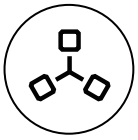
Behavioral Agent Automation Platforms

Addressing the Sequencing and Scaling Challenges in Enterprise Agentic AI

Liminal



AI delivers when it meets people where they are; it stalls when it doesn't.

Most enterprises today remain stuck not because AI underdelivers, but because current approaches demand prediction before discovery, technical translation from non-technical teams, and rigid workflows for a workforce that doesn't work rigidly.

Behavioral Agent Automation Platforms (BAAPs) change this equation, learning from observed behavior and autonomously deploying adaptive, individualized automation at scale.

Executive Summary

Nearly eight in ten companies report using generative AI, but an **equal number report no significant bottom-line impact.**[1] Another study shows that despite $30–40 billion in enterprise investment into GenAI so far, **95% of organizations are getting zero return.**[2]

This isn't a capability problem. Foundation models can reason, write, analyze, and synthesize at near-human levels. Automation frameworks exist. Yet most enterprises see little measurable impact from AI investment.

The next evolution will bring autonomous systems that can orchestrate multi-step workflows,

1 "The state of AI: How organizations are rewiring to capture value," McKinsey, March 12, 2025
2 Chari, Pradyumna, et al. The GenAI Divide: State of AI in Business 2025. MIT NANDA, July 2025

make decisions across organizational boundaries, and operate without constant oversight.

That's the promise of agentic AI, and it's substantial: McKinsey estimates agentic AI could unlock $2.6–$4.4 trillion in annual productivity gains beyond what generative AI already offers.[3]

Yet today, 60% of CEOs remain stuck in pilots.[4] The disconnect between agentic capability and enterprise value realization comes down to two fundamental limitations: **sequencing** and **scalability**.

The sequencing problem

Today's agentic approaches require organizations to *predict* what employees need before observing how they actually work. Design predates discovery; it should be the other way around.

The scaling problem

Current frameworks can't easily bridge individual work patterns to enterprise-wide deployment. **Pre-built agents** assume generic workflows that don't reflect how individuals actually operate. **DIY frameworks** dump complex automation functionalities on time-starved employees and already-stretched IT teams, then ask them to self-assemble their agentic future.

The practical consequences of these limitations are significant.

Administrators must guess which workflows to automate before understanding what behaviors drive inefficiency. And even if they could accurately identify the best candidates for automation, solving it through a DIY framework requires deep domain knowledge of the AI tools and capabilities available, as well as the know-how to assemble, deploy, and maintain custom workflows across their organization — all while those employees have their regular jobs to do. The DIY route can deliver individualized customization, but unless a company already employs an army of readily-available engineers with time to spare, this approach cannot scale.

3 "The economic potential of generative AI: The next productivity frontier," McKinsey, June 14, 2023.
4 IBM Institute for Business Value. *5 mindshifts to supercharge business growth*. IBM Corporation, May 2025.

These structural limitations explain why most organizations see minimal returns despite substantial investment; it's why the path from individual productivity gains to enterprise transformation remains elusive.

Closing this gap requires a fundamentally different approach.

The agentic unlock lives in solving the sequencing problem at scale: **delivering individualized automation while achieving enterprise-wide deployment.**

Adaptive automation inverts the current sequence entirely: observe behavior first, identify friction, *then* assemble agentic capabilities based on evidence rather than assumption. This shift — **from prediction to proof** — changes the enterprise AI value proposition. Instead of requiring employees to assemble multi-step automation sequences or administrators to predict workflows in advance, the system observes how people actually work, identifies automation opportunities from real behavioral data, and deploys capabilities that solve verified problems without manual configuration.

Behavioral Agent Automation Platforms (BAAPs) operationalize this model.

A Behavioral Agent Automation Platform (BAAP) is a horizontal, adaptive automation layer that observes behavior, identifies friction, and autonomously assembles agentic capabilities across the enterprise. It combines observed behavior with organizational memory, then transforms those insights into automated deployments.

This paper defines what BAAPs are, how they work, and why they represent the next architectural evolution of enterprise AI. It examines the structural limitations of current agentic approaches, introduces the behavioral automation model, and provides a framework for evaluating adaptive automation platforms.

Table of Contents

The Enterprise AI Paradox 6
Current Agentic Approaches & Key Challenges 7
Pre-Built Agents 7
The Technical Translation Gap 8
The Architectural Shift Required 9

Behavioral Agent Automation Platforms 11
Defining BAAPs 11
How BAAPs Work 11
Core Capabilities 11
Architectural Principles 12
Why This Architecture Matters 13
BAAPs in Practice 14

The Building Blocks of BAAPs 18
Measurement of Observed Behavior 18
Threshold Triggers for Automation 19
The Five Foundational Pillars 20
The Strategic Case for Model-Agnostic Architecture 21
Addressing Data Fragmentation 22

Governance & Trust 23
The Foundation: Transparency & Trust 23
The Governance Infrastructure 24
The Observability Model & Employee Trust 25

Closing the Gap 27
Conclusion 27
Appendix A: A Maturity Model for Enterprise AI Automation 29
Appendix B: Readiness Assessment for BAAP Adoption 32

The Enterprise AI Paradox

Enterprise AI isn't underdelivering because organizations aren't trying. It's underdelivering because current agentic approaches are structurally limited.

Generative AI is working at the individual level. The tools are delivering on their fundamental promise: incremental productivity gains at the task level. In the right hands, GenAI is great for individual contributors. But the enterprise potential of agentic AI won't be realized at the task level — **it has to come at the system level, and at scale.**

The true promise of agentic AI is autonomous systems that don't just help complete tasks faster, but automate entire workflows with little-to-no babysitting. Agentic AI is designed to eliminate menial, repetitive work, orchestrate across systems, and deliver the exponential productivity gains that justify the investment.

That's the promise. The reality right now is far messier.

Despite the trillions in nascent value McKinsey predicts from AI, 60% of CEOs surveyed are still stuck in the piloting phase when they expected to be well beyond experimentation by now.[5]

The gains that have materialized are "diffuse and hard to measure."[6] Horizontal tools deliver individual, task-level productivity, but enterprise transformation requires system-level change. And while agentic solutions are designed to automate the workflows plagued by friction — cross-functional handoffs, cognitive overload from context-switching, fragmented data scattered across systems — most organizations aren't achieving meaningful returns.

> The problem isn't AI capability; the problem is agentic *architecture*.
>
> Today's agentic tools sit **adjacent** to workflows, not **embedded** within them.

The problem isn't AI capability; the problem is agentic architecture. Today's agentic tools sit adjacent to workflows, not embedded within them.

5 IBM Institute for Business Value. *5 mindshifts to supercharge business growth*. IBM Corporation, May 2025.
6 IBM Institute for Business Value. 5 mindshifts to supercharge business growth. IBM Corporation, May 2025.

Current Agentic Approaches + Key Challenges

Current agentic AI approaches fall into two camps: pre-built "one-size-fits-all" agents, and DIY "build-your-own agent" frameworks.

Pre-Built Agents

The pre-built path puts the burden of discovery on administrators. IT teams are asked to select an off-the-shelf agentic workflow tool and roll it out for departments they don't fully understand. In rigid, script-based environments like call centers, pre-configured agents can deliver real value. But for messier, more strategic, less predictable work, this approach buckles.

These tools ask executives and IT teams to know the answer *before* understanding the problem. They have to **predict** what will be useful before **observing** what actually happens in practice. This is the **prediction > proof problem:** designing solutions for workflows that can only be approximated.

Consultants can help. Organizations can hire management consultants to shadow employees, map workflows, and identify automation opportunities. However, that's time-intensive, expensive, and still suffers from an unavoidable flaw: it captures reported behavior, not lived behavior. Employees perform differently when observed. Self-reported workflows are filtered through what employees think their managers want them to be doing, not what they actually do when nobody is watching. Pre-built agents assume generic workflows, and generic workflows don't reflect how individuals actually work. These solutions operate on the premise that everyone in a given role operates identically: All sales reps prepare for calls the same way. All finance analysts run month-end close the same way. All engineers troubleshoot incidents following the same pattern.

These tools ask executives and IT teams to know the answer **before** understanding the problem. They have to *predict* what will be useful before observing what *actually* happens in practice. This is the **prediction > proof problem:** designing solutions for workflows that can only be approximated.

The way individuals do the same job is as varied as the individuals themselves. Rolling out a pre-built agent that cannot adapt to individual needs results in lagging uptake and diminished returns.

The alternative requires significant investment in change management: training teams to alter their natural workflows and adopt machine-friendly processes. This approach demands standardization so agents can function optimally, but it's expensive, disruptive, and often fails because people resist processes that don't align with how they naturally think and operate.

The Technical Translation Gap

The DIY path shifts the burden to end users and technical teams. Platforms offer low-code builders, agentic workflow frameworks, and tool libraries that enable teams to build their own agentic future. The pitch is empowerment: teams know their jobs better than anyone, so they should design their own automations.

These platforms and frameworks are powerful and genuinely functional. What they struggle with is meeting employees where they are. It is hard to scale custom solutions across an enterprise when each workflow is bespoke. It requires significant time, energy, and expertise from IT administrators to design, deploy, and maintain these systems.

The reality, though, is that most employees and most IT teams today do not have the time or resources to design and implement complex, multi-step agentic workflows.

Asking them to architect their own agentic infrastructure is like handing someone a billion Lego pieces without a manual and expecting them to build an airplane that can actually fly.

Even for technical teams that could theoretically build these workflows, the barriers are prohibitive. Enterprise workflows don't live in one application. They span CRMs, email, Slack, Sharepoint, Google Drive, ERPs, web applications, custom databases, and dozens of other tools. Someone has to translate real-world workflows into technical specs, integrate across all those systems, handle authentication, manage error states, and maintain it all as APIs change and systems evolve.

Technical teams are already stretched thin. They're managing legacy infrastructure, handling security incidents, supporting day-to-day operations, and fielding requests from every department. They do not have the bandwidth to become workflow architects for every role in the organization. Even if they knew everything that needs automating, the implementation burden is overwhelming.

Organizations need deep integrations. They need orchestration across fragmented systems. And critically, they need a horizontal AI interface — **a unified layer through which employees can query all those siloed systems naturally.** Without that infrastructure, employees remain stuck context-switching between native apps, and administrators are left predicting which workflows to automate rather than understanding how work actually happens across these systems.

What's required is a system that can:

- Query organizational memory (Slack, email, Google Drive, SharePoint)
- Interpret employee intent
- Route requests to the right foundation models
- Execute tool calls across multiple systems
- And return results seamlessly

Building that from scratch is a multi-year engineering effort.

The Architectural Shift Required

Both the pre-built and DIY paths struggle because they face the same structural limitations:

- They require prediction instead of proof
- They lack the technical infrastructure to observe behavior and orchestrate across systems
- They force generic workflows onto individuals who work in fundamentally different ways

What's missing is the ability to **observe behavior**, **interpret intent** across systems, and **self-assemble solutions** based on how work *actually* happens.

What's required is a fundamentally different architecture: autonomous, horizontal agentic application layers that don't wait for prompts or require manual configuration. Systems that observe how work actually happens, learn from behavior, and autonomously assemble solutions based on proof, not prediction.

That is the architecture we're calling Behavioral Agent Automation Platforms.

Behavioral Agent Automation Platforms

A Behavioral Agent Automation Platform (BAAP) is a horizontal, adaptive automation layer that **observes behavior** (how employees interact with generative AI and their data), **identifies opportunity**, and **autonomously assembles agentic capabilities across the enterprise**. It combines observation data with organizational memory, then transforms those insights into multi-step automations.

Rather than helping employees complete pre-defined workflows faster, BAAPs identify what work needs to be done in the first place based on behavioral patterns. They don't rely on top-down approximations of what a department or role might need. They build from the bottom up using lived experiences of how work actually happens.

The strategic shift is from task-based to behavior-based automation. By observing how work actually happens rather than predicting what might be useful, BAAPs change the AI value proposition from prediction to proof. Administrators no longer predict what might be helpful and hope it sticks. BAAPs observe behavior and respond to proven need, automatically.

> The strategic shift is from **task-based** to **behavior-based automation.**
>
> By observing how work *actually* happens rather than predicting what might be useful, BAAPs change the AI value proposition from **prediction to proof.**

How BAAPs Work

Core Capabilities

BAAPs operate through five interconnected capabilities:

- **Observe employee behavior within the platform:** Track what users ask, how they configure AI requests, which data sources they query, the frequency with which prompts are submitted, patterns of repetition or friction, and the business outcomes they are working toward

- **Surface automation opportunities proactively:** Identify when a workflow is repeated enough, painful enough, or critical enough to warrant automation
- **Self-assemble agentic capabilities autonomously:** Build individualized agentic workflows by selecting optimal models, connecting relevant data sources, and defining execution logic without requiring users or IT to configure it manually
- **Deploy with human approval:** Suggest completed capabilities to administrators for validation and approval before going live
- **Adapt over time:** Continuously learn from usage patterns, refine and improve deployed capabilities, surface new automation opportunities as work evolves, and sunset outdated automations if/as needed

> ***A note on the observability layer***
> *The observability model is critical to understand. A BAAP does not monitor every keystroke or screen across your entire tech stack. It observes how employees interact with the platform itself: what problems they're trying to solve, what data they're requesting, and how often they repeat certain workflows. When combined with enterprise search that can query across all organizational data, and real-time data flows from MCP connections, the platform builds a comprehensive picture of work patterns without requiring invasive surveillance.*

Architectural Principles

A true BAAP is defined by its architectural principles:

- **Behavioral observability within the platform**: The system logs and analyzes how users interact with AI (what individuals, teams, and departments are asking for, the role they assign to the AI, the manner in which they're asking for the information, what outputs they're seeking, etc.)
- **Enterprise-wide data access:** Deep integration with organizational memory (Slack, Teams, email, Google Drive, SharePoint, CRMs, and other data repositories) through enterprise search capabilities
- **Signal-responsive automation:** The platform automates based on proven behavioral thresholds and human need, not pre-defined task types
- **Individualization:** BAAPs unlock individual agentic customization. The capabilities deployed to one person will look fundamentally different from the capabilities deployed to their colleague, even if they share the same job title and responsibilities
- **Bidirectional triggers:** The platform is interactive at the user level, so individuals can also prompt the system for automations they know they want without having to wait for a signal or intent trigger to kick in
- **Capability-based architecture:** Agents are packaged as deployable bundles that include

optimized model selection, scoped data access, and specific intent definitions
- **Governance by design:** Human-in-the-loop approvals, immutable audit logs, configurable safety rails, and rollback mechanisms baked in from the start, not bolted on after deployment
- **Closed-loop feedback:** Real-time telemetry on what's working, what's unused, and where friction persists

Why This Architecture Matters

BAAPs represent a fundamentally different approach to enterprise automation — not because of what they automate, but because of how they **learn** what to automate.

Traditional automation approaches require organizations to specify workflows upfront. Someone has to:

- Predict which processes are worth automating
- Design the automation logic
- Configure integrations, and
- Maintain the system as conditions change

This prediction-first model creates a structural ceiling: **automation can only scale as fast as human designers can specify it.**

BAAPs invert this model. By observing behavior within the platform and assembling automation based on proven patterns, they remove the specification bottleneck. The system identifies automation opportunities from actual work patterns, not projected ones. It builds capabilities tailored to individual workflows, not generic role assumptions. And it adapts continuously as work evolves, without requiring manual reconfiguration.

This architectural shift unlocks three capabilities that prediction-based approaches cannot achieve:

> Individualization at enterprise scale: Pre-built agents assume everyone in a role works identically. DIY frameworks can deliver customization, but only for the workflows someone has time to build. BAAPs deliver individualized automation across the enterprise because the system observes and responds to each employee's unique patterns. The finance analyst's deployed capabilities look nothing like the engineer's, even though both benefit from automation tailored to how they actually work.

Continuous adaptation without manual intervention: Traditional automation requires ongoing maintenance as workflows, systems, and business conditions change. BAAPs adapt automatically because they're observing behavior in real time. When work patterns shift, the system surfaces new automation opportunities and sunsets outdated ones. The automation layer evolves with the organization, not behind it.

Discovery without prediction: The most significant limitation of current approaches is that they require administrators to know what needs automating before they can build it. BAAPs eliminate this constraint. The system surfaces automation opportunities from behavioral signals — including patterns that administrators might never have identified manually. High-value but infrequent workflows (like the customer success manager's critical triage process) become visible and automatable because the system is observing intensity and impact, not just frequency.

These capabilities matter because they address the core challenges of enterprise automation: the need to **operate on proof rather than prediction**, and the need to **scale individualized solutions without requiring armies of engineers or change management initiatives to force standardization**.

The architectural difference isn't incremental. It's the difference between automation that requires human specification and automation that emerges from how work actually happens. **That shift — from designed workflows to observed behavior — is what makes BAAPs a category, not just a feature.**

BAAPs in Practice

The shift from **manual work** to **behavioral automation** follows a consistent pattern across enterprise functions. The examples below show how BAAPs observe work as it actually happens and autonomously assemble automation tailored to the unique workflows, roles, and working styles of individual employees.

Sales

From Manual Research to Automated Intelligence

A mid-market sales rep spends hours prepping for discovery calls — hunting through the CRM

for past touchpoints, skimming Slack threads for context, Googling company news, and piecing together a mental model of what matters before assembling it all into a sales dossier.

When the organization deploys a horizontal AI interface, the burden lessens. The rep can now ask natural language questions like "What's our history with Acme Corp?" and the platform queries across CRM, Slack, and the web to surface answers in minutes. It's faster and less painful.

But the real unlock comes with a BAAP. The platform notices the pattern: the rep preps for every high-value call the same way. She pulls CRM history, searches internal comms, reviews company financials, and drafts a call agenda. She does this 4–5 times per week, it's high-value, and automation can help. So, the BAAP flags it as an automation opportunity.

When approved, the system begins assembling "deal intelligence" dossiers automatically. Now, two hours before the next scheduled call, the rep receives a summary: CRM activity log, key Slack mentions, competitive intel from recent filings, and a draft agenda with suggested discovery questions based on the prospect's industry and pain points surfaced in prior conversations. The rep reviews it, tweaks two questions, and walks into the call prepared. No hunting. No context-switching. No manual queries.

Finance

From Data Hunting to Strategic Analysis

For an analyst in finance, month-end reporting means pulling data from three systems (ERP, Salesforce, and a custom SQL database), normalizing formats in Excel, running variance analysis, and building summary decks for the CFO. It takes hours. Attempts to automate parts of it fail because the systems don't talk to each other cleanly, and every month there's some edge case that breaks his scripts.

When the organization deploys an AI platform with enterprise search, the analyst starts querying it instead of logging into each system individually. "Pull Q3 revenue by region from Salesforce" or "Get last month's expense data from the ERP" becomes the new workflow. The platform handles the retrieval, and the analyst gets answers in minutes instead of navigating three different interfaces. It was a meaningful improvement, even if he still had to assemble everything manually.

With a BAAP in the mix, the system flags this month-end reporting workflow as high-volume, high-repetition friction and suggests automation. Once deployed, the BAAP autonomously retrieves data from all three systems on the 28th of each month, normalizes formatting, runs variance calculations, and pre-populates the reporting template. The analyst spends 90 minutes reviewing the output, adjusting assumptions where needed, and adding narrative context. The rest of the week opens up for strategic analysis instead of data wrangling.

Customer Success

From Reactive Response to Proactive Triage

A customer success manager handles a portfolio of 200 enterprise customers. Most interactions are routine, but once or twice a quarter, a high-severity issue lands — a critical bug, a contract escalation, or a sudden drop in product usage that signals churn risk.

When the organization introduces an AI platform, the manager starts using it to pull context faster. Instead of manually digging through support tickets, Slack threads, and usage dashboards, she can ask: "What's the sentiment on recent tickets from Acme Corp?" or "Show me their usage trend over the last 90 days." The platform queries across systems and surfaces answers in minutes, shaving meaningful time off the emergency response interval.

Once a BAAP is deployed at the organization, it quickly notices a pattern: High-severity customer issues trigger the same emergency workflow: pull support ticket history, analyze sentiment, check usage analytics, search Slack for engineering updates, draft an escalation plan. Even though these events are rare (maybe 8-10 times per quarter), the intensity is high. The manager's prompts are urgent, follow-ups are rapid, and the business impact is material.

The platform bundles this into a "critical customer triage" capability. Now, the next time a high-severity ticket lands, the BAAP proactively surfaces the full context package within minutes: sentiment trends, usage data, engineering updates, and a recommended next step. The manager is able to jump straight to problem-solving.

IT / Engineering

From Manual Incident Response to Automated Diagnosis

A site reliability engineer spends a chunk of every week troubleshooting incidents. When something breaks, she follows a familiar pattern: pull logs from the monitoring system, search internal documentation for similar past incidents, check recent deployments, correlate timestamps, identify root cause, implement a fix. It's repetitive, high-stakes work where every minute spent searching is a minute the system stays down.

When the team deploys an AI platform with access to logs, docs, and deployment history, the engineer starts querying it instead of jumping between tools. "Show me errors in the last hour from the payments service" or "Find incidents similar to this stack trace" becomes the new workflow. The platform can search across systems faster than manual navigation, cutting initial data-gathering time in half.

The real transformation comes when the BAAP starts observing the workflow. Over six weeks, it notices the engineer is querying the same log sources, searching the same documentation repos, and cross-referencing deployment histories in nearly identical sequences every time an incident fires. The platform identifies this as a high-value automation candidate: frequent enough to matter, structured enough to automate, critical enough to justify priority.

Once deployed, the BAAP activates the moment an incident alert fires. It auto-pulls relevant logs, surfaces the three most similar past incidents (with their resolutions), flags recent deployments that correlate with the timestamp, and drafts a preliminary root-cause hypothesis. The engineer still owns the fix, but she's starting from analysis instead of data gathering.

> These examples span different functions and workflow types, but the underlying pattern is consistent: the platform observes how work actually happens, identifies where automation can help, and automatically assembles capabilities tailored to individual workflows rather than generic role assumptions.
>
> **What makes that possible? The answer lies in the architecture.**

The Building Blocks of BAAPs

A BAAP is not a single technology but an integrated system of interconnected components. Understanding what makes up a BAAP — from the behavioral observation mechanisms that identify automation opportunities to the foundational infrastructure that makes those observations actionable — reveals **why this architecture can deliver what prediction-based approaches cannot.**

Measurement of Observed Behavior

Observed behavior provides crucial context that self-reported workflows cannot: What are employees actually trying to do when they engage with their data? What are they searching for? What business problems are they trying to solve?

In a BAAP environment, every interaction generates logged metrics that feed the platform's intelligence layer:

- **User Identity**: Who is using AI (role, department, seniority). This allows the system to identify both universal patterns ("all sales reps want X") and organizational patterns ("the finance team specifically needs Y").
- **Intent & Patterns**: What they're asking, patterns of repetition, triggers, and timing. If a user requests the same type of analysis every Thursday at 2pm, that's a signal. If fifteen users across a department ask variations of the same question within a week, that's a different signal.
- **Interaction Quality**: How they ask it (prompt structure, follow-ups, iterations), including AI creativity and temperature settings. A user who constantly adjusts temperature or rephrases prompts multiple times is signaling friction. The platform should notice.
- **Business Outcome**: Why it matters (the business problem they're trying to solve). This separates curiosity from urgency and experimentation from critical workflow needs.

The platform uses statistical regression analysis of these data points to identify embryonic automation potential. It looks for patterns that indicate risk, opportunity, or potential for

successful automation. Not every interaction becomes an insight. Not every pattern becomes a capability. The system is designed to surface signal, not noise. The business outcome also measures how likely an agentic capability could successfully address that particular problem.

Threshold Triggers for Automation

A BAAP must capture nuanced signals because a critical use case might happen only once but with high intensity or severity. Determining when to suggest automation isn't just about frequency.

The pattern recognition model analyzes four key factors:

- **Volume**: How often does this question/behavior occur?
- **Intensity**: How deeply felt is the need? (measured through interaction patterns, follow-up urgency, configuration adjustments)
- **Severity**: How important is solving this problem to business outcomes?
- **Successful Outcome Likelihood:** How likely is automation to successfully address the pattern?

A high-intensity interaction that happens only once may still be flagged if the severity is high and the solution is achievable. Conversely, a low-stakes behavior that happens daily might not warrant automation if the ROI doesn't justify the deployment.

When the system identifies both a repeatable pattern and a proposed solution that matches expected outcomes, it bundles the capability for administrator approval. **This is key; a BAAP does not just spin up endless agents without supervision or oversight. Every capability bundle is approved by the system administrator before deployment**. If the answer is unclear, the platform continues observing rather than deploying functionality for its own sake.

Three Types of Behavioral Thresholds

BAAPs respond to three types of behavioral thresholds:

- **Universal threshold**: Automations that apply broadly across role types (“all marketers want X,” “all sales reps want Y”). These often ship as out-of-the-box capabilities.
- **Organizational threshold**: Multiple people within an organization asking the same question or performing similar workflows. These are custom to the company's unique needs.

- **Individual threshold:** Personal repeated actions that trigger automation specific to one employee. This is where BAAPs deliver truly bespoke value; each BAAP instance becomes unique because it reflects how individual employees work, not how someone predicted they would work.

The Five Foundational Pillars

A BAAP comprises five interconnected components:

1. Secure User Interface & Delivery Hub

A unified, enterprise-wide hub that gives the entire organization a single entry point for securely interacting with AI and organizational data. The integrated delivery hub is not only the access layer for multi-model querying but also the observability aperture, the automation approval interface, as well as the agent deployment & management platform.

2. Model-Agnostic Orchestration

The ability to algorithmically route prompts to the model best positioned to deliver a quality response. This prevents vendor lock-in and allows for multi-model pressure testing. Rather than forcing users to know which LLM is best for code generation vs. creative writing vs. data analysis, the platform handles routing automatically. And considering that workflows often entail research, writing, analysis, formatting, refinement, etc., having multi-model capabilities means optimal output within your workflow.

> **Note: Model-agnosticism isn't strictly required for a BAAP to function. You could theoretically build a BAAP on a single foundation model provider, but doing so introduces significant strategic risk.*

3. Role Consistency and Customization

Providing a consistent way for AI to play specific roles (coach, debate partner, research assistant, emotional support). This does more than improve UX; it allows the platform to derive deeper sentiment and behavioral data than a standard search interface would capture. When users explicitly select how they want AI to engage with them, the platform learns not just what they're asking but how they're thinking about the problem.

4. Enterprise Search + Organizational Memory

The capability to connect to vast, distributed data sources (Slack, Teams, Google Drive, SharePoint, email, CRMs, ERPs) and index that data into a comprehensive, dynamic vector store that can be queried naturally. Organizational memory includes access to both dynamic data (live conversations, real-time updates) and static stored data (historical documents, archived decisions). Without this layer, a BAAP is just an interface. With it, the platform becomes the nervous system for institutional knowledge.

5. An Insights Engine

A system built on all observed behavior and log data that turns patterns into intelligent nudges and identifies emerging risks or opportunities. This is the brain of the BAAP. It doesn't just log activity; it interprets it, identifies friction, and surfaces automation candidates proactively. The insights engine is what shifts the platform from reactive tool to proactive partner.

The Strategic Case for Model-Agnostic Architecture

While a BAAP can technically function on a single foundation model, the best implementations will separate the model layer from orchestration logic. This architectural choice future-proofs the platform and shifts competitive value to where it belongs: how tools are chained, how data is governed, and how agents cooperate.

A model-agnostic approach delivers several advantages:

Cost optimization: Route simple queries to cheaper, faster models; reserve expensive, complex models for tasks that justify the cost.

Performance hedging: As new models emerge, plug them in without rebuilding workflows.

Negotiating leverage: Single-model dependency gives that foundational model provider significant pricing power over the entire automation stack. Multi-model architecture keeps model providers competing for workload.

Future-proofing: The foundation model landscape will continue to evolve rapidly. The winners today may not be the winners in 18 months. Model-agnostic architecture ensures a BAAP survives model churn.

It also removes the cognitive burden of tracking which models excel at which tasks. Employees and administrators don't need to know that Claude is better at long-form reasoning while GPT excels at structured output. The platform handles routing automatically, optimizing for quality and cost behind the scenes.

Implementing a BAAP built on a single model provider sacrifices flexibility, introduces vendor risk, and limits the ability to adapt as the market evolves. The most resilient BAAPs treat foundation models as commodities, not dependencies.

Addressing Data Fragmentation

BAAPs address one of the most urgent priorities in enterprise AI: disconnected and siloed data. **Knowledge workers spend about a fifth of their time (one full day each work week) searching for and gathering information.**[7] Meanwhile, 50% of CEOs admit that recent digital investments have resulted in **disconnected, piecemeal technology across their organization.**[8]

BAAPs meet organizations where they are, indexing unstructured data wherever it lives and making it queryable without requiring wholesale system overhauls.

BAAPs overcome this by utilizing enterprise search to pull dispersed organizational memory (including APIs, Slack archives, and legacy systems) into a unified index. This structured foundation is crucial: **72% of CEOs say leveraging proprietary data is key to unlocking generative AI value**.[9] Disconnected data limits the potential of AI-driven business reinvention.

The BAAP model operates on a "come as you are" principle for data readiness. Enterprises do not want to rebuild their entire data stack to realize AI lift. The platform meets organizations where they are, indexing unstructured data wherever it lives and making it queryable without requiring wholesale system overhauls.

7 Chui, Michael, et al. "The social economy: Unlocking value and productivity through social technologies." McKinsey Global Institute, 1 July 2012.
8 IBM Institute for Business Value. *5 mindshifts to supercharge business growth.* IBM Corporation, May 2025.
9 IBM Institute for Business Value. 5 mindshifts to supercharge business growth. IBM Corporation, May 2025.

Governance & Trust

Enterprise adoption of autonomous agents depends on trust, and trust depends on transparency. As agentic systems scale across organizations, the risk of operational chaos grows without proper governance infrastructure. This section examines the governance mechanisms that enable BAAPs to operate autonomously while maintaining organizational control, employee trust, and regulatory compliance.

The Foundation: Transparency and Auditability

As AI agents scale across enterprise environments, the risk of "agent sprawl" and operational chaos grows. BAAPs must prioritize monitoring, audit logs, and human-in-the-loop oversight to avoid scenarios where autonomously deployed capabilities create more problems than they solve.

In the BAAP model, every user prompt and subsequent action is logged. The system tracks tool calls, data platform queries, and the business problems being addressed. This creates a **comprehensive audit trail** that satisfies both internal governance requirements and external regulatory obligations.

Transparency isn't optional; it's fundamental. Organizations need to know what their agents are doing, what data they're accessing, and what decisions they're influencing. Without this visibility, trust erodes quickly. Trust is the currency that determines whether employees adopt these systems or work around them.

Human-in-the-Loop Approvals

When a BAAP identifies a pattern and self-assembles a potential automation capability, the user and administrator are proactively consulted for approval before deployment. The system might surface a recommendation like: "We've noticed you prepare sales dossiers before every discovery call. Would you like us to automate this workflow?"

The user or administrator (based on organizational policy and preference) reviews the proposed capability, validates that it aligns with their actual need, and either approves or

declines. This checkpoint prevents the system from deploying incorrect automations, misinterpreting intent, or building capabilities based on faulty pattern recognition.

Automated deployment without oversight creates serious risk. Human-in-the-loop gates ensure that automation serves real needs rather than compounding errors at scale.

The Governance Infrastructure

Enterprise-grade BAAPs require a comprehensive governance stack:

Access controls: Who can see what data, who can approve which capabilities, and who has administrative authority over the platform.

Human-in-the-loop approvals: Checkpoints before any new capability goes live.

Explainability: The ability to surface why the system suggested a particular automation, what data informed the decision, and how the capability functions.

Audit logs: Immutable records of every action taken by agents, every data query executed, and every approval granted or denied.

Safety rails: Guardrails that prevent agents from taking actions outside defined parameters (e.g. no external communications without approval, etc.).

Rollback mechanisms: The ability to disable or reverse a deployed capability if it's not performing as expected or creating unintended consequences.

Containment boundaries: Limits on what data agents can access, what systems they can interact with, and what actions they can autonomously execute.

Organizational red lines: Each company can determine from the outset what types of conversations, what types of agents, and what types of deployments are off limits to the platform, as well as which administrators are pulled into the loop for approval or escalation.

Drift monitoring: Continuous evaluation of whether deployed capabilities are still solving the problems they were designed to address, or if user behavior has shifted.

Data residency / compliance requirements: Ensuring that data handling meets jurisdictional and regulatory standards (GDPR, HIPAA, SOC 2, etc.).

The Observability Model and Employee Trust

The observability model in a BAAP is fundamentally different from keystroke logging or screen monitoring. The platform doesn't monitor every action an employee takes across their entire tech stack. It observes how employees interact with the platform itself: what problems they're trying to solve, what data they're requesting, how often they repeat workflows, and when they signal friction through their prompts.

This is behavioral observation in service of automation, not performance surveillance in service of discipline. The distinction is critical. Employees are more likely to adopt systems that visibly make their work easier than systems that feel like digital oversight. The platform earns trust by delivering value, not by demanding compliance.

Excessive employee monitoring via AI can erode trust and may violate regulations. The European Union has banned covert AI monitoring in workplaces, and other jurisdictions are likely to follow. Successful BAAP implementations must balance AI-driven observability with employee consent and transparency.

This is behavioral observation in service of automation, not performance surveillance in service of discipline.

Transparency is the key mechanism for building trust. When employees understand what's being observed, why it's being observed, and how it benefits them, adoption follows. When the system feels opaque or punitive, resistance is inevitable.

Well-designed BAAPs use aggregation to protect individual privacy while still surfacing organizational patterns. Aggregate insights reveal systemic friction without exposing individual behavior. **This balance — useful signal without invasive surveillance — separates BAAPs that succeed from those that get sidelined by employee pushback.**

What Aggregate Patterns Reveal

BAAPs can surface organizational dynamics that have nothing to do with AI performance. By analyzing aggregate interaction patterns, the platform identifies cultural issues, structural friction, and systemic problems that might otherwise remain hidden.

For example, if a high percentage of employees are selecting "emotional support" as their preferred AI role, that's a signal worth investigating. It might indicate burnout, low morale, or interpersonal conflict within teams. The platform isn't diagnosing the problem directly, but it's surfacing a pattern that leadership should examine.

This visibility is powerful, but it must be wielded carefully. **The goal is to identify systemic issues and improve working conditions, not to surveil individual employees or create a culture of distrust.** Aggregate analysis protects individual privacy while providing organizational insight — another reason why the observability model must be designed with transparency and consent as foundational principles.

Closing the Gap

The gap between agentic AI's promise and enterprise reality isn't a capability problem — it's an architecture problem. Current approaches to enterprise automation require organizations to predict workflows before observing them and force standardization onto work that doesn't standardize. The result is visible in the data: **despite substantial investment, most organizations remain stuck in pilots, and those that have deployed agentic systems report diffuse, hard-to-measure gains.**

Behavioral Agent Automation Platforms represent a fundamentally different architectural approach:

- By observing how work actually happens rather than predicting what might be useful, BAAPs invert the traditional automation sequence.
- Instead of requiring administrators to specify workflows upfront, the system identifies automation opportunities from proven behavioral patterns.
- Rather than forcing employees to adapt to generic processes, it assembles capabilities tailored to individual work styles.
- Instead of requiring manual configuration and maintenance, it adapts continuously as work evolves.

This architectural shift addresses the two core challenges that limit current agentic approaches: the **sequencing problem** and the **scaling problem**. BAAPs operate on **proof rather than prediction**, eliminating the need to guess what employees need before understanding how they work. And **they scale individualized automation across the enterprise** without requiring armies of engineers or change management initiatives to force standardization.

The building blocks that enable this shift — behavioral observation mechanisms, model-agnostic orchestration, enterprise search, and insights engines — are operational today. What makes BAAPs a category rather than a feature is how these components work together: observed behavior informs what to automate, organizational memory provides the context, and autonomous assembly deploys capabilities without manual specification. Governance isn't bolted on afterward — it's embedded in the architecture through human-in-the-loop approvals, audit logs, and transparency by design.

The strategic implication is clear. Organizations evaluating enterprise AI investments should look beyond whether a platform can execute workflows and **ask whether it can discover them.**

- Can it observe how work actually happens across fragmented systems?
- Can it identify friction from behavioral signals rather than requiring prediction?
- Can it assemble and deploy individualized automation at scale?
- Can it do all of this within governance frameworks that maintain trust and control?

These questions matter because the shift from task-based automation to behavior-based automation changes the enterprise AI equation. It moves the burden of discovery from humans to platforms, turns organizational memory into deployed capabilities, and makes automation adaptive rather than static. Most importantly, it creates a path from the 60% of CEOs stuck in pilots to the enterprise-wide transformation that justifies the investment.

The architectural difference between BAAPs and prediction-based approaches isn't incremental. It's the difference between automation that requires human specification and automation that emerges from how work actually happens. That shift — from designed workflows to observed behavior — is what makes enterprise-wide agentic AI achievable rather than aspirational.

Liminal

Appendix A

A Maturity Model for Enterprise AI Automation

Understanding where an organization sits on the AI maturity curve helps contextualize both the BAAP opportunity and the architectural leap it represents.

Most enterprises today operate somewhere between Level 1 and Level 4. Employees use generative AI for individual tasks, organizations have deployed copilots, some have built scripted automations, and in relatively rare cases, AI-forward firms have experimented with multi-step agentic workflows.

BAAPs represent Level 6 — a significant architectural advancement that typically requires years of sequential investment to reach organically. Organizations must first build observability infrastructure, implement enterprise search, develop insights engines, and layer in governance frameworks before adaptive automation becomes possible.

However, a true BAAP integrates all of these capabilities into a unified platform. **Rather than building each layer sequentially, organizations gain access to behavioral observation, enterprise memory, insights engines, and governance infrastructure through a single deployment.**

The maturity model below illustrates both where most organizations are today and what capabilities BAAPs provide as an integrated system.

Level 1: Reactive / Ad Hoc Use of AI

Employees use consumer AI tools (ChatGPT, Claude, Gemini) on personal accounts for one-off tasks. No enterprise oversight, no governance, no integration with organizational data. AI is a curiosity, not a strategy. Usage is inconsistent and unmonitored.

Level 2: Task-Level Copilots and Enterprise AI Subscriptions

The organization deploys AI assistants embedded in existing tools (Microsoft Copilot, Google Workspace AI, GitHub Copilot) or provides enterprise-wide subscriptions to foundation models (ChatGPT Enterprise, Claude for Work, Gemini Advanced). These tools speed up individual tasks like drafting emails, writing code, summarizing documents, and conducting research. Productivity improves at the edges, but workflows remain largely unchanged. There's no cross-functional coordination, no behavioral observation, and limited customization beyond what the individual employee configures.

Level 3: Manual Workflows + RPA / Low-Code

IT teams build scripted automations using RPA platforms or low-code tools (UiPath, Zapier, Power Automate). These workflows are human-designed, tightly structured, and require ongoing maintenance. They solve narrow, repetitive tasks well but struggle when conditions change. Scaling requires significant manual effort, and each new workflow must be built from scratch.

Level 4: Tool-Based Orchestration

Organizations adopt agentic platforms that allow chaining of AI actions across multiple tools. Administrators or power users manually configure multi-step workflows. This delivers real value for predictable, high-volume processes, but the burden of discovery and deployment remains on humans. Companies are still predicting what will be useful rather than observing what verifiably is.

Level 5: Behavioral Insight + Centralized Memory

The platform begins observing how employees interact with AI and organizational data. An insights engine surfaces patterns, risks, and opportunities based on aggregate behavior. Enterprise search connects disparate data sources into a unified, queryable layer. The organization has visibility into AI usage and friction points, but automation deployment remains manual. This level represents the foundation required before autonomous capability assembly becomes possible.

Level 6: Autonomous Capability Assembly & Deployment (BAAP)

The platform autonomously identifies automation opportunities, assembles capabilities, and suggests deployment to employees or administrators. Human-in-the-loop approvals remain, but the system is proactive rather than reactive. Capabilities are personalized to individual workflows, not just roles or departments. The platform continuously learns and improves based on usage patterns.

At this level, organizations gain integrated access to:

- Behavioral observation mechanisms that identify friction from actual work patterns
- Enterprise search that unifies organizational memory across fragmented systems
- Insights engines that surface automation opportunities proactively
- Governance infrastructure with audit logs, approvals, and transparency by design
- Model-agnostic orchestration that routes tasks to optimal foundation models
- Autonomous assembly that deploys individualized capabilities at scale

This is where agentic AI begins delivering exponential rather than incremental gains, and where the architectural shift from prediction-based to behavior-based automation becomes operational.

Appendix B:
Readiness Assessment for BAAP Adoption

Not every organization is positioned to implement a BAAP effectively. While the architecture integrates foundational capabilities like behavioral observation and enterprise search, successful deployment requires organizational alignment, governance maturity, and cultural readiness. The framework below helps organizations assess their preparedness across three dimensions.

Technical Readiness

Unlike traditional enterprise automation approaches that require extensive infrastructure buildout, **BAAPs are designed to integrate with existing systems as they are.** A true BAAP provides the core technical capabilities — behavioral observation, enterprise search, insights engines, and model orchestration — as part of the platform itself.

Organizations do not need to build observability infrastructure, develop custom integrations for every data source, or implement governance frameworks before deployment. The platform connects to organizational data where it already lives (Slack, Teams, email, Google Drive, SharePoint, CRMs, ERPs), provides employees with multi-model AI access, and begins surfacing behavioral insights immediately.

For most enterprises, the technical barriers to BAAP adoption are substantially lower than traditional automation approaches because the platform provides the infrastructure rather than requiring organizations to build it first.

Organizational Readiness

Leadership commitment to AI-driven automation: Executive leadership demonstrates commitment through budget allocation, strategic prioritization, and active sponsorship. Organizations where AI initiatives lack executive support or compete for resources with numerous other priorities will struggle to achieve meaningful adoption.

Willingness to shift information retrieval workflows: Employees are open to using a centralized AI interface as a primary source of information rather than navigating directly to native applications. For behavioral automation to function effectively, the platform must

become the primary interface through which employees access organizational data and complete workflows. Organizations where employees are deeply entrenched in existing tool usage patterns will face adoption friction.

Tolerance for iterative improvement: The organization accepts that early automation capabilities will improve over time based on usage and feedback. BAAPs learn from behavior and refine deployed capabilities continuously. Organizations that demand perfect automation from initial deployment will be disappointed by the learning curve inherent in adaptive systems.

Cross-functional collaboration: Departments collaborate on shared processes rather than operating in rigid silos. BAAPs deliver maximum value when they can orchestrate workflows that span functions — sales to finance, customer success to engineering, HR to IT. Organizations where departments rarely coordinate or share information will underutilize the platform's horizontal capabilities.

Internal champions and change advocates: Leaders exist within the organization who can communicate the value of behavioral automation, address employee concerns, and drive adoption across teams. Without advocates who understand both the technology and the organizational dynamics, even capable platforms struggle to gain traction.

Governance Readiness

Regulatory compliance framework: The organization understands its regulatory obligations (GDPR, HIPAA, SOC 2, etc.) and has established frameworks to meet them. BAAPs must operate within these boundaries, and organizations with immature compliance postures will need to strengthen governance before deploying autonomous agents.

Data security and access controls: Robust security protocols govern data access, external communications, and system modifications. BAAPs interact with sensitive organizational data and can execute actions on behalf of employees. Weak security models create unacceptable risk.

Transparency and observability alignment: Leadership and IT are comfortable with the level of visibility BAAPs require to function. The platform logs employee interactions with AI, tracks behavioral patterns, and surfaces insights from aggregate data. Organizations uncomfortable with this level of transparency — even when designed to protect individual privacy — will face internal resistance.

Established trust in AI systems: Employees trust that AI recommendations serve their interests rather than functioning as performance surveillance. Trust determines whether employees adopt BAAP capabilities or work around them. Organizations where employees are skeptical of AI motives or fearful of job displacement will struggle with adoption regardless of technical capability.


Liminal