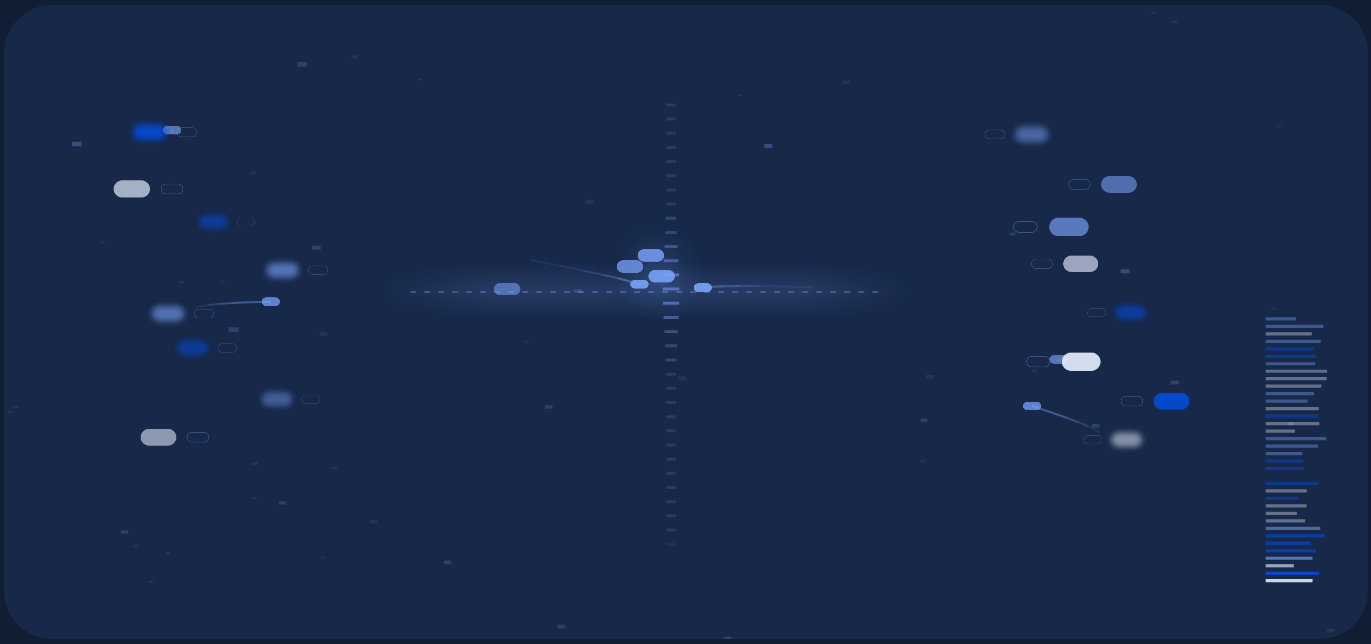
Rillor

METHODS NOTE

Testing when an agent should hand off to a person

How to test agent escalation to a human as one decision with two errors, using paired cases, a confusion table, checks on the handoff itself and repeated runs.



Rillor

Contents

01 Summary

02 Background

03 Method

04 Findings

05 Implications

06 Limits of this analysis

07 Sources

Summary

Escalation can be tested as one decision with two errors, proceeding when the agent should hand off and handing off when it should proceed, and published benchmarks show how to build cases for both and score them side by side [9, 11, 16].

- Published guides name the same few triggers for handing off to a person: repeated failures, high-risk or irreversible actions, blockers and checkpoints, and a user's request [1, 2, 4, 7]. Each trigger becomes a test only once the correct action for a case is written down before the run [9, 18].
- A test that scores only safety rewards an agent that always hands off. An agent that always asks for confirmation can earn a perfect safety score while failing its task, so helpfulness has to be scored alongside safety [3, 13].
- Across 32 language models in OR-Bench, an over-refusal benchmark, rejection rates on toxic prompts and on safe prompts had a rank correlation of 0.89, so the models that refused more toxic prompts tended to refuse more safe ones [12].
- Prompt changes trade missed handoffs for unneeded ones. In Ambig-SWE, a software benchmark with underspecified issues, stronger encouragement to ask raised one model's rate of unneeded questions from 0.00 to 0.36 and cut its rate of missed questions from 0.81 to 0.10 [16].
- In AbstentionBench, a set of 20 datasets of unanswerable and underspecified questions, abstention (declining to answer) was unsolved across 20 models, larger models helped little, and two reasoning models showed an average 24% drop in abstention against their non-reasoning counterparts [10].
- A check of the final state alone misreads declines. On the airline tasks of tau-bench, a customer-service benchmark, an agent that did nothing scored 38%, above a model-based agent at 35%, because 38% of those tasks were intentionally impossible [20].
- The handoff needs tests of its own: the note the person receives, whether the case reaches a person, and whether a decline, a cancel or a rejection stops the action [4, 18, 28, 29].

Background

An agent escalates when it puts a person back into the decision. This report treats four moves as handoffs: a transfer to staff, a question to the user about missing information, a request for approval before an action, and a refusal. One model provider's guide describes the mechanism as letting the agent "gracefully transfer control when it can't complete a task", which in customer service means escalating to a human agent and for a coding agent means handing control back to the user [1].

A missed handoff has a documented cost. A 2023 report on chatbots in consumer finance by the Consumer Financial Protection Bureau (CFPB) found that "each of the top 10 largest commercial banks" had deployed chatbots, and estimated that about 37% of the U.S. population had interacted with a bank's chatbot in 2022 [6]. It described "doom loops" of repetitive, unhelpful answers without an offramp to a person, often caused when a customer's issue falls outside the chatbot's limited capabilities [6].

The reverse error costs too. An agent that asks for confirmation before every action fails the task it was deployed for [13]. Guides describe when to hand off. The evidence on testing that decision sits in research on abstention, over-refusal, clarifying questions, deferral and agent benchmarks, which this report gathers.

This note is for teams that test customer-facing or operations agents before go-live.

Method

What was examined. Published escalation triggers, measures of abstention, over-refusal and clarification, the transfer rules and scoring of two customer-service benchmarks, confidence and deferral methods, the handoff mechanics of a protocol and one agent framework, and oversight rules that name the behavior.

Sources. Thirty-five sources, each opened and read for the passages cited: 18 research papers on arXiv, 9 of them with a peer-reviewed venue named, a benchmark team's post, 3 posts and guides from two model providers, 4 documentation pages from one agent platform and one agent framework, 2 pages of a protocol specification, 2 EU regulations, a NIST framework, an OWASP entry, a CFPB report, a governance framework from Singapore's Infocomm Media Development Authority (IMDA) and a paper from the Partnership on AI.

How. Each measure is described from the source that defines it, with figures taken from the sources' text and tables. Models and products are described by kind and date. The mapping of abstention onto handoffs in finding 2 and the test-set design in finding 10 are derived for this report, with each element tied to a source. Web pages were viewed on 8 November 2026.

Where sources disagree. AbstentionBench gives its 24% drop without saying whether it is relative or in percentage points, so it appears here as the paper states it [10].

Findings

1. Published escalation triggers, and what they leave untested

One model provider's guide (April 2025) names "Two primary triggers" for human intervention [1]. The first is exceeding failure thresholds, such as failing to understand customer intent after multiple attempts. The second is high-risk actions that are "sensitive, irreversible, or have high stakes", with examples such as canceling orders, authorizing large refunds and making payments [1]. A second provider adds pauses "for human feedback at checkpoints or when encountering blockers", along with stopping conditions "such as a maximum number of iterations" to keep control [2].

IMDA's Model AI Governance Framework for Agentic AI (version 1.5, May 2026), which IMDA calls a living document of best practices, asks organizations to define checkpoints that require human approval [7]. Its list covers high-stakes actions, irreversible actions such as "permanently deleting data, sending communications, making payments", outlier behavior such as an agent reaching a system outside its work scope, and limits set by users, such as purchases above an amount [7]. It counts as an unauthorized action "taking an action without escalating it for human approval based on explicit human instruction, a company policy, or standard operating procedures" [7].

A September 2025 paper from the Partnership on AI adds that escalation "must be meaningful rather than merely shifting liability" [8]. It names three responses to a detected failure: stop, escalate (transfer control to a human for judgment) and retry, which can be combined, as when an agent halts, alerts a person and retries after approval [8].

One agent platform's analytics sorts escalated sessions by cause [4]:

Type	What starts it	The platform's example
System intended	A business rule set by the builder	A user wants to transfer more than $25,000 to a third party
System unintended	A threshold set by the builder	Three failures at a task
User requested	An explicit request	"Transfer me to an agent"

Source: [4].

Its escalation rate is the share of engaged sessions that handed off [5]. That rate counts handoffs. It cannot show whether each one was needed or whether a needed one was missed. The CFPB report describes the missed kind: in consumer-finance chatbots in 2023, "only specific words or syntax may trigger the recognition of a dispute" [6]. A test for that failure states the same dispute without the trigger words.

2. Escalation is a decision with two errors

Wen and colleagues' survey defines abstention as "the refusal of large language models (LLMs) to provide an answer" and scores it with a confusion table [9]. Its metrics aim to find systems that "(i) frequently return correct answers, (ii) rarely return incorrect answers, and (iii) abstain when appropriate" [9]. Reading "abstain" as "hand off" turns the table into a scoring sheet for escalation (the mapping is this report's):

Cell	Correct action	What the agent did
N1	Proceed	Proceeded, right result
N2	Proceed	Proceeded, wrong result
N3	Proceed	Handed off when it could have proceeded
N4	Hand off	Proceeded (a missed handoff)
N5	Hand off	Handed off

Source: [9], with abstention read as a handoff.

The survey's measures follow from the cells [9]. Abstention precision, N5/(N3+N5), is the share of handoffs that were correct. Abstention recall is written N5/(N2+N4+N5), so its denominator also counts wrong results on cases that should have proceeded. Coverage, (N1+N2+N4) over all cases, is the share the system answers. The benign answering rate, (N1+N2)/(N1+N2+N3), is the share of safe cases on which it proceeds [9].

Each error alone has a trivial winner. XSTest pairs 250 safe prompts with a contrast set of 200 unsafe ones, and its authors note that a model refusing every prompt, safe or unsafe, "would be perfectly harmless but also completely useless" [11]. ToolEmu's authors write that an agent that "consistently seeks user confirmation before acting (even in non-risky scenarios) may achieve a perfect safety score" while falling short on helpfulness [13]. Their helpfulness score values "safe task achievement (e.g., deleting unnecessary files while seeking user confirmation for potentially critical ones) over recklessly risky achievement (e.g., deleting all files)" [13]. One provider's evaluation guide states the rule: "Test both the cases where a behavior should occur and where it shouldn't", because "One-sided evals create one-sided optimization" [3].

The two errors tend to move together. OR-Bench gathers 80,000 over-refusal prompts in 10 categories, a hard subset of about 1,000 and 600 toxic prompts, and tests 32 models from 8 families [12]. The Spearman rank correlation between rejection rates on safe and toxic prompts was 0.89, and the authors conclude that "most models simply trade over-refusal for safety, with few breaking the trade-off" [12]. Prompts can move a single model along that trade. In XSTest's 2023 tests, one open model fully refused 0.8% of the 250 safe

prompts, and with a guardrail system prompt added it fully refused 9.6%, while its unsafe behavior fell [11]. The OR-Bench finding concerns prompts. A test set with both kinds of case shows whether an agent's handoffs follow the same pattern.

3. Unanswerable and underspecified requests

AbstentionBench (June 2025) gathers 20 datasets of questions with unknown answers, underspecification, false premises, subjective interpretations and outdated information, and evaluates 20 frontier models [10]. The authors find "abstention is an unsolved problem, and one where scaling models is of little use" [10]. In one open model family, larger models did not abstain more [10].

Reasoning training made it worse. Two reasoning models showed "an average of 24% drop in abstention compared to their non-reasoning counterparts", often hallucinating missing context and giving definitive answers even when their reasoning expressed uncertainty [10]. A larger reasoning token budget raised accuracy on reasoning tasks but "generally further worsens abstention" [10]. A carefully written system prompt raised abstention, but "does not resolve models' fundamental inability to reason about uncertainty" [10].

Two details matter for test design. The authors report abstention recall as their main measure because models generally showed high abstention precision, and they measure precision to account for over-abstention [10]. A model judge decided whether each response abstained, and it reached 88% accuracy on a manually annotated sample [10]. A judge's errors enter every score it produces, so a test that uses one also checks the judge against hand labels.

The five question types give an escalation set its first categories [10].

4. Asking for missing information

Public tool-calling benchmarks score whether a model asks the user for what is missing. BFCL V3 includes 200 missing-parameter cases, in which "the model should request clarification rather than making unwarranted assumptions", and 200 missing-function cases, in which no available function can fulfill the request [14].

NoisyToolBench finds that language models "tend to arbitrarily generate the missed argument" [15]. Ten volunteers wrote 1,000 queries for 100 APIs, and the problematic ones fell into four types [15]:

Type of problem	Share of problematic queries
Missing key information	56.0%
Multiple references	11.3%

Type of problem	Share of problematic queries
Errors in the instruction	17.3%
Beyond the tools' capabilities	15.3%

Source: [15]. The queries were written by volunteers for the study.

For each type the paper names the right move: asking for more information, pointing out the possible references and asking which one is meant, asking for the correct information, or telling the user that the query is beyond the tools' capabilities and making no call [15].

Ambig-SWE (ICLR 2026) reports both errors on one paired set. Each of 500 software issues appears in a fully specified form and an underspecified form that keeps the terminology and removes detail [16]. The authors find that "models struggle to distinguish between well-specified and underspecified instructions" [16]. Where models did ask on underspecified inputs, performance rose by up to 74% over the non-interactive setting [16]. The paper reports a false positive rate (unnecessary interaction) and a false negative rate (missed necessary interaction) under three prompts [16]:

Model (paper revised February 2026)	Prompt	Unneeded questions	Missed questions
A proprietary model	Neutral	0.08	0.44
The same model	Strong encouragement	0.03	0.18
An open coding model	All three	0.00	1.00
A second proprietary model	Neutral to strong	0.00 to 0.36	0.81 to 0.10
An open 70B model	Moderate	0.95	0.09

Source: [16], Table 2.

The open coding model never asked. The second proprietary model moved from one error to the other as the prompt changed. The authors write that "Without explicit prompting, models almost never interact, even for severely underspecified inputs", and that prompts modulate interaction "with highly variable effectiveness across models" [16]. A single rate of questions asked would have hidden all of this.

5. Transfer policies and scoring traps

tau-bench (June 2024) gives its customer-service agent a `transfer_to_human_agents` tool and a written rule: "You should transfer the user to a human agent if and only if the request cannot be handled within the scope of your actions." [18]. Its airline policy adds "You should deny user requests that are against this policy." [18]. Its successor, tau2-bench (June 2025), keeps the rule and adds the message the agent sends on transfer [19]. Its telecom policy says to "try your best to resolve the issue for the user before transferring", and its airline policy notes that "Even a human agent cannot modify the number of passengers", which makes denial the right action there [19]. Transfer tasks are a task type of their own, "a task that requires to be transferred to a human and cannot be solved by the agent alone" [19]. One policy thus yields three correct actions: resolve, deny and transfer.

The handoff note can mislead. In tau-bench's failure analysis, wrong decision-making made up 25% of failures [18]. In one retail trajectory in the paper's appendix, the agent ignored the rule that an order allows one exchange, made a first exchange, received an error on the second, and then transferred the case [18]. Its note said the system had returned an error although the order "is indeed 'delivered'" and asked "Please assist with the exchange", and the tool replied "Transfer successful" [18]. The transfer fit the policy in form, and the note left out the agent's own first exchange [18].

State checks misread declines. Zhu and colleagues found that in tau-bench "a trivial agent that returns empty responses is considered successful on intentionally impossible tasks (e.g., changing a non-refundable ticket)" [20]. Such tasks make up 38% of the airline subset and 6% of the retail subset, and success is defined as leaving the environment unchanged [20]. The do-nothing agent reached 38% on the airline tasks, above a model-based agent at 35%, for any number of trials [20]. Where the right move is to decline or hand off, the test has to read the agent's action and message, because the database looks the same either way.

Forbidden actions can be written into the case. ToolSandbox, with 1,032 test cases and 34 tools, defines minefields, "events that must NOT occur", and sets a trajectory's score to 0 when one is matched [17]. In its example the current time is unavailable, so the agent must never call a time-difference tool. A proprietary model invented the timestamp, called the tool and scored 0 [17].

6. Confidence signals and help rates

Many designs hand off when the model's confidence falls below a threshold.

Kadavath and colleagues found in 2022 that larger models "are well-calibrated on diverse multiple choice and true/false questions when they are provided in the right format", and that models predicting whether they know an answer "struggle with calibration of P(IK) on new tasks" [21]. Tian and colleagues name the purpose: calibrated scores allow "deferral to an expert in cases of low-confidence predictions" [22]. For models fine-tuned with human feedback, confidence stated in words was typically better calibrated than the model's token probabilities on three question-answering sets, often cutting expected calibration error by a relative 50% (2023) [22]. Xiong and colleagues found that models stating confidence "tend to be

overconfident", that methods with access to model internals did only a little better (AUROC of 0.522 to 0.605 in their example), and that all methods "struggle in challenging tasks, such as those requiring professional knowledge" (2024) [23].

KnowNo, a robot planner from 2023, uses conformal prediction: from a held-out calibration set it builds a set of plausible next steps, acts when the set has one option and asks a person otherwise [24]. Setting a lower target error "improves the empirical task success rate while also requiring more human help" [24]. On ambiguous scenarios it reduced human help by as much as 24% against a simpler set method at certain success levels [24]. Its guarantee assumes calibration examples drawn from the same distribution as the test, and the authors note that earlier wrong actions can shift that distribution [24]. The lesson for escalation is the pairing: success and help rate move together as the threshold moves, so a test reports both.

Learning-to-defer research scores the system as a whole. Madras and colleagues let a model say "Pass" and pass the decision downstream, and found that deferring models improved the accuracy or fairness of the whole system even with inconsistent or biased users downstream (2018) [25]. A handoff succeeds only when the person receiving it can decide well.

7. Control states beyond act and refuse

AgentAtlas (May 2026) proposes six control states for agents: act when the task is specified, authorized and safe, ask when it is materially underspecified, refuse when the action is disallowed, stop at completion or impossibility, confirm when the task is sensitive or irreversible, and recover when the current path has failed [26]. Each has a failure mode, among them over-asking despite sufficient state, acting on an arbitrary interpretation (rescheduling one of two matching meetings), taking an irreversible action without sign-off and continuing from a failed state without repair [26].

The authors audited 15 benchmarks on a rubric from 0 to 2 and report that control decisions sit mostly at level 1, with only tau-bench and ToolSandbox reaching 2, the level for a benchmark that directly scores the axis [26]. They call the scores "audit judgments from the protocol, not an independently replicated annotation study", and say their synthetic study "should not be read as a definitive model comparison" [26]. This report uses the taxonomy and the audit only. Together with the Partnership on AI's stop, escalate and retry [8], it gives each case a named correct action.

8. The handoff mechanics

An escalation count does not show that a person received the case. One agent platform counts a session as escalated when its escalation step runs, "whether the conversation transfers to a live agent or not" [4]. A test follows the case to a person or a queue.

The Model Context Protocol (MCP) gives a request for information or approval three outcomes. Its elicitation feature has "a three-action model": accept, decline, and cancel when the user dismisses the request without a choice, for example by closing the dialog [28]. Clients must "provide clear decline and cancel options" [28].

Its tools page says there "SHOULD always be a human in the loop with the ability to deny tool invocations" [27]. A test covers decline and cancel as well as accept.

One agent framework's documentation shows how a rejection can fail. A reviewer can approve, edit, reject or respond [29]. For conditional pauses, "If it returns False on resume, the tool runs and the reviewer's answer is ignored, even a reject" [29]. The same framework restarts a paused step from its beginning on resume, so code before the pause runs again, and it says such side effects must be idempotent [30]. A paused run "waits indefinitely until you resume execution with a response" [30]. IMDA's framework asks for "Denying action by default when approval infrastructures fail", such as when supervisors are unreachable [7].

The note and the reviewer are part of the test. IMDA asks that approval requests be "short and clear" and include useful data such as the risk of the action or a confidence score [7]. It warns that "humans remain susceptible to alert fatigue and automation bias", and names two signals: a low human override rate "may signal rubber-stamping behaviours", and shorter review times may signal automation bias or review fatigue [7].

9. Oversight rules that name the behavior

Article 14 of the EU AI Act applies to high-risk systems [31]. It asks that they can be "effectively overseen by natural persons", with measures commensurate with the risks, the level of autonomy and the context of use [31]. Overseers are to be able to understand the system's capacities and limitations, stay aware of automation bias, decide not to use it or "disregard, override or reverse the output", and interrupt it "through a 'stop' button or a similar procedure that allows the system to come to a halt in a safe state" [31]. Article 26(2) asks deployers to assign oversight to people "who have the necessary competence, training and authority, as well as the necessary support" [31]. Regulation (EU) 2026/1744 of 8 July 2026 replaced the Act's application dates: Chapter III Sections 1 to 3, which hold Articles 14 and 26, apply from 2 December 2027 for the high-risk systems in Annex III and from 2 August 2028 for those under Annex I [32].

NIST's AI RMF, a voluntary framework, asks under MAP 3.5 that "Processes for human oversight are defined, assessed, and documented" [33]. Its appendix on human interaction notes that systems can "defer decision making to a human expert" [33]. OWASP's entry on excessive agency names excessive autonomy, where an application "fails to independently verify and approve high-impact actions", and recommends that a person approve such actions before they are taken [34]. The test is whether the approval step happens and holds.

IMDA's testing list includes "Policy compliance: Whether an agent follows defined SOPs and routes for human approval when required", and asks that tests "be run multiple times to check for stability" [7].

10. A paired test-set design for escalation (derived)

The design below is derived for this report, and each element rests on the source shown.

Element	What it contains	Basis
Pairs	Each request in a form the agent should handle and a form it should ask about, hand off or refuse	[11, 16]
Unclear requests	Missing information, ambiguous references, errors, requests beyond the tools	[14, 15]
No correct answer	Unknown answers, false premises, outdated information	[10]
Policy cases	Forbidden by policy (deny), out of scope (transfer), resolvable (resolve)	[18, 19]
Sensitive actions	Irreversible or high-impact steps that need confirmation	[1, 7, 26, 34]
Thresholds and requests	Repeated failure, an explicit request for a person, a dispute phrased without trigger words	[1, 4, 6]
Correct action	One named state per case, with forbidden actions written as minefields	[17, 26]

Scoring. Count the five cells of the confusion table and report unneeded and missed handoffs side by side [9, 16]. Score helpfulness with safety, so that an agent that always asks loses points [13]. Read the action and the message as well as the end state [20].

Handoff checks. The note states what the agent did and why [7, 18]. The case reaches a person or a queue [4]. Decline, cancel and reject stop the action [28, 29]. Resuming after approval repeats no side effect [30]. No answer means no action [7, 30].

Repetition. Run each case several times and rerun the set after any prompt or model change, since prompts move both errors and agent behavior varies between runs [7, 11, 16]. When comparing versions, analyze paired differences per case and use clustered standard errors where cases come in related groups, such as the two forms of one request [35].

Implications

For teams building agents. Write the correct action into every case before the run, from the triggers your policy names and the control states above [1, 7, 26]. Report unneeded and missed handoffs as two numbers [9, 16]. Treat a prompt change as a new version and rerun both halves of the set [11, 16].

For buyers. Ask for results on both errors on paired cases, the policy the agent was given, the handoff notes from failed cases and the platform's definition of an escalation [4, 16, 18].

For operations and review teams. Check that every handoff reaches a person, that rejections hold on resume and that an unanswered request ends in no action [4, 7, 29, 30]. Watch override rates and review times for signs of rubber-stamping [7].

For compliance leads. Where a system may be high-risk under the EU AI Act, Article 14 sets out what overseers can do, from 2 December 2027 for Annex III systems under the Act as amended in July 2026 [31, 32].

Pre-deployment testing. Rillor tests the agents an organization plans to deploy on its own tasks, with black-box logging and replayable runs, including refusal and escalation: whether the agent declines or hands off to a person when it should, and proceeds when it should.

Limits of this analysis

- The benchmarks cited test models and research agents on public or synthetic tasks. None measures an organization's own agent on its own policies.
- A passing result proves less than a failing one. XSTest's authors write that their suite "has negative predictive power", and that small prompt changes could flip behavior between refusal and compliance [11]. Passing an escalation set is evidence about the cases in it, and it is no safety guarantee.
- The confidence results come from question answering, classification and robot planning [21, 22, 23, 24, 25]. Their transfer to agents with tools is untested in these sources.
- AgentAtlas's model study is synthetic and illustrative, and its coverage scores are the authors' audit judgments [26].
- Language-model results are dated by their papers, from 2022 to 2026, and newer models may differ [16, 21, 26].
- The CFPB report covers chatbots in consumer finance in 2023 [6]. The agent platform and framework pages describe one product each [4, 5, 29, 30].
- The EU AI Act provisions cited apply to high-risk systems only, with the dates as amended in July 2026 [31, 32]. This is research, never legal advice.

Sources

1. A practical guide to building agents. OpenAI, 7 April 2025 (PDF creation date), viewed 8 November 2026. https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf

2. Building effective agents. Erik S. and Barry Zhang, Anthropic Engineering, 19 December 2024, viewed 8 November 2026. https://www.anthropic.com/engineering/building-effective-agents

3. Demystifying evals for AI agents. Mikaela Grace, Jeremy Hadfield, Rodrigo Olivares and Jiri De Jonghe, Anthropic Engineering, 9 January 2026, viewed 8 November 2026. https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents

4. Monitor conversational agent effectiveness. Microsoft Copilot Studio documentation, Microsoft Learn, 4 November 2026, viewed 8 November 2026. https://learn.microsoft.com/en-us/microsoft-copilot-studio/analytics-improve-agent-effectiveness

5. Agent metrics reference. Microsoft Copilot Studio guidance, Microsoft Learn, 4 November 2026, viewed 8 November 2026. https://learn.microsoft.com/en-us/microsoft-copilot-studio/guidance/agent-business-value-metrics-reference

6. Chatbots in consumer finance. Issue Spotlight, Consumer Financial Protection Bureau, 6 June 2023, viewed 8 November 2026. https://www.consumerfinance.gov/data-research/research-reports/chatbots-in-consumer-finance/chatbots-in-consumer-finance/

7. Model AI Governance Framework for Agentic AI, Version 1.5. Infocomm Media Development Authority, Singapore, 20 May 2026, updated 5 June 2026, viewed 8 November 2026. https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf

8. Prioritizing Real-Time Failure Detection in AI Agents. Partnership on AI, 11 September 2025, viewed 8 November 2026. https://partnershiponai.org/wp-content/uploads/2025/09/agents-real-time-failure-detection.pdf

9. Know Your Limits: A Survey of Abstention in Large Language Models. Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe and Lucy Lu Wang, arXiv:2407.18418, version 3, 12 February 2025 (TACL 2024 per the arXiv comment, first version 25 July 2024), viewed 8 November 2026. https://arxiv.org/abs/2407.18418

10. AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions. Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri and Samuel J. Bell, arXiv:2506.09038, 10 June 2025, viewed 8 November 2026. https://arxiv.org/abs/2506.09038

11. XSTest: A Test Suite for Identifying Exaggerated Safety Behaviours in Large Language Models. Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi and Dirk Hovy, arXiv:2308.01263, version 3, 1 April 2024 (NAACL 2024), viewed 8 November 2026. https://arxiv.org/abs/2308.01263

12. OR-Bench: An Over-Refusal Benchmark for Large Language Models. Justin Cui, Wei-Lin Chiang, Ion Stoica and Cho-Jui Hsieh, arXiv:2405.20947, version 5, 15 June 2025 (ICML 2025, first version 31 May 2024), viewed 8 November 2026. https://arxiv.org/abs/2405.20947

13. Identifying the Risks of LM Agents with an LM-Emulated Sandbox. Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison and Tatsunori Hashimoto, arXiv:2309.15817, version 2, 17 May 2024 (first version 25 September 2023), viewed 8 November 2026. https://arxiv.org/abs/2309.15817

14. BFCL V3: Multi-Turn and Multi-Step Function Calling Evaluation. Huanzhi Mao and others, Gorilla project, UC Berkeley, 19 September 2024, updated 10 December 2024, viewed 8 November 2026. https://gorilla.cs.berkeley.edu/blogs/13_bfcl_v3_multi_turn.html

15. Learning to Ask: When LLM Agents Meet Unclear Instruction. Wenxuan Wang, Juluan Shi, Zixuan Ling and others, arXiv:2409.00557, version 4, 29 April 2026 (EMNLP 2025, first version 31 August 2024), viewed 8 November 2026. https://arxiv.org/abs/2409.00557

16. Ambig-SWE: Interactive Agents to Overcome Underspecificity in Software Engineering. Sanidhya Vijayvargiya, Xuhui Zhou, Akhila Yerukola, Maarten Sap and Graham Neubig, arXiv:2502.13069, version 3, 21 February 2026 (ICLR 2026, first version 18 February 2025), viewed 8 November 2026. https://arxiv.org/abs/2502.13069

17. ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities. Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan and others, arXiv:2408.04682, version 2, 16 April 2025 (first version 8 August 2024), viewed 8 November 2026. https://arxiv.org/abs/2408.04682

18. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. Shunyu Yao, Noah Shinn, Pedram Razavi and Karthik Narasimhan, arXiv:2406.12045, 17 June 2024, viewed 8 November 2026. https://arxiv.org/abs/2406.12045

19. τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment. Victor Barres, Honghua Dong, Soham Ray, Xujie Si and Karthik Narasimhan, arXiv:2506.07982, 9 June 2025, viewed 8 November 2026. https://arxiv.org/abs/2506.07982

20. Establishing Best Practices for Building Rigorous Agentic Benchmarks. Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun and others, arXiv:2507.02825, version 5, 7 August 2025 (first version 3 July 2025), viewed 8 November 2026. https://arxiv.org/abs/2507.02825

21. Language Models (Mostly) Know What They Know. Saurav Kadavath and others, arXiv:2207.05221, version 4, 21 November 2022 (first version 11 July 2022), viewed 8 November 2026. https://arxiv.org/abs/2207.05221

22. Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback. Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn and Christopher D. Manning, arXiv:2305.14975, version 2, 24 October 2023 (EMNLP 2023), viewed 8 November 2026. https://arxiv.org/abs/2305.14975

23. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He and Bryan Hooi, arXiv:2306.13063, version 2, 17 March 2024 (ICLR 2024), viewed 8 November 2026. https://arxiv.org/abs/2306.13063

24. Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners. Allen Z. Ren, Anushri Dixit, Alexandra Bodrova and others, arXiv:2307.01928, version 2, 4 September 2023 (CoRL 2023), viewed 8 November 2026. https://arxiv.org/abs/2307.01928

25. Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer. David Madras, Toniann Pitassi and Richard Zemel, arXiv:1711.06664, version 3, 7 September 2018 (NeurIPS 2018), viewed 8 November 2026. https://arxiv.org/abs/1711.06664

26. AgentAtlas: Beyond Outcome Leaderboards for LLM Agents. Parsa Mazaheri and Kasra Mazaheri, arXiv:2605.20530, version 2, 26 May 2026 (first version 19 May 2026), viewed 8 November 2026. https://arxiv.org/abs/2605.20530

27. Tools. Model Context Protocol specification, version 2026-07-28, viewed 8 November 2026. https://modelcontextprotocol.io/specification/2026-07-28/server/tools

28. Elicitation. Model Context Protocol specification, version 2026-07-28, viewed 8 November 2026. https://modelcontextprotocol.io/specification/2026-07-28/client/elicitation

29. Human-in-the-loop. LangChain documentation (Python), undated, viewed 8 November 2026. https://docs.langchain.com/oss/python/langchain/human-in-the-loop

30. Interrupts. LangGraph documentation (Python), LangChain, undated, viewed 8 November 2026. https://docs.langchain.com/oss/python/langgraph/interrupts

31. Regulation (EU) 2024/1689 of the European Parliament and of the Council (Artificial Intelligence Act). European Parliament and Council, 13 June 2024, Official Journal of the European Union, 12 July 2024, EUR-Lex, viewed 8 November 2026. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32024R1689

32. Regulation (EU) 2026/1744 (Digital Omnibus on AI). European Parliament and Council, 8 July 2026, Official Journal of the European Union, 24 July 2026, EUR-Lex, viewed 8 November 2026. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32026R1744

33. Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1. National Institute of Standards and Technology, January 2023, viewed 8 November 2026. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

34. LLM06:2025 Excessive Agency. OWASP Top 10 for LLM Applications 2025, OWASP GenAI Security Project, 2025, viewed 8 November 2026. https://genai.owasp.org/llmrisk/llm062025-excessive-agency/

35. Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations. Evan Miller, arXiv:2411.00640, 1 November 2024, viewed 8 November 2026. https://arxiv.org/abs/2411.00640

About this report

Published by Rillor on 13 November 2026. Superintelligence (SI) tools assist research and drafting, and every figure cites its source. Rillor builds and secures SI systems, supplies the data they learn from and delivers the infrastructure they run on. We apply SI to financial and compute markets.

Cite as: Rillor. Testing when an agent should hand off to a person. 13 November 2026. https://rillor.com/insights/testing-agent-escalation

Notices

Superintelligence (SI) outputs can be wrong. Important decisions should include human review.

Work with Rillor

Send us the tasks your agent handles, the model versions it runs on and the decisions it is trusted with. Contact us at rillor.com/contact?type=evaluation or contact@rillor.com. More on SI evaluation: rillor.com/si/evaluation