# GenHAPI: Conversational Orchestration over MCP to Unlock Complex Legacy Workflows

# GenHAPI: Conversational Orchestration over MCP to Unlock Complex Legacy Workflows

Andrea Ferracani[1][0009−0006−7567−3228], Filippo Principi[1][0009−0006−7793−6044], Pavan Kartheek Rachabathuni[1][0009−0004−0606−8118], and Marco Bertini[1][0000−0002−1364−218X]

MICC - DINFO, University of Florence, IT, Viale Morgagni 65, Firenze
info@micc.unifi.it
http://www.micc.unifi.it

**Abstract.** Legacy enterprise applications remain functionally powerful but are often difficult to use, relying on fragmented workflows, dense forms, and expert knowledge. GenHAPI introduces a *chat-to-action* orchestration layer that enables users to interact with legacy systems through natural-language conversation while preserving control, auditability, and compliance. The system translates conversational requests into structured tasks defined in a declarative intent knowledge base and executes them via the Model Context Protocol (MCP) under explicit human confirmation. GenHAPI combines conversational orchestration, parameter governance, and optional vision-based UI grounding to support mixed-initiative interaction across heterogeneous enterprise environments. We report the system architecture, an intent engineering pipeline derived from real user traces, and a multi-agent workflow for safe tool invocation. Preliminary evaluation results include system-level benchmarks and a pilot user study, indicating reduced interaction complexity and improved user control compared to traditional workflows.

**Keywords:** Conversational AI · Human–AI Interaction · Chat-to-Action Orchestration · Model Context Protocol (MCP) · Vision-Based UI Guidance · Legacy Enterprise Systems · Human-in-the-Loop Interaction · Privacy-by-Design.

## 1 Introduction

Conversational interfaces are increasingly explored as a means to reduce the complexity of interacting with digital systems. At the same time, enterprise and public-sector applications pose specific challenges that limit the applicability of fully autonomous conversational agents, particularly in terms of control, accountability, and compliance. We present **GenHAPI**, a conversational orchestration layer designed to mediate between natural-language interaction and enterprise system execution. GenHAPI translates user requests into declaratively defined intents, incrementally resolves and validates required parameters, and executes operations through auditable tool calls using the Model Context Protocol (MCP).

Rather than replacing existing systems, GenHAPI overlays legacy applications with a *human-in-the-loop* chat-to-action interaction that preserves user control and traceability.

This paper makes the following contributions:

- a reusable *chat-to-action* orchestration pattern for legacy enterprise systems based on MCP;
- a data-driven intent engineering pipeline that derives declarative task schemas from real user interaction traces;
- a conversational, multi-agent workflow for parameter governance and safe execution with explicit human confirmation;
- an initial evaluation combining system-level benchmarks, a preliminary assessment of vision-based grounding accuracy, and a pilot user study.

## 2 Background and Motivation

Enterprise and public-sector organizations increasingly rely on complex legacy information systems that encapsulate critical business logic, regulatory constraints, and institutional knowledge. Although these systems are often robust and functionally rich, they are typically characterized by dense user interfaces, fragmented workflows, and a strong dependence on procedural expertise. As a result, effective use frequently requires specialized training and continuous practice, creating barriers for novice users, occasional users, and individuals with accessibility needs. At the same time, recent advances in large language models (LLMs) have made conversational interaction a viable paradigm for accessing complex digital services. Conversational interfaces promise to lower the entry barrier by allowing users to express goals in natural language rather than navigating rigid interface structures. However, applying this paradigm to enterprise environments introduces requirements that go beyond those of consumer-facing assistants. In regulated and mission-critical contexts, systems must guarantee auditability, data protection, reversibility of actions, and clear accountability. Purely autonomous agents that directly operate user interfaces or execute actions without explicit confirmation often conflict with these requirements, raising concerns related to trust, safety, and compliance. Existing market solutions demonstrate that it is technically feasible to deploy LLM-based agents that interact with legacy systems via APIs while respecting strict regulatory constraints such as GDPR. Common architectural strategies include European or on-premise deployment, fine-grained access control, detailed logging, selective data retention, and explicit human supervision for sensitive operations. These approaches highlight an important design insight: in enterprise settings, conversational AI is not primarily about autonomy, but about *controlled mediation* between human intent and system action. From a Human-Computer Interaction perspective, this shift reframes the role of conversational interfaces. Rather than acting as autonomous operators, conversational systems can function as cognitive and interactional scaffolds: helping users articulate goals, retrieve relevant information, validate parameters, and understand the consequences of actions before they are executed. This aligns

with mixed-initiative interaction models, where control is dynamically negotiated between human and system, and with human-in-the-loop design principles that emphasize transparency and recoverability. Another challenge specific to legacy environments is the gap between conversational intent and executable system operations. Enterprise tasks are rarely atomic; they involve multiple parameters, dependencies on existing records, and domain-specific constraints. Bridging this gap requires explicit representations of tasks and parameters that can be validated, audited, and evolved over time. Ad-hoc prompt engineering or end-to-end autonomous agents struggle to meet these requirements, particularly when workflows must be explainable to auditors or system owners. These considerations motivate the design of **GenHAPI** as a conversational *orchestration layer* rather than a replacement for existing systems. By combining a declarative intent knowledge base, structured tool invocation through the Model Context Protocol (MCP), and explicit human confirmation rules, GenHAPI aims to reduce interactional complexity while preserving user control. The goal is not to maximize automation, but to improve usability, accessibility, and trust in complex legacy workflows, making conversational interaction a viable and responsible interface paradigm for real-world enterprise systems.

## 3 Related Work

### 3.1 Conversational Interfaces for Task Completion

Conversational interfaces have long been investigated as a means to reduce interactional complexity and lower the barrier to system use. From an HCI perspective, natural language interaction supports task articulation, clarification, and error recovery, particularly for novice or occasional users. More recent work frames conversational assistants as collaborators rather than replacements, emphasizing mixed-initiative interaction, transparency, and trust. Empirical studies by Kuang et al. [7] show how conversational modality (voice vs. text) influences how users formulate requests and evaluate system reliability, while Heo et al. [8] highlight the challenges designers face when mapping conversational interaction onto real-world, multi-channel service workflows. These challenges become particularly pronounced in enterprise settings, where conversational systems must interface with complex procedures, heterogeneous backends, and legacy interfaces.

### 3.2 LLM Tool Use and Action Orchestration

Recent advances in large language models have enabled systems that combine natural language reasoning with the invocation of external tools. ReAct, introduced by Yao et al. [1], interleaves explicit reasoning traces with action execution, enabling models to iteratively plan and act in interactive environments. Similarly, Toolformer by Schick et al. [2] shows that language models can learn, in a self-supervised manner, when and how to invoke external APIs, effectively

extending their capabilities beyond pure text generation. Related approaches such as WebGPT and SayCan further explore language-guided action execution in web-based and embodied environments, respectively Nakano et al. demonstrate that models can navigate the web through a browser interface to retrieve information [9], while Ahn et al. ground language instructions in robotic affordances to select feasible physical actions [10]. Collectively, these systems primarily frame tool use as an autonomous decision-making problem, in which the agent internally selects and executes actions to optimize task completion. In contrast, GenHAPI conceptualizes tool invocation as a conversationally mediated, mixed-initiative process, where control over task progression is shared between the system and the user rather than delegated entirely to the agent [16]. Instead of embedding tool execution solely within the model's internal reasoning loop, GenHAPI explicitly surfaces parameter validation, ambiguity resolution, and execution confirmation as part of the dialogue. This design aligns with established Human–AI Interaction guidelines that emphasize transparency, user control over system actions, and mechanisms for preventing and recovering from errors [17]. By positioning tool use as an interaction protocol rather than an autonomous control policy, GenHAPI prioritizes accountability and user agency, which are critical for enterprise and public-sector deployments where external actions must remain inspectable and human-authorized.

### 3.3 Conversational UX in Enterprise and Multi-channel Systems

Within enterprise environments, conversational interfaces must operate across multiple channels and organizational constraints. Heo et al. describe how conversational UX designers struggle with maintaining coherence between forms, flows, and conversational layers in real-world services [8]. These findings emphasize the importance of gradual intent refinement, explicit system feedback, and clear boundaries between suggestion and execution. From an HCI standpoint, such requirements align with long-standing principles of mixed-initiative interaction. Horvitz et al. [13] argue that effective intelligent interfaces dynamically balance initiative between user and system, rather than delegating full control to automation. This perspective motivates conversational systems that assist users in achieving goals while preserving agency and oversight, particularly in high-stakes or regulated contexts.

### 3.4 GUI Agents and Vision-Based Grounding

Parallel to conversational approaches, recent work investigates autonomous agents that operate directly on graphical user interfaces using visual grounding. Mind2Web [3], proposed by Deng et al., provides a large-scale benchmark for mapping natural language goals to UI actions on real-world websites. BrowserGym further systematizes this line of research by offering a unified environment for evaluating web-based agents [6]. More recent vision-based models aim to improve action grounding by explicitly parsing interface elements. Ferret–UI by You et al. [4] introduces multimodal representations for mobile UI understanding,

while OmniParser by Lu et al. [5] proposes a purely vision-based approach to GUI action prediction across platforms. These systems demonstrate impressive autonomy and performance on benchmark tasks. However, such approaches typically assume full agent control over the interface and optimize primarily for task success. As discussed by Hoffman et al., high levels of autonomy can undermine user trust if system behavior is opaque or difficult to inspect [14]. In regulated enterprise settings, this raises concerns about safety, auditability, and accountability.

### 3.5 Positioning Summary

In summary, prior work spans conversational assistants, LLM-based tool use, and autonomous GUI agents. GenHAPI positions itself at the intersection of these strands, emphasizing conversational orchestration over autonomous execution. By combining declarative intent representations, MCP-based tool invocation, and optional vision-guided assistance within a human-in-the-loop framework, GenHAPI aligns with established HCI principles of mixed initiative [13] and supports transparency and trust as discussed by Hoffman et al. [14].

The remainder of the paper details the design, implementation, and evaluation of GenHAPI. Section 4 introduces the system architecture and interaction paradigm, outlining how conversational input is translated into auditable enterprise actions. Section 5 describes the core architectural components, including intent engineering from user traces, the multi-agent orchestration workflow, and MCP-based execution. Section 6 presents the vision-based UI module for grounded guidance. Section 7 reports system-level benchmarks and a pilot user study. The final sections discuss implications, limitations, ethical considerations, and future directions.

## 4 System Overview

**GenHAPI** is a conversational orchestration layer that enables users to perform complex operations on legacy enterprise systems through natural-language interaction. Rather than acting as an autonomous agent or replacing existing interfaces, GenHAPI overlays current web-based systems with a controlled *chat-to-action* interaction that mediates between user intent and backend execution. The system translates free-form user requests into structured tasks defined in a declarative intent knowledge base. Through an incremental, mixed-initiative dialogue, GenHAPI identifies the intended operation, resolves and validates required parameters, and executes the task via auditable tool calls exposed through the Model Context Protocol (MCP)[1]. Dispositive actions are always gated by explicit user confirmation, supporting accountability, reversibility, and trust.

[1] MCP is an open protocol for LLM–tool interaction via structured manifests and tool calls; it separates conversation from execution. Docs: `https://modelcontextprotocol.io`

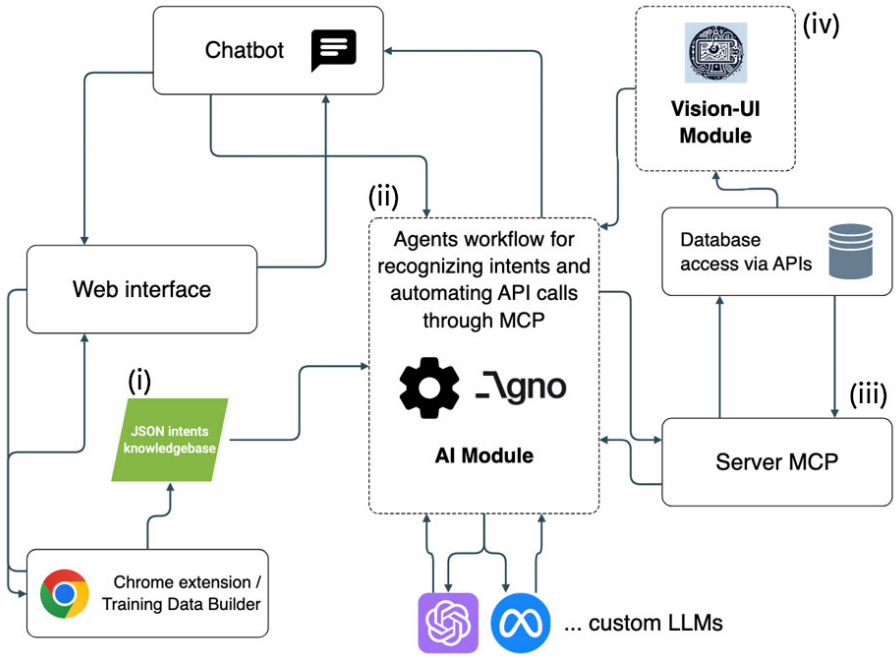


**Fig. 1.** GenHAPI system architecture.

At a high level, the GenHAPI architecture (Fig. 1) comprises four cooperating components: (i) a declarative intent layer that formalizes enterprise tasks and their parameters, (ii) a conversational orchestrator that manages intent recognition, parameter governance, and dialogue flow, (iii) a tool execution layer that invokes enterprise services via MCP and records auditable traces, and (iv) an optional Vision-UI module, described in in Section 6, that provides grounded, step-by-step visual guidance on existing interfaces when API coverage is incomplete.

## 5 Architecture and Implementation

### 5.1 Declarative Intent Modeling from Interaction Traces

GenHAPI is built around a *declarative intent knowledge base* that defines which operations can be performed conversationally. Each intent is specified as a structured JSON schema describing: the task semantics, required and optional parameters, parameter types and sources, validation rules, and the target MCP tool to be invoked. This explicit representation enables parameter governance, inspection, and evolution over time, addressing limitations of ad-hoc prompt-based tool use. The intent schemas are derived through an *intent engineering pipeline* informed by real user interaction traces. A training-only browser extension captures user–system interaction on authorized domains, including UI events

(e.g., clicks, form edits, navigation) and the corresponding legacy web API calls. After on-device pseudonymization and least-privilege filtering, these traces are processed to correlate user actions with backend operations, yielding candidate intents and their parameter fields. This process grounds conversational capabilities in existing enterprise workflows, ensuring alignment between natural-language requests and operational primitives.

### 5.2 Conversational Orchestrator and Mixed-Initiative Workflow

The conversational logic is implemented as a stateful, multi-agent workflow in Python, built on the Agno orchestration framework for composing agent workflows and maintaining session state[2]. The orchestrator maintains a persistent session state and coordinates a small set of specialized agents, each responsible for a distinct phase of the interaction. This separation of concerns improves robustness, maintainability, and interpretability compared to monolithic conversational agents.

The workflow follows a staged pipeline:

(1) **Intent recognition.** An intent recognizer maps the user utterance to one of the declaratively defined intents and extracts candidate parameter values. When no intent is confidently identified, the system falls back to a conversational agent to handle generic dialogue.
(2) **External parameter resolution.** Parameters marked as originating from enterprise archives (e.g., customers, documents) are resolved via controlled database queries. The system handles unambiguous matches automatically and manages ambiguity through explicit disambiguation prompts.
(3) **User-driven parameter completion.** Remaining required parameters are requested one at a time using predefined, intent-specific questions. Inputs are validated and normalized (e.g., natural-language dates converted to ISO format, numeric checks).
(4) **Confirmation and execution.** Once all parameters are resolved, the system presents a concise pre-action summary and requires explicit user confirmation before executing the task.

Throughout the interaction, the orchestrator supports interruption handling, allowing users to cancel, reformulate, or switch tasks without losing conversational coherence. This design operationalizes mixed-initiative interaction: the system actively structures the task and enforces constraints, while the user retains control over commitments and execution.

### 5.3 MCP Integration and Auditable Execution

GenHAPI executes operations through MCP, which exposes enterprise capabilities as typed tools with explicit input and output schemas. The conversational orchestrator composes a normalized execution payload and invokes the corresponding MCP tool only after user confirmation. This strict separation between

[2] `https://github.com/agno-agi/agno`

conversational reasoning and execution ensures that all actions are traceable and inspectable. For each interaction, the system records an auditable trace including the selected intent, parameter sources and validation outcomes, confirmation decisions, and tool responses. This trace supports debugging, compliance checks, and post-hoc analysis, which are essential in regulated enterprise environments. When backend APIs are incomplete, GenHAPI can optionally switch to a guided UI mode that assists users visually while keeping dispositive actions within the same confirmation and execution framework.

## 6 Vision-UI Module for Grounded Guidance

While GenHAPI primarily operates through conversational orchestration and structured tool invocation, certain enterprise environments expose only partial or unstable APIs. In these cases, purely backend-driven execution is insufficient to complete user tasks. To address this limitation without granting the system full autonomous control over the interface, GenHAPI integrates an optional *Vision-UI module* that provides grounded, step-by-step visual guidance directly on the existing web interface. The Vision-UI mode can be explicitly activated by the user from within the conversational chat interface, or automatically triggered by the system once an intent has been recognized and determined to be unsuitable for reliable MCP-based execution. In both cases, intent recognition and parameter resolution are completed conversationally before switching interaction modality. Rather than executing UI actions autonomously, the Vision-UI module generates visual overlays (through the Driver.js library[3]) that indicate where and how the user should interact with the interface, preserving human supervision and control. Fig. 2 shows an example of conversational and vision-guided interaction in a fleet management use case.

The Vision-UI module in GenHAPI supports two complementary grounding approaches that differ in how interaction structure is derived and how guidance steps are generated. In this context, a *guidance step* denotes a single, ordered instruction that visually highlights a specific UI element and suggests the corresponding user action required to progress toward task completion. The first approach (Sec. 6.1) relies on interaction traces collected from successful user sessions and treats them as procedural ground truth. The second approach (Sec. 6.2) adopts a vision-based strategy that combines UI parsing with large language model inference to generate plausible guidance when such traces are unavailable or unreliable. Together, these approaches balance precision and robustness under interface variability.

### 6.1 Tracking-Based Grounding as Procedural Ground Truth

GenHAPI supports a *tracking-based grounding* strategy derived from real user sessions that successfully completed a target task. During an analytics phase,

[3] `https://driverjs.com/`

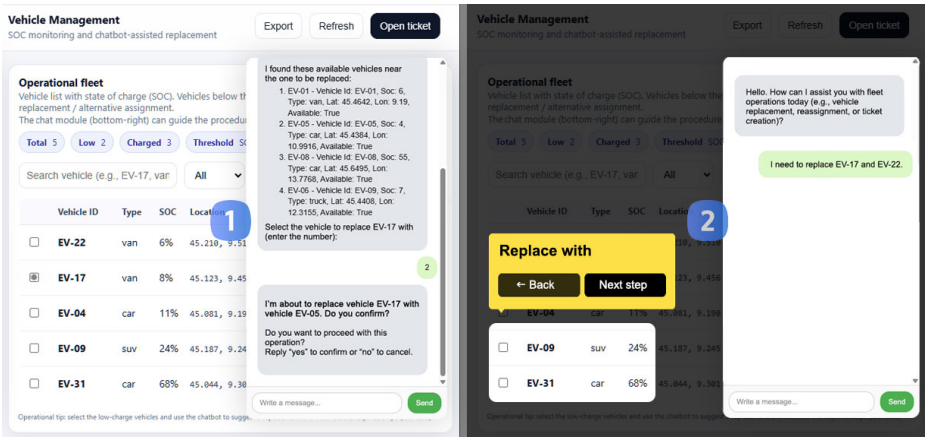


**Fig. 2.** Conversational and vision-guided interaction in a vehicle management use case. (**1**) Conversational mode: the user interacts via chat to manage low-charge electric vehicles, request replacements, and confirm actions through incremental parameter clarification and explicit confirmation. (**2**) Vision-UI mode: once the intent is recognized, the system overlays guided visual cues on the existing fleet management interface to assist vehicle selection and replacement, while preserving manual user control.

interaction sessions from authorized users are recorded only when the task reaches a valid completion state. Each session is associated with a previously recognized intent and stored as a structured JSON trace. These traces encode the ordered sequence of user interactions, including page transitions, actionable UI elements, and interaction metadata. Each step references concrete DOM elements through stable selectors and contextual attributes, resulting in an explicit procedural representation of how a given intent is executed in practice. At runtime, the recorded JSON traces are replayed as guided visual tours as soon as a specific intent from the user is recognized. Because the guidance is grounded in interaction sequences that are known to lead to successful task completion, tracking-based grounding constitutes a form of procedural ground truth. The highlighted elements, their ordering, and their spatial localization directly correspond to observed user behavior, yielding high precision and deterministic execution paths. The main limitation of this approach lies in its dependency on interface stability. Changes in layout or DOM structure require the collection of new sessions to regenerate valid traces.

### 6.2 Vision-Based Grounding via UI Parsing and LLM Inference

To improve robustness under UI variability and reduce reliance on pre-recorded traces, GenHAPI also supports a *vision-based grounding* strategy. In this mode, the system captures a screenshot of the active interface and processes it using a UI parsing model (OmniParser [4]), which extracts a structured description of

[4] `https://github.com/microsoft/OmniParser`

visible elements, including their type, textual labels, and bounding boxes. This step is critical because large language models are effective at interpreting the *semantic role* of interface elements (e.g., identifying which button is likely to trigger a given action), but are not reliable at inferring precise spatial localization directly from raw images or DOM hierarchies. Without explicit spatial grounding, LLM-generated coordinates tend to be approximate or inconsistent across layouts. The structured output of OmniParser provides explicit spatial anchors that are then combined with the recognized intent and passed to a language model. The LLM infers a plausible sequence of interaction steps and associates each step with one or more candidate UI elements, expressed in normalized screen coordinates. These coordinates are converted into pixel-space overlays on the client side and rendered as visual guidance. Empirical testing (see Section 6) reveals common failure modes in vision-based grounding, including inaccurate element localization, selection of visually similar but incorrect elements, omission of intermediate steps, and generation of redundant or reordered sequences. These errors highlight the distinction between semantic understanding and spatial grounding in current multimodal systems.

### 6.3 Complementary Roles and Safety Considerations

The two grounding strategies play complementary roles within GenHAPI. Tracking-based grounding prioritizes precision and reliability and is treated as the preferred mode whenever valid interaction traces are available. Vision-based grounding serves as a flexible fallback when such traces are unavailable or when interfaces vary across deployments. Crucially, both modes are integrated within the same safety framework. The system never performs clicks or form submissions autonomously, and visual guidance remains advisory. By separating semantic intent understanding from spatial execution and by keeping the user in control of all actions, GenHAPI mitigates the risks associated with erroneous visual grounding while extending conversational assistance to otherwise inaccessible legacy interfaces.

### 6.4 Preliminary Accuracy Assessment

We evaluated *guidance step localization and ordering accuracy* for the vision-based grounding module through a preliminary assessment on a small set of representative tasks ($N = 10$ interaction sessions). For each session, the system-generated visual guidance was compared against the corresponding tracking-based ground-truth interaction trace derived from successful user executions. We report two complementary metrics. *Element localization* is assessed using an Intersection over Union (IoU) criterion between the predicted and ground-truth bounding boxes, counting a step as correct when IoU exceeds a fixed threshold ($\geq 0.3$). *Sequence correctness* is measured using Exact Match (EM), which requires the predicted sequence of interaction steps to exactly match the ground-truth order needed to complete the task. Across the evaluated sessions, vision-based grounding achieved approximately 70% IoU-based localization accuracy and 58%

exact sequence match accuracy. Although these results are based on a limited sample, they are consistent with observed qualitative failure modes and support the use of vision-based grounding as a flexible fallback mechanism rather than as a replacement for trace-based guidance.

## 7 Evaluation

### 7.1 System-Level Benchmarks

We evaluated GenHAPI at the system level using a working prototype exposing MCP tools for invoicing, document retrieval, and representative multi-step enterprise workflows characterized by parameter dependencies and conditional execution. Anonymized interaction logs were used to seed the declarative intent knowledge base underlying conversational orchestration. End-to-end task execution was tested using both a cloud-based LLM and an on-premise open-weight model, following identical orchestration logic. The benchmarks focus on orchestration-level capabilities and interaction robustness rather than domain-specific task performance or UI-level perception.

**Preliminary lab metrics** (internal benchmarks):

- **Intent recognition accuracy:** $> 90\%$ on seeded intents across evaluated tasks.
- **Parameter resolution completeness:** $> 95\%$ of required parameters resolved within at most two clarification turns, including external database lookups.
- **Execution robustness:** $> 98\%$ of generated JSON payloads and MCP tool calls were valid; remaining errors were handled through coordinator-driven recovery.
- **Latency (p95):** within enterprise-acceptable bounds for both cloud-based and on-premise deployments, with lower variance observed for on-premise execution.

For tasks that could not be reliably completed through MCP-based execution alone, the system successfully transitioned to guided UI interaction while preserving conversational context, parameter state, and confirmation semantics (see Section 6).

### 7.2 Pilot User Study

We conducted a pilot user study to obtain initial evidence on the usability and perceived control of GenHAPI compared to traditional interaction with legacy enterprise systems. The study was exploratory in nature and aimed at assessing interaction quality rather than statistical generalization.

**Participants.** We recruited $N = 8$ participants (4 novice and 4 experienced users) with prior exposure to enterprise web applications but no previous experience with GenHAPI.

**Tasks and procedure.** Participants completed a set of representative enterprise tasks, including document retrieval and multi-step workflows involving parameter

dependencies. Each participant performed the tasks in two conditions using a within-subject design: (i) the original legacy interface and (ii) the GenHAPI conversational interface. Task order was counterbalanced to mitigate learning effects.

**Measures.** Perceived usability was assessed using the System Usability Scale (SUS) [15]. In addition, short semi-structured interviews were conducted to collect qualitative feedback on perceived control, trust, and clarity of system behavior.

**Results.** SUS responses indicated higher perceived usability for the GenHAPI condition compared to the legacy interface, particularly for novice users. Qualitative feedback highlighted the benefits of incremental parameter clarification, explicit confirmation before execution, and the ability to understand the consequences of actions prior to system invocation. Participants also reported increased confidence when interacting with complex workflows through the conversational interface.

**Qualitative observations.** Participants emphasized the value of "asking only what is missing" and appreciated the option to switch to guided visual interaction for complex tasks while retaining manual control over final actions.

## 8 Discussion

The results of the system-level benchmarks and the pilot user study suggest that GenHAPI effectively reduces interactional complexity in legacy enterprise workflows while preserving user control. From an HCI perspective, the conversational orchestration model supports accessibility by allowing users to express goals in natural language rather than navigating dense and fragmented interfaces. This is particularly beneficial for novice and occasional users, who reported clearer understanding of required inputs and system behavior. The explicit human-in-the-loop design plays a central role in fostering trust. Incremental parameter clarification and confirmation before execution help users reason about the consequences of actions, aligning with mixed-initiative interaction principles. Rather than shifting control to an autonomous agent, GenHAPI maintains a negotiated balance between human intent and system execution, supporting transparency and recoverability. The integration of optional vision-based guidance further extends this model to environments where backend automation is incomplete. By providing grounded visual cues without autonomously performing actions, the system preserves user agency while reducing procedural burden. Together, these results indicate that conversational orchestration can function as an effective interaction scaffold for complex enterprise systems, complementing rather than replacing existing interfaces.

## 9 Limitations and Future Work

The current evaluation is limited to a prototype implementation with a small number of representative tasks and participants. System-level benchmarks were conducted in controlled settings, and the pilot user study was exploratory in

nature, limiting the generalizability of the results. Future work will involve longitudinal studies with industry partners to evaluate GenHAPI in real operational contexts. These studies will measure task completion time, error rates, perceived usability, trust, and adoption over extended use by both novice and expert users. We also plan to expand the intent knowledge base to cover a broader range of enterprise workflows and to further refine the vision-based guidance module to improve robustness under UI variability and interface evolution.

## 10 Ethical, Privacy, and Compliance Considerations

GenHAPI follows a *privacy-by-design* approach. User interaction data are pseudonymized at collection time and in system logs, and dispositive actions are always gated by explicit human confirmation. The architecture supports both cloud-based and on-premise deployments, enabling GDPR-aligned processing in regulated environments. Ongoing and future evaluations will include assessment of consent-related user experience and red-teaming activities targeting prompt injection and tool misuse.

## 11 Conclusion

This paper presented GenHAPI, a conversational orchestration layer that enables safe and auditable interaction with legacy enterprise systems. By combining declarative intent representations, MCP-based tool invocation, and optional vision-guided assistance within a human-in-the-loop framework, GenHAPI supports mixed-initiative interaction while preserving control and accountability. A key design aspect is the separation between cloud-based execution, relying on strict pseudonymization, and on-premise deployment, which enables full data locality for highly regulated contexts. Together, these results suggest that conversational orchestration can provide a practical and responsible interaction paradigm for complex enterprise workflows.

**Acknowledgments.** This work is part of GenHAPI (Generative Human API for Prompt Integration), funded by Regione Toscana under PR FESR 2021–2027, Bandi RS (Call n. 2). Funding IDs: CUP CIPESS D17H24004170009; Local CUP ST 27717.29122023.043000288. Partners: RCP Vision S.r.l. (lead), Krein S.r.l., Neumus S.r.l., Univ. of Florence (DINFO/MICC).

## References

1. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: ReAct: Synergizing Reasoning and Acting in Language Models. In: Proc. of ICLR 2023, Kigali (2023). https://openreview.net/forum?id=WE_vluYUL-X
2. Schick, T., Dwivedi-Yu, J., Tong, S., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language Models Can Teach Themselves to Use Tools. arXiv preprint (2023). https://doi.org/10.48550/arXiv.2302.04761

3. Deng, X., Gu, Y., Zheng, B., et al.: Mind2Web: Towards a Generalist Agent for the Web. In: NeurIPS 2023 (Datasets and Benchmarks Track), New Orleans (2023). https://papers.nips.cc/paper_files/paper/2023/hash/5950bf290a1570ea401bf98882128160-Abstract-Datasets_and_Benchmarks.html
4. You, K., Zhang, H., Schoop, E., Weers, F., Swearngin, A., Nichols, J., Yang, Y., Gan, Z.: Ferret–UI: Grounded Mobile UI Understanding with Multimodal LLMs. In: Computer Vision – ECCV 2024, LNCS, Springer (2024). https://doi.org/10.1007/978-3-031-73039-9_14
5. Lu, Y., Yang, J., Shen, Y., Awadallah, A.: OmniParser for Pure Vision Based GUI Agent. arXiv preprint (2024). https://doi.org/10.48550/arXiv.2408.00203
6. Le Sellier de Chezelles, T., Gasse, M., Lacoste, A., et al.: The BrowserGym Ecosystem for Web Agent Research. Transactions on Machine Learning Research (TMLR) (2025). https://openreview.net/forum?id=5298fKGmv3
7. Kuang, E., Jahangirzadeh Soure, E., Fan, M., Zhao, J., Shinohara, K.: Collaboration with Conversational AI Assistants for UX Evaluation: Questions and How to Ask them (Voice vs. Text). In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), pp. 1–15. ACM, New York (2023). https://doi.org/10.1145/3544548.3581247
8. Heo, J., Lee, U.: Form to Flow: Exploring Challenges and Roles of Conversational UX Designers in Real-world, Multi-channel Service Environments. Proceedings of the ACM on Human-Computer Interaction (CSCW), 7(CSCW1), Article 95 (2023). https://doi.org/10.1145/3610189
9. Nakano, R., et al.: WebGPT: Browser-assisted question-answering with human feedback. arXiv preprint (2021). https://doi.org/10.48550/arXiv.2112.09332
10. Ahn, M., et al.: Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. arXiv preprint (2022). https://doi.org/10.48550/arXiv.2204.01691
11. Shi, T., et al.: World of Bits: An Open-Domain Platform for Web-Based Agents. In: Proc. ICML 2017 (2017). https://proceedings.mlr.press/v70/shi17a.html
12. Yao, S., et al.: WebShop: Towards Scalable Real-World Web Interaction with Language Agents. In: NeurIPS 2022 (2022). https://openreview.net/forum?id=ddf0K_6ry
13. Horvitz, E.: Principles of Mixed-Initiative User Interfaces. In: Proc. CHI 1999, pp. 159–166 (1999). https://doi.org/10.1145/302979.303030
14. Hoffman, R. R., et al.: Explaining Explanation for Explainable AI. In: Human Factors, 60(2), pp. 197–211 (2018). https://doi.org/10.1177/0018720818769426
15. Brooke, J.: SUS: A "Quick and Dirty" Usability Scale. In: Jordan, P.W., Thomas, B., Weerdmeester, B.A., McClelland, A.L. (eds.) *Usability Evaluation in Industry*, pp. 189–194. Taylor & Francis, London (1996).
16. Horvitz, E.: Principles of Mixed-Initiative User Interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99), pp. 159–166. ACM, New York (1999).
17. Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P.N., Inkpen, K., Teevan, J., Kikin-Gil, R., Horvitz,