Technical report

Do AI agents verify the facts before processing bank transactions?

Flitto Agent Bench (Finance – Global BankOps)

Published by Flitto Agent Data Researcher Team

23 November 2026

Abstract

Global BankOps evaluates how AI agents reconcile records and carry out back-office work in a simulated bank. Its tasks span trade finance, loan servicing, treasury, accounting, and pension and trust operations. Agents compare ledger entries, contracts, and SWIFT messages, resolve discrepancies with the team that holds the relevant information, and use the evidence to decide how to proceed. The evaluator checks when the agent obtained the evidence, what it posted or sent, and whether its report accurately describes the outcome. Eight model configurations attempted 50 tasks twice each. Of 800 runs, 342 passed. Outcomes differed between trials in 40 of the 400 model–task pairs (10.0%). In 413 of the 458 failed runs, the model missed a verification or action-order requirement. Reviewing reply reads alongside payments and ledger postings reveals failures that the final report alone cannot establish.

1. Introduction

Bank back-office work can require checking records held by different teams before changing a ledger or communicating a decision. An officer may need to determine which source governs, obtain confirmation, and ensure the recorded action reflects that evidence. An evaluation of this work must inspect the order and effects of actions as well as the final report.

Global BankOps places each evaluated model in a simulated bank. A task gives the model a role, an assignment, and tools to inspect records and act. The benchmark records the decisions, their order, and their consequences.

1.1 Related work and research question

FinQA tests reasoning over financial reports [1], and Bank-World evaluates agents in simulated banking workflows [2]. τ-Bench checks the database state after an agent acts under a domain policy [3]. Global BankOps examines whether an officer reconciles conflicting records before acting and whether the resulting entries and reports agree.

This study asks how often outcomes change when an agent repeats a banking task, and how verifying the governing facts before a transaction relates to successful completion [6].

2. Banking workflows and task design

2.1 Work across five divisions

Depending on its assignment, a model can read agreements, ledgers, and messages, query role-scoped records, contact another team, and inspect the reply. The environment records actions such as payments, postings, reports, and decisions to hold a transaction.

The Budapest subsidiary hosts 40 tasks. The Seoul head office hosts 10, all in pension and trust. Thirty-one tasks stay within the simulated bank and 19 involve SWIFT exchanges with another institution.

Table 1. Work and supporting records across the five divisions

Division	Tasks	Officer responsibilities
Trade Finance	10	Check letters of credit and interbank claims, then communicate payment or hold decisions.
Loan Servicing	10	Reconcile loan agreements with transaction records to verify interest and repayment terms.
Treasury Operations	10	Reconcile account statements with trade records and settle FX and money-market transactions.
Accounting	10	Post month-end accruals and valuation adjustments and reconcile reporting schedules to the ledger.
Pension & Trust	10	Check pension eligibility and trust instructions before paying out and updating records.

2.2 Resolving discrepancies before acting

A stale record or a discrepancy can leave the assigned officer without the governing fact. The officer can request the relevant information from its owner. Where the task permits an up-to-date system record as equivalent evidence, the evaluator accepts the relevant record read before the action. Each recorded run includes its settings, tool trajectory, final environment snapshot, and grade.

The task environments contain documents, ledger entries, and correspondence, including SWIFT message examples. We specify which records each role can access, what information other teams can supply, and how actions change the recorded state. Some tasks simplify operational steps such as internal approval.

2.3 Sources for the tasks

We drew on practitioner interviews, materials supplied by practitioners, and published procedural references. Each task has a source note describing the basis for its workflow.

Public references, including Oracle FLEXCUBE documentation [4] and South Korean retirement-benefit rules, informed specific procedures. The institutions, people, and transaction figures in the scenarios are fictional.

We used the difficulty observed in pilot runs to help select tasks.

3. Evaluation procedure

3.1 Paired trials and scoring

The v1.0.0 evaluation uses workplace framing, seed 0, a default 1,800-second timeout per model run, and a task-specific budget of 40 to 90 tool calls (60 in 42 tasks) with one call per turn. Provider reasoning settings

are listed in Table B1 and were the same in both trials.

Selection used the first two scored runs for each configuration and task, ordered by start time and then run ID. We excluded one provider connection error, one timeout and four later scored runs. The comparison covers the eight configurations that completed the full task set. We used certified grades on the original snapshots where citation repair or adjudication was necessary.

Global BankOps uses hybrid scoring version 6. When a verdict depends on what an officer communicated, Gemini 3.8 Flash (google/gemini-3.8-flash, temperature 0, low reasoning effort) applies the written rubric to eligible messages and reports. It cites the text supporting its decision.

3.2 Measuring completion across two trials

Trials are ordered by start time within each model–task pair. Pass@1 counts tasks passed on the first trial, pass@2 counts tasks passed at least once, and pass² counts tasks passed on both trials. The denominator is 50 tasks overall or ten within each division.

Every configuration has exactly two counted trials on every task, so the three measures use the same 400 model–task pairs. The division results, the task difficulty counts, the confirmation analysis, and the statistical tests use pass@1. Analyses of individual runs, such as failure counts, cost, and tool calls, use all 800 runs.

4. Task completion across two trials

The results use the current runs on the frozen 50-task release. A run made on an earlier validated release remains current when its task inputs and evaluator are unchanged, and a validated regrade of the same task snapshot can supply the effective verdict while the original run and grade remain in the evaluation records. The results contain 800 runs, two per configuration and task: 342 passes and 458 failures.

Table 2. Task-averaged first-trial, at-least-once and both-trial pass rates

Model	pass@1	pass@2	pass²	Tasks where trials differ
GPT-6.2 Astra	78.0%	84.0%	70.0%	7
GPT-6 Astra	74.0%	74.0%	68.0%	3
Fable 5.2	56.0%	62.0%	50.0%	6
Gemini 4 Carbon	48.0%	54.0%	46.0%	4
Opus 5.5	34.0%	40.0%	26.0%	7
Gemini 4 Argon	28.0%	30.0%	24.0%	3
Grok 4.7	16.0%	22.0%	10.0%	6
Qwen3.8 Max Prime	14.0%	16.0%	8.0%	4

Table 2 compares the three measures. The ranking of the eight configurations is the same under each. The two trials differ on 40 of the 400 model–task pairs, from three tasks for GPT-6 Astra and Gemini 4 Argon to seven for GPT-6.2 Astra and Opus 5.5. A second trial raised GPT-6.2 Astra from 78.0% to 84.0% but did not change GPT-6 Astra's 74.0%. Requiring both trials to pass lowers every configuration: GPT-6.2 Astra to 70.0% and GPT-6 Astra to 68.0%, and Grok 4.7 and Qwen3.8 Max Prime to 10.0% and 8.0%.

GPT-6.2 Astra passed 39 tasks on the first trial, 42 at least once and 35 on both. The increase to 42 measures what a second attempt recovered. The 35 repeated passes describe tasks the model completed on both occasions.

5. Verification before payment and posting

5.1 Receiving a reply and reading it

In 44 tasks, at least one rubric check requires the officer to request information from a designated team or SWIFT correspondent and read the reply before a protected action. Four of these tasks have several such checks. In three other tasks, a current system record read before the action is also accepted, and three tasks have no such check.

Across all 458 failed runs in both trials, verification and action-order checks failed in 413, written or saved output checks in 351, payment and ledger checks in 160, action restrictions in 114, and the completion check in 8. A run can fail several checks. Across both trials, 401 runs failed at least one mandatory confirmation check. In 247 runs, no recorded request matched the recipient and reply rule for at least one failed check. Of these, 29 had contacted an eligible recipient for at least one such check, and 218 had no recorded contact with an eligible recipient for any of their unmatched checks. Contact alone does not tell us whether the question asked for the required fact. In 92 other runs, the requests matched but at least one reply was never opened. In the remaining 62, the agent opened a reply after acting.

On first trials, 139 of the 155 runs that verified every required reply in time also passed the task. Reading the reply did not guarantee completion. The resulting payment, posting and report still had to satisfy the other requirements.

5.2 Payment approval and ledger postings

Table 3. Applicable runs and banking control outcomes in the original 800 trials

Banking control	Observed / applicable trials
At least one required confirmation was missed or late	401 / 704
Released the target payment before reading its approval	0 / 31
No target payment recorded on an approval-gated task	1 / 32
Failed a posting or net-ledger requirement	103 / 560

Table 3 counts only the trials to which each control applies [7]. Of 32 trials requiring approval for a target payment, 31 recorded that payment. All 31 releases followed the approval read. The remaining trial had no target payment for which the order could be assessed.

6. Limits of interpretation

The 50 tasks represent selected banking workflows, with simplified approvals and information access in some cases. Two trials per task give a limited estimate of repeatability, and the statistical tests use pass@1 alone. Replay verifies the simulator's response to a recorded action sequence. Differences in model outputs between

trials can reflect variation in generation as well as API routing, execution backend, or caching. Reasoning settings and response budgets also vary across models. The judge comparison in Table A4 used one solver's outputs and scoring version 5. Human reviewers assessed selected criteria during development; agreement with independent human judgments across the full version 6 evaluator remains unmeasured.

Simulated teams answer a matching request accurately, after a short fixed delay and at no cost. Real requests can be slow, incomplete, or costly, so the benchmark may reward asking more than practice does. A valid paraphrase can also miss a reply rule and leave a required confirmation unrecorded. GPT-6 Astra passed all ten pension and trust tasks in both trials and nine of the ten treasury operations tasks on the first trial. All eight configurations passed three tasks on the first trial and two of them on both trials, which leaves little room to measure further gains there.

7. Conclusion

Global BankOps tests whether models verify evidence, act on it, and report the result across five banking divisions. In 413 of 458 failed runs, models missed a verification or action-order requirement, and on must-ask tasks the number of timely confirmations followed the overall ranking of the eight configurations except for the two lowest. Requiring both of two trials to pass lowered every configuration's pass rate by 2 to 8 points without changing the ranking. An evaluation of back-office agents therefore needs the sequence of requests and actions: the final report alone does not show whether the governing fact was confirmed before funds moved.

Ethics and data

Banking practitioners described how back-office teams divide records and reconcile information, and their input shaped the task design. The institutions, counterparties, accounts, and people in the task environments are fictional, and the materials contain no customer records. Public sources are cited where they informed procedures. All model actions took place in the simulated environment; no message or payment reached a real system.

The author used AI tools in preparing the benchmark and manuscript and reviewed the resulting work.

References

1. FinQA: A Dataset of Numerical Reasoning over Financial Data (2021).
2. Bank-World: A Benchmark for Evaluating Agents on Banking Workflows (2026).
3. τ-Bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains (2024).
4. Oracle. FLEXCUBE Enterprise Limits and Collateral Management User Guide.
5. A Coefficient of Agreement for Nominal Scales (1960).
6. Global BankOps. Research manuscript, 23 November 2026.
7. Flitto. Confirmations, payment authority and ledger postings in eight configurations. 2026.

Appendix A. Work environment and grading validation

In all 800 evaluated runs, the file tool returned worksheet data to the agent at least once.

Table A1. Composition of the 50 tasks

Item	Count
Tasks by division · TF / LS / TO / AC / PT	10 / 10 / 10 / 10 / 10
Budapest subsidiary / Seoul head office	40 / 10
Tasks with SWIFT exchanges with another institution	19
Shared-folder documents	605 (median 12.5, range 3–18)
Structured workbooks / worksheets in 49 tasks	146 / 289
Initial mailbox messages / received SWIFT messages	132 / 66
Internal teams that answer mail / reply-selection rules	143 / 403
Rubric checks	585 (median 11.5, range 3–22)
Reference solutions / incorrect variants	50 / 744
Tool-call budget per run	60 in 42 tasks, 40–90 in eight

Each task includes a reference solution and incorrect variants, both written as tool-call sequences. The 50 reference solutions and 744 incorrect variants (median 14 per task) are replayed through the environment and graded. Each must reach its expected verdict, and a variant can also specify the criteria it is expected to fail. The validation suite adds 276 generated control workflows derived from the reference solutions, as well as judge fixtures and regressions from reviewed outputs.

We reviewed each division task by task for solvability, consistency between documents, and whether each hidden fact can be found in a document or by asking its owner. We also reviewed model trajectories after evaluation runs. When a review found an environment defect, such as a reply rule that missed a natural phrasing of a valid request, we corrected the task and revalidated the pack. Regression tests cover alternative ways of naming a transaction, compound requests, and similar messages that should receive a different reply. Runs on a changed task are excluded from the current results.

Table A2. Rubric checks by group

Group	Checks	Count
Payment and ledger	Transfers, payments, net ledger movements and transaction references	124
Written and saved outputs	Reports, mail, SWIFT, saved records and files, notices and cash forecasts	168
Verification and order	Request to the responsible team and reply read, or governing signed record read, before acting	64
Action restrictions	Allowed messages, transfers, payments and SWIFT releases	180
Completion	Whether the officer reported completion	49

Table A3. Grading responsibilities and evidence checks

Component	Decision or check
Programmatic checks	Verify recorded amounts, currencies, accounts, recipients, value dates and state changes. Compare evidence and action steps to check order.
Rubric-based LLM evaluation	Judge whether eligible messages, reports or saved records satisfy a written requirement. Identify the relevant action when order depends on the meaning of a message.
Evidence validation	Verify cited excerpts against eligible records and check figures required by the rubric.
Run verdict	A run passes when all required criteria pass. A missing or invalid required judge verdict leaves the run unscored.

Replay and agreement between judges

We collected the 800 runs on 12–23 November 2026. Across all 400 model–task pairs, both trials used identical task briefs, simulated worlds, and rubrics, verified by their SHA-256 hashes. Each pair also used the same seed, workplace framing, reasoning effort, and tool-call limit. To check the simulator, we replayed the 23,561 recorded actions with the recorded inputs. Every initial observation, tool response, and final state matched its saved record across all 800 runs. The replay used recorded actions and made no model calls.

Using cached judgments, the frozen task pack reproduced all 1,214 expected outcomes across 794 authored workflows, 276 controls, and 144 additional fixtures and regression cases. A development audit used scoring version 5 and compared Gemini 3.8 Flash with Claude Opus 5.5 (max) on 45 saved executions on the earlier task set, with one Opus 5.5 run per task. We used the development sample to refine rubric wording and citation scope. The sole criterion disagreement after clarification did not change the run outcome.

Table A4. Earlier 45-task set · outcomes scored by both judges

Audit stage	Criterion agreement	Cohen’s κ	Run agreement	Unscored by Opus
Initial audit	144/146 (98.6%)	0.962	40/40	5
After clarification	151/152 (99.3%)	0.982	45/45	0

A human reviewer also judged 35 individual criteria in a separate development sample. It included 30 cases where the two judges agreed and five where they disagreed. Flash matched the submitted judgment on 31 items and Opus on 28. This selected sample does not estimate accuracy across the full benchmark.

Appendix B. Division results and required confirmations

Each model configuration has two counted trials on every one of the 50 tasks. Table B1 reports pass@1 by division (Section 3.2). On the first trial, all eight configurations passed three tasks, and none passed eight tasks. After Holm correction, 15 of the 28 pairwise differences in pass@1 are significant at the 5% level. GPT-6.2 Astra and GPT-6 Astra are not separated from each other or from Fable 5.2, and each is separated from every other configuration.

Table B1. First-trial pass rates for 50 tasks and ten per division. Final two columns pool both trials.

Model	Effort	Overall	TF	LS	TO	AC	PT	Runs	Passed
GPT-6.2 Astra	max	78.0%	70.0%	70.0%	80.0%	80.0%	90.0%	100	77
GPT-6 Astra	max	74.0%	60.0%	50.0%	90.0%	70.0%	100.0%	100	71
Fable 5.2	max	56.0%	40.0%	50.0%	70.0%	50.0%	70.0%	100	56
Gemini 4 Carbon	high	48.0%	40.0%	40.0%	50.0%	40.0%	70.0%	100	50
Opus 5.5	max	34.0%	20.0%	20.0%	50.0%	30.0%	50.0%	100	33
Gemini 4 Argon	high	28.0%	20.0%	0.0%	50.0%	10.0%	60.0%	100	27
Grok 4.7	xhigh	16.0%	10.0%	10.0%	30.0%	10.0%	20.0%	100	16
Qwen3.8 Max Prime	xhigh	14.0%	20.0%	0.0%	50.0%	0.0%	0.0%	100	12

TF trade finance · LS loan servicing · TO treasury operations · AC accounting · PT pension and trust.

Table B2. Scope and missed requirements by division

Division (ten tasks each)	Work evaluated	Missed checks across failed runs in both trials
Trade Finance	Trade documents, payments and bank correspondence	92 of 104 missed verification or ordering
Loan Servicing	Loan records, terms and servicing decisions	111 of 112 missed verification or ordering
Treasury Operations	Reconciliation, FX and money-market settlement, and payment release	Of 64 failures, 50 missed verification or order and 37 missed a payment or ledger check
Accounting	Month-end accruals, revaluation, schedules and head-office returns	Of 107 failures, 95 missed verification or order and 86 missed a report, schedule or saved record
Pension & Trust	Eligibility, account instructions and record updates	70 of 71 missed a report, message or saved record

Table B3. Tasks by number of configurations that passed on the first trial

Division	0	1	2	3	4	5	6	7	8	Tasks
Trade Finance	2	1	2	2	1	1	0	0	1	10
Loan Servicing	3	1	0	3	1	2	0	0	0	10
Treasury Operations	1	1	1	0	1	1	2	1	2	10
Accounting	2	1	1	1	2	3	0	0	0	10
Pension & Trust	0	0	1	2	1	3	2	1	0	10
All	8	4	5	8	6	10	4	2	3	50

Across the eight configurations, 17 tasks were passed by at most two on the first trial, 14 by three or four and 19 by five or more.

Table B4. First trials on 44 mandatory-confirmation tasks, and passes on the other six

Model	Passed / 44	On time	Late read	Unread	No match	Other passed / 6
GPT-6.2 Astra	34	35	1	1	7	5
GPT-6 Astra	32	33	6	3	2	5
Fable 5.2	24	26	7	2	9	4
Gemini 4 Carbon	19	20	3	6	15	5
Opus 5.5	12	14	4	9	17	5
Gemini 4 Argon	10	12	2	9	21	4
Grok 4.7	4	7	2	6	29	4
Qwen3.8 Max Prime	4	8	5	13	18	3

The four confirmation outcomes sum to 44 for each configuration. Each trial counts once, prioritizing an unmatched request, then an unread reply, then a late read.

Grok 4.7 had no matching request for at least one required confirmation on 29 first trials. Qwen3.8 Max Prime had 18 such trials and another 13 with an unread reply. On the six tasks without a must-ask check, every configuration passed at least three. On must-ask tasks, the number of on-time verifications falls with the overall pass rate, from 35 for GPT-6.2 Astra to 7 for Grok 4.7 and 8 for Qwen3.8 Max Prime; these last two are the only pair out of order. A configuration that verified on time passed the task in 139 of 155 cases, and each Astra configuration failed only one task on which it verified on time.

Appendix C. Resource use and references to evaluation

Solver cost and tool use

Both axes of Figure 1 use the same 100 scored runs per configuration. The vertical axis is the fraction of these runs that passed, equivalently the mean of the two individual trial pass rates. The horizontal axis counts actions processed by the environment, including the finish action, divided by 100. A response that issues no executable action adds no tool call.

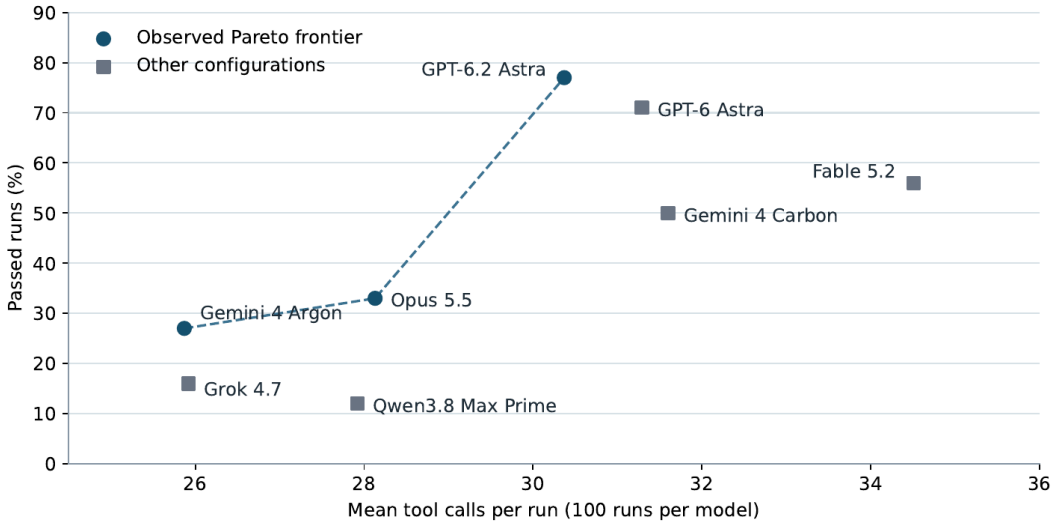

Figure 1. Pass rate and mean tool calls on 100 scored runs per configuration. The dashed line joins configurations for which no other has at least as high a rate and at most as many calls with a strict improvement on one axis.

The observed frontier contains Gemini 4 Argon, Opus 5.5 and GPT-6.2 Astra. Gemini 4 Argon used 25.87 calls per run and passed 27 of 100 runs. Opus 5.5 used 28.13 calls and passed 33. GPT-6.2 Astra used 30.37 calls and passed 77. Grok 4.7 used 25.92 calls and passed 16.

Table C1. Solver cost in US dollars · 100 runs per configuration

Model	Route (runs)	Cost per run	Total cost
GPT-6.2 Astra	OpenAI direct (50), OpenRouter (50)	0.55	54.86
GPT-6 Astra	OpenRouter (96), OpenAI direct (4)	1.07	106.62
Fable 5.2	OpenRouter (62), Anthropic direct (38)	4.81	480.75
Gemini 4 Carbon	OpenRouter (100)	0.68	67.71
Opus 5.5	OpenRouter (94), Anthropic direct (6)	1.70	169.88
Gemini 4 Argon	OpenRouter (100)	0.41	40.87
Grok 4.7	OpenRouter (100)	0.41	40.71
Qwen3.8 Max Prime	OpenRouter (100)	0.71	70.82

Table C1 accounts for solver responses on all 800 selected runs, including billed terminal responses that supplied no executable action. OpenRouter reported US$687.95 in usage charges across 702 runs. For the 98 direct API runs, recorded token usage at the applicable list prices gives an estimated US$344.28. Together these amount to US$1,032.23. Prompt caching and provider route affect the comparison.

Recorded references to evaluation

After scoring, we searched the reasoning blocks on tool-calling turns in all 800 runs for mentions of grading, the task designer, an expected answer, or being tested. We read all 643 matches and set aside 13 that referred to routine bank work, such as evaluating a funding request. These annotations played no part in scoring or in

selecting runs. We compared suspected attempts to game the verifier with the actions the model took and the grades those actions received.

Table C2 counts runs with at least one explicit reference. A further 71 runs described the environment as a simulation without explicitly mentioning evaluation. The system prompt and task materials do not use that word.

Table C2. Runs referring to evaluation and tool-calling turns with readable reasoning

Model	Explicit / 100 runs	Simulation only	Turns with reasoning
GPT-6.2 Astra	4	10	19.3%
GPT-6 Astra	0	0	13.8%
Fable 5.2	42	5	62.4%
Gemini 4 Carbon	10	17	51.3%
Opus 5.5	18	2	62.2%
Gemini 4 Argon	10	4	38.1%
Grok 4.7	9	15	63.7%
Qwen3.8 Max Prime	36	18	100.0%

In the traces reviewed, we found no incorrect work that passed because the model exploited a verifier gap.

The counts also depend on which reasoning each provider exposes. Readable reasoning accompanied every Qwen3.8 Max Prime tool-calling turn and 13.8% of GPT-6 Astra tool-calling turns. The study did not test whether document style prompted these remarks or estimate their effect on pass rates.