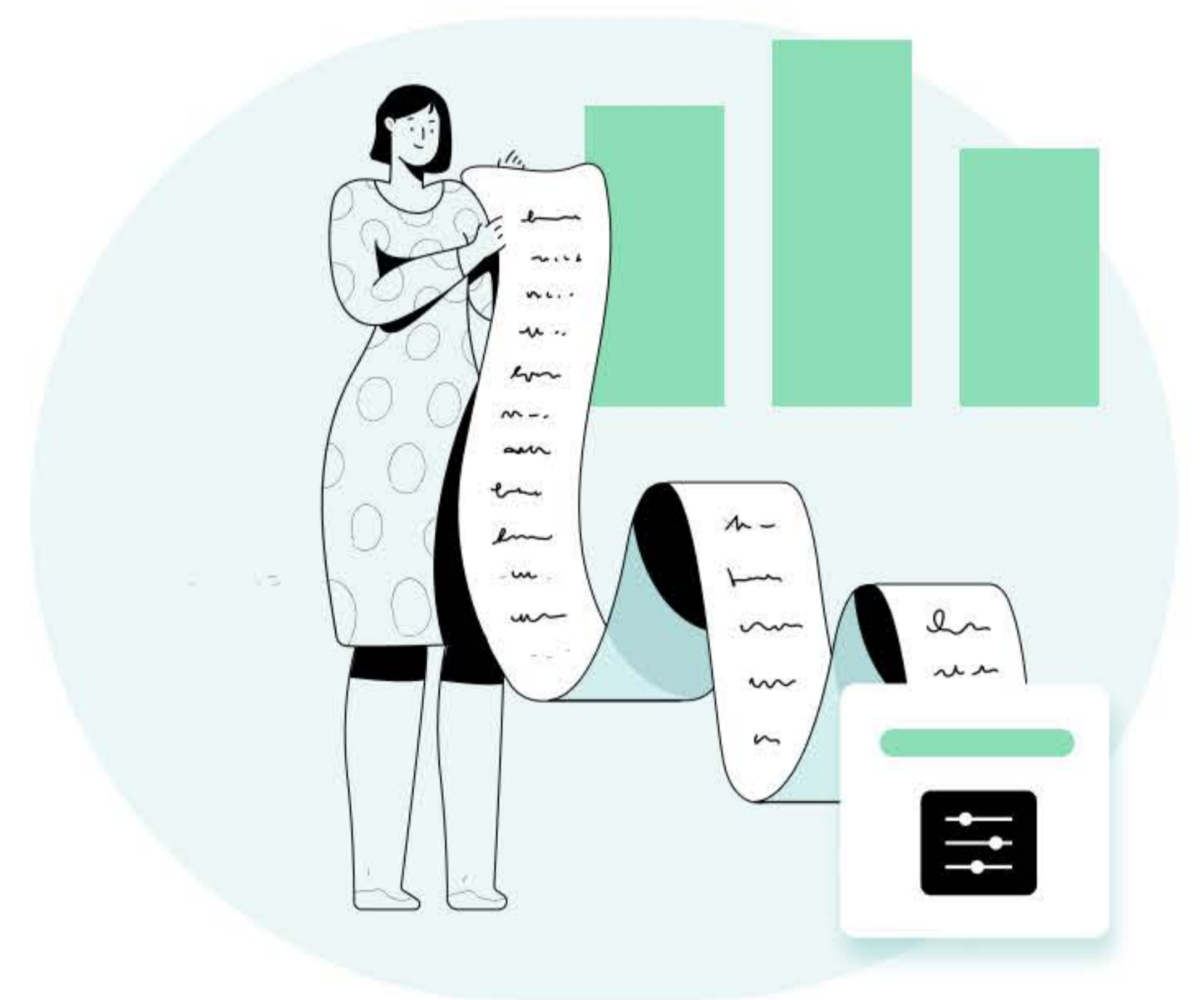
CDPs grab information from multiple sources to create a **single repository** of customer data. This gives marketers, salespeople, and customer service teams all the data they need to increase revenue and drive business growth.

This approach can give your customer-facing teams access to a ton of useful data, but setting it up can take weeks if not months; your engineering teams essentially have to set up integrations for all the data sources users will need. And even then, you'll have a solution that's limited to customer data.

### | Why they've worked (so far)

Traditions don't disappear overnight. The same is true of these traditional approaches to integration. They began to emerge in the early days of software, each designed to address a specific problem.

- **ETL** was developed to manage databases that needed data from multiple sources throughout the organization.
- **Customer data platforms** became popular as customer data became a hot commodity for modern businesses.
- **Automations and RPA** came about to replace repetitive manual tasks.



These traditional approaches to integration were set up, deployed, and maintained by highly technical users, whether that was a developer, a data engineer, or an IT specialist. The average business user couldn't make much use of them on their own.

Since many were deployed on-premise, the high level of technical skill required to do so wasn't an issue. Organizations usually had specialists on hand who could handle these integration requirements.

As long as you had the right technical support and deployed your chosen integration solution for the precise need it was designed for, you wouldn't run into any serious problems. But there was a lot you wouldn't be able to do.



# | Why They Don't Work Anymore

One simple acronym: SaaS.

For decades, businesses deployed software directly on their machines instead of simply accessing it from the cloud. That was the game-changing impact of Software-as-a-Service (SaaS): **any business user can now deploy and adopt software on their own.**

Then came a wave of software adoption, leading to the proliferation of software we know today, and an ever-increasing demand for integration.

While many integration solutions have followed suit and moved from on-premise to the cloud as well, they're still just as technical and complicated. Worse, few of them are anything close to one-size-fits-all. They accomplish very *specific* tasks within a complex infrastructure, but seamless data transfer between tools is still difficult to achieve without professional services or a highly-skilled technical team in-house.



Then, if something doesn't work as expected, you have to sift through dozens – if not hundreds – of individual automations and connections to find the source of the problem. Actually fixing it involves figuring out which systems you have to turn off to make the necessary changes, what you need to pause, and what needs to keep running.

**In summary:** traditional approaches to integration are too technical, built for a single purpose, and a nightmare to fix when things go wrong.

PART TWO

# 2-Way Sync: The Simpler Approach to Integration

## What is 2-way Sync?

**2-way sync is a relationship builder.**

Where traditional integration solutions depend on one-way automation or batched exports that copy and transfer data, 2-way sync platforms create relationships between them. These 2-way relationships make a single data record (a customer, a task, an invoice etc) available anywhere and whenever you need it, exactly as it is everywhere else in the organization's tools. Changes in one place affect the other and vice versa.

These relationships are the foundation of true collaboration, carrying data back and forth with fewer operations than traditional integration options, which means less can go wrong. With 2-way sync in place, a team can access data from your entire tool stack from any tool of their choice.



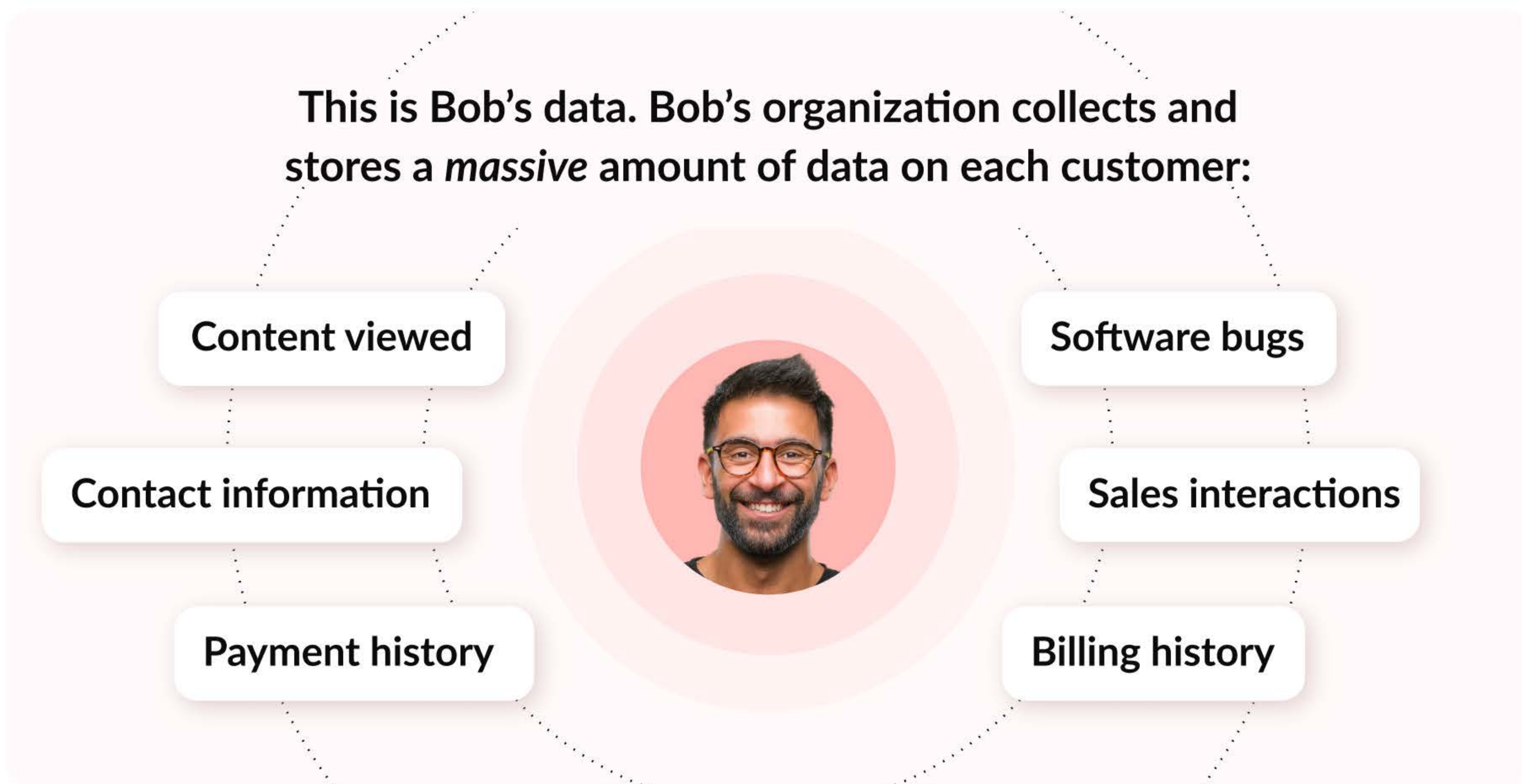
**"2-way sync isn't burdened by assumptions and approaches that are decades-old, and it's built for the cloud-based, AI-powered SaaS era. If used correctly, it can replace an entire integration infrastructure."**

*– Marc Boscher, Cofounder and CEO @ Unito*



Learn more at [www.unito.io](http://www.unito.io)

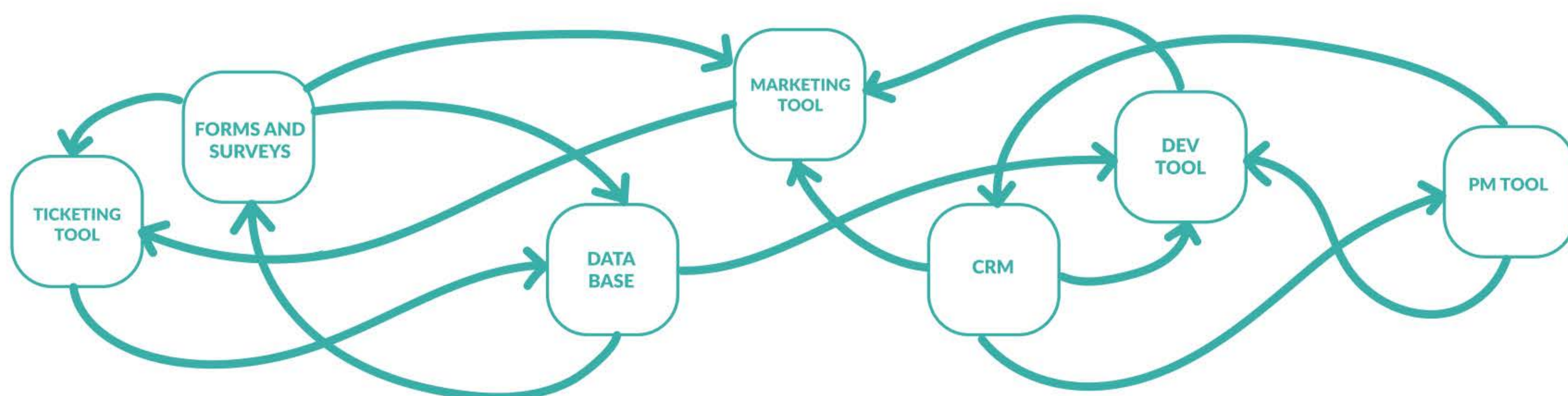
Since we're talking about relationships, let's use customer relationship management (or CRM).



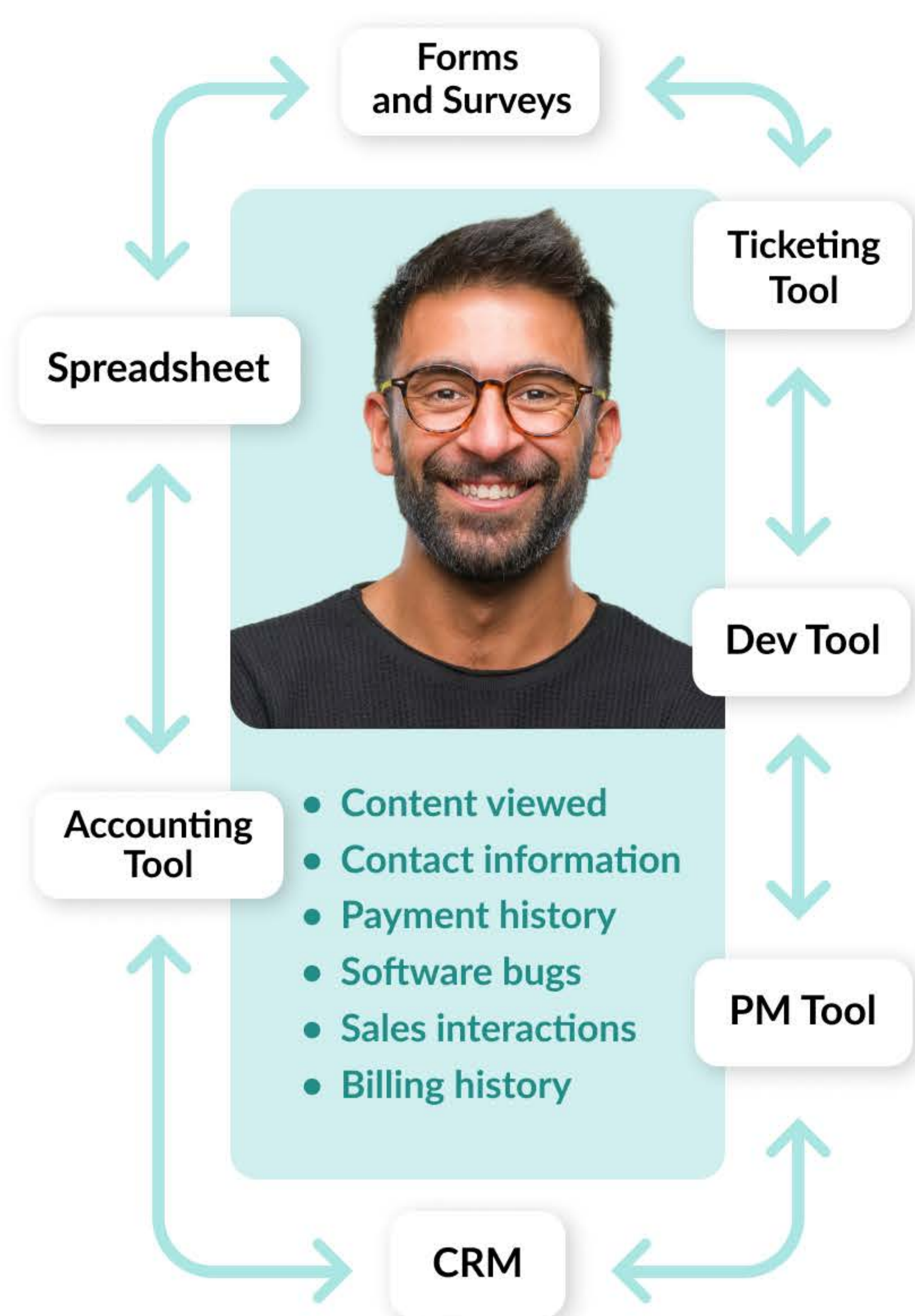
So how would you go about making all that information available to every team with traditional approaches to integration? And how would you make sure it's always exactly where you need it?

Say you need to centralize all information you have about a customer named Bob. First, you need to source Bob's data from anywhere in your organization, then find a way to dispatch it to everyone who needs it.

**If Bob uses traditional integration, AKA 1-way automation, this workflow can get complicated... Quickly.**



Every single arrow represents data moving from one tool to another, and you need an integration to manage each one. Imagine trying to set this up, let alone troubleshoot it. You'd probably get the information you need where it needs to go. Probably.



Compare this to a 2-way sync.

Data is synced **seamlessly** from tool to tool, keeping everything up to date everywhere.

If Bob's phone number changes in your contact management tool, that change can get synced anywhere else it's relevant — without the complex architecture that comes with automations.

A 2-way sync also allows you to **tailor the flow of information** between tools without extensive modifications — or any code.

Only want some of Bob's contact details to make it into other tools? No problem. Don't need Bob's entire life story in your Dev Tool? Only sync updates to support tickets and feature requests.

You can find 2-way sync technology across different kinds of platforms, from dedicated no-code solutions to built-in widgets in other tools. **But they all have the following four elements in common:**

→ **Simplicity**

2-way sync reduces the costs and frustrations that usually come with setting up traditional approaches to integration.

→ **Empowerment**

2-way sync isn't limited to technical users. Anyone can learn how to use this technology.

→ **Adaptability**

A 2-way sync platform can be quickly modified on the fly to adjust the way data flows between tools.

→ **Efficiency**

A single 2-way sync connection can create the same flow of data as dozens of automations.

"Unito grants more visibility to executives and projects managers. Their 2-way sync provided more clarity in our reporting and enables collaboration between teams."

Chris Adams

IT Administrator, Hope International

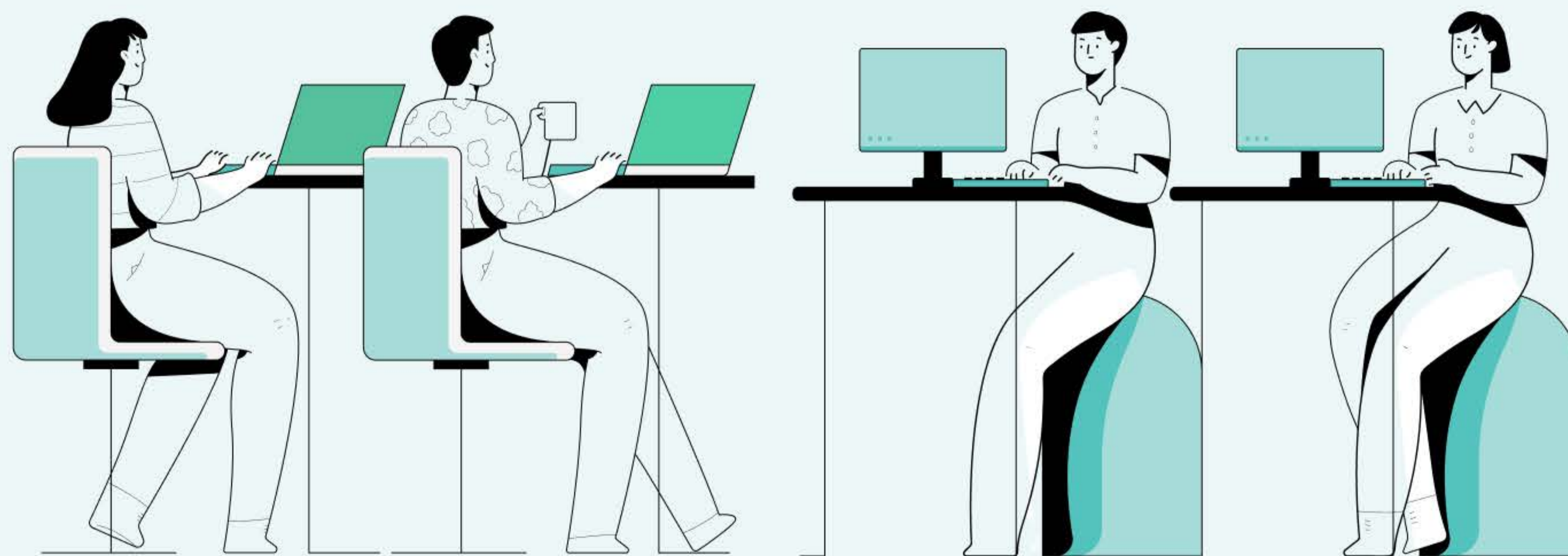
[Read the Case Study](#)

## Where Does 2-Way Sync Excel?

Automation solutions and other traditional approaches at integration have been on the market for decades, so anyone with a technical background is aware of where they excel. But what about 2-way sync? Where is this approach already making a massive difference?

### Collaborative Software Development

One of the greatest challenges of software development is the complete mismatch between tools used by the people writing the software and the people managing the project. This makes every part of the process more complex:



- Gathering requirements from feature requests and support tickets in multiple tools.
- Centralizing requirements for analysis in a project management tool like Jira.
- Designing and whiteboarding features or software to be developed.
- Sourcing feedback and perspectives from developers and leaders who all work in different tools.
- Coding in dedicated development tools.
- Keeping PMs and developers aligned through frequent check-ins in project management tools.
- Testing and deploying code.
- Resulting change management for infrastructure and users

A 2-way sync solution is the best way to maintain alignment between tools used by software teams and their collaborations throughout this process. It allows for all requirements to be centralized in one tool, but they can be consulted from anywhere.

Meanwhile, developers can still get access to input from PMs without leaving the tools they use daily. 2-way sync keeps everyone up to speed in a way no other approach to integration can.

## Contact Management

Instead of keeping contact information in their own files – or worse, remembering it – CRM (customer relationship management) platforms **empower anyone to know everything** about a customer. But that information often comes from a phone's contact list, spreadsheets, marketing automation tools, or even emails.



**“The CRM industry is highly-consolidated, which has led to the emergence of niche needs and gaps that specialized tools aim to fill. If an industry leader doesn’t excel in email sequencing, their clients might seek complementary tools to fulfill those functions.**

**But then the question becomes: how do you connect all those different tools?”**

*– Simon Wahl, VP Marketing @ Flinks*

A 2-way sync platform allows you to centralize your CRM contacts while simultaneously keeping information up-to-date in other tools. That means a salesperson can quickly check for the right phone number of a prospect while a support agent can pull up data on that same customer in HubSpot to properly address an issue.

Other approaches to integration could create new contacts on each platform, but 2-way sync essentially fulfills the promise of your CRM: one place for everything you know about your customers.

## Organization-Wide Reporting

When you think of a report, you're probably thinking of a data set and some charts; a one-time export for someone to look at, turn into some kind of insight, and discard. But while that might have worked in the past, it's hardly a scalable way to do things. Leaders and managers need dynamic dashboards and real-time data without waiting for hours after each request.

*"Achieving transparency across multi-dimensional teams and projects, while empowering each team to self-organize in the project software of their choice, used to be a nightmare for everyone. Executives and PMs were suffocating under the requirement to generate old school "Office Space TPS Reports". Unito solves that nightmare with ease freeing up human capital to Produce vs "cover sheets on all TPS reports."*

Dylan Kennard, CTO, Land Scout



**Usually, there are two solutions to this:**

**Solution one:** a business user tries to create a report themselves, exporting data and wrestling with it until it looks approximately like what they have in mind. That comes with days of headaches as they fight with platforms that aren't built for holistic reporting.

**Solution two:** that same user submits a request to the data team asking for their report to be automated. Not only do they have to wait for their request to be prioritized, but they have to go through a long process of setting requirements, implementing automations, and testing everything before they finally get a report.

Whichever solution you choose, any opportunities your data might have revealed will have likely passed before the report is ready.

With 2-way sync, any business user can set up an integration that automatically generates the report they need when they need it. No more wrestling with uncooperative tools and no support from the data team needed.



Learn more at [www.unito.io](http://www.unito.io)

## Synced Migrations

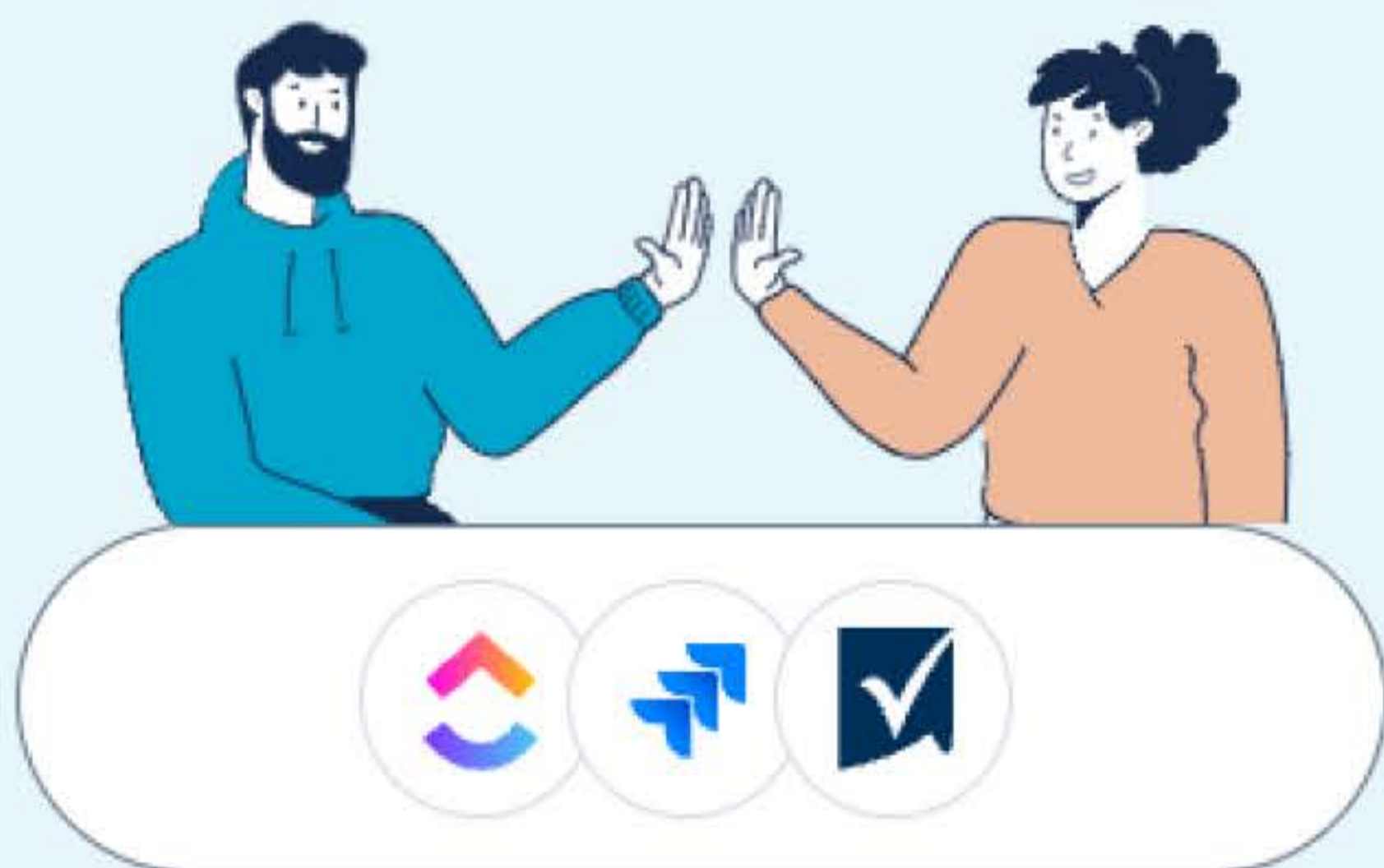
You could have the best integration solution on the market, but there'll always come a time when you'll need to sunset a legacy system in favor of a new one. This can happen because an old, reliable tool isn't so reliable anymore or you need to centralize multiple systems after large, sweeping organizational changes (*e.g. merger and acquisition*).

**Dedicated migration solutions allow technical users to set up a system for transferring data and specific configuration options between platforms.**

This process can take months — if not years — and a single error can lead to rolling back the entire migration for troubleshooting. Even phased migrations, where source data is partitioned and migrated progressively, aren't immune to these issues.

**A 2-way sync platform can enable what's called a synced migration.**

By creating a 2-way relationship between your original system and destination system, you're free to tailor the speed at which data is migrated over. Users can work in either system at any stage of the migration, meaning you can de-couple data migration from user migration as well. Teams can switch systems at their own pace since project-essential data is accessible from both platforms.



**One merchandising team saved \$130k annually and 63 hours per week with Unito**  
[Here's how they did it](#)

## Working seamlessly across any tools

Your organization has no shortage of tools for just about every purpose, and even a single team can be scattered among them. With traditional approaches to integration, you can push work items between these tools, but it won't keep teams on the same page. **No matter the team or workflow, 2-way sync platforms are the best way to work seamlessly across tools.**



## How is 2-Way Sync Unique?

Some organizations using traditional approaches to integration have muddied the waters somewhat by claiming their solutions allow users to achieve a 2-way sync between their favorite tools.

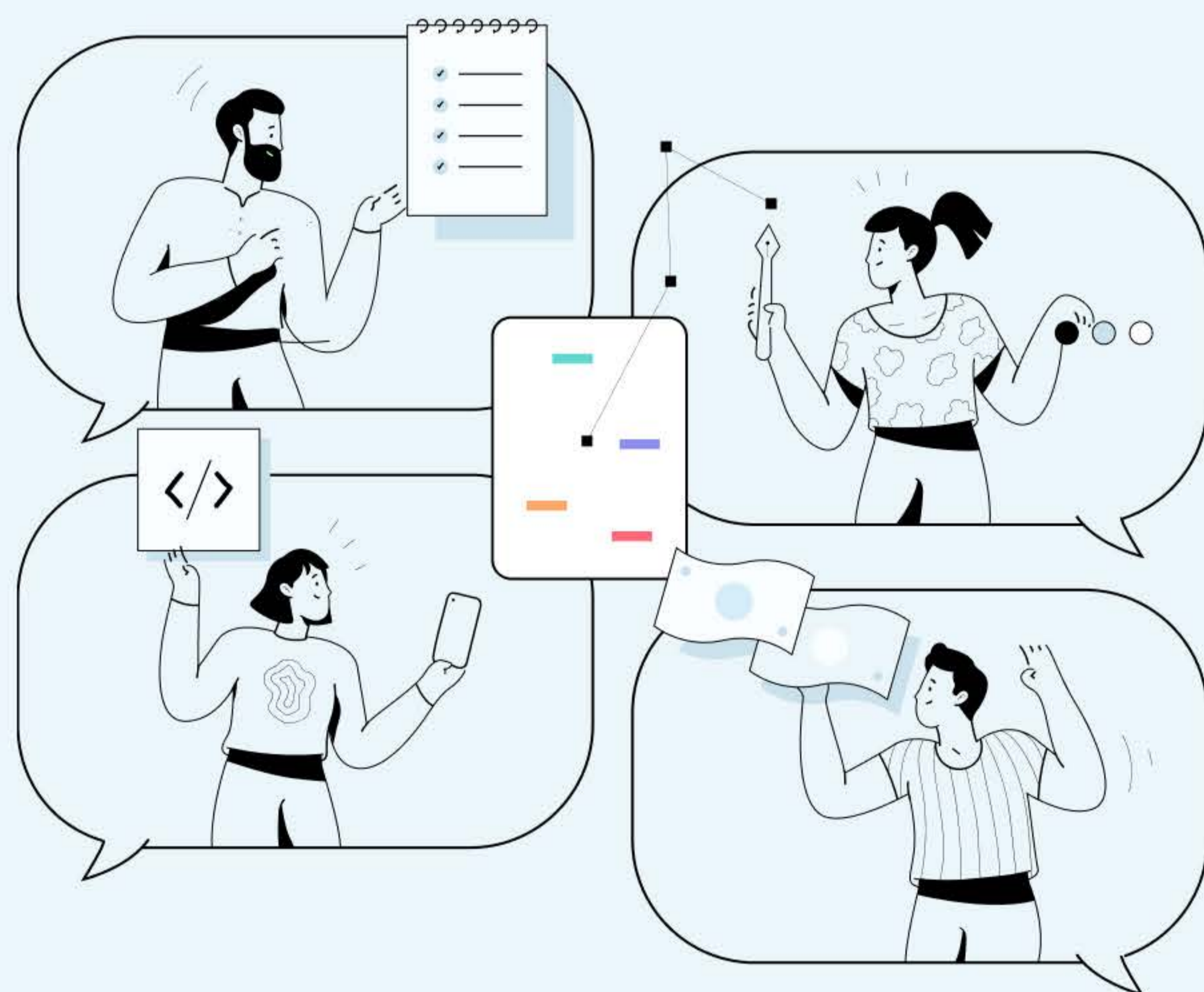
While these solutions can imitate 2-way sync technology, they are still, at their core, generally one-way automations. On the surface, this can seem equivalent to the untrained end user, but the differences usually become apparent with increased troubleshooting, more downtime, and frustrated users.

Here's just a short list of the technical characteristics you'll find in a platform that's built on 2-way technology instead of just simulating it:

**Historical data sync:** Some traditional approaches to integration, like trigger-action automations will only apply to future events. 2-way sync platforms can sync historical data.

### Automatic conflict resolution

If changes are made in equivalent items in different tools at the same time, 2-way syncs can automatically resolve those conflicts by merging the most recent changes at the field level.

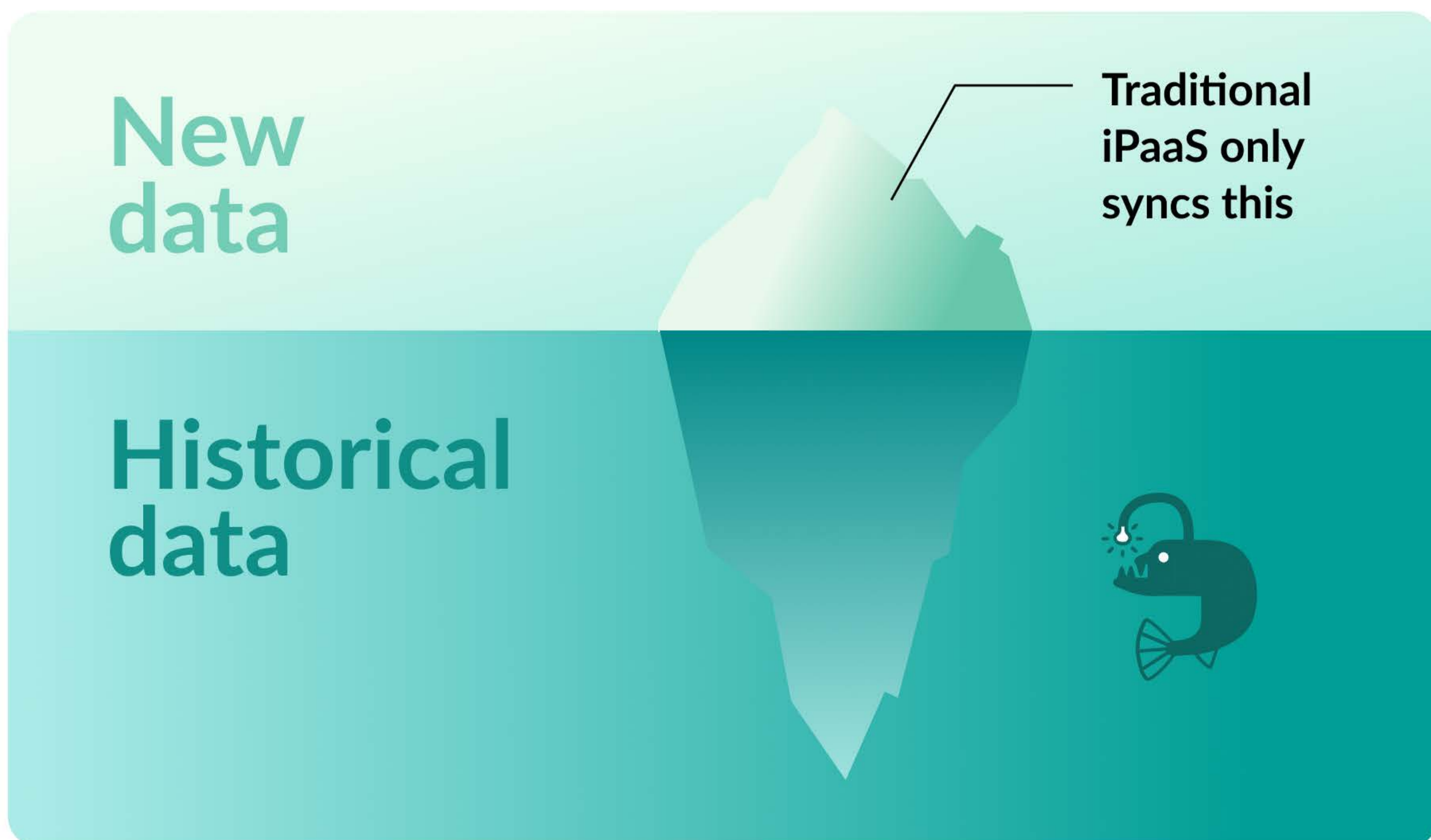


**Infinite loop detection:** A 2-way sync platform understands the relationship between data items across all systems, and thus can prevent the infamous infinite loop problem of trigger-action systems.

**Idempotence:** Traditional approaches to integration are subject to duplicate trigger messages that may be handled independently, causing duplicates. 2-way sync platforms handle these duplicates natively.

**Seamless rich-text conversion:** Different tools offer varied text formatting features, each with its unique syntax. A 2-way sync expertly translates these diverse rich-text formats, ensuring that specific styles exclusive to one platform are preserved and translated appropriately, even after multiple back-and-forth syncs.

**Resyncs:** Many other integration solutions don't allow users to apply configuration changes to historical data retroactively. **That's not an issue with 2-way sync.**



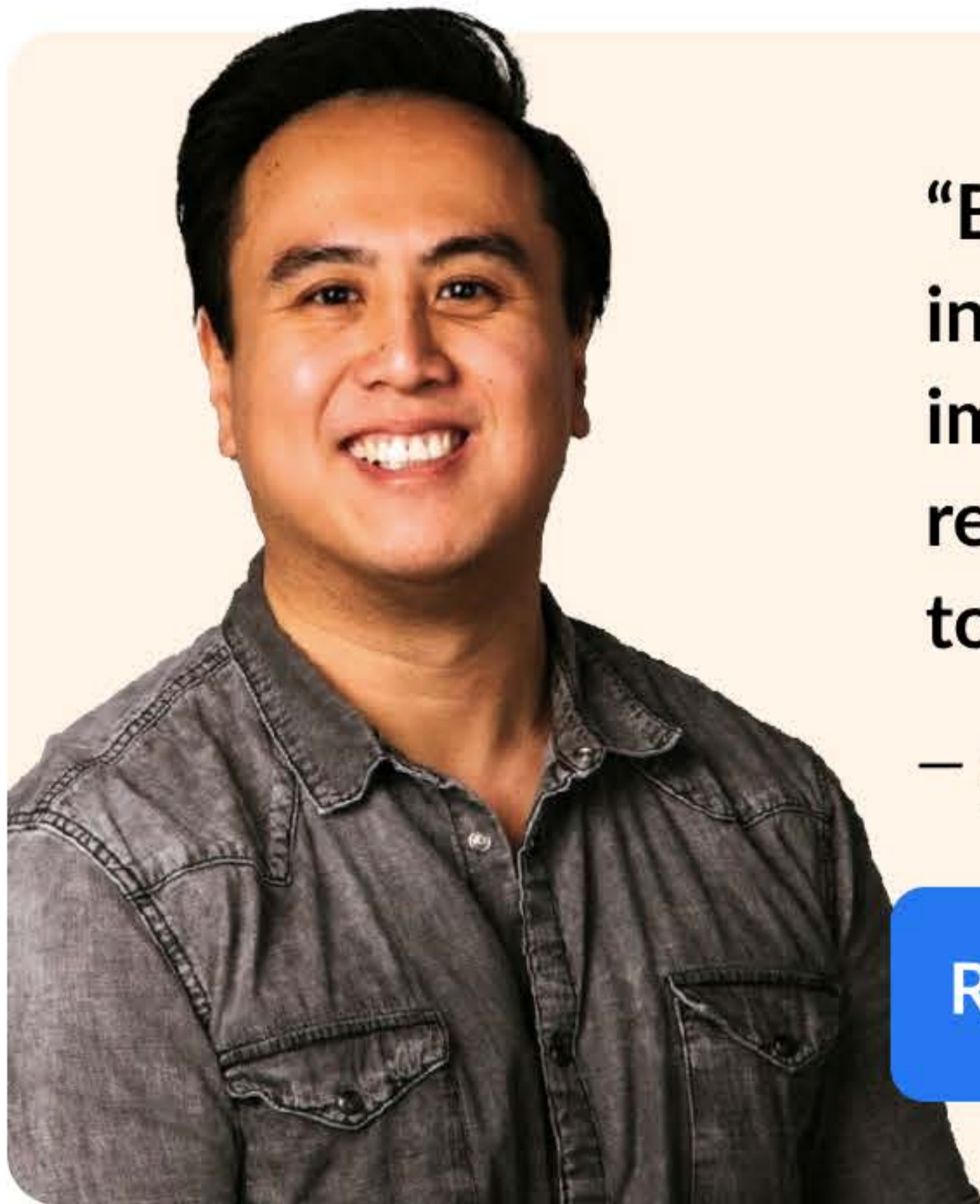
**Order of operations:** Trigger-based automations are based on webhooks or batches, for which the ordering is not guaranteed, leading to inconsistencies in the results. 2-way sync platforms can prevent these issues by always syncing the current state of a work item.

**Live sync:** Most traditional approaches to integration only push data when triggered. Even ETL solutions process data in batches, meaning there are intervals where users will be working with outdated data. A true 2-way sync solution gives you access to the latest data in real-time.

You don't have to dig in this deep to spot the differences between a 2-way sync tool and a traditional automation platform, though. Just try setting two platforms up for the same situation – one traditional approach and one 2-way sync – and you'll immediately spot the difference.

## What Makes 2-Way Sync Simpler?

Since you can meet all your organization's integration needs with one platform, business users can discover new use cases dynamically themselves. That leads to widespread adoption, better metrics in every team, and higher ROI than with a traditional approach to integration.



“Being able to automate client facing boards such as Trello into internal management systems like Asana has been immensely valuable. Using Unito, I am able to significantly reduce processing, project management and handling of day to day in my role as a performance and delivery manager

– *Derek C., Performance & Delivery Manager*

[Read the G2 review](#)



## Unito: Work at the speed of sync. Seriously.

Unito is a new, simpler approach to integration and the leading 2-way sync platform on the market. It has deep integrations for tools like Trello, Asana, ServiceNow, Excel, Google Sheets, Salesforce, PostgreSQL, and many more. Unito takes the technology of 2-way sync and makes it even deeper while maintaining ease of use for end users. You can see this across all four pillars of 2-way sync technology.

- **Simplicity:** A single Unito flow is all you need to map a relationship between two sets of data. And with Unito's built-in workflow designer, you can chain these to match the way you work with ease.
- **Empowerment:** Because Unito is so easy to use, anyone in your organization can learn how it works. That makes finding and training champions in multiple teams a breeze.
- **Adaptability:** Unito is quick to deploy but endlessly customizable. With custom rules and field mappings, you can quickly adapt a Unito flow to just about any use case.
- **Efficiency:** You can deploy a Unito flow in 12 minutes or less. There's no other platform on the market that can match this.



Learn more at [www.unito.io](http://www.unito.io)

# The State of 2-way Sync

You can already find 2-way sync options available in some of the most popular tools on the market. HubSpot's Operations Hub allows its users to sync data in and out of the popular CRM platform. Airtable's Sync enables records to move freely between bases.

But the best option is always a dedicated 2-way sync platform. There aren't many on the market—not nearly as many as those who claim to be a 2-way sync tool—but they have one advantage over built-in 2-way sync features.

**The ability to sync any tool you want to.**

As powerful as HubSpot's Data Sync is, it only works within that ecosystem. The same goes for Airtable and similar tools.

A standalone 2-way sync platform is the best way to make data more available throughout your organization without needing a technical team.



**“Unito performs synchronization every 5 minutes, ensuring that new descriptions, description edits, and comments are synced. The excellent support provided by Unito has meant that we haven't encountered any issues so far.”**

– Nadiia O., E-Commerce Expert, COO

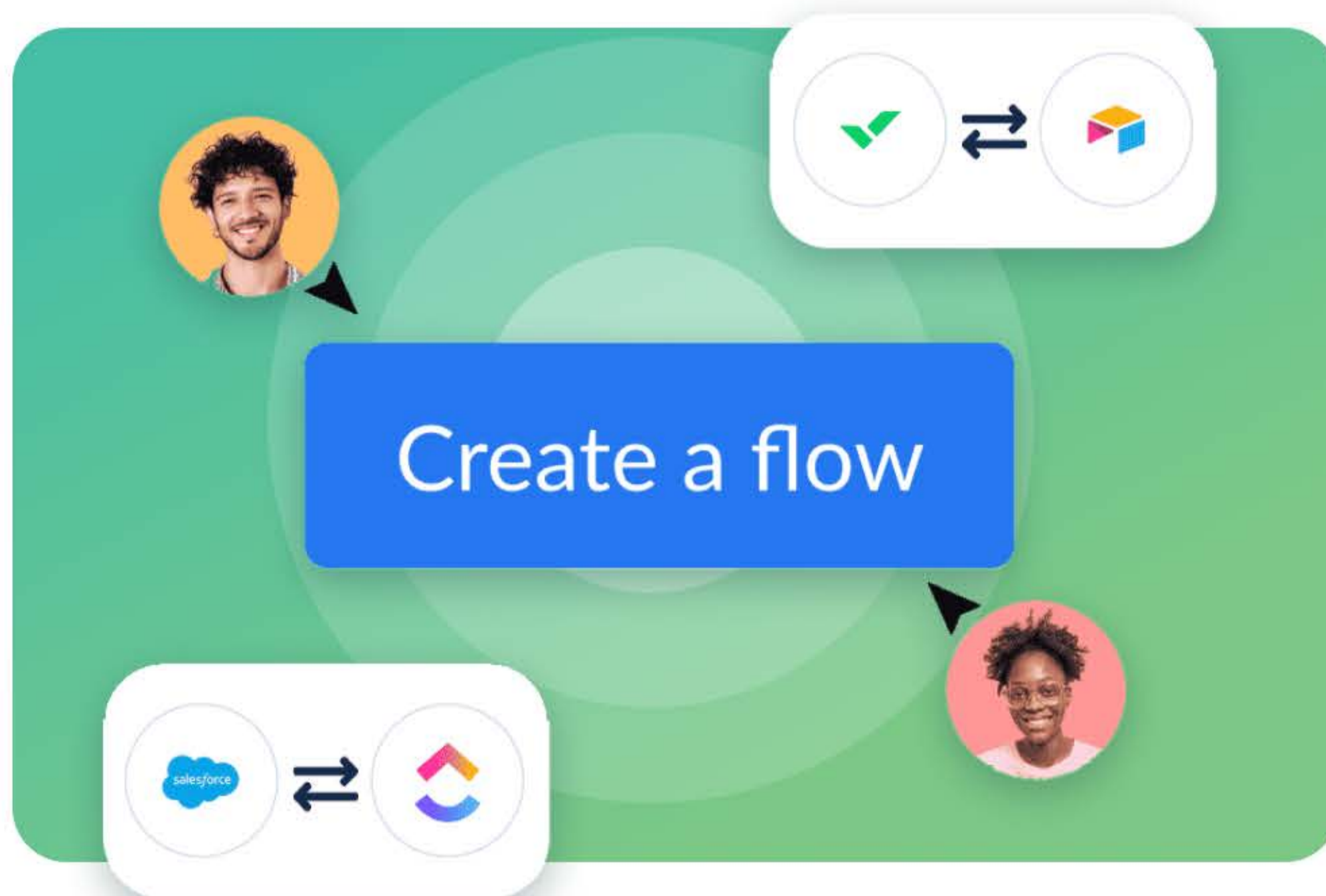
[Read the G2 review](#)



Learn more at [www.unito.io](http://www.unito.io)

# Transform The Way You Work With 2-Way Sync

Rolling out integration solutions for your whole organization doesn't have to take months or dedicated work from your technical teams. Just deploy a 2-way sync solution instead.



- **2-way sync offers a single solution** for all your integration needs instead of a cobbled-together infrastructure with trigger-based automations, built-in widgets, RPA, and ETL tools.

- **End users have everything they need** to get the most out of a 2-way sync solution, instead of needing support from technical users.
- With a 2-way sync platform like Unito, any user can deploy a deep, **completely customizable integration** in 12 minutes..



“Not only is the software smart and nimble, but the best part about Unito is the staff, hands down. This is exactly the kind of thing we want to spend the money on in order to help grow the business and be scalable. Thank you, thank you, thank you for having such a wonderful, human, thoughtful, smart, responsive company.

- A.C. Furey, Executive Director, Thirty-Six Education

[Read the G2 review](#)

