CONNECT THE WHOLE ML SERVICE

DATA

FEATURES

TRAINING

RELEASE

SERVING

MONITORING

INTEGRATE THE ML SERVING DESIGN

A model file is one component of a useful service.

Today: connect data, features, training, release, serving, and monitoring.

Every boundary needs a contract and a failure route.

START WITH THE DECISION

Illustrative project:
prepare a daily review queue for customers at risk of purchase inactivity.

Target: no completed purchase in the next 30 days.

30 days

Staff review the queue; the score does not trigger an automatic action.

Prediction quality is not proof that outreach helps.

WRITE ACCEPTANCE CRITERIA FIRST

QUALITY

Choose quality measures on mature labels.

OPERATIONS

Define the eligible population and daily review capacity.

Set freshness, availability, access, and operating-cost limits.

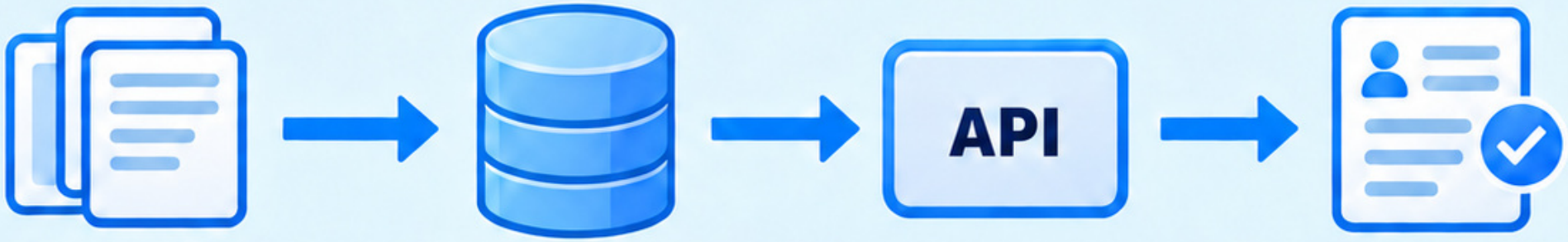
FAILURE BEHAVIOR

Add failure cases with expected behavior.

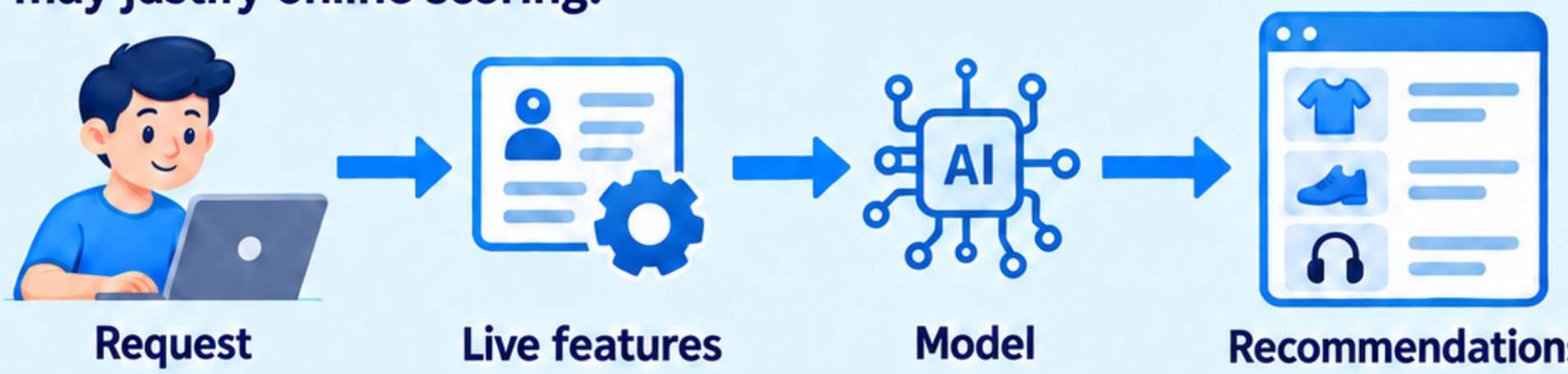
A release passes only the criteria you actually evaluate.

CHOOSE THE SERVING MODE

For a once-daily review queue, nightly batch scoring can be enough.



For recommendations on a live product page, request-time context may justify online scoring.



Choose from decision timing and freshness needs.

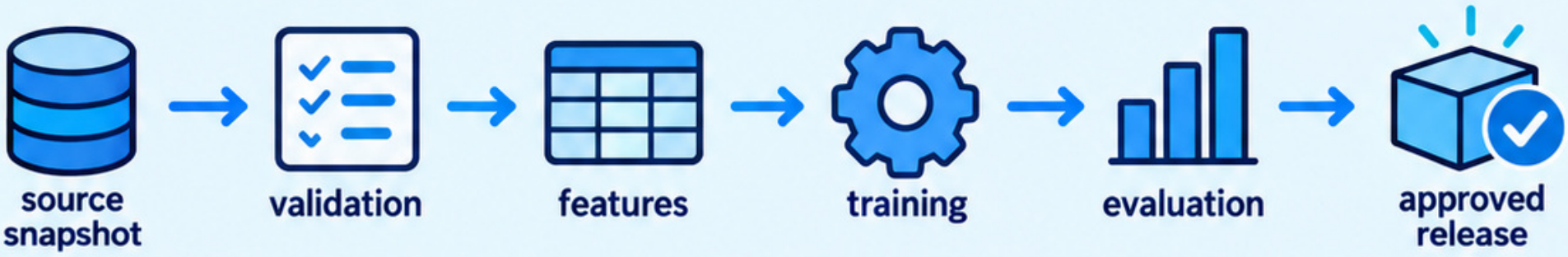
An API can serve stored batch scores without running a model per request.

@learn.machinelearning

DRAW PREPARATION AND SERVING PATHS

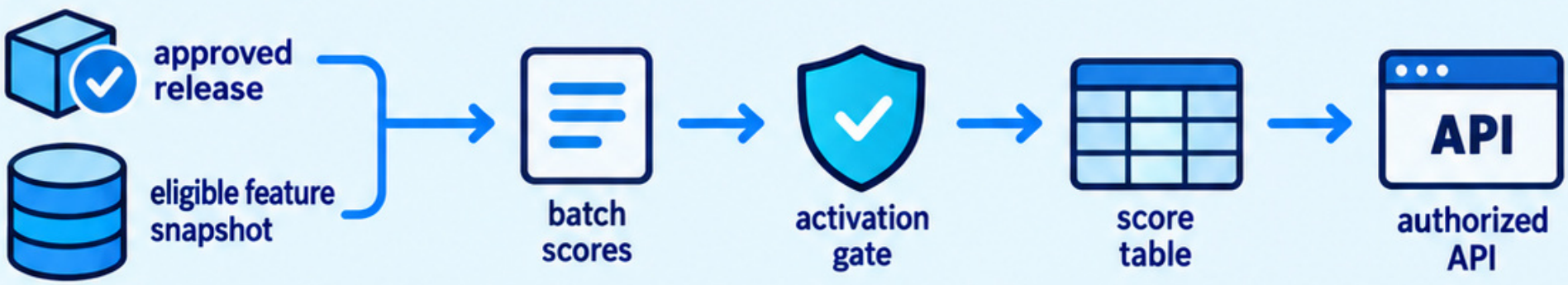
PREPARATION:

source snapshot → validation → features → training → evaluation → approved release.



SERVING:

approved release + eligible feature snapshot → batch scores → activation gate → score table → authorized API.



MONITORING observes both paths and joins later labels.

KEEP THE DATA CONTRACT

Define entity IDs, event meaning, units, timestamps, completeness, and duplicate rules.

Features use only information available at prediction time.

Labels use their declared future window and maturity rule.

Quarantine invalid records and record why they were excluded.

FEATURES (PAST)

Past data
(used for features)

PREDICTION TIME

Make prediction here

FUTURE (LABEL)

Label comes from its declared future window and maturity rule.

Invalid records

VALIDATION & QUARANTINE

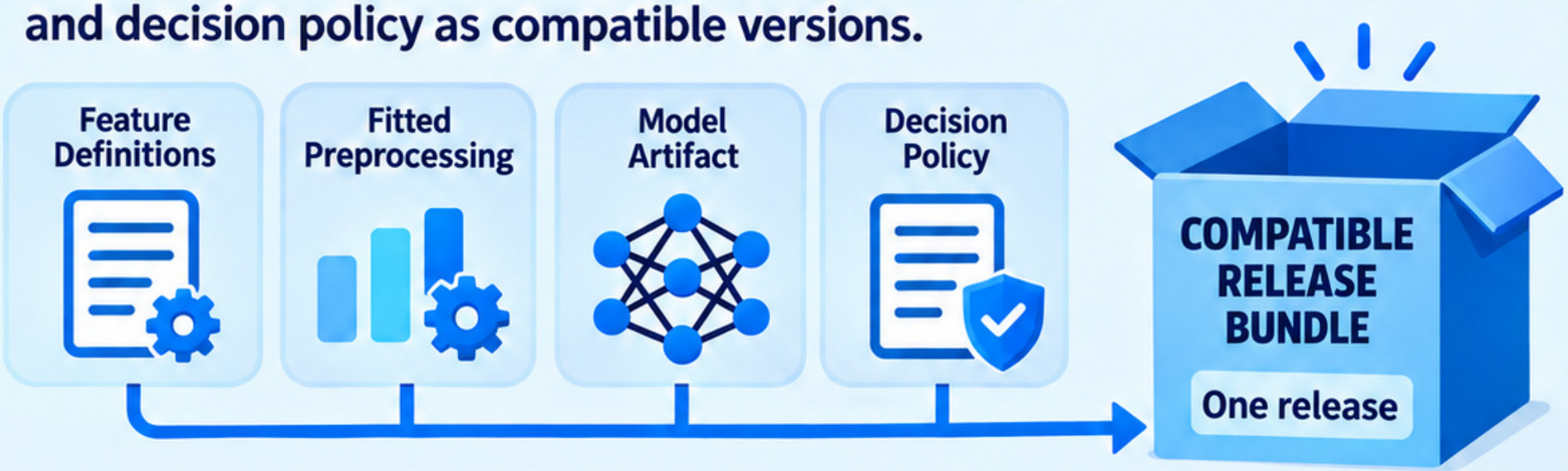
Quarantine invalid records and record why they were excluded.

AI SYSTEM DESIGN • DAY 18

7/18

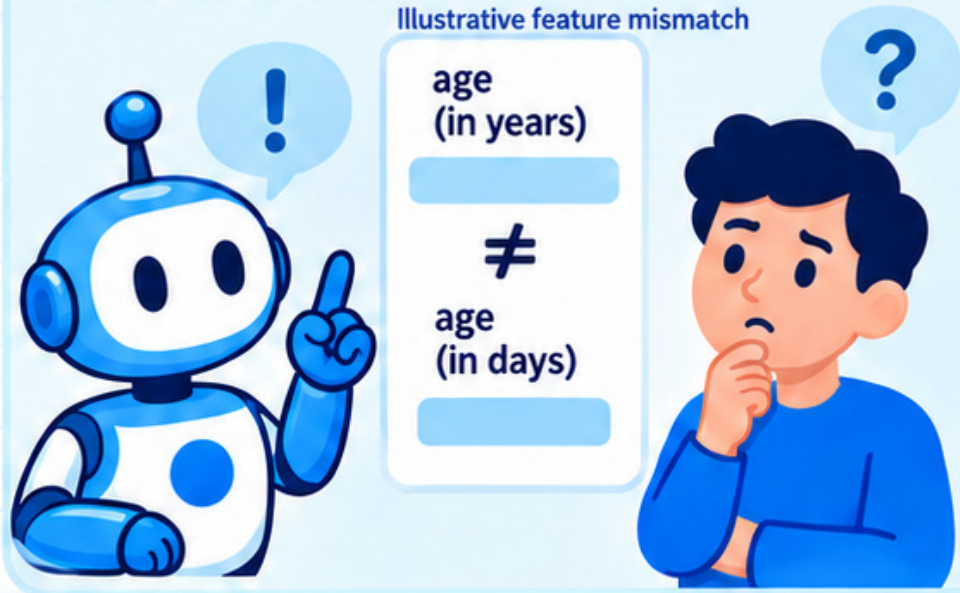
SAVE THE PIPELINE TOGETHER

Package feature definitions, fitted preprocessing, model artifact, and decision policy as compatible versions.



Record the training data snapshot, split manifest, code version, dependencies, and random settings.

Reusing a model with a different feature meaning can change its behavior.



@learn.machinelearning

AI SYSTEM DESIGN • DAY 18

8/18

USE TOOLS BY RESPONSIBILITY

scikit-learn:
fitted preprocessing and model execution.

MLflow:
run metadata and artifacts.

PostgreSQL:
manifests, score records, and application data.

FastAPI:
request validation and application endpoints.

Evidently:
drift and quality reports.

APPLICATION ORCHESTRATOR

Your code connects the contracts and enforces policy.

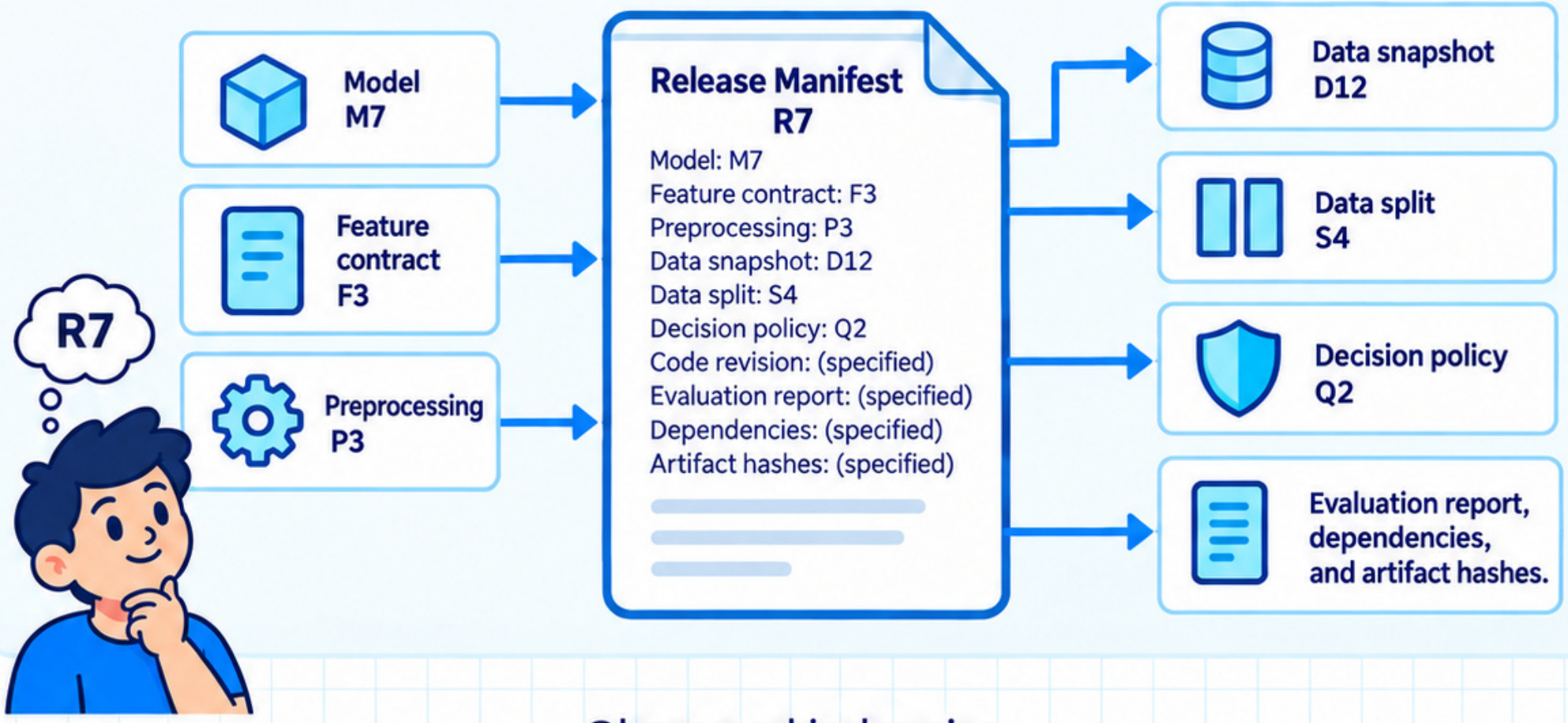
@learn.machinelearning

MAKE A RELEASE MANIFEST

Illustrative release R7 records:

Model M7, feature contract F3, preprocessing P3.
Data snapshot D12 and split S4.
Decision policy Q2 and code revision.
Evaluation report, dependencies, and artifact hashes.
A name alone cannot reproduce a release.

ONE MANIFEST LINKS THE ARTIFACTS



ACTIVATE COMPLETE BATCHES

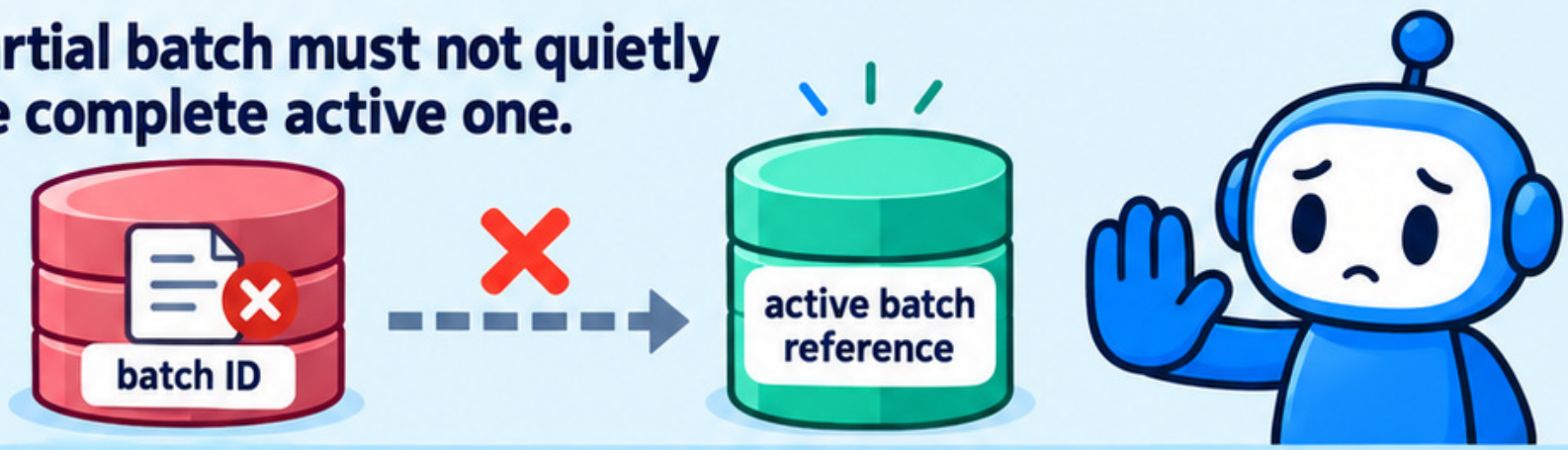
Write candidate scores under a batch ID.

Check partitions, row counts, keys, versions, and expiry.

Activate the batch only after required checks pass.

Readers use the active batch reference.

A failed partial batch must not quietly replace the complete active one.



DESIGN THE SCORE RECORD

Illustrative record fields:
Entity ID, prediction time, model release, batch ID, score, and expires_at.

Entity ID	prediction time	model release	batch ID	score	expires_at

Define a uniqueness key for replayed writes.

A score must carry its freshness and version context.

Do not return an expired score as if it were current.

AI SYSTEM DESIGN • DAY 18

12/18

KEEP ACCESS IN THE APPLICATION

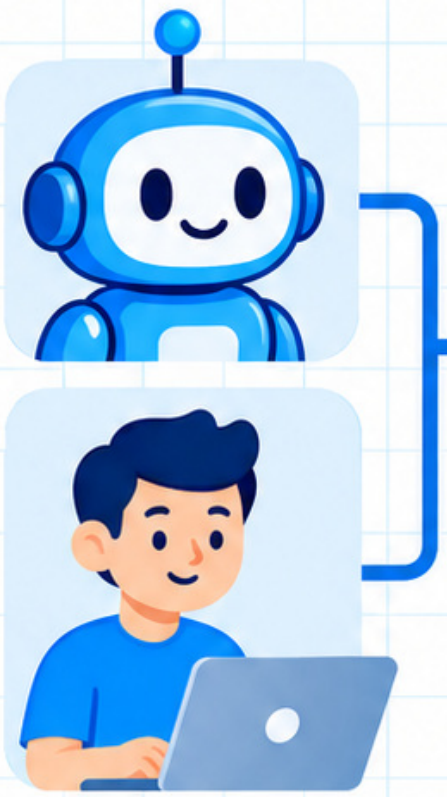
The API resolves authenticated identity.

Application and database rules select allowed records.

The response includes score age and release context when useful.

An LLM or agent may request a score through this boundary.

Model-generated arguments do not grant access.



An LLM or agent may request a score through this boundary.

The API resolves authenticated identity.

Application and database rules select allowed records.

Score read

The response includes score age and release context when useful.

No protected data.

Model-generated arguments do not grant access.

@learn.machinelearning

RUN A FAILURE DRILL

Illustrative faults:

A partition fails during scoring.

Observable response:

Responsible owner:

A stored score expires.

Observable response:

Responsible owner:

The requested customer is outside the caller's access.

Observable response:

Responsible owner:

An artifact has an unexpected hash.

Observable response:

Responsible owner:

For each fault, specify the observable response and the responsible owner.

MAKE THE FAILURE OUTCOMES EXPLICIT

Failed partition → keep candidate inactive.

Expired score → declared unavailable or fallback response.

Unauthorized record → return no protected fields.

Unexpected artifact hash → stop loading that artifact.

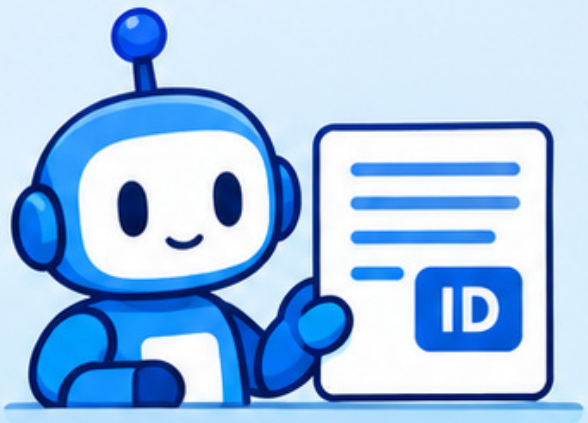
A fallback must obey its own freshness and access contract.

AI SYSTEM DESIGN • DAY 18

15/18

CLOSE THE FEEDBACK LOOP

Record predictions and decisions with stable IDs.



Join only mature, available labels for quality reports.

Monitor data health, score shifts, coverage, quality, and service behavior.

Investigate evidence before retraining.

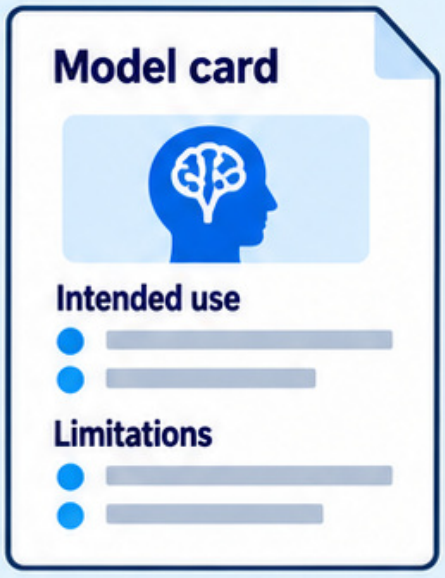
Any new candidate returns through the release gate.

@learn.machinelearning

REVIEW THE DESIGN AS EVIDENCE

Produce a diagram and data contract.

Write a model card with intended use and limitations.



Attach the release manifest and evaluation report.

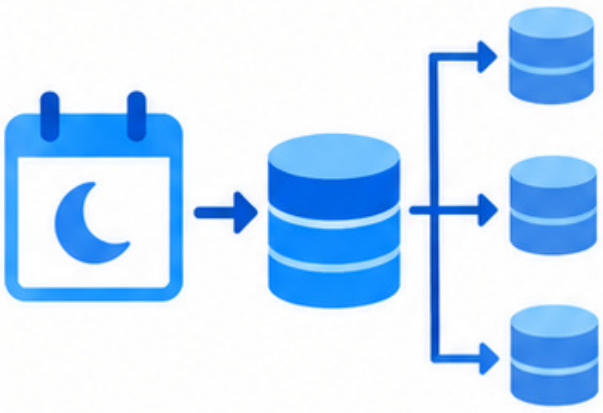
Record failure-drill results against expected behavior.

Mark unmeasured criteria as unmeasured; do not invent a passing score.

YOUR TURN: REVIEW THIS PROPOSAL

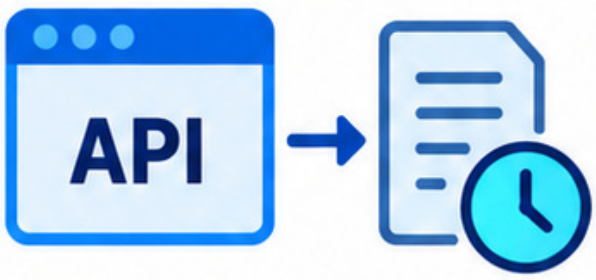
Illustrative proposal:

A nightly job overwrites live scores partition by partition.



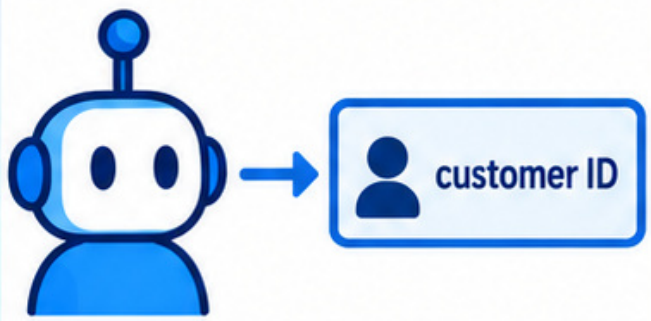
Illustrative proposal:

The API ignores score expiry.



Illustrative proposal:

An agent supplies a customer ID without an access check.

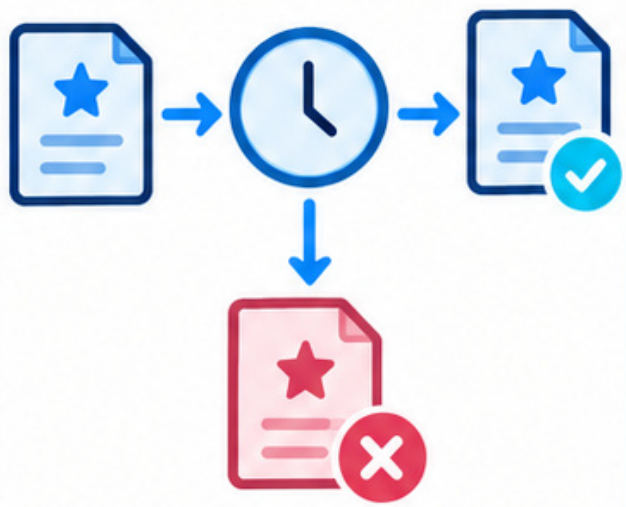


Name three design changes and one verification case for each.

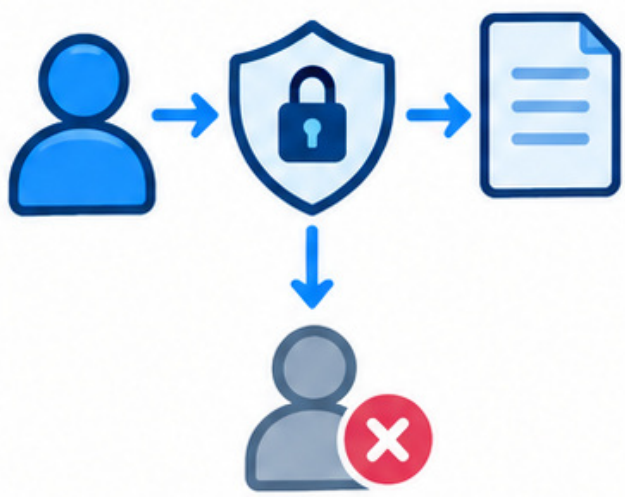
A SERVICE IS A SET OF CONTRACTS

Stage and validate a batch before activation; test partial failure.

Enforce score expiry; test an expired record.



Authorize each read; test a denied customer.



Connect quality, operations, and recovery to the same release.

Next: prompts as versioned interfaces.