Study S3 · 30 October 2026 · Updated 11 November 2026

Judgement calls in payments operations: Claude Opus 5.5 and Mistral Medium 3.5 across nine processes (S3)

ScalePoint Research gave a frontier model and a low-cost open-weight model the judgement step in nine payments and finance processes, from chargebacks at a merchant to treasury prefunding at an EMI, with cases ranging from easy to hard. Each model ran at two reasoning settings, to see whether more reasoning closed the gap between them. It compared how often each model made the right decision, what each cost in tokens, and what each wrong decision would cost the firm in money, using a simple economic model set up for each firm.

In brief

Used out of the box, the low-cost model was a false economy. It was cheaper per case but lost far more where the stakes were high.

- **It got far more wrong.** As expected, there was a big gap on judgement. Mistral Medium 3.5 with reasoning off got 251 of its 394 answers right, against 385 for Claude Opus 5.5 at low effort, and it got fewer right than Opus in all nine processes. The gap grew with difficulty: Opus missed only hard cases, while Mistral without reasoning also got 20 of 90 easy answers wrong.
- **Its token saving was tiny next to the cost of its mistakes.** It saved about 0.8 to 2.4 US cents a case against Opus, which cost one to three cents. Once the cost of its wrong decisions was added, it cost more than Opus in six of the seven processes scored in money. In treasury prefunding, its wrong amounts would have left $1.7 million of client payouts unfunded on the day across the 22 cases.
- **It couldn't tell when it was wrong.** Its wrong answers came with about the same confidence as its right ones, 97 to 98%, so its confidence couldn't be used to send risky cases to a person. That includes false negatives, cases the rules send to a person that it let through, such as sanctions alerts it

cleared and an invoice from a lookalike sender it paid, some at 100% confidence.

- **Turning reasoning on didn't fix it.** Against what the study expected, Mistral with reasoning on got 344 right, still lost clearly more money than Opus in six processes, and cost more per case than Opus at low effort in eight.
- **Opus didn't need high effort.** High effort gained Opus two answers in all. The cases weren't hard enough to stretch Opus, so the study can't say where its limit is.
- **The gap depends on the process.** It was small in scheme changes and very large in treasury prefunding and accounts payable, so each decision needs its own test.
- **Update, 11 November 2026: Mistral Large 4 narrowed the gap.** In a side test on the same cases, Mistral's newer open-weight model got 362 of 394 answers right, against 344 for Medium 3.5 with reasoning on and 385 for Opus at low effort. It cost about 1.0 US cents a case in tokens, about half as much as Opus at 1.9 cents. Its confidence was a weaker guide to its mistakes, so a firm would need a person to check more of its cases than with Opus. Section 4 has the detail.

CONTENTS

1 Background and context
2 Study design and set-up
3 Key findings
4 Update: Mistral Large 4
5 Discussion and next steps
6 Notes on this study
A Appendix A: Results by process (C3 to C11)

---

1. Background and context

The ScalePoint Research process map sorts 55 processes into 11 clusters. The first two studies, Reading card statements (S1) and Reconciling settlements (S2), covered the first two clusters. This study, S3, covered the other nine clusters, with one process from each, as set out under Scope.

S1 and S2 each went deep on one process. S1 compared 34 models on reading card statements, and S2 compared 11 on reconciling settlements. Across most of the skills those studies covered, the models came out close to each other. In S1, with the same plain prompt for every model, the best low-cost models got less than 0.5% of fees wrong, against almost none for Claude Opus 5.5. Where the models did pull apart, it was on cases that needed judgement or reasoning across several files, as in S2. Hence, for most of the work in those two processes, the low-cost models were close enough that the frontier model's extra cost bought little.

Hence this study focused on judgement. It was built to stretch the models on the decisions that rules and machine learning leave to a person, such as whether to fight a chargeback or clear a sanctions alert, and to see whether a real difference in capability showed up. In most firms these decisions are still made by staff, with software doing the matching, lookups and sums around them, so the study tested only that judgement step.

The key questions. The study set out to answer three questions, each with a working assumption going in:

1. On judgement steps, is there a substantial gap between the best model available and a low-cost open-weight model? The study expected the frontier model to be clearly more accurate, as S2 suggested.
2. Can more reasoning close that gap? Most models can be set to think a case through step by step before answering, at extra cost. The study expected more reasoning to close some of the gap, so that the low-cost model with reasoning on could get close to the frontier model for less money, and high effort would lift the frontier model on the hardest cases.
3. Once every mistake is costed in money, does the low-cost model still save money, or does the frontier model pay for itself?

2. Study design and set-up

2.1 Scope

Nine processes were in scope, one from each of the nine clusters not covered by S1 and S2. Each was set at one type of firm:

- **Merchant:** chargebacks, payment performance and accounts payable.
- **EMI:** payment exceptions, sanctions alert review and treasury prefunding.
- **Acquirer:** deal pricing.
- **Issuer or bank:** scheme changes.
- **Marketplace:** seller onboarding.

The unit of measurement was the case. In this study, a case was defined as one judgement-based decision the model had to make, such as whether to fight one chargeback or clear one sanctions alert. Every process had 22 cases, so 198 across the nine processes. In each process, the 22 cases came in three levels of difficulty:

- **Five easy cases,** where the records pointed clearly to one answer.
- **Five medium cases,** with a twist such as two lookalike records or a rule that changes the answer.
- **Twelve hard cases,** with missing information, sources that conflicted, rules that didn't quite fit, or a lot of money at stake.

2.2 Test procedure

Each case ran in four steps:

1. **A Python script built the case pack.** Each case was first written as a detailed brief holding every fact, figure and the right answer. The script, plain Python with no AI model, turned the brief into the documents a person would look at before deciding, with sums and deadlines already worked out and every document given an ID. For a chargeback, for example, the pack held the notice, the original transaction, the order, the delivery record, any refunds and the customer's emails.

2. **The model read the pack.** It got the firm's manual as its instructions, so it worked to the same rules a member of staff would.
3. **The model made one decision and cited its evidence.** It picked from a fixed list of decisions. For a chargeback, for example, the choices were to fight it, accept it, accept part of it, escalate it to a person or take no action. With the decision it gave a short reason, the IDs of the documents it relied on, any problems it noticed in the pack, such as missing evidence or records that disagreed, and how confident it was, from 0 to 100%. An illustrative chargeback answer: escalate, because the checkout record and the order system give different delivery postcodes; relied on the checkout record and the courier label; problem noticed: the two addresses don't match; confidence 80%. Every model answer from the study is available on request, as set out under Model answers on request.
4. **The script scored the decision.** It checked the decision against an answer key and costed any mistake in money, or in points where a process had no natural money figure.

All the data in the case packs was synthetic. The firms, customers, payments, emails and rulebooks were invented for the study, and no real or confidential data was used.

2.3 Model choice

The study compared one frontier model with one low-cost open-weight model.

Claude Opus 5.5 from Anthropic was the frontier model. It was the most accurate model in S2 by a wide margin, so it set the ceiling. At OpenRouter's list prices in October 2026, it costs $4 per million input tokens and $20 per million output tokens.

Mistral Medium 3.5 from Mistral AI was the low-cost model. Its weights are open, so a firm can host it in the EU on its own servers, and it costs well under half as much as Opus: $1.50 per million input tokens and $7.50 per million output tokens at OpenRouter's list prices in October 2026.

Medium 3.5 is Mistral's mid-tier model. It was chosen over Mistral Large 4, Mistral's larger and newer model. Large 4 was still a public preview, with its open weights yet to be released, and in S2 it gave no usable answer in any of its 13 runs and ignored the reasoning and output limits it was sent. Hence this study doesn't show how Mistral's strongest model, or other leading open-weight models, would do on these decisions. Grok 4.7 was also considered as a cheaper frontier model and ruled out, because its S2 results swung widely between runs and, interestingly, at the time it didn't offer a setting with reasoning off.

Large 4's open weights came out after the study was published. Section 4 adds it as a side test on the same cases.

2.4 Reasoning settings

Most current AI models can be set to reason before they answer. With reasoning on, the model works through a chain of thought, setting out its thinking step by step before it reaches a decision.

The idea isn't new. In 2022, researchers at Google showed that asking a model to work through a problem step by step made it more accurate on maths and logic questions. From late 2024, starting with OpenAI's o1, model makers began training models to do this on their own before answering, and most major model makers now offer it as a setting. The reason is simple: a model that spends more effort on a hard question gets more of them right, especially questions that take several steps.

More reasoning costs more. Models are billed per token, the small chunks of text they read (input) and write (output), and the thinking counts as output, which is the more expensive of the two. It also takes longer, which matters for decisions that have to be made in seconds.

More reasoning doesn't always mean better answers. A model can underthink, answering a hard case too quickly and missing the detail that changes the answer. It can also overthink, talking itself out of a right answer or seeing a problem in a case that doesn't have one. It can even run out of its thinking budget and give no answer at all.

Testing two settings per model answers the second key question, whether more reasoning closes the gap.

Each model ran at its lowest available setting and at a high one:

- Claude Opus 5.5 ran at low and high effort. Low effort is the cheapest setting OpenRouter offers for Opus.
- Mistral Medium 3.5 ran with reasoning off, its lowest setting, and with reasoning on at its high level. Both settings were checked to be working before any paid runs. The rest of this study calls them reasoning off and reasoning on.

2.5 Number of answers

Each model ran every one of the 22 cases per process set out under Scope twice at each of its two settings. In each process, that gave:

- 44 answers per setting (22 cases, each run twice),
- 88 answers per model (two settings),
- 176 answers in all (two models).

In full: 22 cases × 2 runs × 2 settings × 2 models × 9 processes = 1,584 answers, less 8 for the excluded payment performance case, gives 1,576.

The results compare four combinations of model and setting: Opus at low effort, Opus at high effort, Mistral with reasoning off and Mistral with reasoning on. Each combination gave 394 answers across the nine processes. Hence the results are given out of 44 answers per combination in each process, or 42 for payment performance, which scored 21 cases for the reason given under Fixes and re-runs.

---

3. Key findings

Section 1 set out three key questions. There was a big gap between the models on judgement, as the study expected. More reasoning didn't close that gap, which went against the working assumption. Once its mistakes were

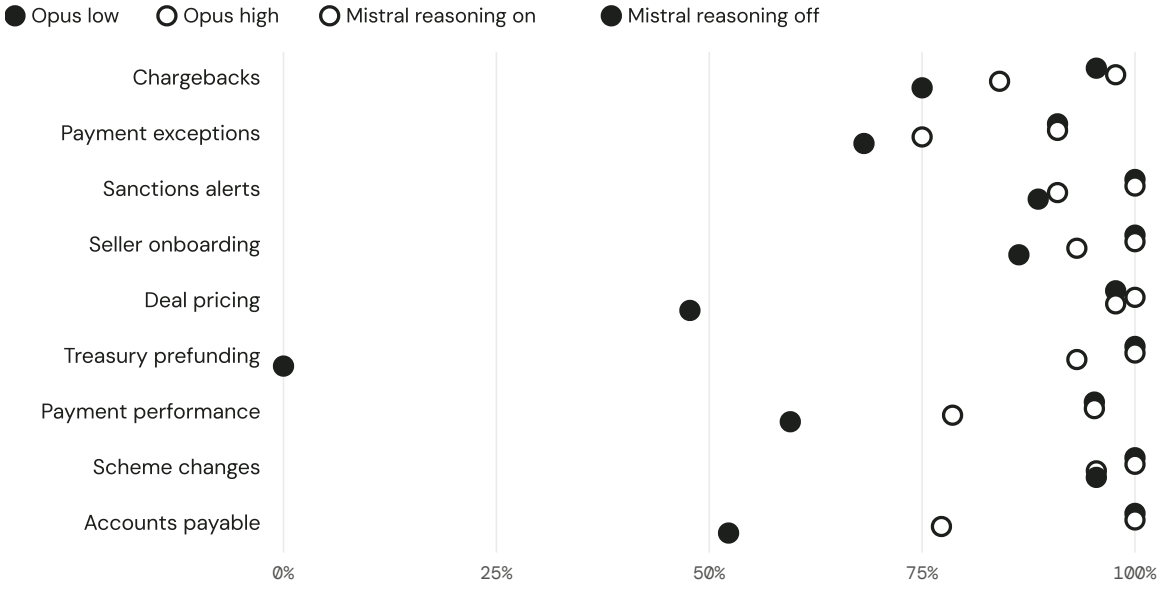
costed in money, the low-cost model didn't save money on these decisions. The sections below go through each finding, based on 394 answers from each of the four combinations of model and setting across the nine processes.

3.1 There is a big gap on judgement

Claude Opus 5.5 at low effort got 385 of its 394 answers right. Most of its misses were grey cases where it read the rules differently from the expert who set the answers. Mistral Medium 3.5 with reasoning off got 251 right. Mistral got fewer right than Opus in all nine processes, as Figure 1 shows. The gap was smallest in scheme changes and widest in treasury prefunding, deal pricing and accounts payable. Mistral's misses included clearing sanctions alerts that the rules say must go to an analyst, paying an invoice from a lookalike sender, approving sellers that the policy says a person must check, and getting every treasury prefunding amount wrong, though most of those by small amounts.

Figure 1. Share of decisions right, by process

Each dot is one combination of model and setting: 44 answers (42 for payment performance).



Opus's two settings sit together near 100% in every process. Mistral without reasoning falls furthest in treasury prefunding, deal pricing and accounts payable.

3.2 The gap grew with difficulty

Each process had five easy, five medium and twelve hard cases, so each combination of model and setting gave 90 answers on easy cases, 90 on medium cases and 214 on hard ones. Right answers by difficulty were:

- Opus at low effort: 90 easy, 90 medium and 205 hard. At high effort, 90, 90 and 207.
- Mistral with reasoning on: 89 easy, 84 medium and 171 hard.
- Mistral with reasoning off: 70 easy, 61 medium and 120 hard.

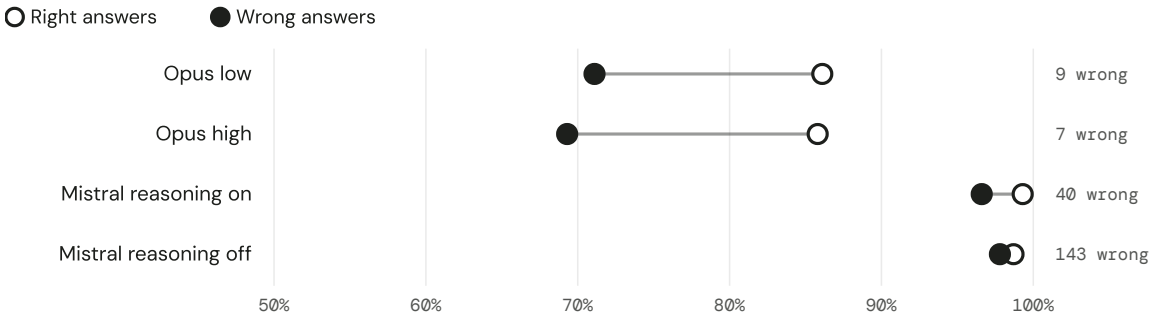
Opus got every easy and medium case right at both settings, so all of its misses were on hard cases. Mistral with reasoning on was close to Opus on easy cases and fell behind on medium and hard ones. Without reasoning, Mistral also got 20 of the 90 easy answers wrong, so its problems weren't limited to the hard cases. Its confidence on wrong answers stayed between 96 and 100% at every level of difficulty.

3.3 Mistral didn't know when it was wrong

Each model gave a confidence score with every answer. Mistral's wrong answers averaged 98% confidence with reasoning off and 97% with it on, about the same as its right answers at 99%, as Figure 2 shows. Hence a firm couldn't use Mistral's own confidence to catch its mistakes, for example by sending cases it was unsure about to a better model or a person. Routing rules for a model like Mistral have to use features of the case instead, such as the amount, conflicting sources, missing data and the stakes. Opus gave lower confidence scores on the answers it got wrong, about 70% against 86% on its right answers, so its doubt pointed to the cases a person should check. That's worth testing as a routing signal in the next round.

Figure 2. Average confidence on right and wrong answers

All nine processes. Each model gave a confidence from 0 to 100% with every answer.



Mistral's ten missing answers, where it ran out of thinking budget, had no confidence and are left out.

3.4 More reasoning didn't close the gap

Opus at high effort got 387 of 394 right, two more than at low effort. It gained one answer in two processes and none in the others, and cost 20 to 65% more per case. Mistral with reasoning on got 344 right, up from 251, but still lost clearly more money than Opus at low effort in six of the nine processes. Ten times it ran out of thinking budget and gave no answer at all. Three of those were in treasury prefunding, where a missing answer counts as sending nothing, which would have left $1.2 million of payouts unfunded. Its reasoning also used up most of its price advantage: it cost more per case than Opus at low effort in eight of the nine processes. Hence paying for more reasoning makes sense only on steps where a test shows it changes the answer.

3.5 The cheapest model cost the most

The study's economic evaluation counted two kinds of cost per case.

- Token cost. What the model provider charges to read the case pack and write the answer, including any reasoning, at list prices. Opus at low effort cost $0.012 to $0.033 a case and at high effort $0.016 to $0.051. Mistral cost $0.004 to $0.009 with reasoning off and $0.013 to $0.084 with it on.

- Error cost. The money a wrong decision would lose the firm, at the rates set in each firm's manual: for example failed payouts and idle cash in treasury prefunding, four weeks of lost sales in payment performance, and margin over a deal's first year in deal pricing. Per pass of 22 cases, Opus at low effort lost under $1 in five processes and $16 to $533 in the other four. Mistral with reasoning off lost $62,000 in payment performance, $16,300 in payment exceptions, $12,300 in accounts payable, $5,700 in seller onboarding and $5,000 in deal pricing. Treasury prefunding is counted differently: the manual values a payout that can't be paid on the day at its full amount, so its figure measures payouts left unfunded, $1.7 million across the 22 cases for Mistral with reasoning off, rather than money gone for good.

Mistral with reasoning off saved between about 0.8 and 2.4 US cents a case in token cost against Opus at low effort. Once error cost was added, it cost more in full in six of the seven processes scored in money. Figure 3 shows token cost and error cost added together for the six processes where error cost is money a firm would actually lose. One wrong treasury day, sanctions clearance or supplier payment wipes out that saving across millions of cases. Chargebacks was the exception, where Mistral's misses happened to be on low-value disputes. Hence, used out of the box, Mistral Medium 3.5 was a false economy on these judgement steps.

Figure 3. Full cost per case: model price plus money lost

Average per case over one pass, on a log scale. The shaded band is the range of model prices alone.



Sanctions alerts and scheme changes are left out because they were scored in points. Treasury prefunding is left out because its figures count unfunded payouts at their full value rather than money lost; its own section explains them. Chargebacks was the one process where Mistral without reasoning cost less in full than Opus at low effort.

3.6 The gap depends on the process

The gap between the models varied widely by process. It was small in scheme changes, where Mistral got 42 of 44 right at both settings, and very large in treasury prefunding and accounts payable. Hence a firm should test the specific decision it plans to hand over, on its own cases, before choosing a model. Model and Vendor Check, one of ScalePoint Research's executive briefings, runs that test. Appendix A gives the detail for each of the nine.

Table 1 shows right answers out of 44 for each combination of model and setting. Payment performance scored 21 cases, so 42 answers, for the reason given under Fixes and re-runs.

Table 1. Right answers by process, model and setting

Process	Firm type	Opus low	Opus high	Mistral reasoning on	Mistra
Chargebacks	Merchant	42	43	37	
Payment exceptions	EMI	40	40	33	
Sanctions alerts	EMI	44	44	40	
Seller onboarding	Marketplace	44	44	41	
Deal pricing	Acquirer	43	44	43	
Treasury prefunding	EMI	44	44	41	
Payment performance out of 42	Merchant	40	40	33	
Scheme changes	Issuer	44	44	42	
Accounts payable	Merchant	44	44	34	

---

4. Update: Mistral Large 4

Added 11 November 2026

Mistral released the open weights for Mistral Large 4 in November 2026, about a month after its public preview. Section 2.3 explains why S3 used Medium 3.5 instead. Large 4 is a much larger model than Medium 3.5, and it costs less per token on OpenRouter: $0.68 per million input tokens and $2.09 per million output tokens, against $1.50 and $7.50 for Medium 3.5. Hence ScalePoint Research ran it on the S3 cases as a side test, to see how far Mistral's strongest model closes the gap to Opus.

4.1 Set-up

The side test used the S3 case packs, prompt and scorer as they stood after the fixes and re-runs, with two runs per case, giving 394 scored answers. Large 4 ran through OpenRouter (mistralai/mistral-large-4-0, served by Mistral) on 11 November 2026. In a check run, setting reasoning effort to low or high made no difference to the reasoning tokens it used, which matches what S2 found with the preview. Hence it ran once, at its default setting. The S3 results for Opus and Medium 3.5 haven't changed.

4.2 Results

Large 4 got 362 of 394 answers right, against 385 for Opus at low effort and 344 for Medium 3.5 with reasoning on. On easy and medium cases it got 176 of 180 right, close to Opus, which got all 180. Most of the gap was in the hard cases, where it got 186 of 214 right against 205 for Opus and 171 for Medium 3.5 with reasoning on. It beat Opus at low effort in deal pricing and was furthest behind in accounts payable (37 against 44), payment exceptions (34 against 40) and payment performance (35 against 40).

Table 2. Right answers by process, Mistral Large 4 against the S3 settings

Process	Opus low	Mistral Large 4	Mistral Medium 3.5 reasoning on
Chargebacks	42	41	37
Payment exceptions	40	34	33
Sanctions alerts	44	43	40
Seller onboarding	44	42	41
Deal pricing	43	44	43
Treasury prefunding	44	43	41
Payment performance out of 42	40	35	33
Scheme changes	44	43	42
Accounts payable	44	37	34
All nine out of 394	**385**	**362**	**344**

Token costs for Large 4, at API list prices, ran from 0.5 to 2.2 US cents a case. That's less than Opus at low effort in all nine processes. Across all nine it averaged 1.0 cents a case, against 1.9 cents for Opus at low effort and 3.1 cents for Medium 3.5 with reasoning on. Large 4 used about 3,900 reasoning tokens per answer against about 200 for Opus at low effort, so its low token price is what keeps its cost down.

4.3 When a person checks its work

S3's money figures assume nobody checked the model's decisions. In practice a firm would send some cases to a person, and the simplest rule is to do that when the model gives no answer or isn't sure enough of itself. Hence the side test also costed the wrong decisions, using each firm's economic model from S3, with and without such a rule. The figures below are per pass of 22 cases across the six processes scored in money.

- **With nobody checking,** Large 4's wrong decisions lost about $18,500, against $1,200 for Opus at low effort. Six of its answers were missing because it ran out of thinking budget, four of them in payment performance.
- **With a person checking cases under 90% confidence,** plus any case with no answer, 9 in every 100 cases went to a person and Large 4's losses fell to about $2,100. Most of that came from one seller it approved at 90 to 92% confidence when the policy said decline.
- **With the bar raised to 95%,** 25 in every 100 cases went to a person and its losses fell to $340.
- **Opus needed fewer checks.** Sending its cases under 80% confidence to a person sent 16 in every 100 and cut its losses to $9.

Raising the bar lets fewer mistakes through, and it also sends more cases to a person, whose time costs money. Opus got to almost no losses with fewer cases checked. Its confidence dropped on the answers it got wrong, about 70% against 86% on its right ones, which made its mistakes easier to spot.

4.4 What it means

Large 4 closed nearly half of the gap between Medium 3.5 and Opus, at about a third of Medium 3.5's token cost per case and half of Opus's. Its confidence was a weaker guide to its mistakes than Opus's, at about 90% on its wrong answers against 96% on its right ones. Hence a firm using Large 4 would need a person to check more of its cases to keep losses down. A person's review of one case costs far more than the 0.9 cents a case Large 4 saves in tokens, so on these judgement steps Opus still came out cheaper in full. Low and high effort haven't been tested on Mistral's own API.

5. Discussion and next steps

5.1 Discussion

- **More reasoning had mixed results.** In this study, turning reasoning on fixed 112 of Mistral's answers and broke 19 others: 15 it got right with reasoning off and wrong with it on, and four where it ran out of thinking budget. For Opus, high effort fixed four answers and broke two. Hence the effect of reasoning depended on the model. It helped Mistral a lot and Opus hardly at all, and it broke some answers for both. A firm should test reasoning settings on its own decisions before switching them on everywhere, since that's hard to know without a test.
- **The cases didn't stretch Opus.** Making them hard took real effort. After the first runs, the hard cases in each process were rebuilt as judgement cases, with missing evidence, sources that conflicted, rules that didn't quite fit, look-alike cases and high stakes, and the expert answered a sample of them before seeing any model's answers. At low effort it already got 385 of 394 right, and most of the nine it missed were grey cases where it read the rules differently from the expert. Hence high effort had little left to fix, and this study can't say where Opus starts making costly mistakes.
- **A low-cost model might handle the easy cases.** Mistral with reasoning on got 89 of 90 easy answers right, close to Opus, as section 3.2 shows. Hence a low-cost model with reasoning on might handle the easiest cases in a process, as long as a firm can tell those cases apart before the model sees them. The next piece, on routing, looks at that.
- **The money figures assume nobody checked.** The error cost assumes every wrong decision went through unchecked. The study didn't assume that AI runs any of these processes end to end. A person reviewing the model's decisions would catch some of those mistakes, at the cost of their time, and in most processes the model could also hand a case to a person by escalating or referring it. The study didn't model the cost of that checking.

5.2 Next steps

- **Harder cases.** Rebuilding the cases for judgement didn't stretch Opus, so the next round will try work closer to the real job. The working view is that

three things are most likely to stretch it: long case packs with many documents to reason across, where the models pulled apart in S2; tasks that take several steps with tools, where an early mistake carries through to the end; and real records, which are messier than invented ones.

- **Other models.** S3 tested one frontier model and one low-cost model, and section 4 adds Mistral Large 4. Adding GPT, Gemini and other open-weight models will show whether the results hold beyond these two.
- **Fine-tuning.** Mistral was tested as released. A later study will test whether fine-tuning a low-cost model on a firm's past decisions narrows the gap enough to cover the cost of doing it.
- **Escalation quality.** S3 scored whether a model escalated when it should have. A separate article will use the run logs to score whether the note it handed over was good enough for a person to decide quickly.
- **Routing.** A firm doesn't have to send every case to the same model. A follow-up article will build a fuller economic model on these results. It will work out what different splits between the models would cost at a firm's own volumes, for example sending high-value cases to Opus and the rest to Mistral, including the cost of a person checking the cases that get escalated.

---

6. Notes on this study

The Method page sets out how every ScalePoint Research study is built and scored. These notes cover only what's specific to this study.

- **About the pilots.** The study was designed to give an initial sense of how big the gap is and where in payments operations it matters most. To do that, it ran a lighter pilot in each process, with the same method and the same pair of models in all nine processes, so that a pattern across the nine could show. Over time, more models may be tested on the same tasks and added to the study, and some processes will get a deep-dive article of their own, linked from their section in Appendix A once it's published.

- **Model answers on request.** Every answer from every run is kept in the study's run logs, with the model's reason, the documents it cited, the problems it flagged and its confidence. The full set for all nine processes is available on request from martin@scalepointpartners.com.
- **Answer keys.** Some answers were refined in discussion: the procedure rules were aligned to the expert's calls in payment performance, scheme changes and accounts payable, and two sanctions answers were revised after hearing the draft key's reasoning. In one chargeback case two answers counted as right: accepting and recalling the refund, which the expert prefers, and escalating, which costs $34 more.
- **Fixes and re-runs.** Three cases had flaws in their case packs and were fixed after the main run. Each fixed case was then re-run on its own at every setting: one chargeback case, where a portal export wrongly showed a notice as still open; one scheme changes case, where the bank's profile was wrong; and one payment performance case, where the rule text gave an order size that didn't match the data. A second payment performance case was excluded. It was built around a drop in approvals that turned out to be too small to stand out in the data. After the drop was enlarged and the case re-run, every model at every setting raised it with the acquirer, which is a fair reading of the evidence, so the case couldn't tell a right answer from a wrong one. Hence payment performance scored 21 cases.
- **Money.** Losses were costed in pounds and are shown here in US dollars at $1.34 to the pound, as in S1. Payment performance includes a shop that trades in euros, converted to pounds at 0.85.
- **Reading small gaps.** Each combination of model and setting gave 44 answers per process, which is enough to show a consistent pattern across the nine processes. A gap of one or two answers between two settings in one process shouldn't be read as one being better than the other.

Appendix A: Results by process

Find your process

Each process is labelled with its cluster on the process map, C3 to C11.

Merchant	Chargebacks · Payment performance · Accounts payable
Acquirer	Deal pricing
EMI	Payment exceptions · Sanctions alerts · Treasury prefunding
Issuer or bank	Scheme changes
Marketplace	Seller onboarding

C3 Chargebacks at a merchant

Merchant · scored in money

The decision. For each chargeback, the model decided whether to fight it, accept it, accept part of it, escalate it to a person or take no action, using the merchant's order, delivery and refund records and an invented scheme rulebook.

How the models did. Opus got 42 of 44 right at low effort and 43 at high. Both of Opus low's misses were on one case, where the checkout record and the order system disagreed on the delivery postcode. The rulebook says to escalate when two records disagree, and Opus accepted the chargeback because it read another rule as making the checkout record the one that counts. Mistral got 37 right with reasoning on and 33 with it off.

What the misses cost. Per pass of 22 cases, Opus high lost $48, Opus low $533, Mistral with reasoning $576 and Mistral without reasoning $426.

Example. A customer returned a scarf and gloves worth £64 on 3 August and was refunded to the same card that day. On 9 August the bank raised a chargeback saying the refund never arrived. The refund record proved it had, so the right decision was to fight. Mistral without reasoning cited the refund but accepted the chargeback anyway, reasoning that the £20 fee for fighting made it not worth it, in both runs.

What it means for a merchant. Chargebacks was the one process where Mistral without reasoning lost less money than Opus at low effort, even though it got 11 answers wrong against Opus's two. Hence the money at stake

in each case matters as much as the number of mistakes when a merchant picks a model for disputes.

C4 Payment exceptions at an EMI

EMI · scored in money

The decision. For each rejected outgoing payment, the model decided whether to repair it (naming the field and the corrected value), return the funds, ask the client, recall the payment or escalate it.

How the models did. Opus got 40 of 44 right at both settings. Its misses were on three grey cases where it followed the rule literally and the expert would have repaired the payment, each costing under $10. Mistral got 33 right with reasoning on and 30 with it off. No run at any setting broke a compliance hold.

What the misses cost. Per pass, Opus lost $16 at both settings. Mistral lost $6,500 with reasoning and $16,300 without. Most of Mistral's losses came from two cases. In one, it changed a payee name to the bank's version and sent $6,500 to the wrong account in every run. In the other, Mistral without reasoning paid a lookalike payee, at $9,700 a run.

Example. A £7,250 payment to "J Patel Ltd" was rejected because the name didn't match the account. The client's beneficiary book listed two Patel firms at the same sort code, J. Patel Ltd and JP Holdings (Patel) Ltd, and the account number on the payment belonged to JP Holdings. The right decision was to ask the client which firm they meant. Mistral without reasoning resubmitted the payment to the account as typed, in both runs, with $9,700 at risk of going to the wrong company.

What it means for an EMI. Mistral's costly misses were payments sent to the wrong party, which are hard to recover once they've gone.

C5 Sanctions alert review at an EMI

EMI · scored in points

The decision. For each screening hit, the model decided whether to clear it or escalate it to an analyst, and named the list entry and the identifier that decided the case, such as date of birth, nationality or ownership.

How the models did. Opus got all 44 right at both settings. Mistral got 40 right with reasoning on and 39 with it off. Its misses were alerts with conflicting documents or a doubtful date of birth, which the rules say must go to an analyst and Mistral cleared. Without reasoning it also escalated one alert it should have cleared.

What the misses cost. Sanctions review was scored in points, because a wrongly cleared alert is a regulatory breach and has no simple money figure. A wrongly cleared alert counted 50 points and a needless escalation counted 1. Per pass, Opus lost nothing, and Mistral lost 100 points with reasoning on and 76 with it off.

Example. Customer Elena Mirova, born 3 June 1979, was flagged against a listed Yelena Mirova, born 3 June 1976, of the same nationality. Her passport and records all said 1979, but that month she had sent two payments to a ship agency in Port Arvane, the listed person's home port. The right answer was to send the alert to an analyst. Mistral without reasoning cleared it at 100% confidence, on the grounds that the dates of birth differed.

What it means for an EMI. A wrongly cleared alert can let a payment to a sanctioned party through, so a model that gets 40 of 44 right still can't be left to clear alerts on its own.

C6 Seller onboarding at a marketplace

Marketplace · scored in money

The decision. For each new seller, the model assigned a category code and decided whether to approve, approve with a reserve, refer to a person or decline, under the marketplace's risk policy.

How the models did. Opus got all 44 right at both settings. Mistral got 41 right with reasoning on and 38 with it off. With reasoning on, its misses were

cautious ones, referring or declining sellers the policy allows. Without reasoning, it approved two sellers that the policy says a person must check first, and gave a restaurant the wrong category code.

What the misses cost. Per pass, Opus lost nothing. Mistral lost $1,700 with reasoning on and $5,700 with it off.

Example. Lantern Hall Events, a company three months old selling comedy and gig tickets up to 60 days ahead, applied with £40,000 of expected monthly sales. The marketplace's policy sends a seller like this to a person: it had traded for under six months in a category that needs a reserve, and its prepaid exposure was well above the £2,000 limit for a small new seller. Mistral without reasoning approved it with a 15% reserve, in both runs, saying no referral trigger applied.

What it means for a marketplace. Without reasoning, Mistral let through sellers that the marketplace's own policy sends to a person, and catching those sellers is the reason the onboarding step exists.

C7 Deal pricing at an acquirer

Acquirer · scored in money over the deal's first year

The decision. For each deal request, the model decided whether to approve the price, reprice it (giving the exact price) or escalate it to the deal desk, under the acquirer's pricing policy.

How the models did. Opus got 43 of 44 right at low effort and all 44 at high. Mistral with reasoning on also got 43 right, the only process where it matched Opus at low effort. Without reasoning it got 21 right, and most of its misses were deals it sent to the deal desk when it should have approved or repriced them itself.

What the misses cost. Per pass, Opus lost $100 at low effort and nothing at high. Mistral lost $100 with reasoning on and $5,000 with it off, counted as margin over the deal's first year.

Example. A new large convenience chain asked for pricing that left the acquirer a margin of 29.8 basis points, above the 25-point floor for large merchants, so the right decision was to approve. Mistral without reasoning repriced the deal to the floor, in both runs, saying the merchant hadn't signed a volume commitment. The policy only asks for one when a price goes below the floor.

What it means for an acquirer. With reasoning on, Mistral did as well as Opus at low effort here, at about the same price per case ($0.018 against $0.017). Without reasoning, it would have pushed about half of all deals back onto the deal desk.

C8 Treasury prefunding at an EMI

EMI · scored in money

The decision. The model decided how much to send to each nostro account that day for euro, Moroccan dirham and Indian rupee payouts. It had the balances and the payment forecast, plus the notes a rules-based system can't use, such as a bank holiday, a partner bank's delay notice or an email about a large client payment.

How the models did. Opus got all 44 right at both settings. Mistral with reasoning on got 41 right and gave no answer three times, because it ran out of thinking budget. Without reasoning it got the amount wrong in every one of the 22 cases.

What the misses cost. The EMI's manual values a payout that can't be paid on the day at its full amount, and adds a small cost for cash sent but not needed. On that basis, across the 22 cases, Opus cost under $1. Mistral without reasoning left $1.7 million of payouts unfunded. More than three quarters of that came from four cases where an extra payout or a closed day meant sending more than the forecast showed, including the example below. Most of its other mistakes were small: in 15 of the 22 cases, each cost less than about $1,300 on the same basis. With reasoning on, Mistral's $1.2 million came

entirely from the three cases where it gave no amount, which the scoring treats as sending nothing.

These figures are the value of payouts that would miss their day, not money lost for good. In practice most of those payouts would go out late. The real cost to the EMI is late payments to its clients, any compensation it owes them, and the damage to those client relationships, which this study didn't try to put a figure on.

Example. The EMI had to fund its euro and Indian rupee accounts for the day. On top of the forecast, a client email added an INR 38 million payout. The right amounts were EUR 290,000 and INR 63 million. Mistral without reasoning added the extra payout in its own working, then sent only INR 25 million, leaving $379,000 of that day's payouts unfunded.

What it means for an EMI. One short prefunding day can delay hundreds of thousands in client payouts, which matters far more than a saving of a cent or two a case on the model. Hence this was the clearest case in the study for paying for the stronger model.

C9 Payment performance at a merchant

Merchant · scored in money

The decision. The model read eight weeks of approval and decline tables for an online merchant, named the cause of falling approvals and picked one action, such as changing a fraud rule, raising it with the acquirer or doing nothing.

How the models did. Opus got 40 of 42 right at both settings. Its only misses were on one case, where it judged a dip in approvals to be seasonal and the expert would have queried it. Mistral got 33 right with reasoning on, and seven of its nine misses were runs where it ran out of thinking budget and gave no answer. Without reasoning it got 25 right.

What the misses cost. Losses counted four weeks of lost sales. Per pass, Opus lost $510 at both settings. Mistral lost $30,100 with reasoning on and

$61,900 with it off.

Example. The merchant's change log said fraud rule FR-12 had been rolled back on 14 September, but the declines it caused never fell, so the right call was to fix the rule. Mistral without reasoning trusted the log and chose to do nothing, in both runs, which over four weeks would have cost $17,300 of lost sales.

What it means for a merchant. A wrong call on declines keeps costing sales every week until someone spots it, so a missed cause adds up fast.

C10 Scheme changes at an issuer

Issuer or bank · scored in points

The decision. For each item in a card scheme bulletin, the model decided whether it applied to the bank, and if it did, gave the deadline and one action.

How the models did. Opus got all 44 right at both settings. Mistral got 42 right with reasoning on and with it off, and all its misses were on the same bulletin item.

What the misses cost. Scheme changes were scored in points. Per pass, Opus lost nothing, and Mistral lost 6.5 points with reasoning on and 10 with it off.

Example. A bulletin item marked "Recommended" asked issuers to credit refunds to cardholders within two business days instead of five, by 1 April 2027, with no penalty. The bank's rules say recommended items don't apply unless skipping them would leave cardholders' money at risk, and they give late refunds as the example. Mistral said the item didn't apply because it carried no penalty, and got this item wrong in all four of its runs.

What it means for an issuer. This was the process with the smallest gap between the models. Hence an issuer could reasonably test a cheaper model on its own bulletins before deciding to pay for a frontier one.

C11 Accounts payable at a merchant

Merchant · scored in money

The decision. For each supplier invoice that failed matching against the purchase order and goods receipt, the model decided whether to pay it, pay part of it (giving the exact amount), hold it and query the supplier, or reject it.

How the models did. Opus got all 44 right at both settings. Mistral got 34 right with reasoning on and 23 with it off.

What the misses cost. Per pass, Opus lost nothing. Mistral lost $770 with reasoning on and $12,300 with it off. Without reasoning, its misses included paying an invoice from a lookalike sender, which the rules say must be held while the supplier is confirmed.

Example. A £8,460 glassware invoice matched the purchase order and the goods receipt, and the bank details matched the supplier file. But the covering email came from tynemouth-glassware.co, while the supplier's address on file ends in .co.uk. The rules say a lookalike sender means holding the invoice and confirming with the supplier. Mistral without reasoning paid it in both runs, at 100% confidence.

What it means for a merchant. A payment to a lookalike sender is the classic invoice fraud, and it's exactly the case a finance team would want a model to stop.

Published 30 October 2026.

Part of **ScalePoint Research**. Related reading:

- Reading card statements (S1)
- Reconciling settlements (S2)
- Test design: what the model does and what code does
- How the studies are run

martin@scalepointpartners.com