2025

ASCEND.IO

Agentic Data Engineering:

Beyond the Modern Data Stack

Redefining data engineering with intelligent agents that understand context, act autonomously, and unlock strategic value.

7x
Boost in team productivity

83%
Reduction in cloud costs

87%
Faster data processing speed

Executive Summary

Data teams across industries face an unprecedented capacity crisis. Despite significant investments in modern data stacks, 95% of data practitioners report operating at or beyond capacity limits, spending half their time on maintenance rather than delivering value. The root cause isn't insufficient tooling—it's tool fragmentation. Organizations now use 5-7 different data tools from 3-5 vendors, creating new metadata silos that worsen the problems modern data stacks were meant to solve.

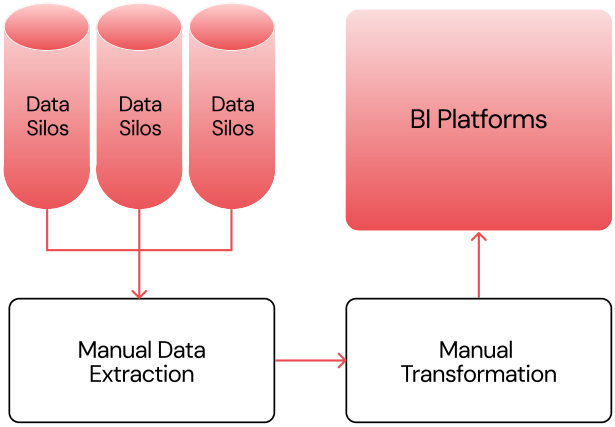
Agentic Data Engineering (ADE) represents the next evolutionary step beyond the modern data stack. By integrating AI agents directly into data workflows, ADE platforms enable intelligent automation that doesn't just execute predefined rules; it reasons, acts, and adapts to changing conditions. Early adopters are already seeing 7x productivity improvements through agent-assisted pipeline development, automated error resolution, and intelligent metadata management.

To understand why ADE represents such a fundamental shift, it's essential to trace how we reached this crisis point and examine the architectural requirements that have made truly intelligent data operations possible for the first time.

Index

The Evolution to Crisis: From Data Silos to Tool Fragmentation

Data Silos Obstruct Data Sharing



Since organizations began collecting digital data, it's been trapped in the systems where it was created mainframes, cloud CRMs, or on prem ERPs. Early data management analyzed each silo independently, and combining data for insights required heavy engineering.

BI systems could display data, but extracting and transforming it across silos demanded specialized skills. Engineers wrote custom ETL scripts or manually merged data, making requests take days, weeks, or months.

Pipelines required constant maintenance. Changes to structures or dependencies added work, leaving engineers fixing problems instead of building new pipelines.
Data remained inaccessible except for high priority use cases. Lower priority analytics that could drive improvements were too costly.

A bigger issue was metadata management. Metadata—data about the data—provides context but was tracked manually, hardcoded in ETL pipelines, or sometimes only in engineers' heads.

Legacy practices made lineage tracking, observability, and auditing impossible. Limited sharing meant metadata scalability wasn't prioritized. As data grows, metadata becomes critical, and traditional approaches can't meet modern needs.

The Promise of the Modern Data Stack

- The mounting challenges with manual data integration and metadata management, combined with explosive growth in data volumes and business demands for faster insights, paved the way for the modern data stack.

- Three technological breakthroughs enabled the modern data stack. Cloud infrastructure delivered unlimited scale and storage. Columnar data warehouses like Snowflake and BigQuery handled massive analytical workloads. ETL/ELT tools became accessible to non-specialists, eliminating the need for armies of specialized developers.

- Cloud native architectures enabled the industry to shift from monolithic, hand-coded solutions to modularity and specialization. Purpose-built tools emerged for each stage of the data lifecycle, with stand-alone solutions emerging for data ingestion, transformation, orchestration, observability, governance, and more.

- The productivity gains were immediate and substantial. Organizations that previously spent months building a single integration could now connect new data sources in days or hours. Data engineers who once wrote thousands of lines of custom ETL code could leverage pre-built connectors and transformation frameworks. Business analysts gained direct access to tools that let them model and analyze data without depending on engineering resources for every request.

Tool Fragmentation Creates New Problems

70%

of data teams use 5–7 tools from 3–5 vendors.

68%

of data teams struggle to integrate tools; 40% cite integration maintenance as highest cost.

42%

of data teams need 26 vendors for DataOps.

Tool Fragmentation Creates New Problems

While these applications represented a significant improvement over manual approaches and boosted engineer productivity, they introduced new complexities of their own. The very modularity that solved integration challenges created an unexpected side effect: tool proliferation.

The market evolved rapidly, and countless startups came to market with point solutions and niche functionality. Every stage of the data pipeline spawned multiple competing tools: data ingestion, transformation, quality monitoring, cataloging, lineage tracking, orchestration, and observability.

The numbers tell a stark story of complexity explosion. A survey by the Modern Data Company found that 70% of data practitioners are using 5-7 tools or products from 3-5 vendors for data quality and dashboarding. Additional research from ESG found that 42% of respondents needed 26 vendors to execute their DataOps strategy.

Operational Complexity

The integration burden became the new bottleneck.

While these stacks enabled data to move across silos with less friction, they required each separate tool to be integrated into the stack. This caused friction. A 451 research survey reported that 68% of organizations struggle to integrate different data tools, with these integration challenges increasing costs and adding operational friction. Furthermore, 40% of respondents agreed that maintaining integrations across tools resulted in higher costs.

This operational complexity compounds exponentially with scale.

Each new tool added to the stack doesn't just create linear complexity; it creates potential integration points with every existing tool. For 10 tools a modest count for many enterprises you're managing 45 potential integration relationships. You also have to manage multiple vendor relationships, security complexity, and engineers need to understand how each tool works.

The Metadata Silo Paradox

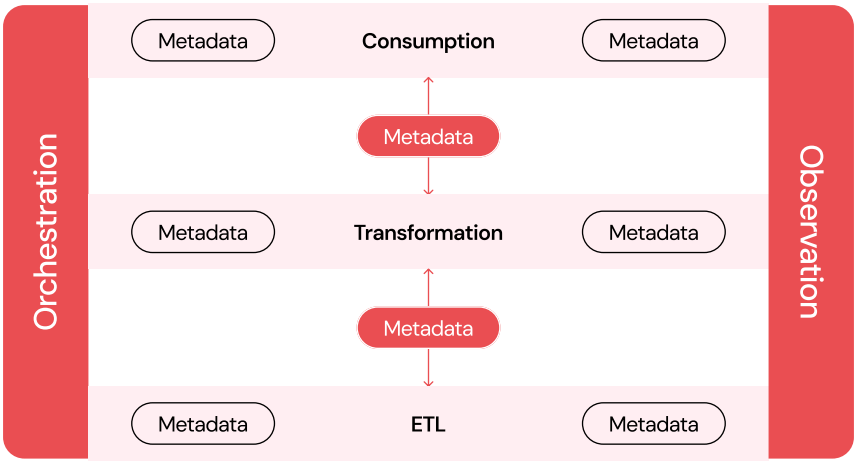
The modern data stack tools sprawl has also created new silos of metadata. Each tool in the stack has data models and metadata that are designed to support the unique objective of that tool. This makes commonality across different tools scarce, leading to challenges in accessing metadata across the modern data stack. The result is that valuable metadata gets trapped in each layer and tool.

ACCORDING TO THE 2025 DATAAWARE SURVEY

50% of all data engineering time is spent on maintaining broken pipelines rather than delivering valuable new datasets.

This metadata fragmentation is happening at the worst possible time. Metadata is no longer static it changes rapidly as business requirements evolve and data sources multiply. Meanwhile, more stakeholders across organizations need access to this metadata: data engineers, business analysts, data scientists, business users and today, even AI all depend on understanding data context to do their jobs effectively.

The solution requires a fundamental architectural shift: treating metadata not as a byproduct of individual tools, but as the intelligent foundation that connects and coordinates the entire data ecosystem.



The Failure of Intermediate Solutions

Multiple approaches have emerged to address the metadata silos and developer productivity challenges, each offering genuine value but ultimately falling short of the comprehensive solution that modern data environments require.

Metadata Management Strategies

The data community has responded with several approaches to metadata fragmentation. Automated metadata management platforms attempt to synchronize information across tools, but struggle to maintain context as data flows through complex transformation pipelines. Semantic layers try to unify metadata above the data stack, but require extensive integration work and ongoing maintenance. Open-source metadata standards promise compatibility, but can't capture the business context and operational nuances that make metadata truly valuable.

Low-Code/No-Code (LCNC)Platforms

Low-Code/No-Code (LCNC) platforms reduce the demand on developers by enabling more self-serve capabilities for business analysts and data scientists. The business value proposition is compelling, but these platforms never cover 100% of business requirements and complex transformations and vendor lock-in make it impossible to cover all requirements. They are also more focused on workflow and less on data quality and governance. This leads to more shadow IT processes, adding to operational complexity.

Observability Platforms

Observability platforms are more focused on tracking metadata. They continuously monitor data pipelines, detect anomalies, and provide rich metadata about data quality, freshness, and usage patterns. However, they typically do not cater to improving the productivity of the data engineer. They can identify issues, but do not provide the capability to fix them. This means that engineers need to integrate additional tools, which leads to more tool sprawl. They also don’t have the capability to fully integrate observed issues with automated fixes.

Data Mesh and Data Fabric Architectures

Data Mesh and Data Fabric represent architectural approaches to managing tool fragmentation by creating clear APIs and abstraction layers. Both approaches offer genuine value in organizing data access and reducing some integration complexity. However, they primarily address data consumption patterns rather than the underlying operational challenges that consume engineering time. The fragmented tool landscape persists beneath these architectural layers, and the fundamental metadata silo problems remain unsolved.

The Fundamental Gap: Context Without Action

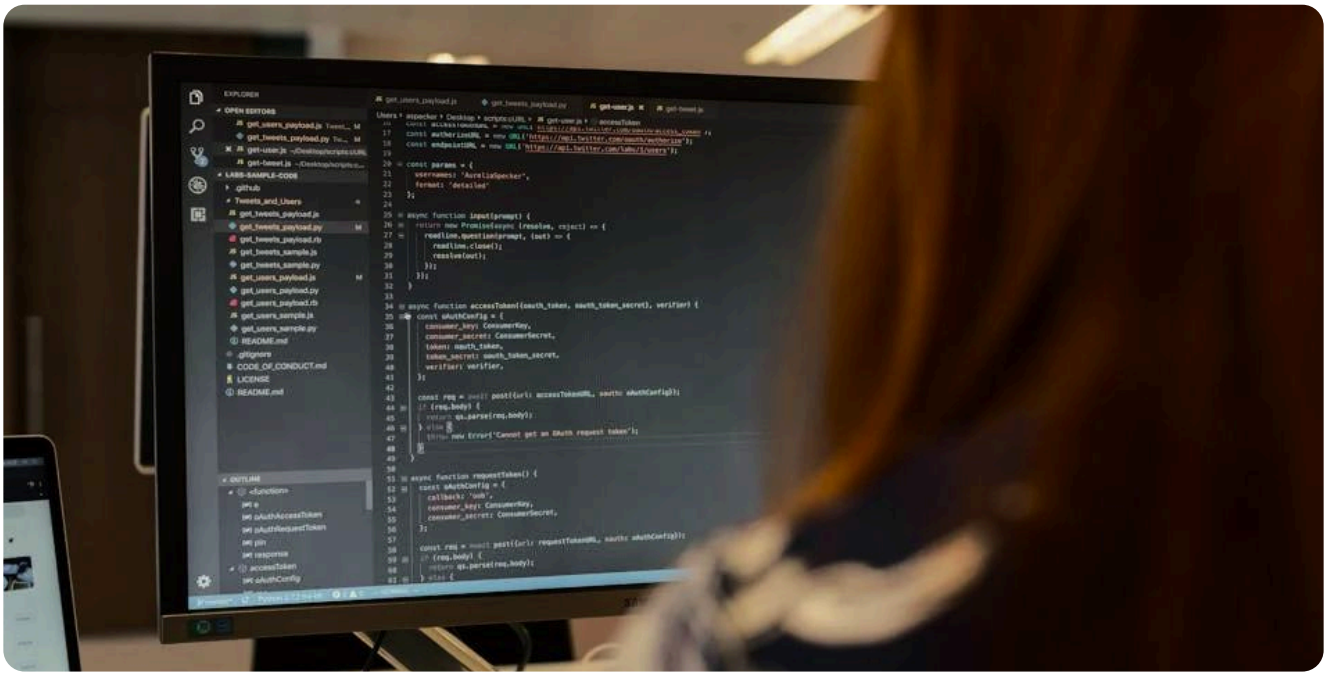
What's clear from the intermediate solutions is a predictable pattern: they boost visibility or lower manual effort but don't fundamentally change how data teams interact with systems.

ACCORDING TO THE 2025 DATAAWARE SURVEY

95% of data practitioners still report operating at or beyond capacity, despite heavy investment in these tools, highlighting persistent fragmentation, metadata silos, and operational complexity.

What's missing is a unified system that blends observability, accessibility, and automation with autonomous intelligence: systems that understand context, make decisions, and act.

The AI Revolution: Why Data Engineers Have Been Left Behind



While data engineers struggle with the capacity crisis, AI is transforming productivity across virtually every other technical role. The contrast is stark—and revealing.

We're already seeing the transformative impact of agentic AI across industries. In software development, GitHub's Copilot has evolved from code suggestion to autonomous pull request creation and code review, with developers reporting 7x productivity gains. In customer service, companies like Intercom are deploying agents that don't just answer questions but proactively identify and resolve customer issues before they escalate. Sales teams use AI agents that automatically qualify leads, schedule meetings, and even conduct initial discovery calls.

The pattern is consistent: AI agents understand context, make intelligent decisions, and take autonomous action to amplify human capabilities.

Yet data engineers—the people building the infrastructure that enables AI everywhere else—have barely benefited from these advances. While their colleagues in software development leverage intelligent copilots and their marketing teams deploy autonomous campaign agents, data engineers still spend 95% of their time on reactive maintenance and troubleshooting.

Fragmentation limits Agentic AI in Data Engineering

The reason isn't a lack of AI capability it's the fragmented architecture of modern data stacks. Agentic AI requires three fundamental capabilities that fragmented environments simply cannot provide:

1

Complete Context: Comprehensive Operational Intelligence

Agents require full understanding of the entire data ecosystem to make informed decisions. This includes technical metadata (schemas, lineage, performance), business context (priorities, SLAs, impact), and operational state (resource availability, ongoing processes, historical patterns). Fragmented systems trap this context in isolated silos. An agent trying to resolve a pipeline failure can see the error message but has no access to the business impact, upstream dependencies, or historical patterns that would inform the best resolution approach.

2

Intelligent Triggers: Event Detection and Scheduling

Effective agents need sophisticated triggering mechanisms. They must detect complex patterns across multiple data streams, understand the significance of combined events, and prioritize actions based on complete business context. In fragmented systems, triggers are scattered across different tools with no unified event correlation. An agent monitoring data quality in one tool has no visibility into pipeline execution status from another tool, making intelligent decision-making impossible.

3

Coordinated Tools: Unified Action Capabilities

Agents must be able to take coordinated action across the complete data infrastructure—from adjusting pipeline parameters and updating schemas to communicating with stakeholders and modifying business rules. In fragmented environments, each tool has its own APIs, authentication mechanisms, and operational models. An agent that identifies a data quality issue might be able to flag it in a monitoring tool but cannot coordinate the fix across the ingestion tool, transformation framework, and downstream analytics platform.

This is why adding AI agents to existing fragmented data stacks typically delivers disappointing results. The agents operate in isolation, making decisions with incomplete information and taking actions that may conflict with other system components. Instead of intelligent automation, organizations get expensive alert systems that still require human intervention for every meaningful action.

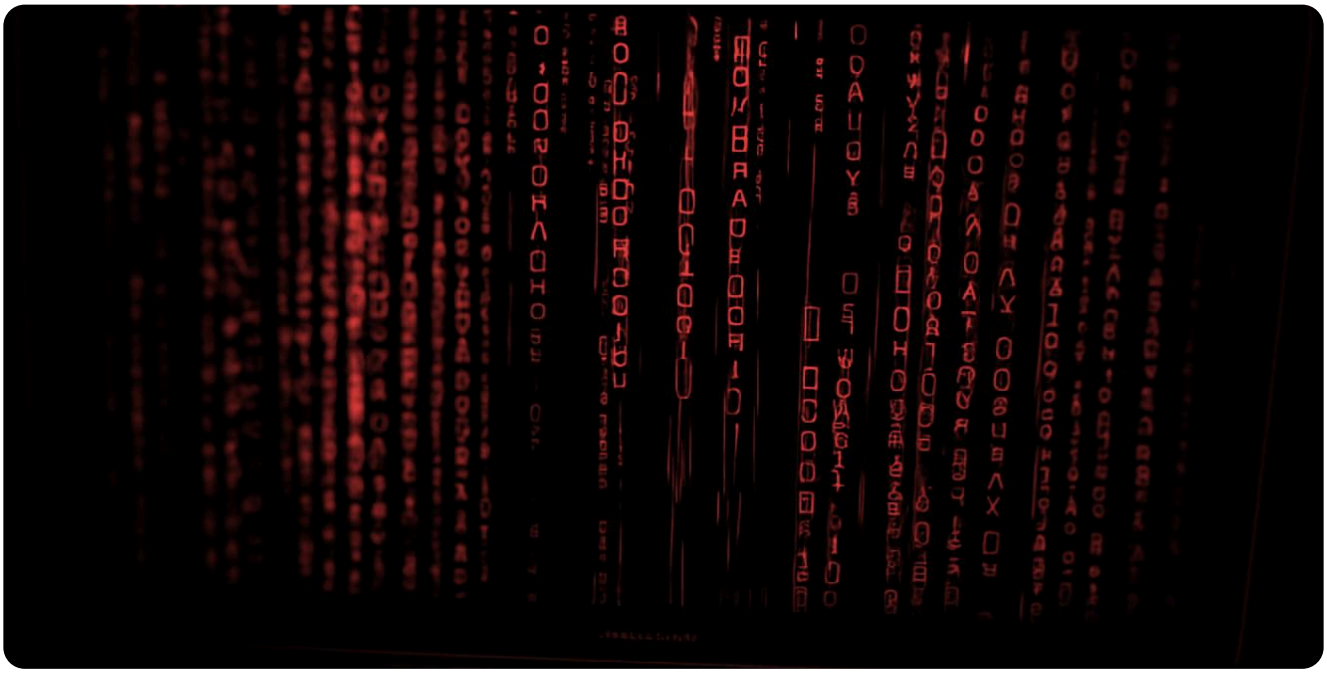
Agentic Data Engineering: A New Way Forward

Agentic Data Engineering (ADE) offers a fundamentally different approach built on human-AI collaboration. Instead of adding more tools to fragmented stacks, ADE platforms give both engineers and AI agents unified access to all metadata and operational context.

This shared understanding enables a new kind of partnership: engineers provide strategic thinking and complex problem-solving while agents handle routine operations, error resolution, and system monitoring. Because agents have the same comprehensive view of data systems that experienced engineers do, they can work as intelligent collaborators rather than isolated tools.

Ascend: The First True ADE Platform

Building a true ADE platform requires solving fundamental architectural challenges that have prevented other vendors from delivering genuine agentic capabilities. Ascend has built the industry's first platform specifically designed to deliver a truly agentic data engineering experience.



How Ascend Delivers ADE

Fundamental to Ascend's design is the integration of automation, intelligence, and AI agents into a single platform architecture. Unlike traditional data platforms that add AI features on top of existing infrastructure, Ascend is architected from the ground up to provide agents with the three fundamental mechanisms they need to operate effectively:

01.

Context: Complete Operational Intelligence

Agents require a comprehensive understanding of the entire data ecosystem to make informed decisions.

02.

Triggers: Event-Driven Deployment and Scheduling

Agents need sophisticated triggering mechanisms that go beyond simple alerts or cron jobs.

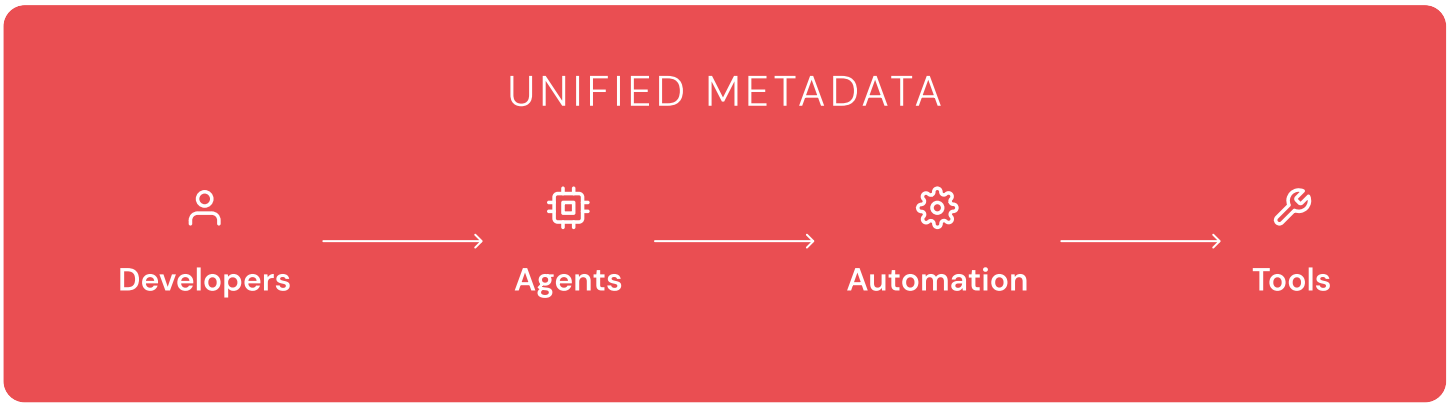
03.

Actions: Unified Capabilities

Agents must be able to take coordinated action across the complete data infrastructure.

Ascend's Unified Metadata Foundation: Complete Context

> **Unified metadata serves as the central nervous system of Ascend's ADE platform, providing complete, real-time context about every aspect of data operations.**
>
> Unlike traditional systems where metadata is scattered across dozens of tools in isolated silos, the ADE platform maintains all context code, execution history, lineage, business rules, operational state, quality metrics, usage patterns, and system relationships is available for every component.



This unified approach solves the metadata silo paradox that plagues modern data stacks and makes ADE possible.

When metadata exists in fragmented tools, agents operate blindly, making decisions with incomplete information. The unified metadata core ensures that every component human engineers, AI agents, and automated systems has access to the same complete context, enabling coordination across the entire data lifecycle.

As data environments evolve rapidly, the unified metadata core adapts dynamically, ensuring that all system components operate with current, accurate context. Schema changes are automatically propagated to dependent systems.

Ascend's DataAware Automation Engine: Intelligent Triggers

Ascend's DataAware Automation Engine serves as the platform's extensible event bus architecture that users can customize for sophisticated data operations.

This goes far beyond simple rule-based automation. The architecture enables complex event-driven workflows where agents can respond to sophisticated patterns across multiple data streams, understand the business significance of combined events, and coordinate responses based on complete system context.

Users can extend this automation through custom sensors and triggers, enabling organization-specific workflows that integrate seamlessly with their infrastructure. This extensibility ensures that the platform can adapt to unique business requirements while maintaining the unified architecture that enables agent coordination.

Ascend's Integrated AI Layer: Coordinated Tools

Integrated AI represents the action layer of the Intelligence Core, enabling autonomous agents to operate with complete context and coordinated capabilities.

Unlike traditional AI implementations that work in isolation, integrated AI agents share the unified metadata foundation and coordinate through the DataAware automation engine to deliver collaborative intelligence that amplifies human capabilities.

The unified metadata foundation is what enables agents to operate effectively. Unlike fragmented systems where agents operate with limited context, ADE platform agents have access to the same comprehensive information that human data team members use to make decisions. They can investigate issues following the same processes, write code with a full understanding of upstream dependencies and existing logic, and coordinate actions across the complete data ecosystem.

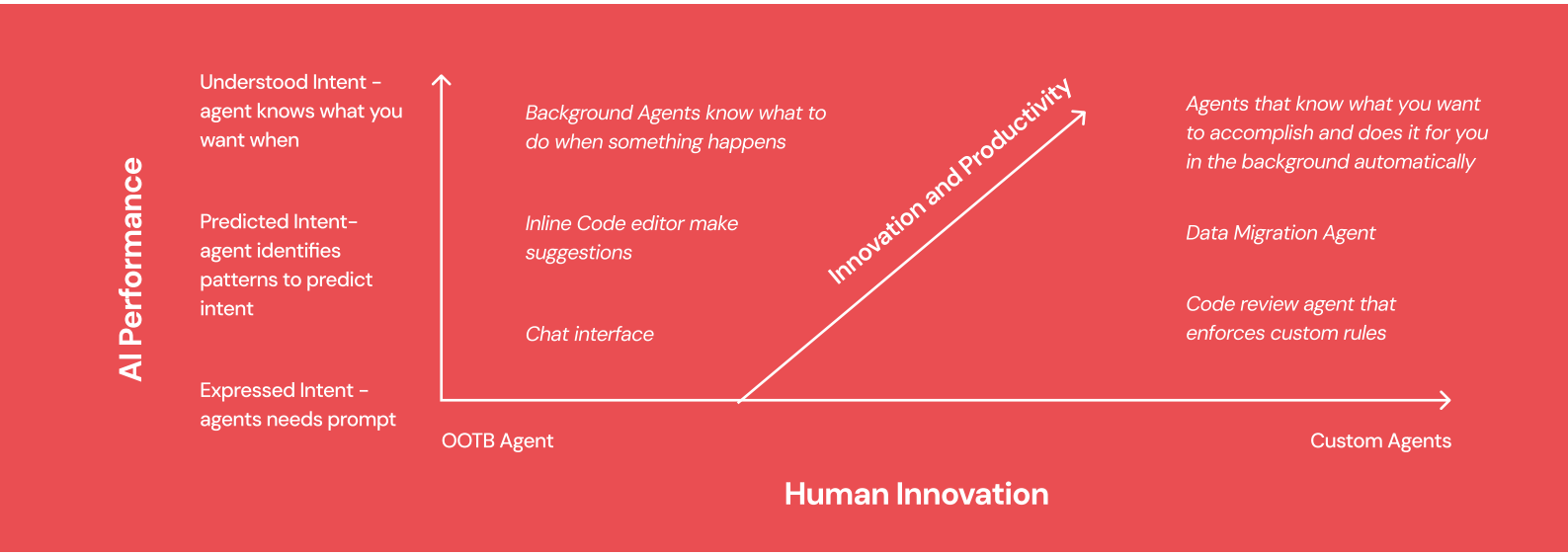
Agentic pipelines fix issues, write commit messages, and even prep pull requests, all before you ask. They don't just automate tasks. They anticipate them. And every move is easy to follow, easy to undo, and always in your control.

Agent Types & Capabilities

Agentic Data Engineering strategies require support for a variety of agent types designed to handle the full range of data engineering scenarios. Ascend's approach enables teams to deploy the right level of intelligence and customization for each use case, creating a natural progression path from basic automation to sophisticated autonomous systems.

Customization Spectrum (Out Of The Box Agents to Custom Agents):

Organizations can start with out-of-the-box agents that handle common data engineering tasks and progressively develop custom agents tailored to their specific business logic, data sources, and operational requirements. This progression is driven by human innovation, the more organizations understand their unique challenges, the more they can leverage custom agents to address them.



Autonomy Spectrum (Supervision to Independence):

Agents can operate with varying levels of human oversight, from requiring explicit instructions for every action to running completely autonomously in the background. This spectrum reflects the agent's ability to understand context and intent:

- **Expressed Intent:** Agents require explicit instructions and direct supervision, typically through chat interfaces or detailed prompts
- **Predicted Intent:** Agents can identify patterns and predict what users want to accomplish, making intelligent suggestions and streamlining workflows (ie. providing in-line code suggestions)
- **Understood Intent:** Agents fully comprehend user objectives and operational context, performing tasks autonomously when specific conditions are met (ie. writing automated Git commit messages based on changes to code)

Organizations can blend these approaches as needed. Basic tasks like automated commit messages will always be well-served by out-of-the-box agents, while complex, business-critical processes benefit from custom agents with deep organizational understanding.

Super Agents: Orchestrating Complex Workflows

The framework's most powerful capability emerges through "super agents"—intelligent orchestrators that coordinate both out-of-the-box and custom agents to execute complex, multi-step workflows. Super agents don't just manage other agents; they create comprehensive execution plans, analyze task requirements, and delegate work to the most appropriate specialized agents for optimal performance.

Task Analysis:

When faced with a complex objective like migrating a critical data pipeline, a super agent analyzes the complete scope of work and creates a detailed execution plan. The agent will break down the scope into individual tasks and define unique requirements.

Dynamic Agent Selection:

Super agents make intelligent decisions about which agents to deploy based on task requirements and agent capabilities. For routine tasks, they leverage efficient OOTB agents. For complex business logic, they engage custom agents with deep organizational context. This dynamic selection ensures that each sub-task is handled by the agent best equipped to deliver quality results efficiently.

Task Analysis:

When faced with a complex objective like migrating a critical data pipeline, a super agent analyzes the complete scope of work and creates a detailed execution plan. The agent will break down the scope into individual tasks and define unique requirements.

Dynamic Agent Selection:

Super agents make intelligent decisions about which agents to deploy based on task requirements and agent capabilities. For routine tasks, they leverage efficient OOTB agents. For complex business logic, they engage custom agents with deep organizational context. This dynamic selection ensures that each sub-task is handled by the agent best equipped to deliver quality results efficiently.

Ensuring Quality and Governance in the Agentic Data Engineering Platform

Agentic Data Engineering platforms are built on a foundation of proven DataOps best practices that create safe, controlled environments where both human engineers and AI agents can operate with confidence. This governance framework ensures that the increased automation and intelligence of agentic systems enhances rather than compromises operational discipline.

1. Environment Isolation: ADE platforms enforce strict separation between development, staging, and production environments, creating isolated spaces where agents can experiment, test, and validate changes without risk to production systems. Development agents can freely explore new approaches and test hypotheses, while production agents operate under stricter controls, monitoring performance only, with human oversight for critical operations.

2. Access Control and Permissions: The governance framework implements role-based access control that applies to both human engineers and AI agents. Different agent types have permissions appropriate to their function and risk level—background agents handling routine tasks have broad operational access, while agents making structural changes require elevated permissions and approval workflows. This ensures that agents can be productive within appropriate boundaries.

3. Change Management Integration: All agent activities integrate with established change management processes, backed by Git. Agents automatically generate change requests for significant modifications, maintain detailed logs of all actions, and coordinate with existing approval workflows. This integration ensures that agentic automation enhances rather than bypasses organizational governance requirements.

4. Compliance and Auditability: The unified metadata foundation creates comprehensive audit trails that meet enterprise compliance requirements. Every data transformation, schema change, and operational decision is logged with full context, including the reasoning chain that led to agent actions. This level of documentation often exceeds what traditional manual processes can provide, improving rather than complicating compliance efforts.

Benefits of Agentic Data Engineering

The transformation from fragmented tool stacks to unified agentic platforms delivers exponential rather than incremental improvements across every aspect of data operations. Organizations implementing ADE are not just seeing productivity gains, they're fundamentally reshaping how data teams create value and how organizations leverage data for competitive advantage.

Exponential Productivity Gains

From Maintenance to Innovation

ADE enables data engineers to focus on strategic initiatives such as designing new data products, optimizing business processes, and driving innovation through data insights.

7x Developer Productivity

Research demonstrates that agents, including in-line developer copilots and intelligent code review systems, can multiply engineer productivity by seven times.

Operational Excellence Through Intelligence

Proactive Issue Resolution: Agents continuously monitor data flows, detect anomalies as they emerge, and can automatically trigger remediation workflows before issues impact downstream consumers. When problems do occur, debugging agents trace issues across complex pipeline networks using complete lineage and execution history, often resolving problems faster than human engineers could even identify them.

Enhanced Data Quality: Agents integrated directly into DataOps workflows automatically validate pipeline logic, data quality, and business rule compliance. These agents don't just flag issues—they analyze root causes, suggest specific fixes, and can implement corrections with appropriate oversight. The result is data quality that improves continuously rather than degrading over time.

Strategic Competitive Advantage

Accelerated Implementation:

While 80% of organizations desire data automation, only 5% have successfully implemented it using traditional approaches. ADE platforms provide the unified architecture and intelligent coordination that make comprehensive automation achievable, enabling organizations to realize benefits that have remained elusive with fragmented tool stacks.

Future-Proof Architecture:

As AI capabilities continue advancing rapidly, organizations with agentic foundations can leverage new developments immediately rather than waiting for vendor implementations across multiple disconnected tools. The unified metadata layer and agent framework create an architecture that amplifies rather than constrains AI advancement.

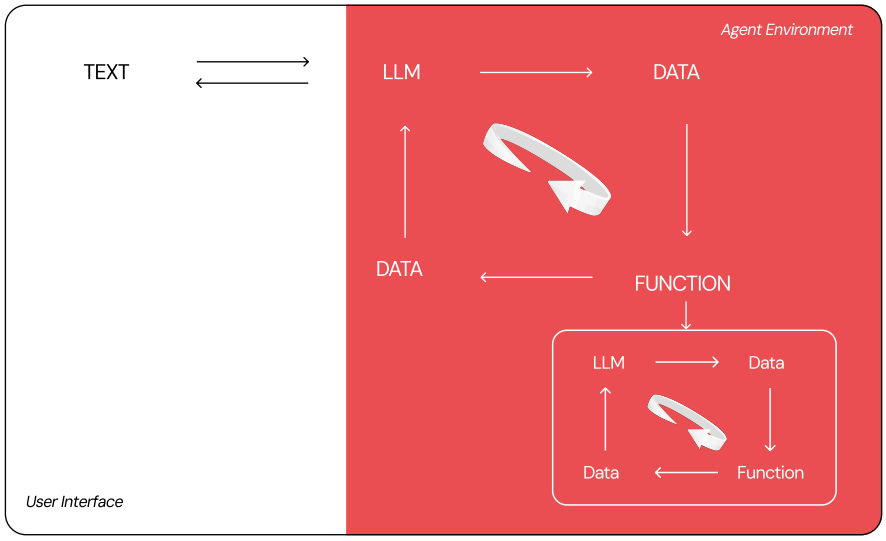
Conclusion

Data and AI technologies are changing and improving rapidly. Engineers focused strategies that are supported by the three pillars of agents, automation, AI, and metadata will provide the agility needed to navigate the choppy waters of the rapidly changing ecosystem.

Data leaders face a stark choice: begin experimenting with agentic workflows today to establish operational advantages, or risk being left behind by competitors who embrace intelligent automation. This transformation requires fundamental architectural decisions that put intelligent coordination at the center of data operations—it cannot be accomplished through incremental tool additions.

Appendix: Understanding Agentic AI

Agentic AI represents a major advancement in how AI systems operate. While traditional LLMs excel at generating content and answering questions, agentic systems can take autonomous action.



From a technical perspective, an agent operates as an iterative loop that incorporates an LLM call. The LLM analyzes data, determines an action to take, calls a function, processes the results, and feeds that information back to decide the next action. This cycle continues until the agent completes its objective or determines it needs human intervention. This architecture also supports multilayered agents, where one agent can call functions containing other embedded agents.

Unlike rule-based automation that fails when conditions change, agents adapt dynamically. When a pipeline errors, traditional automation triggers alerts and waits for humans. Agentic systems investigate the issue, understand downstream impact, adjust transformation logic, and update documentation completing resolution before data consumers are affected.