eBook

Getting Started with TrustArc Integrations

A guide for Privacy, Compliance, Information Technology, and Legal teams who want more connected systems, less noise, and no code.

TrustArc

Table of contents

TrustArc

Welcome to Simple Privacy Automation

IF YOU'VE EVER...

- → waited months for an Information Technology (IT) ticket to sync two systems
- → chased five teams to finish one vendor risk review
- → uploaded a 200-column Comma Separated Value file and hoped it matched your fields
- → or wondered if privacy could run in real time, not retroactively

... THIS GUIDE IS FOR YOU.

You don't need Python to understand APIs, or require development resources.

This guide walks you through how to leverage TrustArc Integrations to connect with the rest of your business processes and systems easily and quickly with no code.

YOU'LL LEARN:

- ✓ What TrustArc Integrations are
- ✓ How they work
- ✓ How to start fast, build confidently, and avoid surprises

Let's begin with what makes TrustArc Integrations different and why privacy teams are using them to move faster.

What Makes TrustArc Integrations Different

Most integrations promise privacy workflow automation. Few make it feel this simple.

YOU'VE SEEN IT BEFORE.

- → A vendor promises automation.
- → Then comes the scope creep.
- → Then the permissions.
- → Then the engineering asks.

TrustArc Integrations is different. They're designed for everyday users, not developers.

THAT MEANS:

- You can launch privacy workflows in minutes using expert-designed, pre-built templates. No coding required.
- You can adapt each template or build your own from scratch to match the exact needs of your operations and privacy team.
- You can design and manage automations through a simple visual interface — no engineering skills needed.
- You can orchestrate integrations across your business stack without relying on IT or developers.

WE FOCUS ON WHAT MATTERS MOST:

- → **Speed** – real-time or scheduled flows that respond when things change
- → **Scale** – build once, reuse often, adjust without rewriting
- → **Savings** – fewer delays, fewer tickets, less duplication

All of this works because we keep one principle at the center: **simplicity**.

Not just a style, but a system. Whether you're sending privacy reports to BI tools, fulfilling Data Subject Requests, or flagging risk across systems, it's all possible without code.

That's why most teams launch their first flow in minutes, not months.

Let's look under the hood, and cover the key building blocks before you start.

Core Concepts: What Every Integration Needs

Before you start building, experimenting, and cooking within your Integrations Workspace, let's nail down some basic concepts of how it works and the key ingredients, or "building blocks," to create an Integration Recipe.

A **Recipe** is where you build your integration workflow and configure your integration job. It includes a trigger that initiates one or more series of actions. You can build various if/then statements within your Recipe.

THINK OF IT THIS WAY:

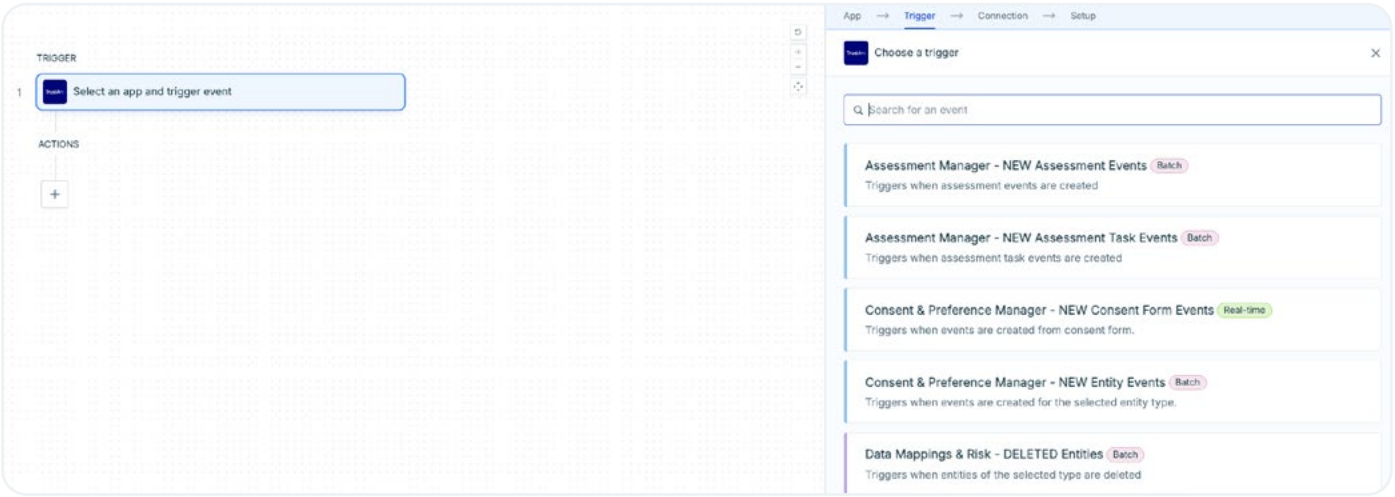
→ A trigger is the "IF" — it specifies the event that starts the Recipe.

→ An action is the "THEN" — it is the step that runs after the trigger fires.

This IF/THEN structure is an easy way to understand the difference between them.

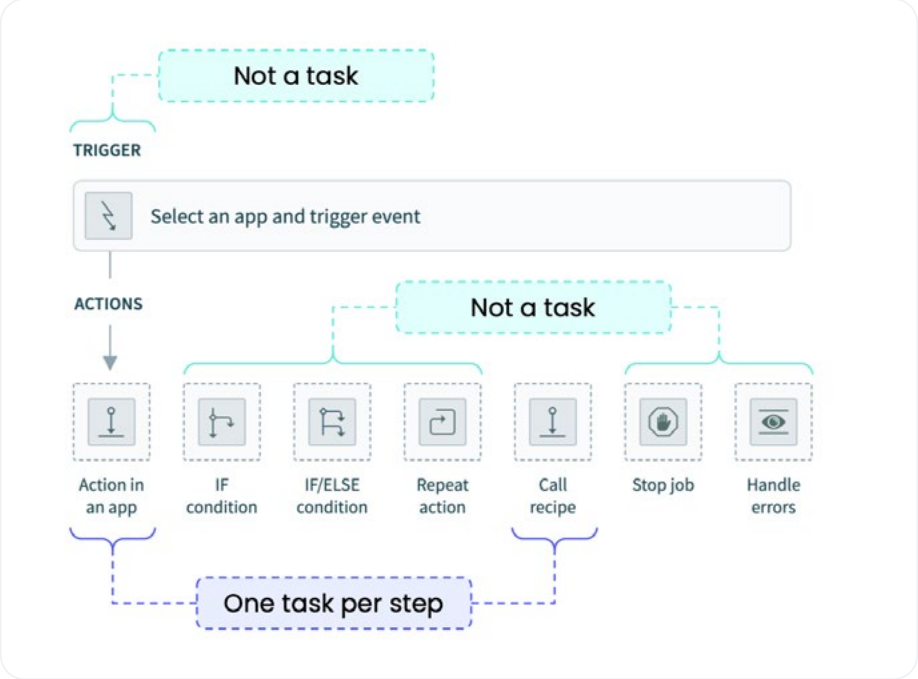
Main "building blocks" of a Recipe includes:

1. **Trigger** = A trigger specifies the event that initiates the action(s) in a Recipe (e.g., on a scheduled basis or a specific event that happens in TrustArc or another system).

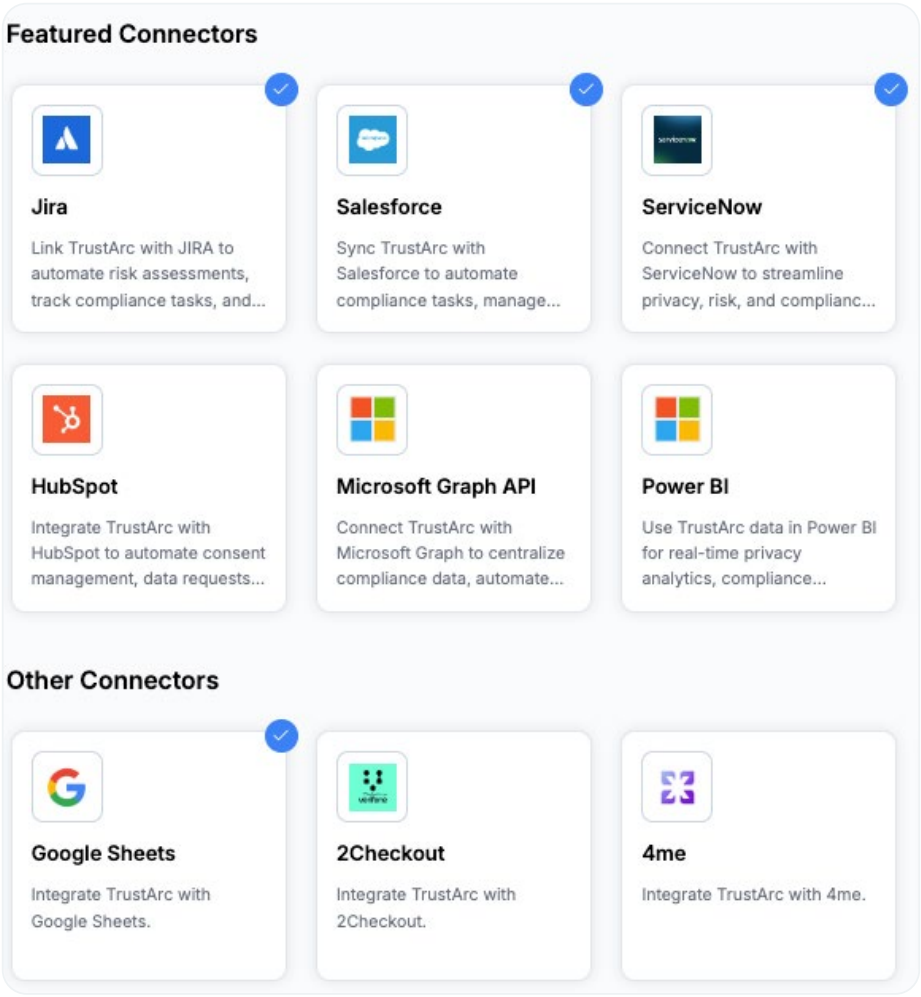


2. **Action** = An action is a single step within a Recipe that performs an operation in a downstream system (e.g., creating a record, updating, performing a search, or triggering an email notification).

3. **Task** = a unit of work whenever a Recipe performs an action requiring compute resources. Each action executed on an individual connector (system) counts as a task. Only successful tasks used count to your billing out. Rerunning jobs are also added to your billing task count. We'll dive into what counts as a task, how to calculate your tasks, and how to optimize task usage in the Recipe Cookbook.



4. **Connector** = a system (e.g., Salesforce, Zendesk)



Each Recipe is made up of these components. Once you know what they are and how they behave, building flows becomes easier.

You won't be tested on this but you'll use it constantly. Now let's head into the workspace where these flows get built.

A Tour of Your Integration Workspace

Start here to build and manage your privacy automations.

1. Go to Integrations

Open the Integrations tab from the main TrustArc navigation panel.

2. Explore Your Workspace Tabs

Inside Integrations, you'll find three main tabs:

- **Connectors**:
  Choose and manage the systems your recipes use.
- **Usage**: Track recipe runs, task usage, and overall performance.
- **Logs**: Review detailed records and troubleshoot errors.

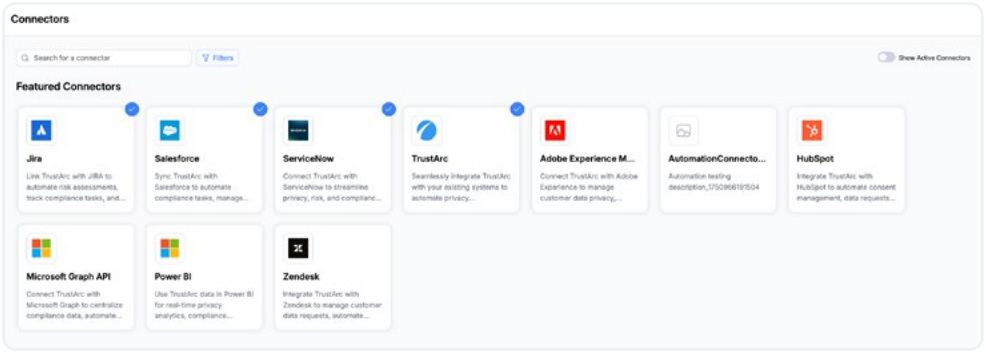
3. Start with Connectors

Connectors is where you select the systems for your automation flows and decide whether to use a prebuilt recipe or start from scratch.

4. Use the Connectors Library

In the Connectors Library, you can:

- View featured systems like Salesforce, Jira, and ServiceNow



- Search across 300+ connectors
- Filter by category or status
- Identify connectors with live recipes (look for the blue checkmark)

5. Click Any Connector to Build

Select a connector to start building or editing your recipes.

Exploring a Connector: TrustArc Example

When you select a connector from the TrustArc Integrations library, you land on a page that organizes everything related to that system.

At the top, you will find two tabs:

1. **Recipe Templates**: Prebuilt automation flows created by TrustArc that you can use or customize.
2. **Saved Recipes**: All Recipes you or your team have built with this connector.

Under **Saved Recipes**, each Recipe is shown as a card with:

- → The systems it connects, displayed with clear icons and a directional flow
- → The Recipe name and current status, such as Inactive or Never Active
- → The number of successful and failed runs
- → A menu that lets you clone, edit, rename, start, or delete the Recipe

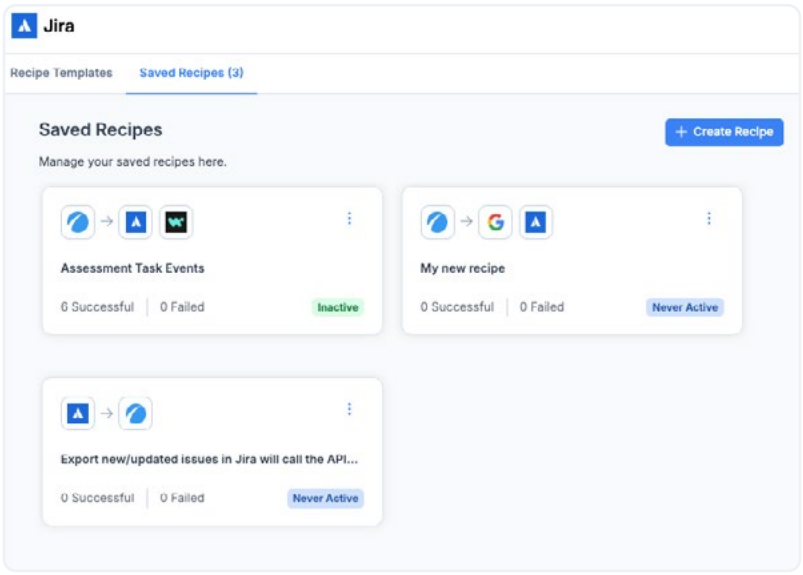
This page lets you keep track of every automation linked to that connector. You can see which Recipes are ready, which ones are still drafts, and which have run successfully. Use Recipe Templates to get started quickly, or manage your existing workflows in Saved Recipes. Everything is organized to make it easy to review, update, and launch the flows that matter most for your privacy program.

Once you've picked a connector, the next step is building your flow.

The Recipe Builder: Where Everything Happens

Your Integration workspace is like a kitchen, and the Recipe Builder is your cutting board. Here is where you lay out the details, design each step, and put your plan into action.

The Recipe Builder is a visual canvas. Each flow starts with a trigger. After that, you sequence actions, add logic, and set up what happens at every stage.



Tip: Download and fill out the TrustArc Workbook before you build. This practical worksheet helps you map out the real design details your Recipe will need. Use the workbook to clarify:

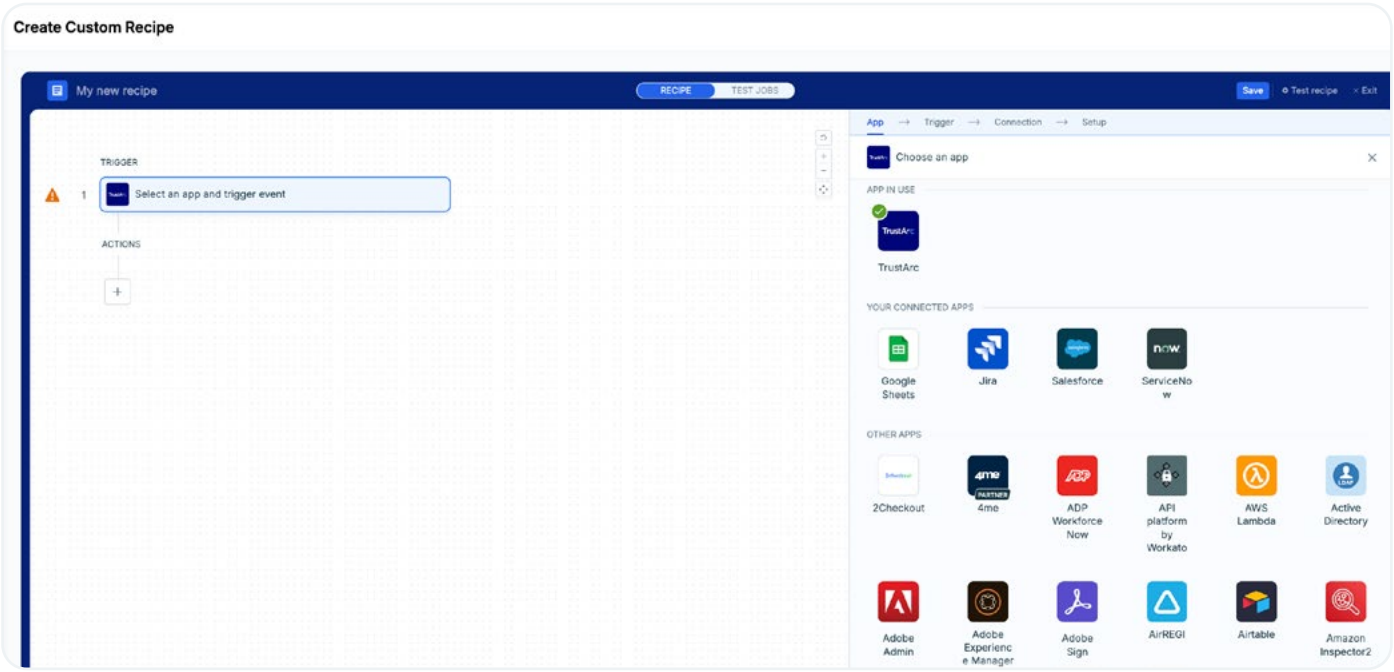
- Which specific fields are required from each system, and their names
- Whether data should be processed in batches or in real time
- What triggers will start your flow (events, schedules, or manual actions)
- The average volume of records or requests you expect for this flow
- Any filters, routing rules, or fallback logic that could shape how the Recipe runs
- Steps for error handling, validation, and notification along the way

Filling out the workbook helps you see what could break, what data you actually need, and how to design each flow to be robust and efficient.

What You'll See in the Builder

Left side: Your Recipe's outline. Each step is numbered — starting with the trigger.

Right side: Your configuration panel. This is where you fill in the "how."



Let's say your Recipe starts with "Add salt."

The right panel answers: *How much? What kind? When?*

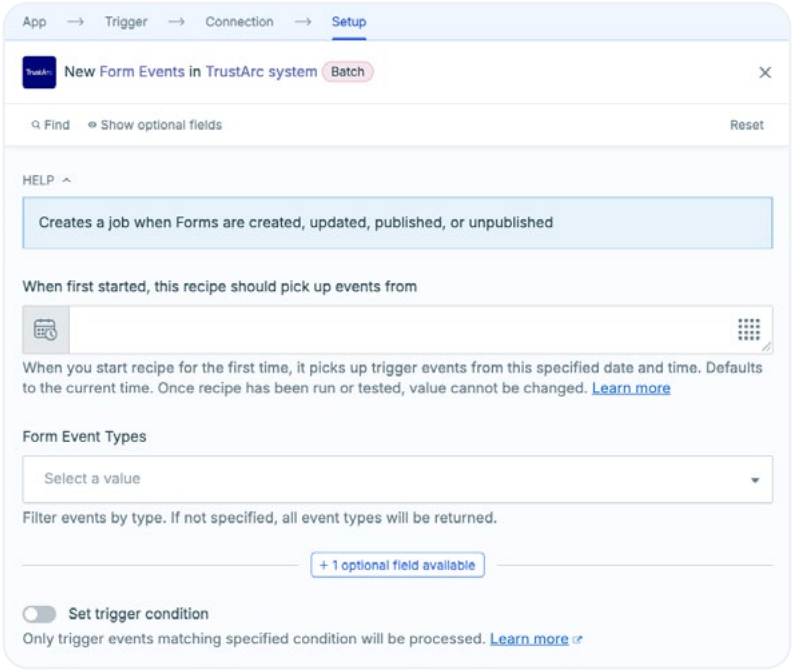
That's what configuration is. Step-by-step, you'll define the details — like which app, what field(s), what logic, and what happens next.

In the builder, you'll choose the systems, assign actions, authenticate the accounts you want to use and add filters — all visually.

Connecting and Authenticating Systems

Before your Recipes can run on any system, each connector — like Salesforce, Snowflake, or Google Workspace, needs to be set up and authenticated. You can draft and save Recipes at any time, but nothing will move data until the connection is complete.

When you add a new system, you'll see a **Setup** prompt. You'll need to enter a connection name, choose a location, pick an authentication method (such as OAuth 2.0), and indicate if you're connecting to a sandbox or production account. Some connectors, like Salesforce, will also ask for your organization's custom domain or advanced access settings.



If you're unsure what credentials or permissions are needed, check with your IT team or system owner. Make sure you have both read and write access for the actions your flows require.

Follow the on-screen steps to authenticate and store credentials according to your organization's security policies.

Tip: If you don't have access yet, just note which Recipes depend on that system and circle back after setup is complete.

Every Recipe follows a clean, structured flow:

- → A trigger
- → An action that happens
- → Where the primary action is performed
- → Subsequent fallback and success actions

You don't need to memorize these. We'll walk through each part in the next section.

It Starts with a Trigger

Before your Recipe does anything, it needs a reason to begin.

That's where triggers come in. A trigger is the starting point. The moment something happens — like a new vendor, a system changing status, or a DSR being submitted — and your flow gets the green light to start.

TYPES OF TRIGGERS

1. **Real-time**: Instant reaction to a change

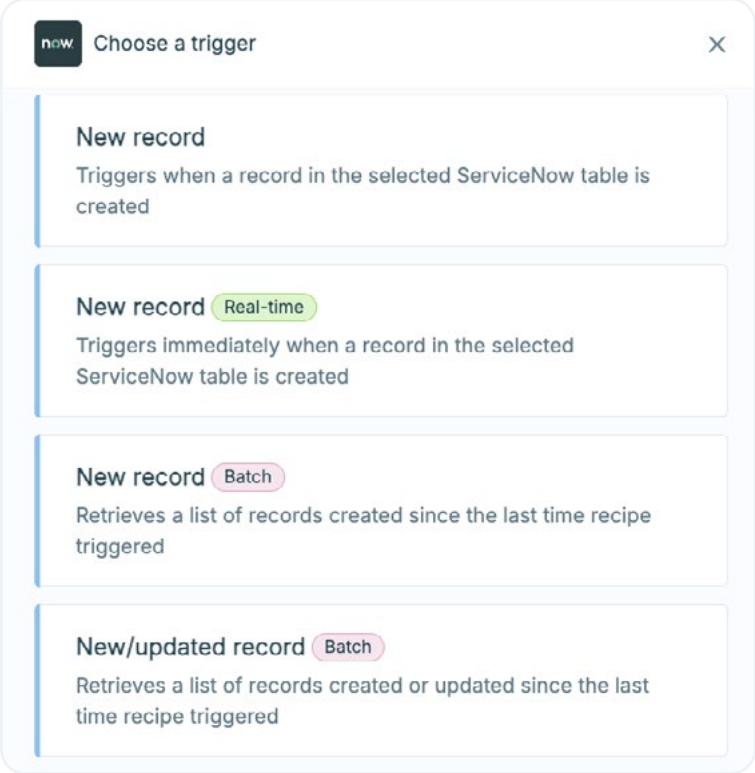
   Example: A DSR request hits your intake form → Recipe starts.

2. **Polling**: Checks at regular intervals

   Example: Every 10 minutes, check if a new system was added.

3. **Scheduled**: Runs on a fixed schedule

   Example: Every Monday at 8 a.m., run a vendor review sync.

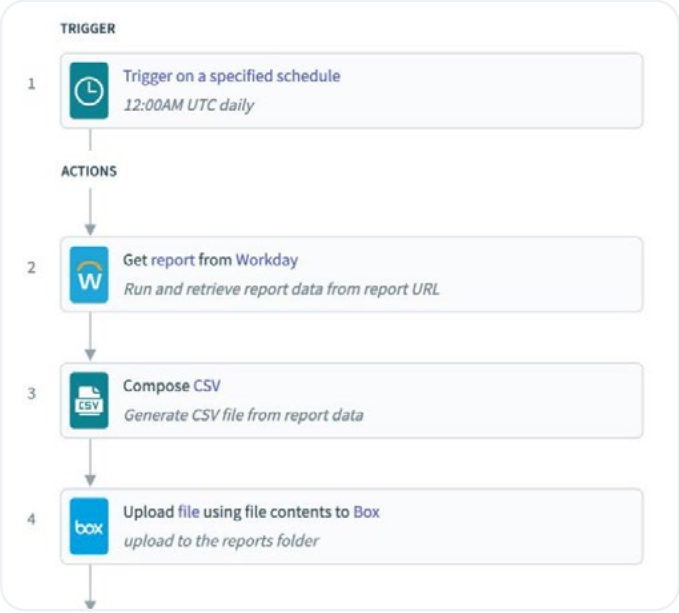


Tip: For heavy data jobs, use **bulk triggers**. These fetch multiple records at once and prevent delays when handling thousands of rows.

After the Trigger: Actions

Actions are what your Recipe does once the trigger fires.

Each one performs a step: **send a message, update a system, copy data, tag a vendor** — whatever the job requires. Actions run in sequence. Step 1 is always your trigger. Step 2 onward, each step is an action.



Here's a simple example above.

- → **Retrieve** a report from Workday → **Compose** it as a CSV → **Upload** it to Box
- → Each step follows the last. This is a **linear Recipe**: one path, no branches. You can add as many actions as needed, and they'll run in the order you set.
- → Each of these counts as a **task**, something we'll unpack more in the Cookbook when we walk through pricing, batching, and optimization.
- → But not every workflow is linear. Sometimes you need the Recipe to make decisions — to follow one path if a condition is met, and a different one if not.
- → That's where **logic** comes in.

Using Conditions to Guide Your Recipes

How "If this, then that" logic drives smart automation

Every useful automation makes decisions. In TrustArc Integrations, you set those decisions using conditions: rules that tell your recipe what to do based on the data it sees.

Conditions follow a clear structure:

- → You choose a field from the data
- → You define a rule (like equals, contains, or starts with)
- → You decide what should happen if the condition is met

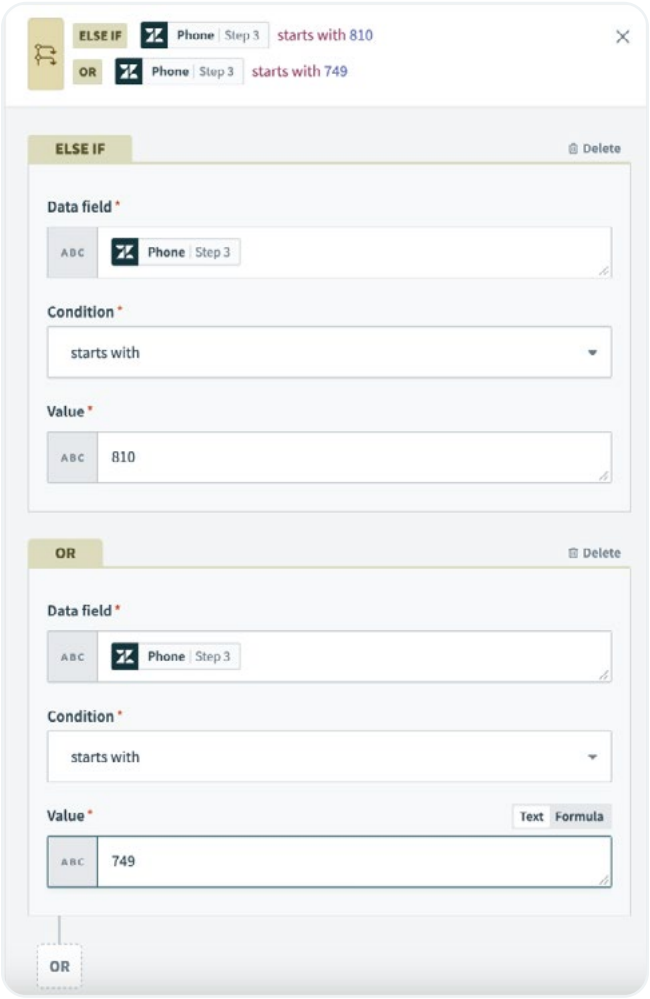
This is how you move from a single flow to a smart one that adapts.

EXAMPLE: ROUTING BASED ON PHONE NUMBER

In the screenshot shown here, the recipe is checking a phone number field.

It asks:

- → Does the number start with **810**?
- → Or does it start with **749**?



If either is true, the recipe takes action. If not, it skips the rest of this branch.

The recipe uses two separate conditions, joined by **OR** logic. This means either rule can qualify. In this case, the ELSE IF group tells the recipe to check both conditions and only proceed if one of them matches.

Real Privacy Example: Syncing Salesforce Vendors to Your Privacy Inventory

Now let's put this into a privacy-specific context using a real integration recipe.

Say your team tracks vendors in Salesforce, but your privacy inventory in TrustArc should only include vendors that are actively in use. You don't want to bring in every vendor, just the ones that are active or engaged.

Here's how that condition might look:

- → If the vendor's status is "Active"
- → Or if the vendor's status is "Engaged"
- → Then sync the record to TrustArc

Once that condition is met, the rest of the recipe takes over:

- → If the vendor **does not** already exist in TrustArc, it creates a new vendor record
- → If the vendor **does** exist, it updates the relevant fields
- → If the status is anything else, it skips the record entirely

This helps you avoid syncing outdated, test, or incomplete vendor records. You keep your inventory current without wasting time or task volume on records that don't matter.

This is one of several inventory-focused recipes we've built into the TrustArc library.

So far, you've seen how recipes decide **what** to do. But what about **how many times** to do it?

When your recipe needs to handle more than one item, like reviewing each DSR or updating vendors one by one, you'll use a loop. We'll walk through exactly how that works in the next section.

Loops: When One Step Isn't Enough

So far, your Recipe runs in a straight line or splits when logic tells it to.

But what if you need to run the same action more than once? Like for every item in a list, or while a condition is still true? That's where loops come in.

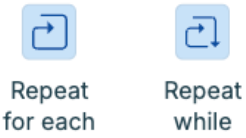
A loop is a way of telling your Recipe: *"Keep going until you've done it for everything, or until something tells you to stop."*

It's not repetition for repetition's sake. It's controlled, purposeful action designed to handle what you can't always count in advance.

THERE ARE TWO TYPES OF LOOPS IN TRUSTARC INTEGRATIONS:

1. **Repeat for Each** — for working through lists
2. **Repeat While** — for continuing while a condition holds

You'll see these in your Recipe builder as a loop symbol.



Let's walk through each one, how it works, and where it applies in privacy workflows.

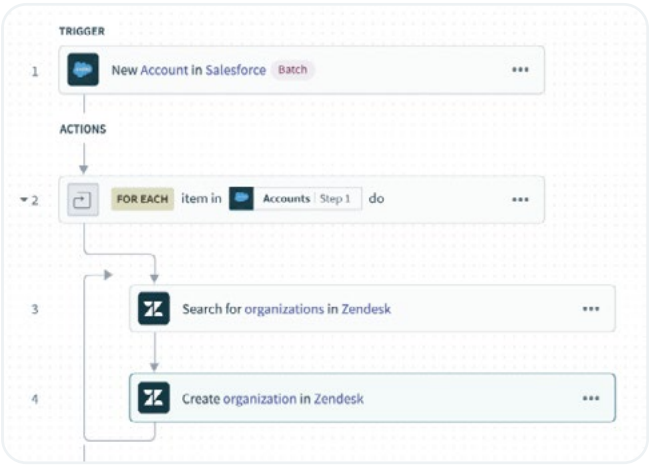
Repeat for Each: When You Have a List

This is the most common loop you'll use. It tells your Recipe: "Do these steps for each item in this list."

That might be a list of vendors, systems, records, or subprocessors. Instead of building the same action 20 times, you build one loop and let it handle every item.

In the image shown here, we see:

- A batch of new accounts is pulled from Salesforce
- For each account:
  - → The Recipe checks if it already exists in Zendesk
  - → If not, it creates it



You're not writing 20 steps. You're writing one structure that runs 20 times.

Privacy-Specific Example: Flagging High-Risk Vendors Across Systems

Let's say your privacy team needs to flag vendors that may present elevated risk, but their risk signals live in different systems:

- **Splunk** captures security events tied to vendor-managed IPs or cloud resources.
- **Salesforce** holds vendor contacts and contract terms.
- **Your privacy platform** (like TrustArc) tracks risk metadata — such as AI usage, cross-border transfers, and processing purpose.

You use a **Repeat for Each** loop to automate vendor-level checks across those sources:

1. **Start by pulling vendor records** from your privacy platform, enriched with contact IDs from Salesforce
2. **For each vendor**, the Recipe:
   - Checks if the vendor uses AI for profiling or decision-making
   - Verifies whether data processing involves cross-border transfers
   - Queries Splunk for any recent incidents involving the vendor
3. **If any of these risk signals are present**, the Recipe:
   - Tags the vendor as "High Risk"
   - Creates a reassessment task in Jira or ServiceNow
   - Notifies the assigned vendor owner

This kind of loop helps your team scale privacy risk management without losing precision. It ensures that every vendor is checked, flagged, and routed for follow-up.

Another Kind of Loop: Repeat While

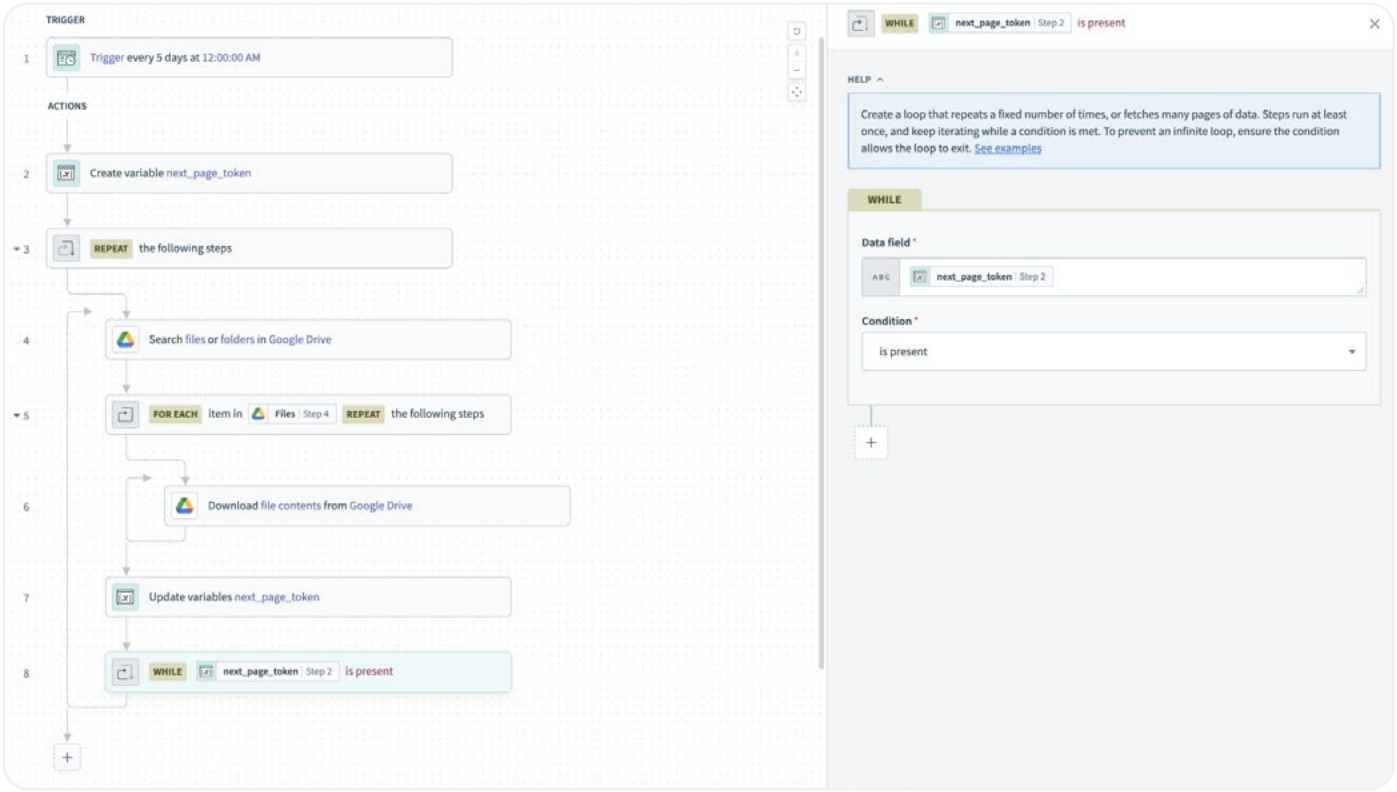
Most loops process a list. But sometimes, you don't have a list — you have a condition.

That's where **Repeat While** comes in.

It tells your Recipe: "Keep going until this condition changes."

YOU'LL USE IT WHEN YOU:

- Fetch results across **multiple pages**
- Retry an action until it **succeeds**
- Repeat something a **fixed number of times**



In the image above, this Recipe checks Google Drive for files, one page at a time. As long as a *next_page_token* is present, the loop keeps running.

HERE'S WHAT'S HAPPENING:

1. It starts by creating a blank variable (*next_page_token*).
2. Then it begins a loop.
3. Inside the loop, it searches Google Drive, downloads content, and updates the token.
4. It checks: *"Is there still a next page to process?"*
   - → If yes, it loops again.
   - → If not, it exists.

You don't need to build five steps for five pages. The loop handles it dynamically.

Where Loops Show Up in Privacy Flows

Most of the prebuilt recipes in TrustArc today focus on one record at a time. That's intentional. We started with the flows that give the biggest lift without needing complexity—like syncing a system, updating a vendor status, or triggering a task when a risk score changes.

But loops are fully supported. And there are real privacy situations where they help.

You'll want to use them when:

- → You're importing large sets of systems or vendors from another platform
- → Your trigger brings in many records that need the same treatment
- → You're fetching paginated data from an external system or report

If you're not sure whether a loop makes sense for your flow, take a moment to think through the volume and structure of your data.

Why Loops Matter

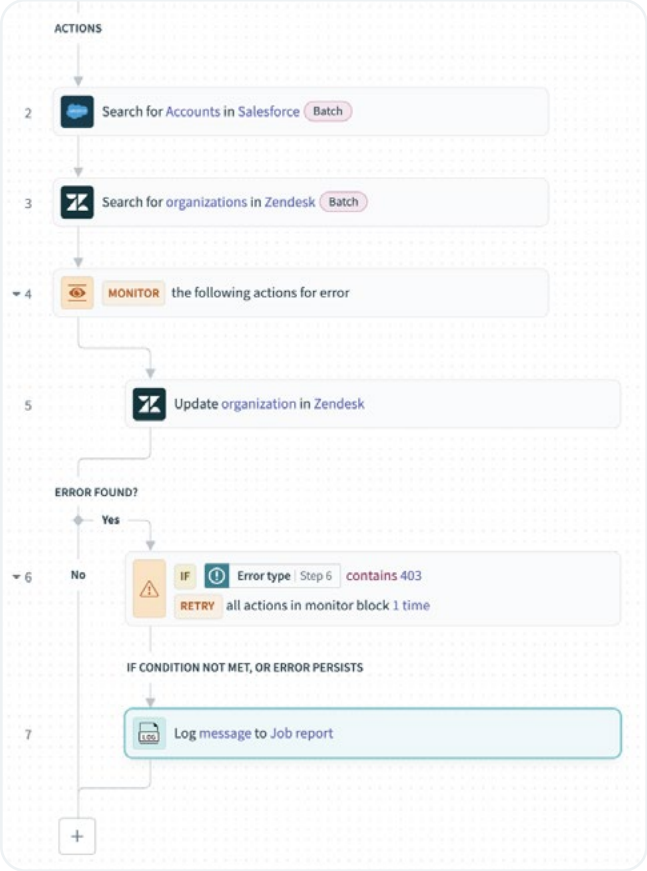
Loops help your automations scale without becoming brittle.

- Repeat for Each lets you work through every item in a list — systems, subprocessors, vendors, or records
- Repeat While lets you keep trying until a goal is met — like completing a sync or pulling every page of data

They eliminate manual rework, reduce oversight, and let your logic run precisely, even when the inputs vary.

Error Handling — What Happens When Something Breaks

Not every automation runs perfectly. Sometimes data is missing, an app is down, or you hit an unexpected error. Good error handling means your workflow keeps moving, or lets you know if something needs attention.



WHAT YOU SEE IN THIS EXAMPLE:

- This Recipe starts by searching for accounts in Salesforce and matching organizations in Zendesk, both in batch.
- Next, it tries to update each organization in Zendesk. If an error happens, the monitor block checks what went wrong.
- If the error is a "403" (often a permissions problem), the Recipe will automatically retry the update one more time. This gives your workflow a chance to recover from temporary issues.
- If the retry still fails, the Recipe logs a message in the job report. You can see exactly which step failed, and what the error was, so you can review or fix it later.

You do not have to guess every possible problem before you start. By adding error handling like this, your Recipes stay reliable and clear, and you stay in control when something goes off track.

What You Can Do

TrustArc Integrations support built-in error handling patterns. You can:

- → **Monitor** critical steps for errors
- → **Retry** only when it makes sense
- → **Stop** the Recipe if key data is missing
- → **Alert** your team with context
- → **Fallback** to another action if needed

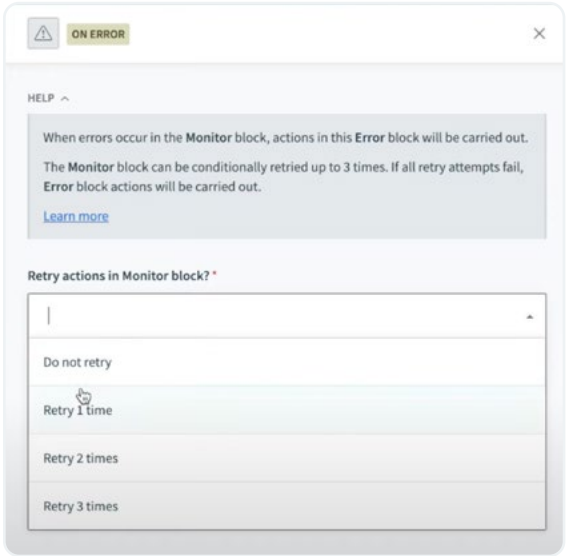
You decide what happens next. The logic stays clean.

Best Practices

Keep error handling lightweight but effective:

- → Wrap important steps in a **Monitor block**

- → Use **Retry** when the error might resolve (e.g., network issues)



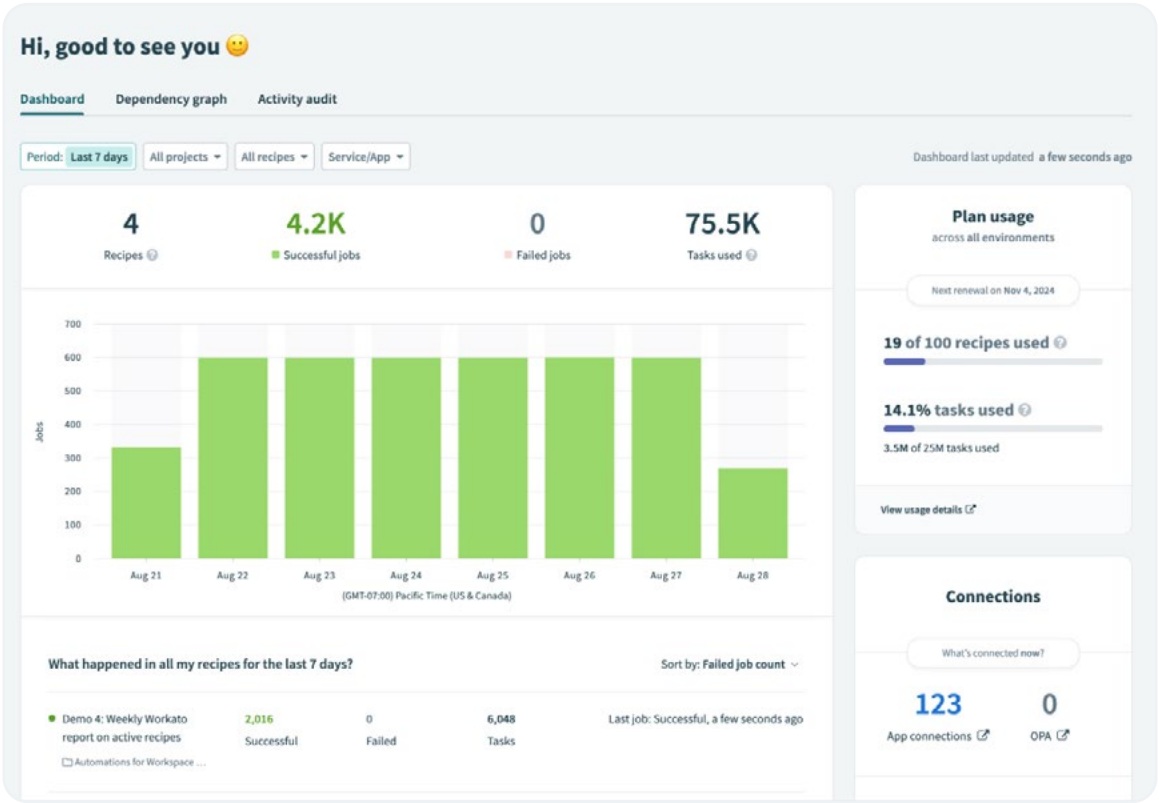
- → Use **IF + Stop** when critical data is missing
- → Send **alerts** only when someone needs to act

You'll learn more advanced patterns in the The Privacy Automation Cookbook.

Usage: Your Operational Command Center

You've learned how Recipes work. Now let's talk about how to track them — clearly and simply.

Click the **Usage** tab under TrustArc Integrations. This is where your operations come into focus.



You'll see a real-time view of everything running behind the scenes:

- → Which **Recipes** are active
- → How many **jobs** have run
- → How many **tasks** were used
- → What's consuming the most time and volume

Why It Matters

Usage helps you answer three key questions:

1. Are our automations working as expected?
2. What Recipes are costing us the most?
3. Where can we streamline?

Whether you're checking a spike, tracking a new flow, or just keeping tabs on capacity, this is your visibility layer.

WHAT YOU CAN DO HERE

- View Recipe usage by project, connector, or time window
- Track how many tasks you've used against your plan
- Spot trends over time with charts and filters
- Dig into job success or failure and see what ran, when, and why

Tip: Each "job" is a run of a Recipe. Each job consumes one or more tasks. Tasks are what count toward your plan.

Where to Go From Here

You've got the basics. Now you're ready to build.

You are not going to become an expert overnight. But now you know how to:

- → Pick a connector
- → Launch or edit a Recipe
- → Understand the flow of triggers, actions, logic, and loops
- → Track jobs, tasks, and usage in real time

Start with the Cookbook

If you want to build smarter, the Privacy Automation Cookbook is your next step. It's designed to help privacy teams understand how tasks work, how to structure recipes efficiently, and how to reason through logic without getting lost in complexity.

Inside the Cookbook, you'll find:

- → Step-by-step breakdowns of filters, loops, and control flow
- → A detailed guide to how tasks are counted, tracked, and optimized
- → Prompts to help you plan your own automations with confidence
- → Mental models that make complex logic easier to build and maintain

If you've ever asked "how should I structure this recipe?" or "how do I avoid task waste?", the Cookbook is where to look next.

Or Try It Yourself

If your account is active, you can try building in the platform right away through a self-serve free trial. You'll get access to:

- → A library of **prebuilt recipe templates** inside the builder
- → All featured connectors
- → The full drag-and-drop recipe editor
- → A real-time task usage dashboard

Start with something simple and familiar, like syncing vendor records from Salesforce or triggering a ServiceNow task when residual risk scores change. These are high-value jobs that are easy to test and adjust.

Prebuilt templates in the builder can be cloned, adapted, or used for reference. Every task your recipe uses is tracked automatically, so you'll see exactly what's running and why.

Want to see TrustArc Integrations in action?

Watch a live walkthrough of real privacy workflows—including vendor risk reviews, inventory management, assessments, and data subject requests. See exactly how privacy teams automate and connect the systems they use every day.

Book a demo

About TrustArc

As the leader in data privacy, TrustArc automates and simplifies the creation of end-to-end privacy management programs for global organizations. TrustArc is the only company to deliver the depth of privacy intelligence, coupled with the complete platform automation, that is essential for the growing number of privacy regulations in an ever-changing digital world. Headquartered in San Francisco, and backed by a global team across the Americas, Europe, and Asia, TrustArc helps customers worldwide demonstrate compliance, minimize risk, and build trust. For additional information visit TrustArc.com.

TrustArc