

ILLUSTRATIVE Fictional feature and hypothetical results, to show the method. Not a benchmark or a customer result.

Is this AI feature ready for our product? How we decide

A completed example of how an AI feature is tested before it ships: the user task, the data boundaries, the tests and the decision rule, agreed before any model is compared.

01 THE FEATURE (FICTIONAL)

ITEM	DEFINITION
Feature	Reply drafts for support agents in a B2B help-desk product
User task	An agent opens a ticket and gets a suggested reply, grounded in that customer’s help articles and past resolved tickets. The agent edits and sends it, or discards it.
Data boundary	Only the tenant’s own articles and tickets, filtered by the agent’s permissions before anything reaches a model
Accepted draft	Sent with no edits or light edits, judged by reviewers against a written rubric
Fallback	If the model is slow, unsure or unavailable, no draft appears and the agent works as today

02 THE TESTS

CATEGORY	WHAT IT CHECKS	HOW IT IS JUDGED	CASES
Groundedness	Every claim in the draft is supported by the cited articles or tickets	Two reviewers score against the rubric; disagreements resolved by a third	240
Answerability	The draft declines when the answer is not in the sources	Correct abstention on questions with no answer in the data	60
Permission leakage	No content from another tenant or a restricted article appears	Any leak fails the whole evaluation	50
Adversarial input	Instructions hidden in ticket text do not change behavior	Draft ignores the injected instruction	50
Fallback	Timeouts and errors leave the agent workflow intact	Forced failures produce no draft and no error to the agent	Scripted

03 THE DECISION RULE (AGREED BEFORE TESTING)

PROCEED ONLY IF	ASSUMED THRESHOLD	AND	ASSUMED THRESHOLD
Groundedness	90% or more	Adversarial pass rate	95% or more
Correct abstention	85% or more	Response time (p95)	4 seconds or less
Permission leaks	Zero	Cost per accepted draft	\$0.05 or less

ILLUSTRATIVE Fictional feature and hypothetical results, to show the method. Not a benchmark or a customer result.

The scorecard: two models, the same tests

The current model and an Amazon Bedrock candidate, run on the same 400 cases. The values below are hypothetical, to show how a result is read.

MEASURE	THRESHOLD	CURRENT MODEL	BEDROCK CANDIDATE
Groundedness	90%+	88% (211 of 240)	86% (206 of 240)
Correct abstention	85%+	80% (48 of 60)	87% (52 of 60)
Permission leaks	Zero	0 of 50	0 of 50
Adversarial pass rate	95%+	94% (47 of 50)	96% (48 of 50)
Response time (p95)	4 s or less	3.1 s	2.2 s
Drafts accepted by reviewers	-	62%	58%
Model and retrieval cost per draft (assumed)	-	\$0.018	\$0.012
Cost per accepted draft	\$0.05 or less	\$0.029	\$0.021

Cost per accepted draft = cost per draft divided by the share of drafts accepted. Green meets the threshold; red does not. All values hypothetical.

READING THE RESULT

- **Neither model passes.** Both miss the groundedness threshold, so choosing between them now would be premature.
- **The failures share a cause.** In this example, most ungrounded drafts cite outdated help articles that retrieval still returns. That is a data problem, not a model problem.
- **The candidate is faster and abstains better here,** but these differences only matter once groundedness passes.

THE DECISION IN THIS EXAMPLE

Proceed

All thresholds met

Change

Fix retrieval, then rerun the same 400 tests

Stop

Leaks, or no path to the thresholds

ILLUSTRATIVE Fictional feature and hypothetical results, to show the method. Not a benchmark or a customer result.

If the retest passes: the first implementation slice

What ships first, how it is protected, and how it can be switched off.

PART	WHAT IT INCLUDES
Application	A draft panel in the agent's ticket view, behind a feature flag that can be enabled per customer
Access	Retrieval filtered by tenant and agent permissions before the model call; no cross-tenant index
Release gate	The 400-case set runs on every change to prompts, retrieval or model. A release ships only if every agreed threshold on page 1 still passes, groundedness has not fallen more than 3 percentage points since the last release, and there are zero permission leaks. Any leak fails the release.
Monitoring	Acceptance rate, edit size, abstention rate, response time, cost per accepted draft and flagged outputs, reviewed weekly
Rollback	Feature flag off per customer or for everyone; timeouts and errors already fall back to no draft

BEFORE CHOOSING A BEDROCK MODEL

- Confirm model availability, quotas and pricing in the chosen Region from current AWS documentation, on the date of the decision.
- Confirm data-handling terms for the specific model and service against your requirements.
- Rework prompts and output parsing for the candidate and rerun the full test set. Models are not swapped as if interchangeable.

WHAT THE BUYER RECEIVES

- The agreed test set and rubric, owned by your team
- A scorecard like page 2, with measured values from your data
- A written proceed, change or stop recommendation, with the evidence behind it
- If proceeding: the implementation slice above, with its monitoring and rollback

Funding. For an eligible proof of concept, AWS program support may help. It is optional, depends on AWS approval, and is considered only after the feature itself makes sense.