ASCEND.IO

What Is Data Pipeline Automation?

WHAT IS DATA PIPELINE AUTOMATION?

Theoretically, data and analytics should be the backbones of decision-making in business. Like mitochondria power a cell, data powers a business.

But for most companies, that's not the reality.

Today, there are no intelligent systems that deliver data at the pace, and with the impact, leaders need to power the business. The processes to consume and transform data are ad-hoc and manual, and the costs are unjustified. As a result, stakeholders limit their reliance on data, making decisions based on gut instinct rather than facts.

To shift this harmful approach, companies need to make fundamental changes to their data engineering function and start running at speed and with agility.

These engineering functions are almost exclusively concerned with data pipelines, spanning ingestion, transformation, orchestration, and observation — all the way to data product delivery to the business tools and downstream applications.

Data teams need a new technology that will empower them to focus on business value rather than fixing code holding together a patchwork of point solutions. Data pipeline automation has the power to not only meet leadership demands but make order-of-magnitude improvements to an organization's productivity and capability.
The time is now to adopt this shift and take a position that redefines how your company competes in the marketplace.

IN THIS GUIDE, WE WILL:

- Review the demands of becoming a data-driven business
- Study examples in the history of automation
- Demonstrate the immediate opportunity to automate data pipelines
- Outline the specific new capabilities that define data pipeline automation

MOUNTING PRESSURE FROM LEADERSHIP

Executives are increasingly taking note of the ability of their engineering organizations to deliver the data and insights they need to steer the business, and are raising the bar with specific demands critical to competing in the marketplace:

1. **Teams need to build data pipelines faster.**
2. **Changes and fixes need to happen in minutes, not days.**
3. **Pipelines need to grow faster than the cost to run them.**
4. **Pipelines need to use underlying compute and storage intelligently.**
5. **Technology needs to be accessible to a much broader user base.**
6. **New hire ramp time needs to be quicker.**

But that same C-suite doesn't see the desperation of data engineering teams struggling to keep their data systems running. What they see is the overall engineering budget skyrocketing while speed, quality, and ROI suffer. And that leaves them with a choice.

Stick with the status quo, fail to hit business goals, and put the business at risk, or get and stay ahead of the competition with automation. Increasingly, it is executives who finally escalate the need for automation.

A BRIEF HISTORY OF AUTOMATION

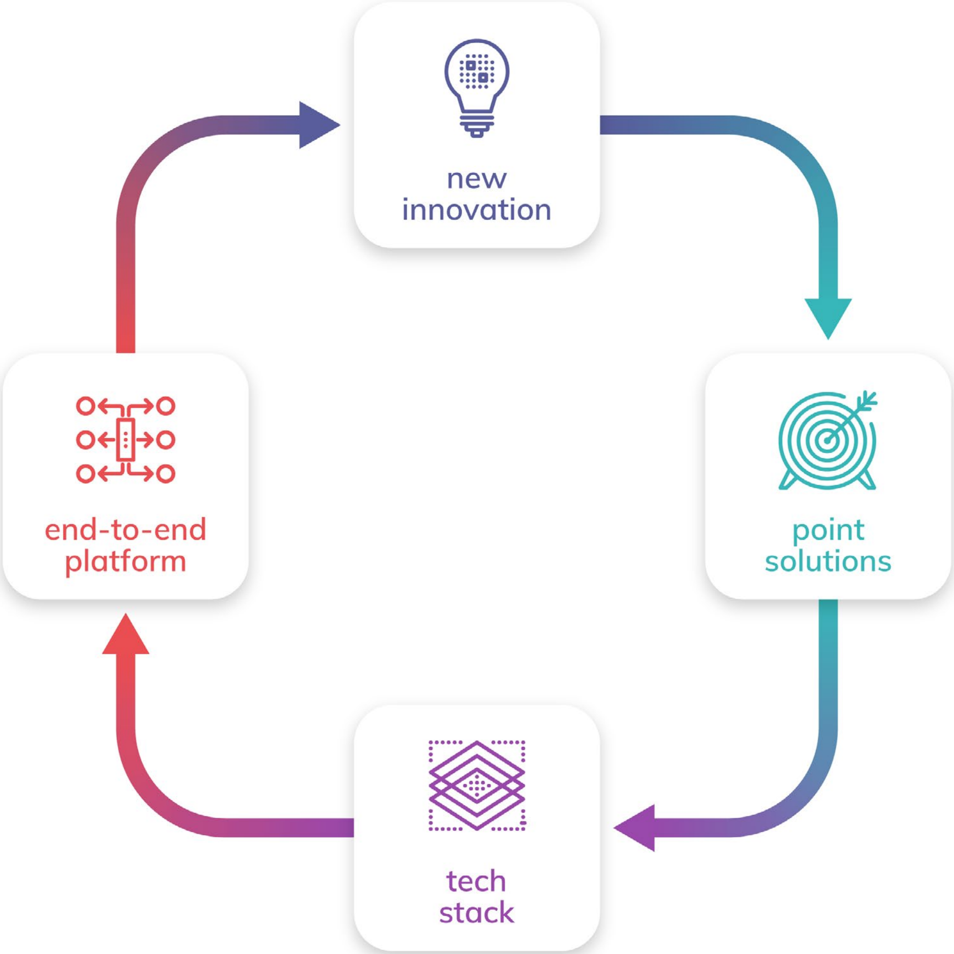
Although the term "automation" has gotten a lot of coverage in the past few years, the idea isn't new.

In fact, it's been a recurring theme in software engineering. When a new innovation triggers a wave of product development, startups and large innovators race to build point solutions. But at that time, the value chain itself is not understood.

It is through the practical use of these point solutions that the key areas of value crystallize. Winning solutions surface to fit individual stages of the value chain, and engineers assemble all the working pieces to form the "perfect" tech stack — driving consensus on common architectures and their necessary components.

But over time, this assortment of tools lacks efficiency, leading to another, more revolutionary wave of solutions that span the entire value chain and introduce end-to-end automation. In every case, these solutions were based on new types of controllers that unlock productivity and scalability that was not previously possible.

To illustrate, let's review three historical examples of this cyclical pattern: RPA, React, and Kubernetes.



Robotic Process Automation (RPA)

Problem:
Back in 2000, many companies — from manufacturing to healthcare — noticed that certain workflows were extremely manual and highly repetitive. Activities like data entry, document management, and payroll were time-consuming and prone to mistakes. Scaling these workflows meant hiring lots of people to do the same thing.

Solution:
As companies realized that paying for that talent (and those mistakes) was becoming prohibitively expensive, the first machine learning techniques were being developed. Entrepreneurs saw an opportunity to leverage ML (and later, AI) to orchestrate repetitive workflows like invoicing, payroll, and inventory management faster and more accurately than humans could — significantly reducing errors and staffing costs. Robotic Process Automation (RPA) software was born.

Results:
Leaders caught on fast, adopting RPA platforms like UiPath, Automation Anywhere, and Blue Prism to take care of routine tasks. The talent in these areas was freed up to focus on customer satisfaction and solving outlier problems that AI and ML couldn't. Today, RPA is heavily used in telecommunications, insurance, banking, finance, and healthcare companies to streamline their reconciliation, invoicing, and governance processes.

React

Problem:
When websites started becoming mainstream, companies embraced them as a new opportunity to reach their customers. To enhance the experience, they began adding more and more features to their sites. But that caused a major problem: the back-and-forth loading of ever larger payloads between browsers and servers exploded, becoming expensive and slow.

This was especially poignant for web developers at companies like Facebook, which were rapidly adding ever-increasing sophistication to their user interfaces and trying to manage steady increases in traffic. For them, a laggy site wasn't an option — users wouldn't stick around for long. On top of that, the incremental cost of serving web pages to each additional user exceeded the network value they generated.

Solution:
Facebook decided to tackle this problem head-on. Their engineers discovered that the same Javascript code was being sent thousands of times to each individual browser or app, creating redundant traffic and blocking faster performance. What if they could automate some of the process?

Facebook's software engineers began turning their code into reusable components they could modify to fit specific requirements. Developers could then render each component dynamically — only loading the part the user cared about, not the entire page. The team released their automated solution as the open-source framework, React. This new engine revolutionized the way web servers, apps, and browsers communicate.

Results:
React drastically reduced design time and load time, while boosting application performance. For these reasons, React is the primary web development language today.

Kubernetes

Problem:
As cloud applications arose, the infrastructure of virtual machines became more and more similar. Yet, the administration of this infrastructure was still manual, only scaling with the addition of more humans performing repetitive, error-prone tasks. To remove this layer of friction, software engineers turned to containerization as their preferred deployment model.

Containers sequester memory, CPU, processing space, and file systems from the underlying hardware to securely power applications. Since they are decoupled from a company's architecture, they can move data from OS to OS and from cloud to cloud.

The hitch: container deployments were a hassle. The thousands of variations in which resources could be combined to securely and efficiently power applications became a configuration nightmare. Orchestration not only took time, it was complex. If one container went down, another needed to start immediately. At scale, containers were becoming too unwieldy.

Solution:
As one of the largest operators of data centers at the time, Google was dealing with this problem on a previously unimaginable scale. So a team of three Googlers took on the many problems with containerization. They committed to creating an automated solution that would:

- Allow them to deploy multiple instances of one application
- Schedule deployments and let them roll failed ones back
- Balance usage loads
- Restart containers that fail, and replace or kill the ones that don't respond
- Store confidential information, like passwords and tokens

The Google team built and began using this new, massively scalable framework internally to great success. Eventually, Google released it to the public as the open-source platform, Kubernetes. The open-source community immediately adopted Kubernetes and contributed greatly over the years.

Results:
The leverage provided by Kubernetes automation is profound. It no longer takes thousands of system administrators to manage the infrastructure for vast applications like Spotify, Asana, and The New York Times. Put bluntly, Kubernetes automation keeps the world's applications running. Today, 3.9 million Kubernetes developers around the world continue to deepen the capabilities of the platform.

AUTOMATION YIELDS HIGH-LEVERAGE CAPABILITIES

In the examples above, automation based on a new type of controller designed for each domain was the key to unlocking unprecedented efficiency.

RPA automated manual tasks, so employees can spend more time with customers. React eliminated significant coding work and shrank website payloads, helping developers get optimized websites up and running quicker. And Kubernetes streamlined container deployment and infrastructure management so product teams could release features faster.

But added efficiency isn't the only result of automation. It also sets companies up to make order-of-magnitude improvements in productivity and capability.

Because RPA solved businesses-as-usual workflows, it also allowed those businesses to innovate. With the extra time employees had, they could focus on understanding their customers better, achieving more strategic objectives, and dreaming up new initiatives they never had a chance to think about before.

React gave web developers the tools to build highly-optimized sites, but the framework also opened up a whole new world of possibilities — developers could get more creative, finding new ways to increase brand presence and dwell time on their sites.

And Kubernetes simplified DevOps and infrastructure management so software engineers could focus on producing features that set their products apart from the competition.

AUTOMATION FOR DATA PIPELINES

So what is data pipeline automation, and how does it work?

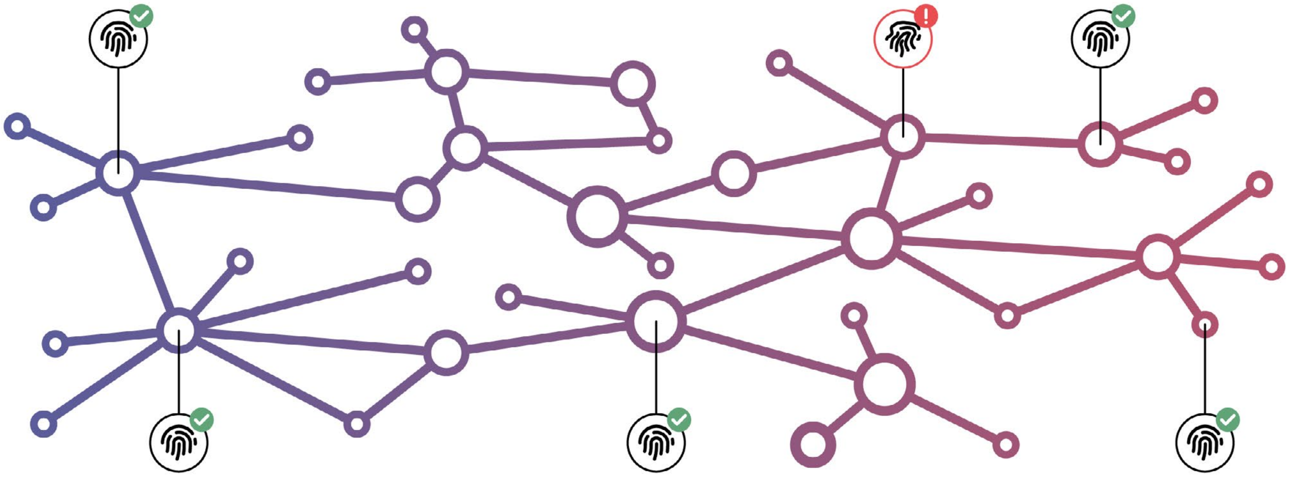
Similar to the consolidation of tools in previous waves of automation, data pipeline automation replaces data stacks that have been assembled from multiple tools and platforms. For data pipeline automation, these previous approaches have foundered on one important barrier: the need to scale.

The Key to Unlock Data Pipeline Automation

It turns out that the key for a controller to break through the scaling barrier is to (1) utilize an immutable metadata model of every aspect of the pipelines and (2) automate every operation with it. This can be done with unique digital fingerprints that map not just every snippet of data, but the data engineers' code as well.

Such a controller can then be programmed specifically for end-to-end data pipeline operations, at near-infinite scale. With the ability to track pipeline state in networks at vast scale, the controller can always know the exact state of every node in every pipeline with certainty. It can constantly detect changes in data and code across the most complex data pipelines, and respond to those changes in real-time.

These fingerprint linkages assure that all dependent pipelines maintain data integrity and availability for all users of the data. For data teams, scalable technology becomes a vehicle for managing change throughout an organization.



DATA PIPELINE AUTOMATION CAPABILITIES

So what capabilities should such a massively scalable data pipeline controller exhibit?

The domain of data pipeline automation is uniquely challenging in that:

- Data is “heavy”, meaning it is costly to move and even more costly to process
- Data in an enterprise has thousands of sources, each of which is well-defined
- Data-driven business outcomes are well understood, but hard to achieve
- The space between sources and outcomes is chaotic and poorly understood

The automation capabilities needed to close this gap should achieve new levels of efficiency, but perhaps more importantly, propel businesses forward with greater productivity and business confidence. Let’s outline what these capabilities should look like.

Automatic Propagation of Change

Definition
At the heart of data pipeline automation is the ability to propagate change through the entire network of code and data that make up a pipeline. A data pipeline automation solution should be instantly aware of any change, in the code or in the arriving data. It should then automatically propagate the change downstream on behalf of the developer, so it is reflected everywhere.

Value
As your network of pipelines increases, this capability alone will save highly skilled technologists days of mundane work assessing and managing even the simplest of changes.

Extensions
When data pipelines are chained together, changes propagate automatically throughout the network. This technique eliminates redundant business logic and reduces processing costs for the whole system. When resources are limited, pipeline automation provides controls to prioritize the pipelines that matter most.

It also provides continuity through different types of failures. Automated retry heuristics should ride through cloud, data cloud, and application failures to reduce administrative human intervention and minimize downtime.

Different clouds and data clouds have different strengths and weaknesses, which large enterprises are keen to harness. Data pipelines should be able to span across these resources, using automation to propagate changes through them seamlessly.

Real-time Operational Visibility

Definition
Data pipeline automation includes a single pane of glass spanning all data pipelines, in all environments, across all teams and resources. Intuitive visualization of the interconnectedness and the real-time states of automated data pipelines is critical.

Value
Automation with a "single pane of glass" breaks down silos across teams and dramatically reduces time to resolution for problems. Aligned on a single source of truth, engineers become more productive.

Extensions
Visibility should reach into every detail of data pipeline operations so engineers can validate data pipeline logic, see operational reports, instantly pinpoint problems, communicate them easily across teams, and accelerate issue resolution.

Subject matter experts and business stakeholders should be able to participate in pipeline logic validation and assume data steward roles to raise company-wide confidence in data quality.

Guarantee Data Integrity

The integrity of data pipelines from source to sink is the foundation for confident decision-making.

Definition
A data pipeline automation controller actively maintains lineage integrity, instantly and precisely pinpoints problems, and automatically notifies the business to hold off data-driven decisions until the problem is fixed.

Value
A controller that can guarantee integrity results in high business confidence. It enables higher pipeline availability and lower monitoring costs. Knowing where and when pipelines break the moment it happens drives timely resolution.

Extensions
After repair, pipeline automation should be able to restart easily, and eliminate the need for manual cleanup, such as rollbacks, tracing of impacted lineage, or orphaned dataset cleanup. These repetitive activities are significant sources of error and delay that undermine data integrity.

Data pipeline automation assures strict adherence to the lineage that maps data through transformation logic from node to node. A historical record of every dataset and processing step should allow data engineers to reconstruct and navigate the "genealogy" of every dataset from its origin to its business-driven destination.

Empower Human Intervention

Definition
Intelligent data pipeline automation makes it easy and efficient for engineers to interact with development and production pipelines, eliminating the cleanup, busy work invoked with every human interaction.

Value
Unlike black box systems, this capability helps engineers constantly verify and grow their trust in automation. Since their intervention incurs no overhead or long restart delays, they are far more likely to be proactive and nip early problems in the bud before they become costly. In addition, corrections incur no additional costs, with no reruns or cleanups.

Extensions
With data pipeline automation, engineers can pause, inspect, and resume pipelines at any moment anywhere, meaning they can diagnose pipeline logic and investigate complex issues — even if there is no obvious failure. Pausing a specific transformation step should automatically ripple through and pause steps that are downstream. Any unimpacted pipelines should continue to run freely, limiting business disruption.

Resuming a pipeline from a pause should resume where it left off, with zero impact on the system, and zero reruns of any part of any pipeline. If an engineer made a change to the code, this change would appear in the fingerprint, and be processed automatically as part of the Automatic Propagation of Change capability.

When a data pipeline is paused, the automation should instantly notify users downstream of the paused step, so they can make an informed decision on whether to use the data they already have, or wait for the system to resume. The automation should notify users again when the pipeline has resumed normal operation.

Enforce Rules During Processing

Definition
Data pipeline automation automatically assesses data in the pipelines against quality rules in real-time. Data quality rules, or assertions, should be configurable at every processing step in every pipeline and evaluate every data record.

Value
Catching data problems in real-time avoids costly reruns and delays. It reduces the amount of rework arising from after-the-fact quality reviews and simplifies data workflows.

Extensions
The moment a record does not meet data quality assertions, the system should automatically take action according to pre-configured rules. Actions can include stopping or continuing processing and alerting engineers. Stopping processing should be limited to a specific node and managed by the Guarantee Data Integrity capability. All other pipelines continue to operate as part of the Automatic Propagation of Change capability.

Users should be able to run data quality reports at any time, to monitor and proactively identify areas that may pose a problem in the future.

Optimize Production Costs

Definition

All the capabilities of data pipeline automation inherently reduce costs wherever possible, and reduce all redundancy in data processing.

Value

Companies can save hundreds of thousands of dollars in compute costs with these techniques, and significantly raise the value returned for compute costs incurred.

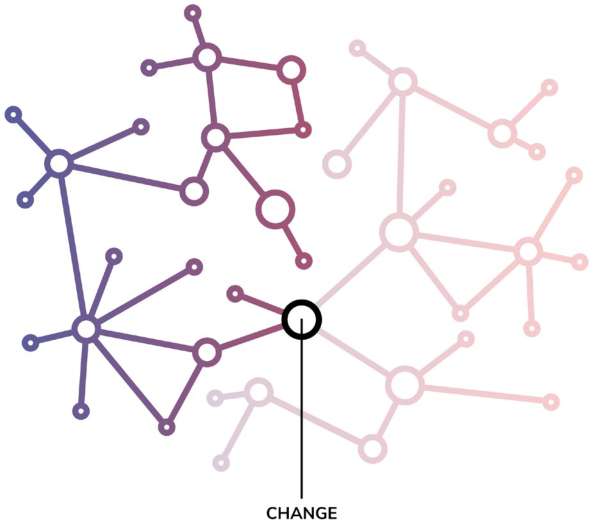
Extensions

The Automatic Propagation of Change capability should apply to every dataset, processing each individually and incrementally. For example, the fingerprinting technique identifies specific datasets that are affected whenever the code of a single processing step is changed and ensures that only affected downstream datasets are reprocessed.

The break-fix features that support the Guaranteed Data Integrity capability should instantly pinpoint points of failure in the data pipelines. When engineers resolve the problem and the system resumes via the Automatic Propagation of Change capability, it automatically monitors the datasets generated by new code. When it detects that the data in a subsequent processing step matches the data that was previously generated at that step, the processing stops, and their lineage is merged. This technique reuses as much existing data as possible and eliminates useless reprocessing costs.

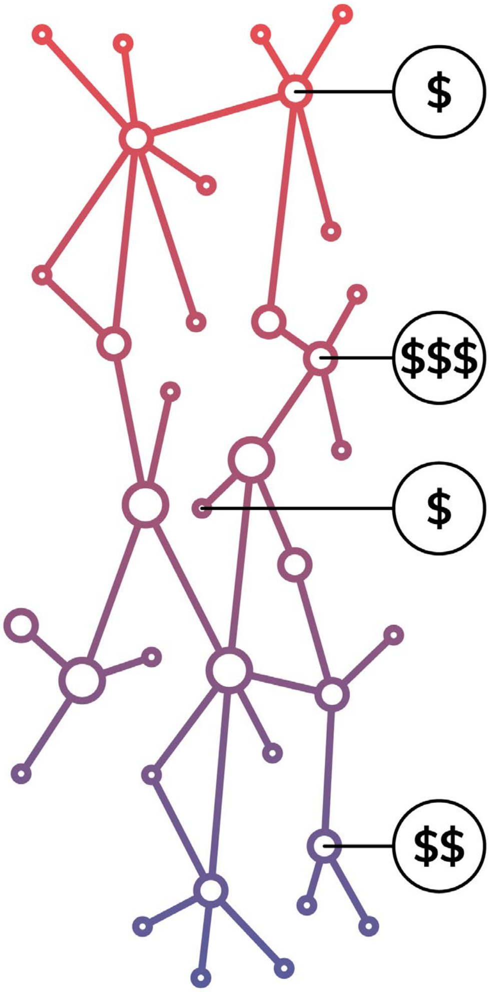
Sometimes an engineer addresses a logic problem by updating the code in a processing step. But if they realize the update was incorrect or incomplete, they need to roll back the change. In this common scenario, the Ease Human Intervention capabilities of data pipeline automation should harness Automatic Propagation of Change capabilities to perform an instant roll-back, recovering all datasets and their lineage with zero reprocessing. Without this technique, engineering teams lose days of productivity as they painstakingly locate and manually restore untracked datasets or rerun huge datasets to create new baselines for their pipelines.

In the background of all these capabilities, data pipeline automation should track every dataset and its lineage, and automatically delete or archive orphaned datasets. Removing them eliminates these sources of confusion and costs while making all history available to engineers for analysis.



Quantify Product Costs

Definition
As data pipelines are processed step by step, data pipeline automation sends workloads to the data clouds that customers have selected for that purpose. The system monitors the processing of each workload for resource utilization and shows the total cost of producing datasets at the end of each pipeline.



Value
All levels of the organization can easily assess the complete financial costs of data production — from source to report — enabling cost attribution for cross-charging and cost analysis for production optimization.

Such financial accountability is also an essential element of shifting into a data product approach that aligns data teams with the business. Engineers can better predict the cost of a data product as they speak with internal stakeholders, identify the savings of reusing existing data pipelines, and weigh the pros and cons of building new ones.

Extensions
By showing workload costs with Real-time Operational Visibility, data pipeline automation helps engineers monitor and pinpoint problems that arise from the workloads in the data clouds. Over time, detecting repeated spikes in processing costs can uncover pipelines that are candidates for refactoring.

Data pipeline automation should send pipeline-level processing cost information to logging and reporting systems, where data pipeline costs can be tied back to financial business benefit and profit.

THE IMPACT OF DATA PIPELINE AUTOMATION

Just like RPA, React, and Kubernetes revolutionized the productivity of business and software processes, early adopters of data pipeline automation are unlocking profound value for their businesses. This is already changing the competitive landscape of several sectors, ranging from healthcare to advertising and to financial services.

As enterprises adopt and realize the benefits of data pipeline automation, they also realize additional advantages that are particularly important to the C-suite:

Accelerate engineering velocity

When the team is no longer worrying about debugging vast libraries of code or tracing data lineage through obscure system logs, speed of delivery increases exponentially. Engineers also gain the capacity to shift into higher-order thinking to solve data problems in conjunction with business stakeholders.

Ease the hiring crunch

Enabled by a comprehensive set of data automation capabilities, companies no longer need to hire hard-to-find esoteric skill sets. Anyone familiar with SQL or Python can design, build, and troubleshoot data pipelines, making data far more approachable and making data engineering teams more affordable and nimble.

Cost reduction in data tools

When data automation is purchased as a single end-to-end platform, data engineering teams can reduce software costs from dozens of point solutions. They also realize dramatic savings in engineering time as engineers shift focus to creating data pipelines rather than maintaining an in-house platform.

ASCEND.IO

AUTOMATE TO SURVIVE

Ascend is the leader in data pipeline automation, building the world's most intelligent data pipelines. We provide a single platform with intelligence to detect and propagate change across your ecosystem, ensure data accuracy, and quantify the cost of data products.

Learn more at Ascend.io or follow us @ascend_io.