AI ARCHITECT & ENABLEMENT LEAD

20 enterprise AI architecture cases

An interview casebook covering data preparation, architecture, evaluation, and when to fine-tune.

Case index

01 Reusing ten years of incident reports · 02

02 Employee policy and benefits assistant · 04

03 Customer service answer drafting · 05

04 Contract clause comparison · 06

05 Engineering repository onboarding · 07

06 Support ticket routing · 08

07 Invoice reconciliation · 09

08 Meeting-to-action tracker · 10

09 Service request fulfillment · 11

10 Multilingual product support · 12

11 Natural-language analytics · 13

12 Inventory demand forecasting · 14

13 Service anomaly triage · 15

14 Duplicate customer records · 16

15 Executive signal summaries · 17

16 Product defect inspection · 18

17 Digital forms and scanned archives · 19

18 Developer pull-request review assistant · 20

19 Standardizing incident postmortems · 21

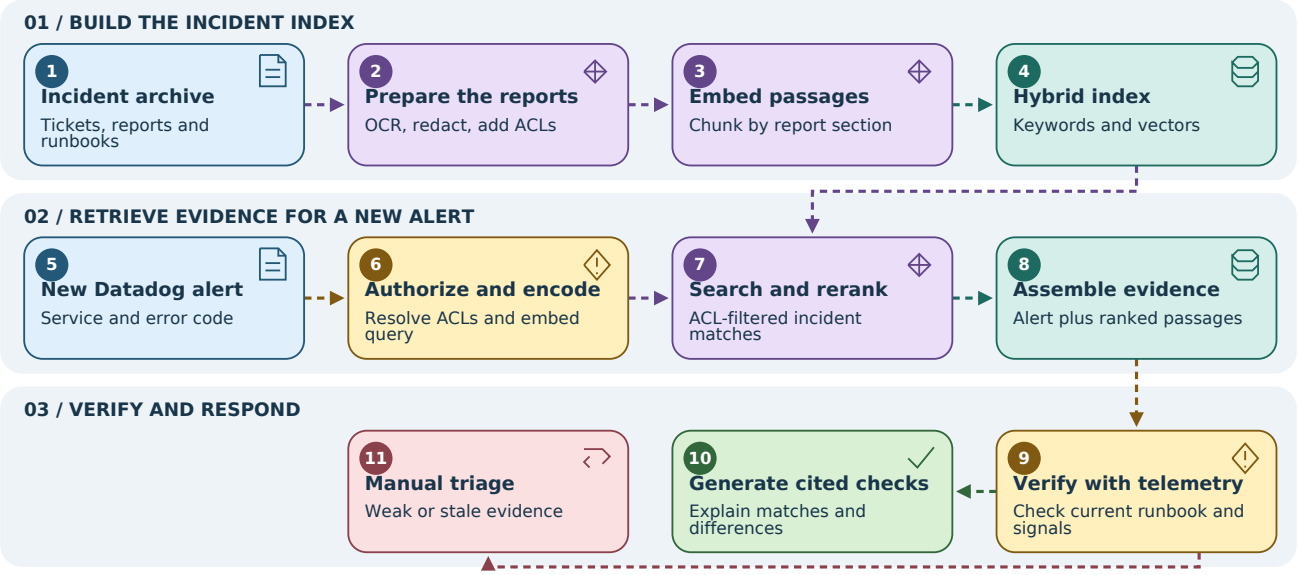
20 Regulated case intake and categorization · 22

Fine-tuning decision guide · 23

CASE 01 · HYBRID RAG + LIVE TELEMETRY

Reusing ten years of incident reports

On-call engineers need to find prior incidents with the same failure signature and compare their causes and workarounds to a current alert.



Read the diagram

The archive supplies historical matches. A separate request path brings in the new alert and the caller’s permissions. Both paths meet at search and reranking.

The verification step checks current runbooks and live evidence before recommending diagnostic checks. A weak or obsolete match goes to manual triage.

CASE 01 / IMPLEMENTATION AND VALIDATION

Putting incident RAG into practice

Prepare the data

Inventory postmortems, tickets, PDFs and runbooks. Deduplicate by incident ID and revision. OCR scans and spot-check extracted error codes, tables and timestamps. Remove secrets, map renamed services, attach access groups, environment, component, date, document version and source URL. Split by sections such as symptoms, timeline, verified cause and remediation. Build keyword and vector indexes. Reconcile additions, permissions and deletions through a versioned ingestion ledger.

Design the workflow

For a Datadog alert, extract the service, error signature and environment. Filter by current user permissions before retrieving chunks. Combine exact-code and semantic search, rerank a short list and group evidence by incident. Show dates, similarities, differences and citations. Check current telemetry and current runbooks separately before proposing a diagnostic step. Mark outdated fixes and abstain when evidence is weak.

Evaluate before rollout

Replay incidents chronologically with future reports excluded. Measure incident-level Recall@5, cited-claim support and time to useful next step. Include restricted reports and renamed services. Start in advisory mode with one team.

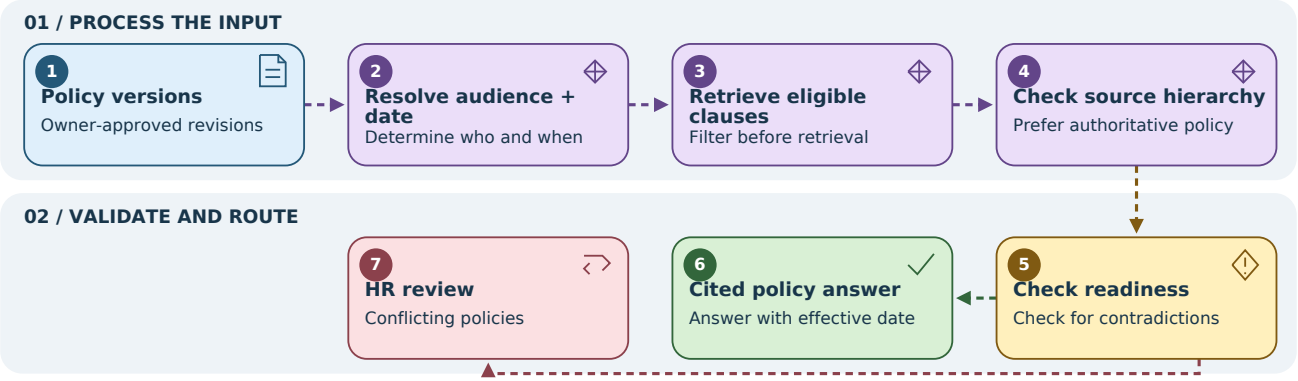
Technique choice

The archive is evidence that changes over time. RAG preserves provenance. If a stable owner-classification rubric still causes repeated errors after prompting, evaluate a separate LoRA adapter.

CASE 02 · EFFECTIVE-DATED RAG

Employee policy and benefits assistant

Staff ask policy questions whose answers vary by location, date or employment category.



Prepare the data

Ingest approved documents with jurisdiction, audience, effective dates, version and policy owner. Store superseded versions for explicit historical queries. Create an approved source hierarchy and define how conflicts reach HR.

Design the workflow

Resolve the relevant employee attributes and requested effective date. Retrieve eligible clauses and present cited answers with any missing eligibility inputs. Escalate conflicts or individualized determinations.

Evaluate before rollout

Test boundary dates, regional exceptions, audience restrictions and conflicting documents. Track HR-reviewed correctness and unsupported claims.

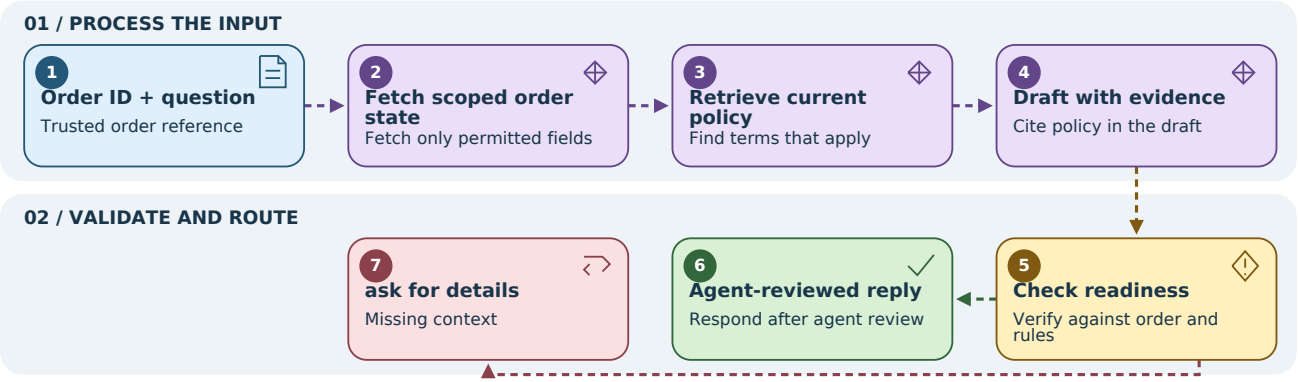
Technique choice

Use retrieval for policy and deterministic rules for eligibility where formal rules exist. Version changes belong in the policy store.

CASE 03 · RAG + SCOPED TOOLS

Customer service answer drafting

Agents need grounded answers that reflect policy and the customer's actual order state.



Prepare the data

Curate current help articles, policy versions and approved response templates. Separate knowledge docs from customer records. Expose permissioned, narrow order-status and entitlement APIs.

Design the workflow

Retrieve policy, fetch the order by a trusted ID, and draft a response with internal citations. Ask for clarifying information when needed. An agent reviews the draft and handles exceptions.

Evaluate before rollout

Measure policy accuracy, agent edit burden and resolution time. Include cross-account denial, stale policy, and returned or partially shipped orders.

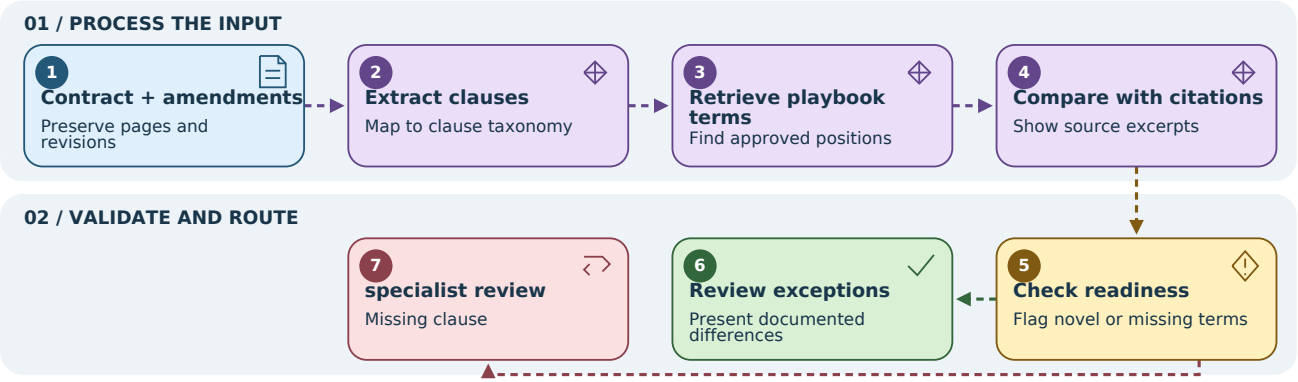
Technique choice

Consider fine-tuning only if a stable support tone or classification rubric underperforms few-shot prompting on held-out examples.

CASE 04 · STRUCTURE-AWARE RAG + EXTRACTION

Contract clause comparison

Procurement wants to compare supplier terms against an approved playbook.



Prepare the data

Parse agreements and amendments into sections, retaining cross-references, page anchors and document hierarchy. Create a clause taxonomy and preserve effective versions and access restrictions.

Design the workflow

Extract clause candidates into a schema, retrieve corresponding playbook text, and show side-by-side excerpts. Route missing, conflicting and unusual clauses to a reviewer.

Evaluate before rollout

Evaluate clause recall, citation precision and missed exception rate on reviewed contracts. Include scanned and amended agreements.

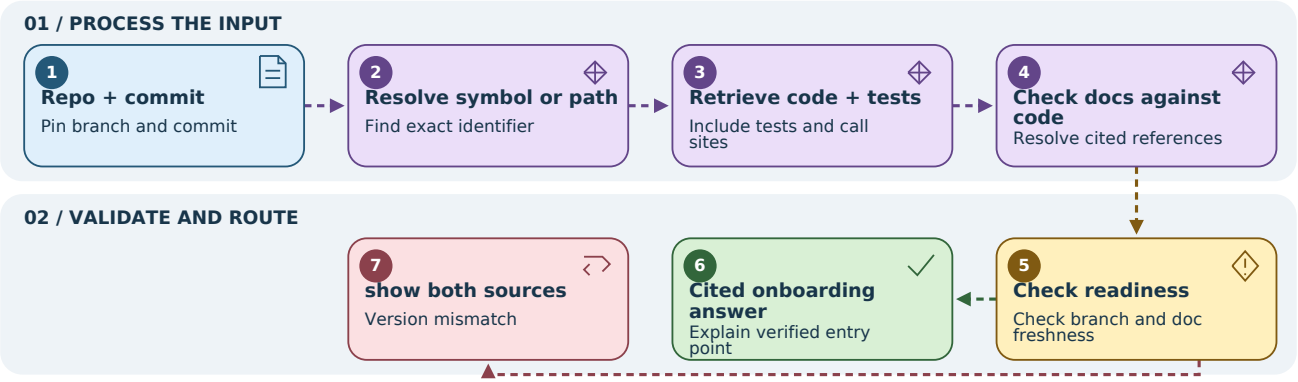
Technique choice

Use the model to surface evidence and differences. Keep acceptance decisions with authorized reviewers.

CASE 05 · CODE SEARCH + RAG

Engineering repository onboarding

New developers need to find where a flow starts, what it touches and which tests demonstrate expected behavior.



Prepare the data

Index approved repositories by commit and branch. Parse symbol definitions and links between callers, tests, database migrations and architecture records. Enforce repository permissions in retrieval.

Design the workflow

Resolve identifiers with symbol search, then retrieve nearby implementation and documentation. Cite repository path, line and commit. Flag where docs and code disagree.

Evaluate before rollout

Use known onboarding questions and check answer accuracy, link validity and time to locate the right entry point. Include branch version mismatches.

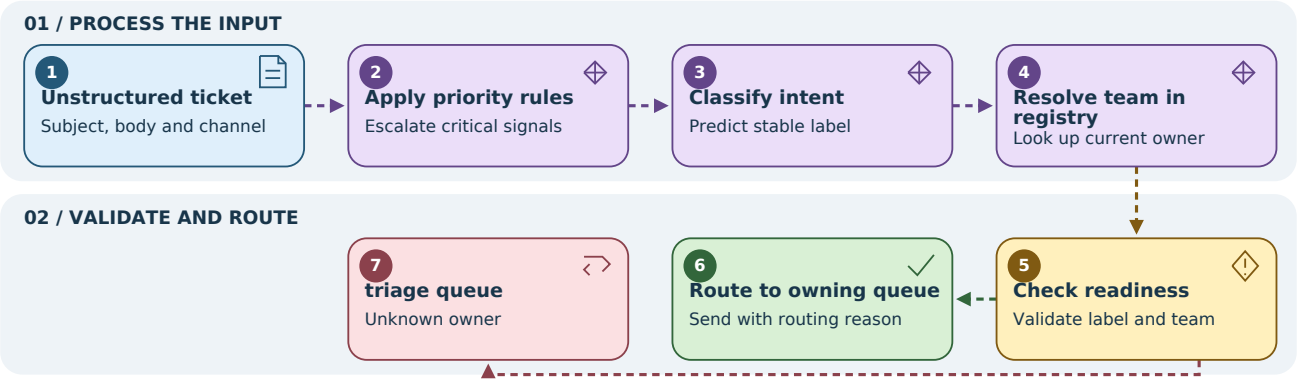
Technique choice

Use a fresh index for changes to code. Fine-tuning can teach formatting but cannot keep repository facts current.

CASE 06 · RULES + CLASSIFICATION

Support ticket routing

Incoming tickets need an owner, urgency rating and escalation path.



Prepare the data

Define an owned-team taxonomy and create labeled examples, including unknown and multi-team cases. Group near-duplicate cases and split historically. Maintain current ownership in a separate registry.

Design the workflow

Apply deterministic priority rules, then route ambiguous text through a classifier or an LLM with structured output. Resolve owner IDs via the registry and send uncertain cases to triage.

Evaluate before rollout

Compare keyword, conventional classifier and few-shot baselines. Track macro-F1, critical-class recall and manual reroute rate.

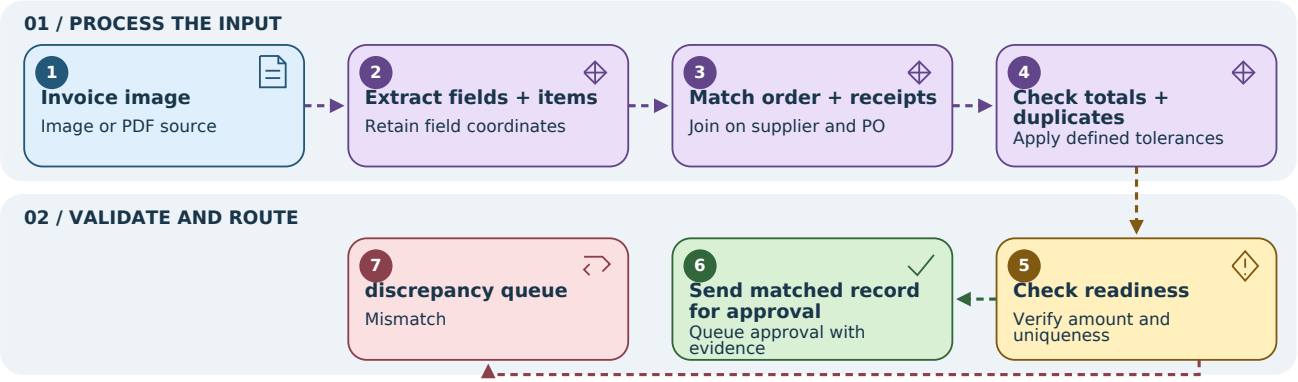
Technique choice

Start with the cheapest adequate classifier. LoRA is reasonable when stable label distinctions remain hard and training improves the held-out test.

CASE 07 · DOCUMENT AI + DETERMINISTIC MATCHING

Invoice reconciliation

Accounts payable wants to match invoices against purchase orders and receipts.



Prepare the data

Extract line items, quantities, currencies, tax and supplier identifiers. Keep page coordinates, field confidence and the original document. Normalize units and currencies according to approved reference data.

Design the workflow

Match against trusted records by supplier and purchase order. Apply explicit totals and tolerance rules, detect duplicates and queue discrepancies with evidence. Use the model to explain mismatches.

Evaluate before rollout

Measure field accuracy, false matches and review time. Test multi-page items, duplicate submissions and currency mismatches.

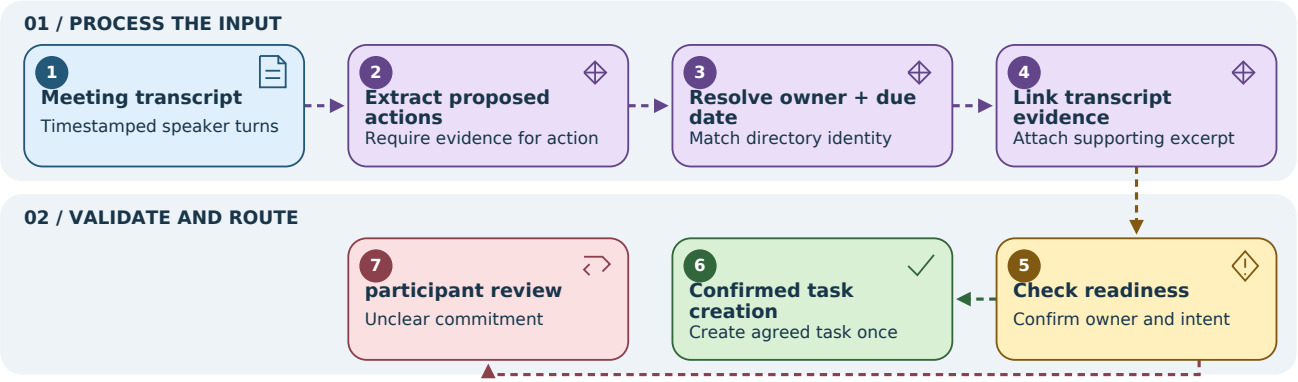
Technique choice

Use a document model for unstructured input. Arithmetic, approval and payment belong to governed business services.

CASE 08 · TRANSCRIPTION + STRUCTURED EXTRACTION

Meeting-to-action tracker

Teams need accepted commitments from long meetings.



Prepare the data

Obtain approved transcripts with timestamped speaker turns. Define a schema for action, owner, due date, evidence and uncertainty. Maintain participation and retention rules.

Design the workflow

Extract proposed items with citations to transcript spans. Resolve identities against an approved directory, flag uncertain owners and let participants confirm before creating tasks.

Evaluate before rollout

Track precision of accepted commitments, owner correction and duplicate task rate. Include hypothetical suggestions and shifting deadlines.

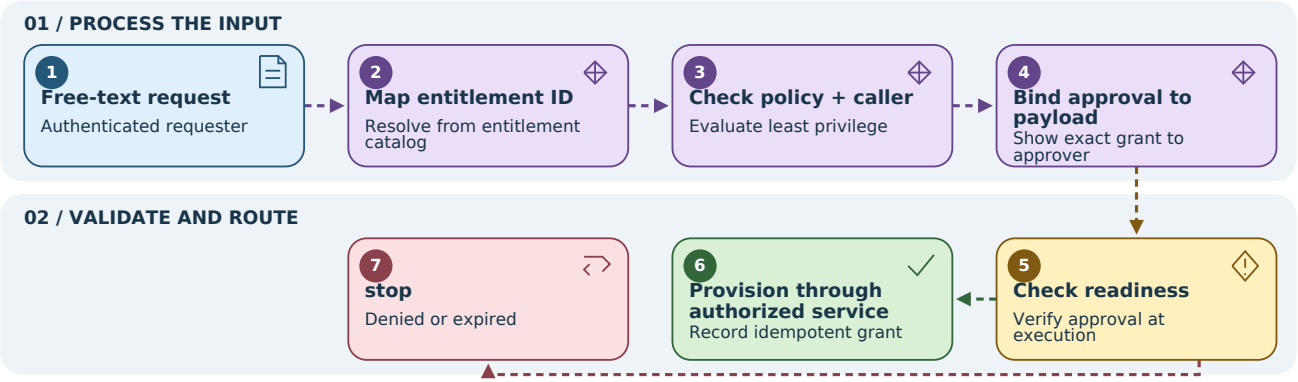
Technique choice

A small adapter may improve extraction conventions after a strong schema and prompt baseline. Human confirmation addresses ambiguous intent.

CASE 09 · BOUNDED TOOLS + ORCHESTRATION

Service request fulfillment

Employees request access to an internal application in free text.



Prepare the data

Model the process as a durable state machine. Define a catalog of allowed entitlement IDs and authorized approvers. Bind requests to the caller's authenticated identity.

Design the workflow

Interpret the request into catalog IDs, verify policy, prepare an approval with the exact requested entitlement and submit it through an authorized service. Use idempotency and reconcile timeouts.

Evaluate before rollout

Test duplicate deliveries, expired approval, modified payloads and unavailable provisioning. Track completion and incorrect grants.

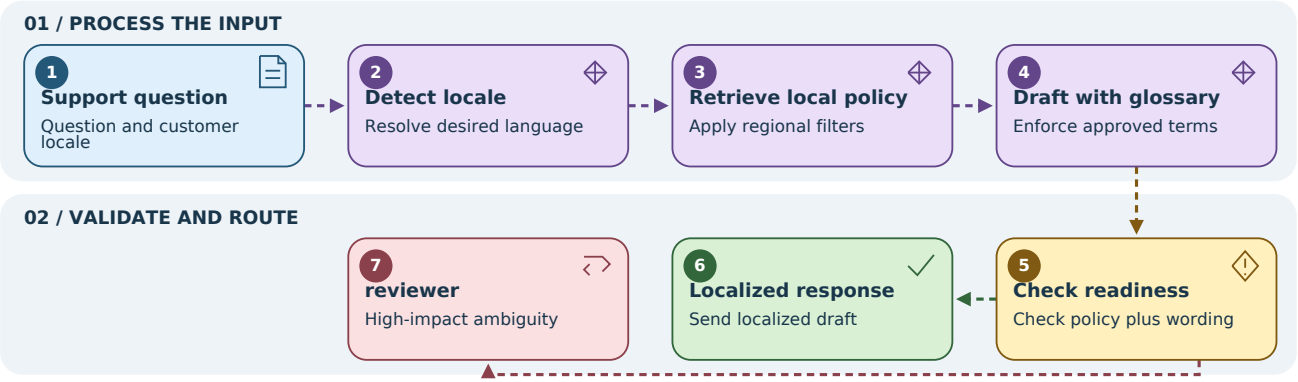
Technique choice

Use AI for intent interpretation. State, authorization and writes require deterministic controls.

CASE 10 · TRANSLATION + GLOSSARY + RAG

Multilingual product support

A support team must answer accurately across several languages with consistent product terminology.



Prepare the data

Collect approved glossary entries, locale-specific articles and reviewed bilingual examples. Define how to handle untranslated feature names, formatting and policy variants.

Design the workflow

Detect language, retrieve locale-appropriate evidence, and draft using glossary constraints. Route uncertain wording or high-impact content to a human reviewer.

Evaluate before rollout

Score semantic accuracy and terminology by language and locale. Compare throughput and correction burden to the existing process.

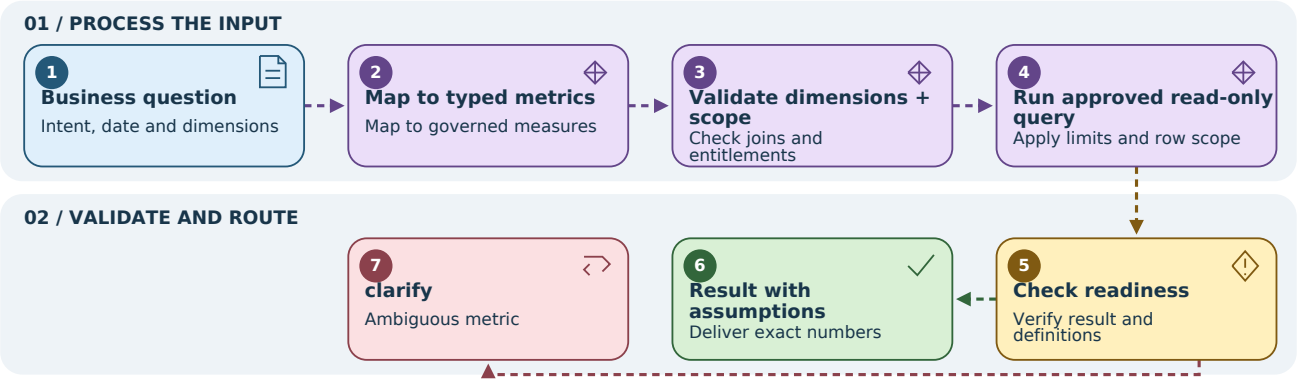
Technique choice

Use glossary and retrieval for evolving products. LoRA can help persistent terminology or style errors if reviewed parallel data is available.

CASE 11 · SEMANTIC LAYER + CONSTRAINED QUERY GENERATION

Natural-language analytics

Managers want to ask business questions without writing SQL.



Prepare the data

Define approved metrics, grain, dimensions and permitted joins. Expose data with caller-scoped access. Prepare examples and a set of queries with known results.

Design the workflow

Map the question to a typed query plan, validate it, and compile it through the semantic layer. Execute read-only with row-level controls and cost limits. State metric definitions and filter assumptions.

Evaluate before rollout

Test ambiguous metrics, time zones, duplicate joins and unauthorized columns. Compare returned numeric results to known answers.

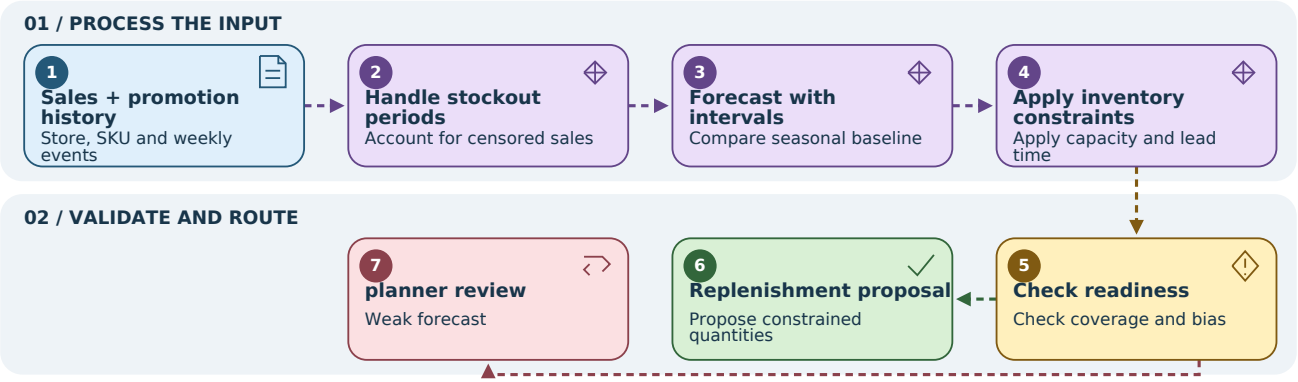
Technique choice

Use an LLM for interpretation and deterministic execution for arithmetic. Start with a narrow set of governed metrics.

CASE 12 · TIME-SERIES FORECASTING

Inventory demand forecasting

A retailer needs weekly forecasts that reflect promotions and stockouts.



Prepare the data

Build a time-indexed store and SKU dataset with holidays, promotions, price and stock availability. Distinguish observed sales from unconstrained demand and define a treatment for censored periods.

Design the workflow

Compare seasonal naive baselines with regression or forecasting models. Produce intervals and connect forecasts to a deterministic replenishment rule with lead-time and capacity constraints.

Evaluate before rollout

Use rolling backtests, horizon-specific error and bias metrics, and simulated stockout and overstock results. Track promotions separately.

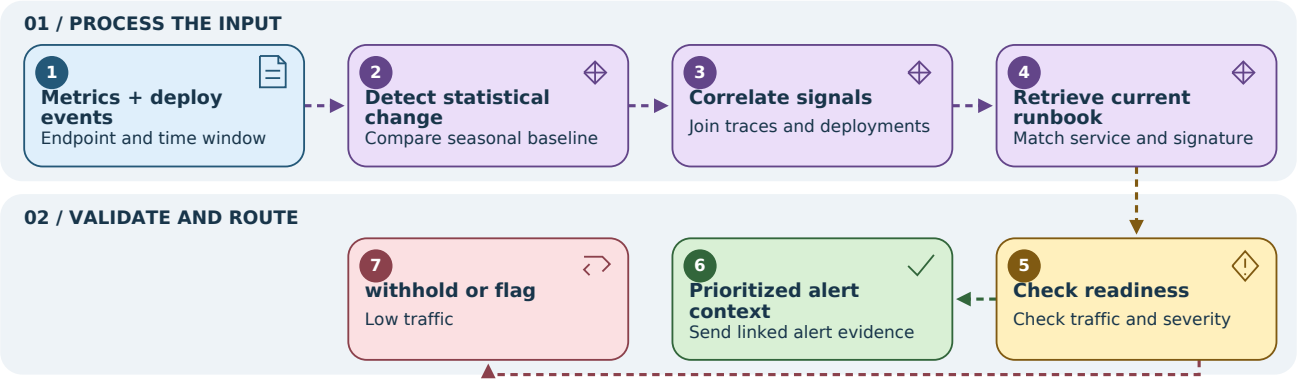
Technique choice

Use forecasting methods for numeric prediction. An LLM can explain inputs and assumptions.

CASE 13 · STATISTICAL DETECTION + EXPLANATORY RAG

Service anomaly triage

A platform team wants fewer noisy alerts and better incident context.



Prepare the data

Curate metrics by endpoint and traffic pattern. Identify maintenance and deployment windows. Link traces, alerts and runbooks by service and timestamp.

Design the workflow

Detect changes with seasonal baselines and minimum-volume rules. Group correlated signals, then use a model to summarize linked telemetry and cite applicable runbooks.

Evaluate before rollout

Track alert precision, missed events and on-call review time. Evaluate high p95 on low volume separately from persistent degradation.

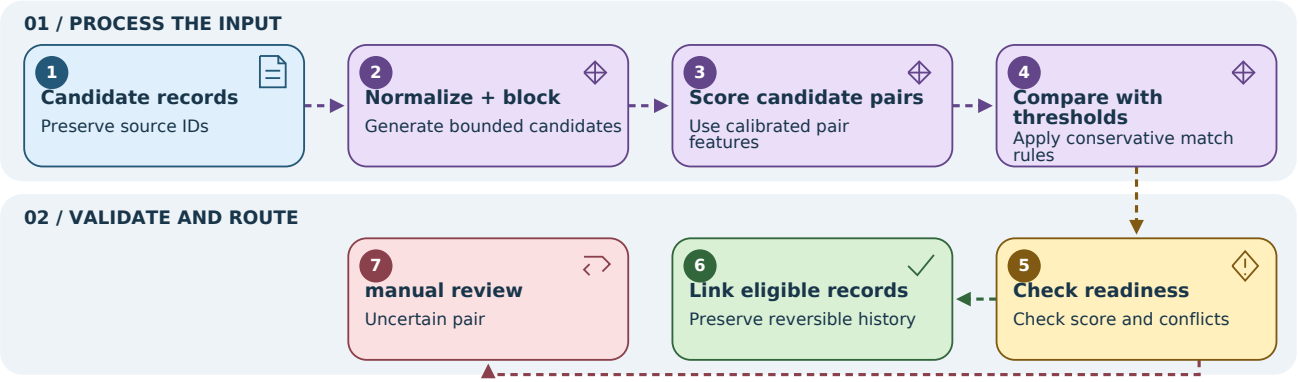
Technique choice

Separate numeric detection from narrative explanation. Keep service level and paging rules explicit.

CASE 14 · ENTITY RESOLUTION + REVIEW

Duplicate customer records

Operations needs to merge duplicate customer profiles safely.



Prepare the data

Normalize names, organization identifiers and contact fields. Build candidate sets using blocking rules. Label matches, near matches and hard negatives.

Design the workflow

Score pairs with deterministic and learned features. Auto-link only narrowly defined high-confidence pairs. Show uncertain evidence to reviewers and preserve source IDs and a reversible merge history.

Evaluate before rollout

Measure precision and recall at each threshold and review false merges. Test households, shared addresses and regional naming conventions.

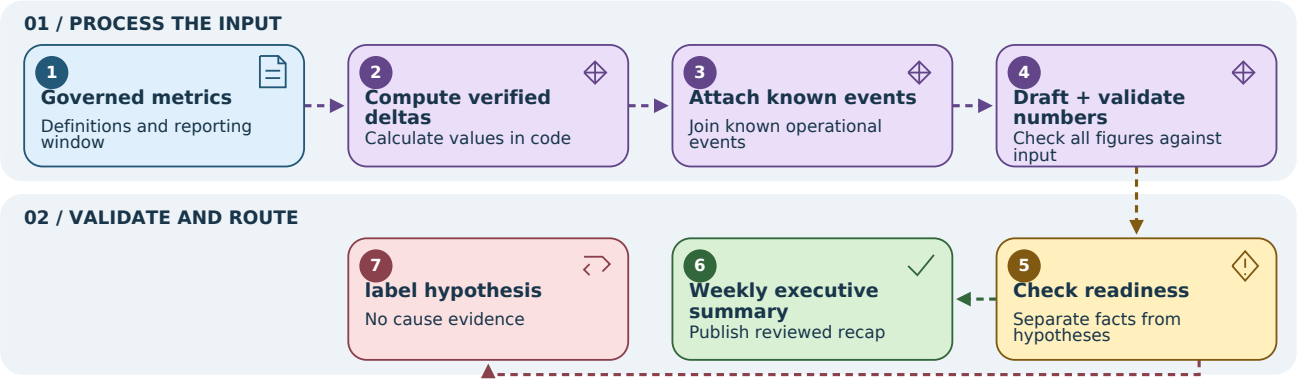
Technique choice

Start with matching features and a calibrated scorer. Use an LLM only where unstructured evidence adds value.

CASE 15 · VERIFIED METRICS + CONSTRAINED GENERATION

Executive signal summaries

Leaders need a weekly summary of trends and the reasons teams can verify.



Prepare the data

Define metrics in the semantic layer. Store approved thresholds, reporting windows and annotations for launches and disruptions.

Design the workflow

Compute values and deltas in code. Provide the model with a small verified table and available annotations. Validate numbers in the draft and mark explanations as hypotheses when evidence is missing.

Evaluate before rollout

Track numeric fidelity, omission rate and analyst edits. Include late-arriving data and changed definitions.

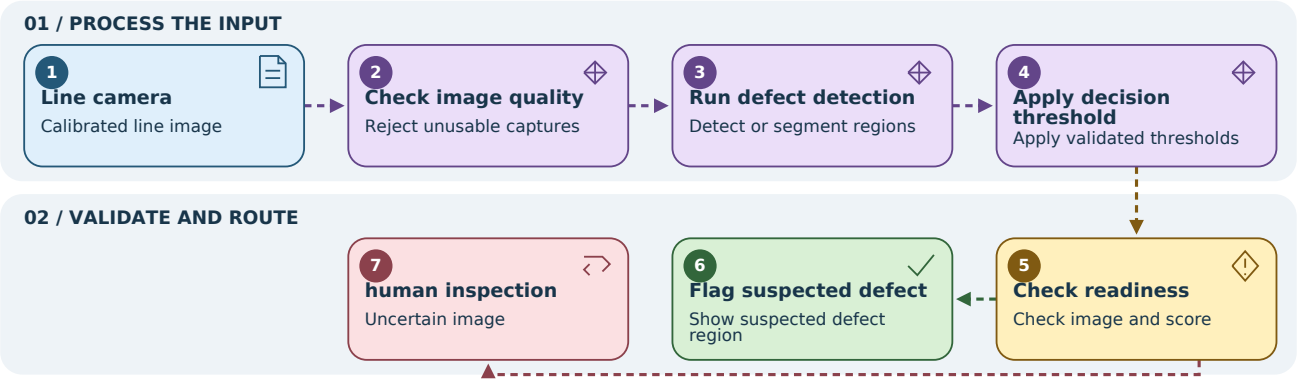
Technique choice

Generate narrative around verified facts. Avoid learning changing metric values in the model weights.

CASE 16 · COMPUTER VISION + REVIEWER QUEUE

Product defect inspection

A manufacturer needs to flag visible defects on a production line.



Prepare the data

Standardize camera placement and lighting. Collect images across plants, batches and defect types. Label defect regions and reserve unseen batches and time periods for test.

Design the workflow

Run detection or segmentation at the edge where latency matters. Send uncertain items for inspection. Log model version, image quality and factory conditions.

Evaluate before rollout

Measure missed defects, false rejects and inference time by line and defect class. Monitor changes in lighting and packaging.

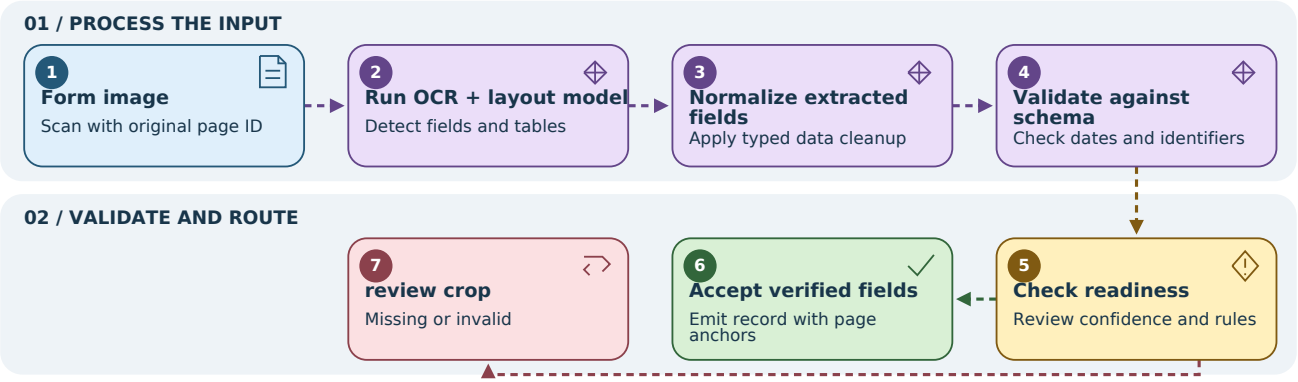
Technique choice

A supervised vision model fits the core task. A multimodal LLM can help draft reviewer notes when useful.

CASE 17 · OCR + LAYOUT EXTRACTION

Digital forms and scanned archives

Operations needs to convert mixed paper forms and PDFs into validated records.



Prepare the data

Inventory form variants and languages. Retain originals and page anchors, and collect manually verified field labels. Identify restricted fields and retention rules.

Design the workflow

Use OCR and layout models to extract fields and tables. Validate dates, totals and identifiers. Route missing or low-confidence fields with image crops for review.

Evaluate before rollout

Track exact field match, document-level completion and manual corrections by form variant. Include skewed scans and handwritten edits.

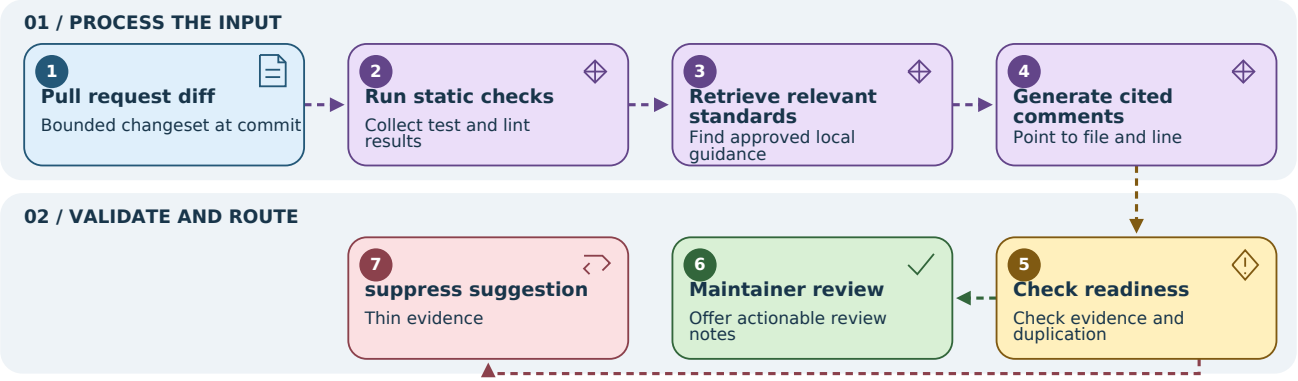
Technique choice

Start with OCR and validation. Consider targeted fine-tuning only for recurring extraction errors with reliable labels.

CASE 18 · REPOSITORY RAG + STATIC ANALYSIS

Developer pull-request review assistant

A team wants faster review of changes against its own engineering standards.



Prepare the data

Index approved standards and relevant repository context by commit. Run existing linters, tests and static analyzers. Define a schema for proposed comments and evidence.

Design the workflow

Supply a bounded diff, eligible context and existing check results. Suggest review questions and cite code locations. Keep review decisions with maintainers.

Evaluate before rollout

Measure actionable comment precision, repeats of existing checks and reviewer time. Include large diffs and outdated guidance.

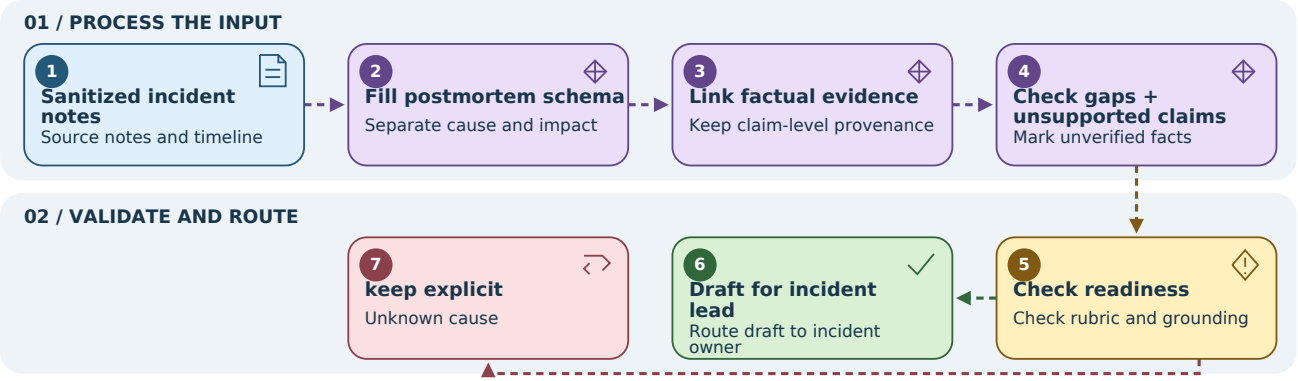
Technique choice

Use deterministic analysis for proven rule checks. Use generation to explain and prioritize review concerns.

CASE 19 · TEMPLATES + OPTIONAL LORA

Standardizing incident postmortems

Incident leads write inconsistent drafts using the same review rubric.



Prepare the data

Define fields for impact, timeline, evidence, confirmed cause and owner-approved actions. Curate approved drafts paired with sanitized incident notes. Split by incident family and date.

Design the workflow

Generate a draft from cited notes into a schema. Require evidence for factual claims and leave unknowns explicit. Compare a prompt baseline with LoRA if rubric or house-style failures persist.

Evaluate before rollout

Score accuracy, evidence support, completeness and reviewer edit time. Test that the model never fills gaps with invented causes.

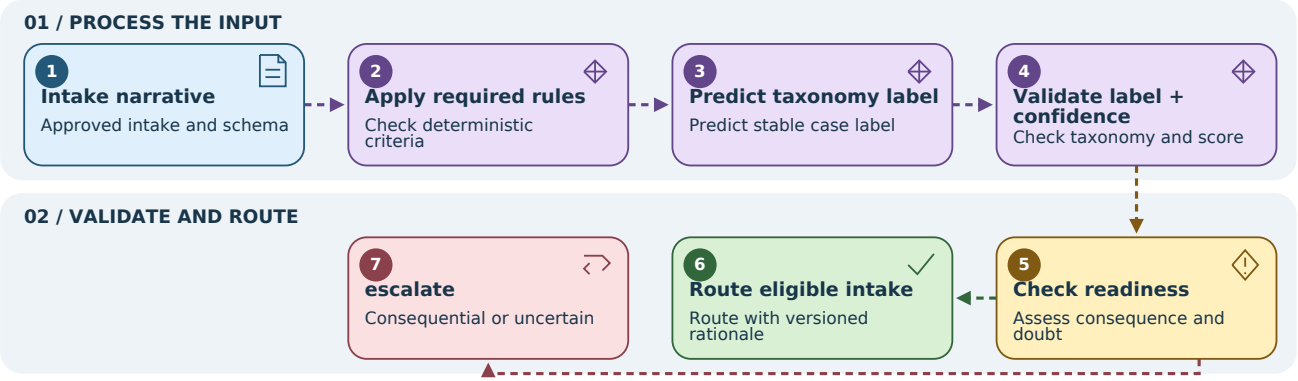
Technique choice

A stable writing rubric is a candidate for adaptation. Use QLoRA when base-weight memory is the limiting factor and the deployed artifact preserves quality.

CASE 20 · RULES + SUPERVISED ADAPTATION

Regulated case intake and categorization

A review team must map incoming narratives to a stable case taxonomy with clear handoffs.



Prepare the data

Create a label guide with reviewed examples, disputed labels, rare classes and unknown cases. Separate identities from authorized training data where possible and keep access and provenance records.

Design the workflow

Apply deterministic intake requirements and compare a conventional classifier, few-shot LLM and adapter. Validate output IDs against the taxonomy and escalate uncertain or consequential cases.

Evaluate before rollout

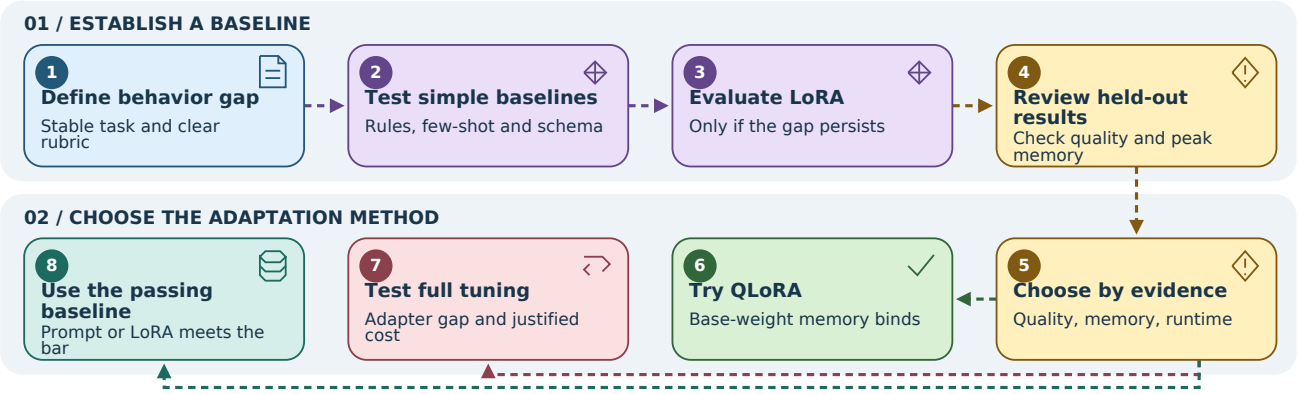
Measure macro-F1, critical class recall, abstention, and human review burden. Test with a newer intake period and inspect uneven performance across relevant groups.

Technique choice

Fine-tune if there is measurable benefit on the stable taxonomy. Prefer LoRA first when the platform supports adapters. Evaluate QLoRA if memory materially limits training.

DECISION GUIDE

When to fine-tune



Fine-tuning is the umbrella term. Full fine-tuning updates the base weights. LoRA and QLoRA are parameter-efficient ways to fine-tune a model.

Prompting, retrieval, and structured output first

Use prompt examples for behavior you can demonstrate in a few examples. Use RAG for private, changing, permissioned or attributable knowledge. Use a schema for output shape and application code for business rules. A live API is the better source for exact current state.

LoRA for stable behavior

LoRA freezes the base and adds trainable low-rank updates. It fits repeated classification, extraction conventions and house style when you have quality labels and a measurable prompt-baseline gap. Version the adapter with the compatible base revision, tokenizer, prompt template and evaluation results. Verify runtime support for loading or merging it.

QLoRA for a training memory constraint

QLoRA trains LoRA adapters with a quantized frozen base. The original method uses 4-bit NF4, double quantization and paged optimizers. Consider it when base-weight memory makes conventional LoRA expensive or impractical. Benchmark peak memory, runtime and task quality, then validate the actual serving artifact after any merge or conversion.

Full fine-tuning for a demonstrated gap

Evaluate full fine-tuning when strong prompt and adapter baselines miss a justified target, you have representative data and compute, and the required customization or serving runtime favors an updated checkpoint. Require evidence of benefit against its additional training, evaluation and operational cost.

Compare the techniques

1. Choose a narrow behavior, such as assigning incident categories from a stable rubric. Keep service ownership in a live registry.

2. Group near duplicates, split by incident and time, and reserve an untouched test set with rare classes and unknown cases.

3. Compare rules or a conventional classifier, a few-shot prompt, LoRA, and QLoRA if memory is a constraint. Keep test inputs and label definitions fixed.

4. Review macro-F1, critical-class recall, abstention mistakes, cost per successful task, latency and reviewer workload. Release only if the retained serving artifact clears the agreed gates.

Primary references

Amazon Bedrock retrieval documentation

LoRA paper

QLoRA paper