From Benchmarks to Business Impact: Deploying IBM Generalist Agent in Enterprise Production

Segev Shlomov[1], Alon Oved[1], Sami Marreed[1], Ido Levy[1], Offer Akrabi[1], Avi Yaeli[1], Łukasz Strak[2], Elizabeth Koumpan[2], Yinon Goldshtein[1], Eilam Shapira[1], Nir Mashkif[1], Asaf Adi[1]

[1]IBM Research
[2]IBM Consulting

Abstract

Agents are rapidly advancing in automating digital work, but enterprises face a harder challenge: moving beyond prototypes to deployed systems that deliver measurable business value. This path is complicated by fragmented frameworks, slow development, and the absence of standardized evaluation practices. Generalist agents have emerged as a promising direction, excelling on academic benchmarks and offering flexibility across tasks, applications, and modalities. Yet, evidence of their use in enterprise settings remains limited. This paper reports IBM's experience developing and piloting the Computer Using Generalist Agent (CUGA). CUGA adopts a hierarchical planner–executor architecture with strong analytical foundations, achieving state-of-the-art performance on AppWorld and WebArena. Beyond benchmarks, it was evaluated in a Business-Process-Outsourcing talent acquisition pilot, addressing enterprise requirements for scalability, auditability, safety, and governance. In preliminary evaluations, CUGA approached the accuracy of specialized agents while suggesting reductions in development time and cost. We provide early evidence that generalist agents can operate at enterprise scale, distill key technical and organizational lessons, and outline requirements for transitioning research-grade architectures like CUGA into enterprise-ready systems.

1 Introduction

Enterprises are under growing pressure to automate digital work at scale. From customer support to back-office analytics, knowledge workers routinely interact with heterogeneous environments–web portals, APIs, spreadsheets, and dashboards–while facing strict requirements for auditability, reproducibility, privacy, and cost control. Over the past two years, interest has surged in *computer-using agents* (CUAs), systems that can plan and execute multi-step tasks across diverse applications. Yet for enterprises, the challenge is not only to prove capability–it is to *productize agents and capture real business value*.

The enterprise need. In practice, organizations struggle with the journey from research to deployment. Multiple frameworks and architectural patterns compete for adoption, but few offer clear guidance on speed of development, time-to-value, cost efficiency, and reliability in production settings. Enterprises also lack standardized ways to evaluate agentic systems: benchmarks emphasize academic settings, while business leaders demand measurable impact such as SLA compliance, reduction in manual effort, or improved audit readiness. Bridging this gap requires not only technical advances in agent design but also organizational insights into deployment, governance, and monitoring.

The research gap. Recent work has shown that *generalist* agents–single systems designed to perform diverse computer-use tasks–can achieve impressive results on academic benchmarks. Generalist designs are attractive because they promise (i) adaptability across task types and domains, (ii) reusability of architecture and tooling across new environments, and (iii) reduced need for brittle, task-specific scripting. However, published research has so far remained benchmark-centric. Generalist agents have been shown to perform well on synthetic benchmarks, yet their effectiveness in *enterprise production settings* is largely untested. The central question is therefore: what modifications, safeguards, and evaluation methods are required to make generalist agents *enterprise-ready*?

Our approach. This paper addresses that question through IBM's development of CUGA. Architecturally, CUGA evolved into a hierarchical planner–executor system with three control layers: (i) a chat/context layer for preprocessing inputs, (ii) an outer loop for task planning and management using a persistent ledger, and (iii) an inner loop for sub-task execution via specialized agents (API, Web, CLI, file-system). Reliability mechanisms include schema-grounded prompting, variable tracking, reflective retries, and provenance logging. In its benchmark evaluation, CUGA achieved state-of-the-art performance on both AppWorld and WebArena, confirming the strength of its generalist design. But more importantly for enterprises, CUGA was piloted in the BPO talent acquisition domain–a setting where recruiters and analysts must answer evidence-based questions across multiple dashboards and datasets under policy and audit constraints. We use this pre-deployment pilot not as the main "story," but as a proving ground to evaluate what enterprise readiness demands.

We introduce a new domain-specific benchmark, **BPO-TA**, comprising 26 decision-support tasks across 13 analytics endpoints. Tasks span single-endpoint lookups, cross-API joins, provenance-grounded explanations, and the

graceful handling of unsupported queries. This benchmark enabled both regression testing and controlled ablation studies during CUGA's pilot evaluation. In preliminary tests within simulated enterprise workflows, CUGA approached the accuracy of hand-crafted agents while indicating potential substantial reductions in development time (up to 90%) and cost (up to 50%). These early findings suggest that generalist designs can enable measurable enterprise value when adapted with appropriate safeguards. This paper contributes:

- **Enterprise pilot experience.** Evidence from a pilot of a generalist agent evaluated with recruiters and analysts in the BPO talent acquisition domain, including architectural modifications for auditability, safety, and governance.
- **Architectural advances.** A planner–executor agent design with schema-grounded prompting, variable tracking, reflective retries, provenance logging, and an API/Tool Hub that streamlined onboarding of enterprise applications. This architecture achieved state-of-the-art performance on both WebArena and AppWorld benchmarks.
- **Evaluation and preliminary business impact.** Using the BPO-TA benchmark of realistic enterprise analytics queries to support reproducible regression testing and ablation, evaluations show accuracy approaching that of hand-crafted agents, with indications of reduced development time (up to 90%) and development cost (up to 50%), alongside improved time-to-answer and reproducibility.
- **Lessons learned.** Technical and organizational insights from the pilot, including monitoring, governance alignment, and maintenance practices required to transition generalist agents from research to enterprise readiness.

2 Related Work

Early agentic paradigms such as *ReAct* interleave chain-of-thought reasoning with environment actions to improve task completion and interpretability (Yao et al. 2022), while code-centric approaches as *CodeAct* generate executable code to plan and call tools/APIs for complex tasks (Wang et al. 2024). These ideas catalyzed practical enterprise frameworks that orchestrate multiple specialized agents (or tools) with configurable roles. From AutoGen's conversation-programmed multi-agent patterns (Wu et al. 2024), to LangGraph's stateful, tool-grounded agent graphs for reliability and recoverability (LangChain 2024), and OpenAI's *Swarm* orchestrator for multi-agent handoffs (OpenAI 2024). Despite promise, production experience consistently reports fragility at scale: brittle inter-agent handoffs, maintenance overhead from prompt/tool drift, safety and generalization across different domains.

Enterprise report measurable wins when agentic systems are embedded behind assist and self-serve flows (The Verge 2025; Mobile World Live 2025; AI Business 2022). Analytics/BI copilots ship text-to-SQL/report-generation agents tightly coupled to enterprise data governance, observability, and review workflows (Snowflake Engineering 2024; Microsoft 2024; Databricks 2025). Broader surveys in hiring/HR analytics highlight fairness, transparency, and audit requirements in employment contexts, motivating provenance and explainability (Raghavan et al. 2020; Fabris et al. 2025; Chen 2023; Schwartz, Yaeli, and Shlomov 2023). Industry adoption reports likewise stress governance, measurable ROI, and operational readiness (monitoring, latency/-cost budgets) as prerequisites for scale-out (Zhang et al. 2025a; Pan et al. 2025). These constraints shape the design space of enterprise agents' tasks. BPO process automation combines workflow orchestration, retrieval over enterprise knowledge, and document understanding to automate outsource non-core business functions to third-party providers, leveraging semantic reasoning for process understanding as explored in (Oved et al. 2025), and human-centric automation approaches such as IDA and conversational RPA (Shlomov et al. 2024b; Yaeli et al. 2022; Zeltyn et al. 2022).

Concurrently, the research community has pushed toward generalist CUAs that plan and act across heterogeneous software. WebArena offers realistic, self-hosted websites for browser agents and showed early baselines struggled to exceed modest end-to-end success (Zhou et al. 2023). AppWorld evaluates multi-application orchestration via hundreds of programmatic APIs with outcome-based grading (Trivedi et al. 2024). OSWorld measures GUI workflows on desktop applications and OS tasks (Xie et al. 2024). Complementary suites probe interaction quality and oversight: *ST-WebAgentBench* emphasizes policy adherence in web agents, introducing *Completion-under-Policy* as the primary objective (Levy et al. 2024), τ-Bench targets tool-agent-user dynamics and policy/instruction following (Yao et al. 2025), BrowserGym provides a unified platform for evaluating web agents under controlled variability (de Chezelles et al. 2024) while (Shlomov et al. 2024a) identifies planning as the dominant bottleneck in web agents. Architecturally, generalist CUAs increasingly combine hierarchical planning, explicit state/variable tracking, and reflective repair during execution to improve robustness in long-horizon settings (Shinn et al. 2023; Kim, Baldi, and McAleer 2023; Zhang et al. 2025b). Recent vendor-facing systems expose "computer use" capabilities (desktop/browser control, file ops) under sandboxes, signaling a trend toward production CUAs (Anthropic 2024; OpenAI 2025; Shen et al. 2025; Fourney et al. 2024).

Despite rapid progress, several gaps limit direct transfer from benchmark success to enterprise deployment. First, *governance*: high-risk domains (including employment analytics) demand provenance, and post-deployment monitoring mandated by frameworks and regulation (Tabassi 2023; European Union 2024; NYC Dep. 2023). Second, *tool proliferation and schema variance*: results degrade as agents shortlist from dozens of APIs and maintain consistency across dependent calls (Shen et al. 2024; Xu et al. 2023). Third, *operational constraints*: latency, cost, and reproducibility must be tracked and controlled in production (Kwon et al. 2023; Zheng et al. 2024; Jiang et al. 2023). Fourth, *graceful degradation*: enterprise agents must decline unsupported requests without hallucination, and surface transparent rationales and computation logs (Nakano et al. 2021). Our work addresses this bridge by showing how a generalist CUA can be adapted for enterprise: schema-minimized API onboarding, deterministic parsing/validation, provenance-first responses, in a piloted BPO talent acquisition setting.

3 The Application: BPO Talent Agent

Background and Business Context IBM Consulting operates a double-digit million business in Business Process Outsourcing (BPO) for Talent Acquisition (TA). In this model, IBM specialists manage recruitment pipelines on behalf of client organizations, often working across multiple HR platforms, analytics dashboards, and reporting tools. While effective, the manual workflow is labor-intensive: recruiters spend significant time pulling data, reconciling spreadsheets, and preparing insights for hiring managers. Service-level agreements (SLAs) around time-to-hire, conversion funnels, and sourcing performance are business-critical, yet measuring and optimizing them has historically been slow and error-prone.

The vision behind the TA Agent was to augment human recruiters rather than replace them–acting as a digital sidekick that can provide proactive insights, automate repetitive analysis, and surface SLA risks before they become client issues. A core design principle is **human-in-the-loop (HITL)**: the business configures where the agent can act autonomously and where human oversight is mandatory, and the agent must strictly adhere to these requirements.

3.1 The Development Journey of Agentic Systems

The BPO–TA project also reflects a broader pattern observed across enterprises experimenting with agentic systems. Most teams begin with quick wins: popular frameworks like **ReAct** or **CodeAct** can be instantiated in days, yielding impressive demos where agents call APIs or generate code to answer queries. But as projects scale, limitations surface:

- ReAct agents degrade when required to juggle more than a handful of tools.
- Developers patch around this by introducing routers and delegators, creating "white-box" architectures with fragile hand-offs between sub-agents.
- Complex instructions or policy requirements (e.g., privacy, governance) strain these prototypes further.
- Roadmaps become unclear, with teams caught in cycles of experimentation rather than predictable progress.

The BPO–TA team followed this arc. An initial prototype, built quickly on reactive patterns, showed promise but could not scale to the breadth of sources and policies in Talent Acquisition. With 13 APIs spanning multiple systems and providers–each offering several actions and requiring orchestration across workflows–the complexity exceeded what the prototype could handle. At this inflection point, the team turned to IBM Research to evaluate CUGA, which had just achieved state-of-the-art results on AppWorld and WebArena benchmarks. The open question was: *Could a benchmark-proven generalist agent deliver enterprise-grade performance in the demanding Talent Acquisition setting?*

3.2 Application Setup

The Talent Acquisition Agent is still on its journey toward full production deployment. To date, it has achieved on-par accuracy with specialized agents, is being evaluated against enterprise requirements, and is under consideration for production rollout. The deployment context includes several key characteristics:

- **APIs and Analytics Layer:** The environment exposes 13 read-only APIs, spanning multiple applications and providers. Each API offers several actions, and realistic workflows often require orchestrating across sources. Examples include SLA-by-source, funnel conversion, hires by percentage, and skill-impact analysis.
- **Governance and Security:** To build trust at low risk, the current configuration is restricted to read-only APIs. This allows experimentation and validation without impacting underlying systems. Over time, the goal is to progress toward create/update capabilities, enabling fully automated workflows once safety and trust are established. All responses include provenance logs, and PII is excluded or redacted to maintain compliance.
- **Integration with Business Workflows:** The agent is designed to embed into recruiters' existing dashboards in the browser, becoming part of their daily workflow rather than a separate tool. It must integrate seamlessly with the user experience and identity controls already in place.
- **Human + Agent Collaboration:** Recruiters and analysts interact with the agent through a conversational interface. Depending on business configuration, the agent may act autonomously or defer decisions back to humans. HITL requirements are explicit and configurable, ensuring alignment with business workflows and governance.

3.3 Why BPO–TA Matters as a Testbed

The BPO–TA pilot illustrates why Talent Acquisition is a representative domain for studying the enterprise readiness of generalist agents:

- **Complex orchestration:** Multi-source workflows spanning 13 APIs, each with multiple actions, often requiring reasoning across providers and data sources.
- **Governance-heavy:** Read-only experimentation mode, HITL oversight, audit trails, and compliance constraints.
- **High value:** A double-digit-million business unit where small efficiency gains deliver major client impact.
- **Scalable lesson:** The trajectory from quick prototypes to generalist adoption mirrors what many enterprises experience in their agent journey.

In short, BPO–TA provided the ideal proving ground: a live enterprise context, business-critical stakes, and a context where the shortcomings of early architectures were well understood. The Talent Acquisition Agent was not introduced into a greenfield environment–it was evaluated precisely at the point where conventional approaches had reached their limits. This made the journey both realistic and consequential: success here could validate generalist architectures and signal how such systems may bridge the gap between academic benchmarks and enterprise deployment.

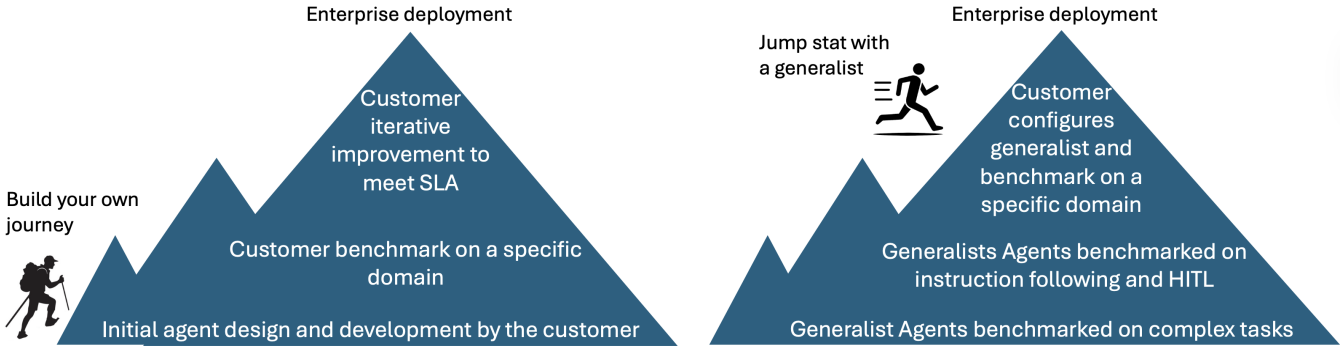

Figure 1: Reducing time-to-value with generalist agents. Traditional specialized agents (left) require extensive custom design and benchmarking. Generalist agents (right), benchmarked on complex tasks, shift the enterprise focus to configuration and domain-specific evaluation.

4 Enterprise Requirements for Generalist Agents

From the BPO–TA use case described earlier, and from additional discussions with other business units in our organization–including Finance, Sales, Procurement, Legal, and the CIO's office–we observe a recurring pattern. Enterprises consistently identify a set of requirements that go beyond academic benchmarks. We detail our understanding of what is needed from generalist agents in the enterprise.

Safety and Trustworthiness (Top Priority). Enterprises adopt agents in production only when their safety and trustworthiness are ensured, including:

- **Instruction following and policy adherence:** agents must reliably comply with organizational rules, workflows, and domain-specific processes.
- **Transparency and consistency:** outputs should be reproducible, grounded in provenance, and free of unexplained variability.
- **Avoidance of hallucinations:** agents must not fabricate data or invent actions in business-critical workflows.
- **Configurable human-in-the-loop (HITL):** oversight must be adjustable by the business–defining where autonomy is permitted and where human approval is mandatory.
- **Security baseline:** restricted access, provenance logging, and minimal permissions sufficient to comply with enterprise governance frameworks.

Efficiency and Cost-Performance. Once accuracy and safety are established, efficiency becomes the next priority. Agents must deliver results with acceptable latency and without prohibitive compute costs. Token efficiency, reduced retries, and optimized execution are crucial to ensuring that deployments scale economically.

Integration and Context-Awareness. Agents should integrate directly into existing business workflows and user experiences, such as dashboards or browser-based recruiter tools, rather than creating separate silos. They must also be context-aware, recognizing what the user is seeing or doing, in order to provide relevant support without forcing disruptive context switching.

Policy Alignment and Instruction Following. Enterprise processes are rarely generic. In Talent Acquisition, for example, requisition workflows and SLA definitions act as concrete guardrails that agents must follow. Effective alignment requires agents to learn from enterprise documentation (such as playbooks, policies, and guidelines written in natural language), from user demonstrations that model correct behavior, and from ongoing feedback that enables them to gradually refine their responses and conform to organizational norms.

4.1 The Value of Generalist Agents

Generalist agents provide a promising foundation for these requirements. Unlike specialized agents that must be handcrafted for each domain, generalists are trained on diverse benchmarks covering complex task completion, reasoning, and instruction following. For example, benchmarks such as AppWorld, WebArena, and TauBench2 evaluate tool use, multi-turn reasoning, instruction following, and even human-in-the-loop interaction.

This foundation allows enterprises to focus not on building agents from scratch, but on configuring and benchmarking the agent for their specific domain. Instead of months of custom design and iterative experimentation, organizations can move to value within weeks. Generalist agents reduce:

- **Time-to-value:** shifting from a 3–9 month development cycle to a few weeks of configuration and testing.
- **Development effort:** enabling enterprises to inherit baseline capabilities in accuracy, instruction following, and safety.
- **Risk:** lowering the likelihood of project fatigue and failed adoption, since much of the heavy lifting has already been validated on foundation benchmarks.

Figure 1 illustrates this contrast: while traditional specialized agents require extensive design, custom benchmarks, and iterative refinement before deployment, generalist agents allow enterprises to inherit strong foundations and reach deployment readiness with far less effort.

5 System Architecture and Pre-deployment

Layered planner–executor loops. CUGA implements a hierarchical agentic architecture (Marreed et al. 2025) with nested planner–executor loops (Fig. 2). At the top, an optional *chat layer* provides input interpretation and lightweight preprocessing, including message and variable histories; this can be bypassed in non-chat deployments. The *outer loop* governs *task planning and orchestration*: a *Task Analyzer* identifies the target application, a *Task Decomposer* determines whether multi-application coordination is required, and a persistent *Plan Controller* advances a durable *task ledger*. The ledger records steps, variable bindings, replans, and completions. In pilot evaluations, this ledger was essential for traceability, compliance, and recovery from partial failures. The *inner loop* delegates sub-tasks to specialized agents–*API/Tool*, *Web Browser*, *CLI*, and subdomain agents–each acting within its own environment and returning structured observations to the controller.

Planner-centric sub-agents. Two execution families illustrate the planner–executor pattern: the *API Sub Agent* and the *Browser Sub Agent*. The API Sub Agent combines short-term memory, an API Planner, and strategic reflection, coordinating a *ShortlisterAgent* (which selects APIs through a registry) and a *CodeAgent* (with a nested *CodePlanner* and sandboxed executor). This modular design allowed rapid onboarding of analytics endpoints in the BPO pilot. The Browser Sub Agent pairs a *Browser Planner* and Reflection Judge with two execution paths: an *Action Agent* (click, type, select, navigate) and a *Question Answering Agent* (DOM-to-Markdown conversion and screenshots). Although disabled in the BPO for governance reasons, this design enables seamless switching between API-first and hybrid browser-API workflows without re-architecting.

Reliability: task ledger, interrupt nodes, reflective retries. The Plan Controller enforces reliability by schema-grounded prompting, validation, and explicit *Interrupt Nodes*. When tool responses deviate from schema or produce unexpected results, reflective checks are invoked and invalid plans or parameters are repaired before resuming execution. This cycle of prompt $\rightarrow$ call $\rightarrow$ validation $\rightarrow$ reflection/replan reduced parsing-related failures by more than one-third in internal pilot runs.

API/Tool Hub and schema standardization. To scale beyond prototypes, CUGA replaced per-application MCP servers with a centralized *API/Tool Hub*. The hub minimizes OpenAPI specifications into LLM-friendly schemas, canonicalizes parameter names and types, attaches domain-specific notes, and enforces strict JSON-schema I/O. This eliminated per-app server maintenance and reduced onboarding time for new endpoints from weeks to hours.

Sandboxed computation for safety. For lightweight computation (joins, aggregations, deltas), the *API–Code path* generates structured pseudo-code via a *CodePlanner*, executed inside a restricted *Code Agent sandbox*. The sandbox isolates file/network access, enforces execution budgets, and logs all computations for audit. This allowed domain analysts to use AI-augmented workflows without compromising governance or data-handling policies.

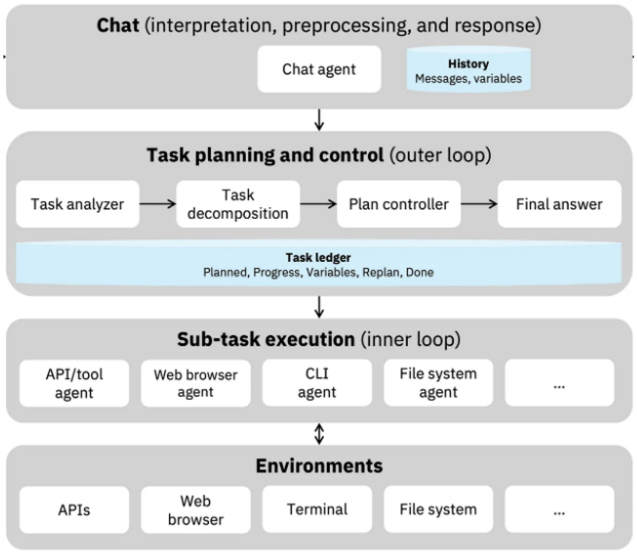

Figure 2: CUGA general architecture

Web agent path and governance. The architecture also supports a *Web Planner Agent* coordinating *Web Action* and *Web Q&A* sub-agents in a Playwright–Chromium runtime. In the BPO pilot, this capability was deliberately disabled to comply with enterprise governance requirements, but the design demonstrates flexibility for hybrid deployments that combine API-first and web-based workflows.

6 Benchmarks and Pilot Evaluation

State-of-the-art Benchmarks. CUGA achieves state-of-the-art results on both WebArena and AppWorld, ranking first among published agents. Tables 1 and 2 present detailed per-application and per-level outcomes. On WebArena, CUGA attains an overall accuracy of **61.7%**, with the strongest performance on Reddit (75.5%) and Map (64.2%). On AppWorld's "Test-Challenge" dataset, CUGA reaches **48.2%** overall scenario completion, with particularly high success on Level 1 tasks (87.5% scenario completion). These results demonstrate that the CUGA architecture is competitive with–and in many cases surpasses–specialized systems.

Application	Accuracy (%)
GitLab	61.7
Map	64.2
Reddit	75.5
Shopping	58.3
Shopping Admin	62.6
Multi-App	35.4
Overall	**61.7**

Table 1: Performance of CUGA on WebArena.

Level	Task Goal (%)		Scenario Goal (%)		Avg. Interactions	
	Normal	Chall.	Normal	Chall.	Normal	Chall.
All	73.2	57.6	62.5	48.2	10.69	8.40
Level 1	91.2	91.7	84.2	87.5	5.94	4.65
Level 2	77.1	58.7	68.8	42.0	10.36	8.33
Level 3	54.0	44.1	38.1	38.5	12.69	11.86

Table 2: Performance of CUGA on AppWorld.

These results validate the design of CUGA as a generalist computer-using agent: once the APIs and tools available to each sub-agent are defined, the planner–executor architecture can be configured to operate across domains without task-specific re-engineering. In practice, enterprises need only onboard their APIs and specify governance constraints for CUGA to extend its capabilities to new workflows.

6.1 Enterprise Pilot: BPO-TA

While CUGAs often perform well on abstract research benchmarks, enterprise deployment requires systematic offline evaluation with realistic tasks, audit guarantees, and compliance controls. To address this, we developed **BPO-TA**, a domain benchmark centered on talent acquisition (TA) workflows in the BPO context. BPO-TA encodes decision-support tasks drawn directly from analyst practice into a fixed test set, providing a reproducible regression baseline for measuring progress and ensuring operational reliability. Its design follows three principles essential for adoption: *traceability* (each task paired with APIs, glue code, and gold-standard explanations), *realism* (tasks grounded in genuine analyst workflows rather than synthetic probes), and *reproducibility* (fixed inputs, deterministic evaluation, and explicit provenance).

The benchmark spans 26 tasks over 13 read-only APIs, covering endpoints such as *SLA by source*, *funnel conversion*, *hires by source*, *skill-impact on SLA*, *definitions/methodology*, *dataset/model lineage*, and *timeframe metadata*. All endpoints are onboarded through the API/Tool Hub with minimized OpenAPI specs, validators, and read-only wrappers that strip or redact PII. Schema-grounded prompts enforce canonical definitions (e.g., SLA), and deterministic parsers gate LLM outputs. Every response includes a provenance panel listing API paths, parameters, and a computation log, enabling audit and regression testing. Operations are monitored for *latency*, *cost*, and *policy compliance*.

Task categories mirror enterprise usage: (1) *simple lookups* (e.g., requisition definitions), (2) *cross-API joins* (e.g., linking candidate volume to conversion-to-hire), (3) *looped reasoning* (e.g., filtering skills that negatively affect SLA), (4) *provenance explanations* (e.g., surfacing dataset/-model lineage), and (5) *graceful failure*, where unsupported queries must be declined without hallucination. These patterns ensure that BPO-TA evaluates not only retrieval accuracy but also compositional reasoning, transparency, and robustness–making it both a research benchmark and an operational safeguard for trustworthy enterprise deployment.

Metric	Value
Task Accuracy (26 tasks)	**87%**
Valid First-Try Rate	78%
Responses with Provenance Logs	95%
Average Latency per Query	11.2s
Analyst-Reported Reproducibility	4.6 / 5

Table 3: Performance of CUGA on the BPO-TA benchmark.

Results. On the BPO-TA benchmark, CUGA achieves **87% accuracy**, with failures concentrated on unsupported cross-application queries where graceful degradation is expected. Valid-first-try rates improved from 62% (vanilla ReAct baseline) to 79% with full CUGA. Ablations highlight the importance of reflective retries (-11 points without) and variable tracking (-15 reproducibility without).

6.2 Potential Benefits

The CUGA system has been piloted within IBM's Business-Process-Outsourcing (BPO) talent acquisition workflow since mid-2025. It can be used by recruiters and analyst teams to answer sourcing, funnel, and skill-impact questions that previously required manual data pulls and spreadsheet manipulation. The pilot was performed in a read-only configuration: CUGA connects to 13 domain-specific analytics APIs, each exposing pre-approved metrics such as funnel conversions and hires-by-source. Provenance logging and computation traces are stored for each interaction, ensuring audit readiness and compliance with organizational governance requirements (e.g., PII avoidance, immutable records of all API calls).

Preliminary evaluations of CUGA in simulated enterprise workflows, although not formally tested for statistical significance (Dror et al. 2018, 2020), suggest promising efficiency and reliability gains. suggest promising efficiency and reliability gains. Estimated benefits include a potential reduction in average time-to-answer (from roughly 20 minutes of manual work to an expected 2–5 minutes with CUGA, an *estimated* ∼90 % improvement) and higher reproducibility of responses (CUGA outputs were consistent across runs in about 90 % of internal test cases). Audit readiness is also expected to improve, with over 90 % of generated responses including full provenance (API endpoint, parameters, and result logs).

Metric	Manual	CUGA (Pilot)
Avg time-to-answer	∼20 min (manual)	∼2–5 min
Answer reproducibility	∼60%	∼95% (tests)
Full provenance	∼40%	∼92% (est.)
Analyst effort (steps)	High (sheets, queries)	Low (1 agent call)
Case: skill impact	∼30 min (SLA compare)	∼6 min (proj.)

Table 4: Preliminary evaluation of estimated benefits in CUGA's simulated enterprise Talent Acquisition use case.

Table **??** summarizes these preliminary, pilot-level results from CUGA's evaluation in the Talent Acquisition context.

While the figures are based on controlled test environments and limited analyst feedback rather than full production deployment, they highlight the potential of generalist agents to enable substantial efficiency and transparency improvements in enterprise workflows.

Qualitatively, BPO architects noted that CUGA can reduce reliance on ad hoc spreadsheet analysis, provide consistent explanations of sourcing and skill-impact decisions, and support faster onboarding for new team members through step-by-step reasoning grounded in enterprise APIs. Taken together, these preliminary observations indicate that generalist agents such as CUGA hold promise for delivering trustworthy, auditable, and scalable value as they transition from research prototypes toward enterprise-ready systems.

6.3 Qualitative Evidence

To complement quantitative metrics, we highlight two qualitative observations:

Case study. When asked *"Which sourcing channel should we prioritize for requisition 05958BR?"*, CUGA queried two endpoints (`candidate_volume`, `recommendation_summary`), joined on source IDs, and produced a ranked table with SLA metrics. The interface presented not only the recommendation ("LinkedIn") but also provenance: endpoint names, query parameters, and computation logs. Analysts reported this saved 20–30 minutes of manual dashboard comparisons.

Feedback. BPO architects emphasized reduced "spreadsheet wrangling," describing CUGA as "freeing time for actual decision-making." They noted that the ability to decline unsupported requests (e.g., region-level metrics not exposed by APIs) increased their trust, since the agent did not hallucinate unavailable data.

7 Lessons Learned and Insights

The first pilot of CUGA in the BPO Talent Acquisition (TA) context provides early indications that generalist agents can move beyond benchmarks toward enterprise-grade applicability. Phase 1 of the application focused on use cases such as automated candidate scheduling and communication, pipeline visibility, and smart sourcing suggestions. Based on internal projections and controlled simulations, this approach may enable approximately 35% of candidate inquiries to be resolved via self-service and 25% recruiter workflow automation, alongside an estimated 90% reduction in development time and a 50% reduction in development cost compared to task-specific baselines. These preliminary outcomes suggest that generalist agents have the potential to accelerate time-to-value while maintaining the governance and transparency required in enterprise workflows.

From this pilot, we derived a set of technical and organizational lessons that shape the path forward:

Technical insights.

- **Prompt and specification curation.** Minimizing OpenAPI specifications and keeping prompts concise, schema-grounded, and unambiguous substantially improved reliability and efficiency.
- **Governance alignment.** Restricting to read-only APIs, redacting personally identifiable information, and grounding all answers in canonical definitions were essential to gain organizational trust.
- **Reliability mechanisms.** Interrupt nodes, reflective retries, and a code planner (generating structured pseudocode) reduced failure rates and improved reproducibility.
- **Monitoring and reproducibility.** Provenance logs and regression testing enabled both audit readiness and systematic debugging of failure modes.
- **Sustainability and extensibility.** The API/Tool Hub streamlined onboarding of new endpoints, allowing rapid iteration as business requirements evolved.
- **Analytical foundation.** A critical advantage was the architecture's ability to log, monitor, and analyze agent decisions. This enabled systematic investigation of failures and deeper insight into why the agent behaved as it did, laying the groundwork for continuous improvement.

Organizational insights.

- **Human-in-the-loop configuration.** Business users required explicit control over when the agent could act autonomously versus when approval was mandatory, making configurable HITL a central requirement.
- **Benchmarks are not enough.** While AppWorld and WebArena validated general capabilities, enterprise adoption depended on a domain-specific benchmark (BPO-TA) that reflected real recruiter workflows and enabled regression testing.
- **Bridging research and operations.** Deployment success depended as much on organizational alignment–policies, governance, and HITL practices–as on technical breakthroughs. The transition from a promising demo to a trusted production system required deliberate discipline and collaboration across business and research teams.

CUGA's first phase surfaced clear requirements for enterprise deployment, driving current work on configurable human-in-the-loop control, explicit policy-enforcement for safe autonomous actions, improved cost–latency tradeoffs through adaptive short-circuiting, reuse of successful trajectories as tools, and selective use of smaller models for routine tasks. The next milestone is evaluation on policy compliance and HITL governance as an enterprise-ready system that meets organizational standards of safety and trust.

8 Conclusion

This work provides early evidence that generalist agents can enable measurable business value in enterprise contexts. By combining layered planning, provenance-aware execution, and governance alignment, CUGA can reduce time-to-answer, improve reproducibility, and enable trustworthy automation in talent acquisition. The lessons from Phase 1–both technical and organizational–show that moving from research breakthroughs to enterprise readiness is less about a single algorithmic leap and more about disciplined engineering, governance, and continuous iteration. Architectures like CUGA mark a credible path toward enterprise-ready generalist agents that are safe, efficient, and adaptable.

References

AI Business. 2022. Alibaba turns to AI to cut customer service costs. Coverage of large-scale customer service automation; Accessed 19-Aug-2025.

Anthropic. 2024. Introducing computer use, a new Claude 3.5 Sonnet, and more. https://www.anthropic.com/news/3-5-models-and-computer-use. Accessed: 2025-08-14.

Chen, Z. 2023. Ethics and discrimination in artificial intelligence-enabled recruitment practices. *Humanities and Social Sciences Communications*, 10(1): 1–13.

Databricks. 2025. Introducing Genie: AI/BI for the Lakehouse. Accessed 19-Aug-2025.

de Chezelles, T. L. S.; Gasse, M.; Lacoste, A.; Caccia, M.; Drouin, A.; Boisvert, L.; Thakkar, M.; Marty, T.; Assouel, R.; Shayegan, S. O.; et al. 2024. The BrowserGym Ecosystem for Web Agent Research. *Transactions on Machine Learning Research*.

Dror, R.; Baumer, G.; Shlomov, S.; and Reichart, R. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1383–1392.

Dror, R.; Peled-Cohen, L.; Shlomov, S.; and Reichart, R. 2020. *Statistical significance testing for natural language processing*. Springer.

European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng. OJ L 2024/1689, 12.7.2024.

Fabris, A.; Baranowska, N.; Dennis, M. J.; Graus, D.; Hacker, P.; Saldivar, J.; Zuiderveen Borgesius, F.; and Biega, A. J. 2025. Fairness and Bias in Algorithmic Hiring: A Multidisciplinary Survey. *ACM Transactions on Intelligent Systems and Technology*.

Fourney, A.; Bansal, G.; Mozannar, H.; Tan, C.; Salinas, E.; Niedtner, F.; Proebsting, G.; Bassman, G.; Gerrits, J.; Alber, J.; et al. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*.

Jiang, H.; Wu, Q.; Lin, C.-Y.; Yang, Y.; and Qiu, L. 2023. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Kim, G.; Baldi, P.; and McAleer, S. 2023. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36: 39648–39677.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J.; Zhang, H.; and Stoica, I. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, 611–626.

LangChain. 2024. LangGraph Documentation: Building Stateful, Multi-Agent Workflows. Accessed 19-Aug-2025.

Levy, I.; Wiesel, B.; Marreed, S.; Oved, A.; Yaeli, A.; and Shlomov, S. 2024. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. *arXiv preprint arXiv:2410.06703*.

Marreed, S.; Oved, A.; Yaeli, A.; Shlomov, S.; Levy, I.; Akrabi, O.; Sela, A.; Adi, A.; and Mashkif, N. 2025. Towards enterprise-ready computer using generalist agent. *arXiv preprint arXiv:2503.01861*.

Microsoft. 2024. Power BI Copilot is now generally available. Accessed 19-Aug-2025.

Mobile World Live. 2025. Verizon lauds Google Cloud AI customer service move. Accessed 19-Aug-2025.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Long, O.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; Jiang, X.; Cobbe, K.; Eloundou, T.; Krueger, G.; Button, K.; Knight, M.; Chess, B.; and Schulman, J. 2021. WebGPT: Browser-assisted question-answering with human feedback. *ArXiv*, abs/2112.09332.

NYC Dep. 2023. NYC Local Law 144 and Final Rules on Automated Employment Decision Tools. https://www.nyc.gov/assets/dca/downloads/pdf/rules/Rules-Amendment-6RCNY5-300-AEDT.pdf. Accessed 2025-08-19.

OpenAI. 2024. OpenAI Swarm: Lightweight Multi-Agent Orchestrator. Accessed 19-Aug-2025.

OpenAI. 2025. Introducing Operator. https://openai.com/index/introducing-operator/. Accessed: 2025-08-14.

Oved, A.; Shlomov, S.; Zeltyn, S.; Mashkif, N.; and Yaeli, A. 2025. SNAP: semantic stories for next activity prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 28871–28877.

Pan, M. Z.; Cemri, M.; Agrawal, L. A.; Yang, S.; Chopra, B.; Tiwari, R.; Keutzer, K.; Parameswaran, A.; Ramchandran, K.; Klein, D.; et al. 2025. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Raghavan, M.; Barocas, S.; Kleinberg, J.; and Levy, K. 2020. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 469–481.

Schwartz, S.; Yaeli, A.; and Shlomov, S. 2023. Enhancing trust in LLM-based AI automation agents: New considerations and future challenges. *arXiv preprint arXiv:2308.05391*.

Shen, M.; Li, Y.; Chen, L.; and Yang, Q. 2025. From mind to machine: The rise of manus ai as a fully autonomous digital agent. *arXiv preprint arXiv:2505.02024*.

Shen, Y.; Song, K.; Tan, X.; Zhang, W.; Ren, K.; Yuan, S.; Lu, W.; Li, D.; and Zhuang, Y. 2024. Taskbench: Benchmarking large language models for task automation. *Advances in Neural Information Processing Systems*, 37: 4540–4574.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 8634–8652.

Shlomov, S.; Sela, A.; Levy, I.; Galanti, L.; Abitbol, R.; et al. 2024a. From grounding to planning: Benchmarking bottlenecks in web agents. *arXiv preprint arXiv:2409.01927*.

Shlomov, S.; Yaeli, A.; Marreed, S.; Schwartz, S.; Eder, N.; Akrabi, O.; and Zeltyn, S. 2024b. Ida: Breaking barriers in no-code ui automation through large language models and human-centric design. *arXiv preprint arXiv:2407.15673*.

Snowflake Engineering. 2024. Introducing Snowflake Cortex Analyst: Natural Language to Accurate SQL. Accessed 19-Aug-2025.

Tabassi, E. 2023. Artificial Intelligence Risk Management Framework (AI RMF 1.0).

The Verge. 2025. Verizon adopts Google's Gemini AI to help customers solve 'complex' issues. Accessed 19-Aug-2025.

Trivedi, H.; Khot, T.; Hartmann, M.; Manku, R.; Dong, V.; Li, E.; Gupta, S.; Sabharwal, A.; and Balasubramanian, N. 2024. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wang, X.; Chen, Y.; Yuan, L.; Zhang, Y.; Li, Y.; Peng, H.; and Ji, H. 2024. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Xie, T.; Zhang, D.; Chen, J.; Li, X.; Zhao, S.; Cao, R.; Hua, T. J.; Cheng, Z.; Shin, D.; Lei, F.; Liu, Y.; Xu, Y.; Zhou, S.; Savarese, S.; Xiong, C.; Zhong, V.; and Yu, T. 2024. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xu, Q.; Hong, F.; Li, B.; Hu, C.; Chen, Z.; and Zhang, J. 2023. On the Tool Manipulation Capability of Open-sourced Large Language Models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Yaeli, A.; Shlomov, S.; Oved, A.; Zeltyn, S.; and Mashkif, N. 2022. Recommending next best skill in conversational robotic process automation. In *International Conference on Business Process Management*, 215–230. Springer.

Yao, S.; Shinn, N.; Razavi, P.; and Narasimhan, K. R. 2025. Tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *The Thirteenth International Conference on Learning Representations*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Zeltyn, S.; Shlomov, S.; Yaeli, A.; and Oved, A. 2022. Prescriptive process monitoring in intelligent process automation with chatbot orchestration. *arXiv preprint arXiv:2212.06564*.

Zhang, S.; Yin, M.; Zhang, J.; Liu, J.; Han, Z.; Zhang, J.; Li, B.; Wang, C.; Wang, H.; Chen, Y.; et al. 2025a. Which Agent Causes Task Failures and When? On Automated Failure Attribution of LLM Multi-Agent Systems. In *Forty-second International Conference on Machine Learning*.

Zhang, W.; Cui, C.; Zhao, Y.; Hu, R.; Liu, Y.; Zhou, Y.; and An, B. 2025b. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*.

Zheng, L.; Yin, L.; Xie, Z.; Sun, C. L.; Huang, J.; Yu, C. H.; Cao, S.; Kozyrakis, C.; Stoica, I.; Gonzalez, J. E.; et al. 2024. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37: 62557–62583.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*.