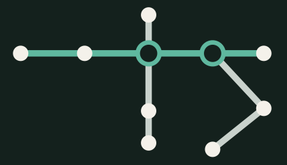
CASE STUDY · PART A — CLASSIC ML

Forecasting Munich subway ridership

Hourly passengers per station, a week ahead — so operations can put resources where demand will be.

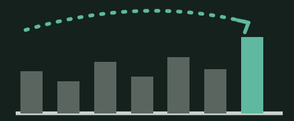
Daniel Fridljand · danielfridljand.de

23 June 2026



10 stations

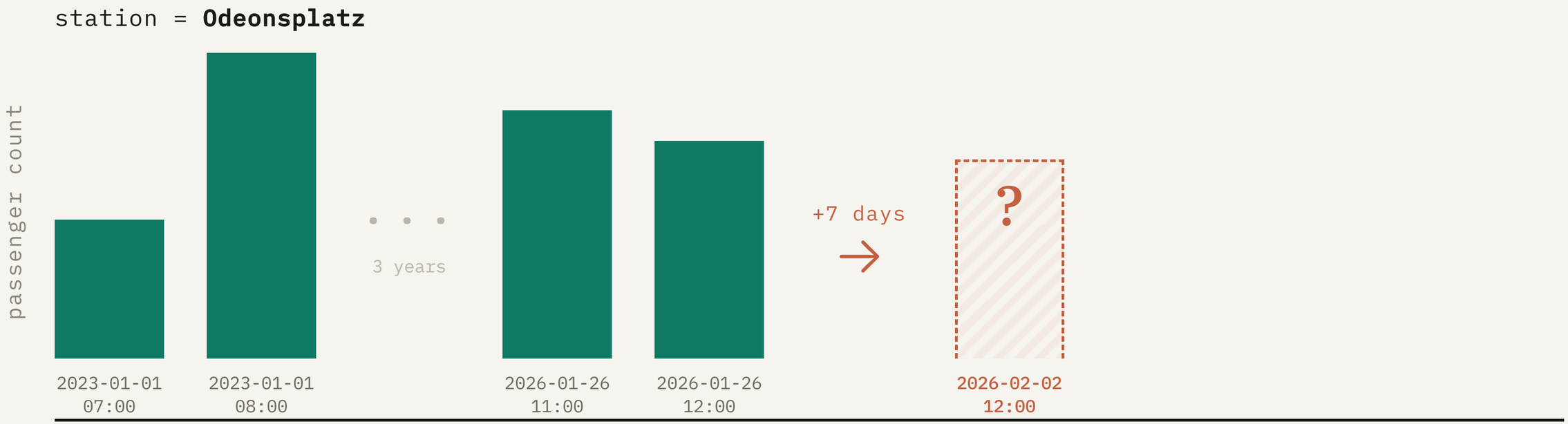
7am – 12pm window



7 days ahead

PROBLEM FRAMING

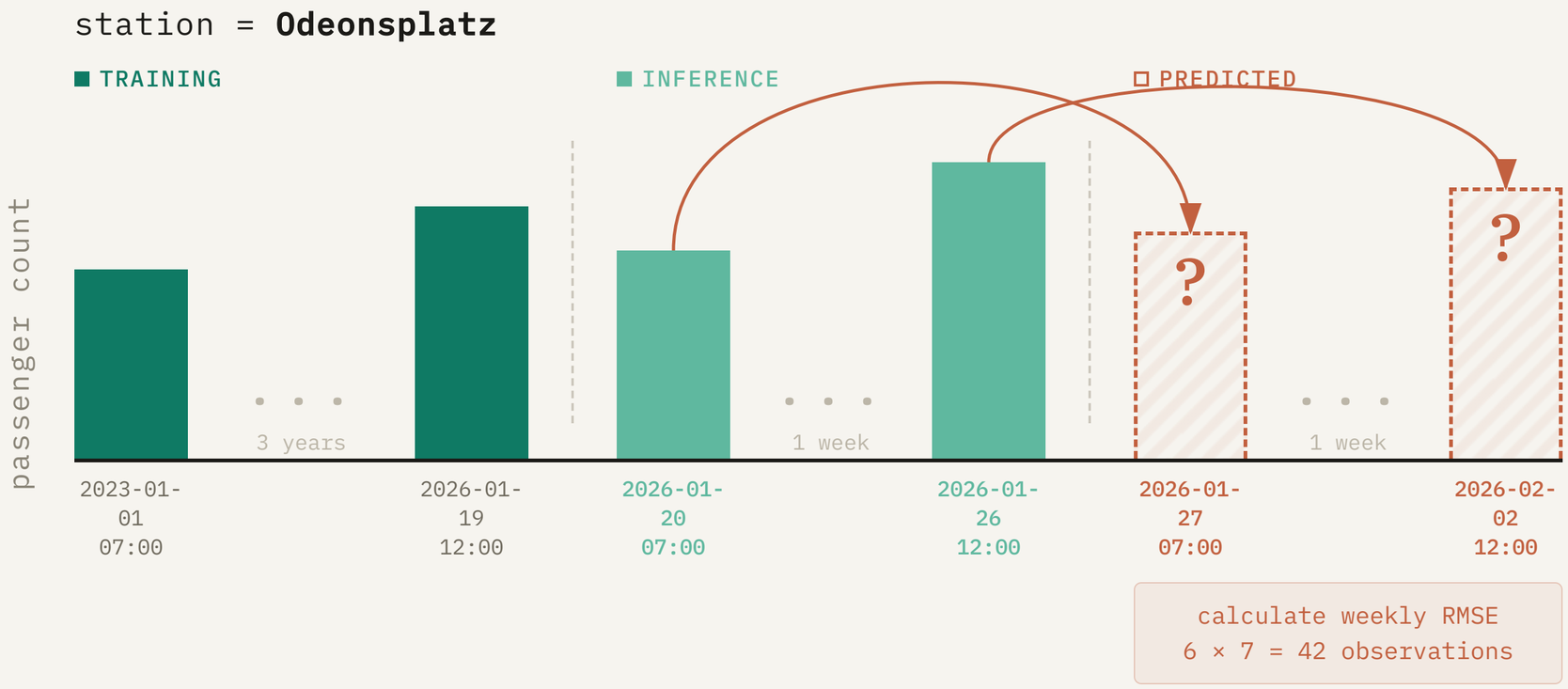
Forecast next week's demand



For **every station**, **every operating hour** (7am–12pm), predict the passenger count **7 days ahead** — so staff and resources are in place before demand arrives.

PROBLEM FRAMING

Three windows: train · infer · evaluate



MODEL CHOICE

One model: gradient-boosted trees

THE CHOICE

LightGBM

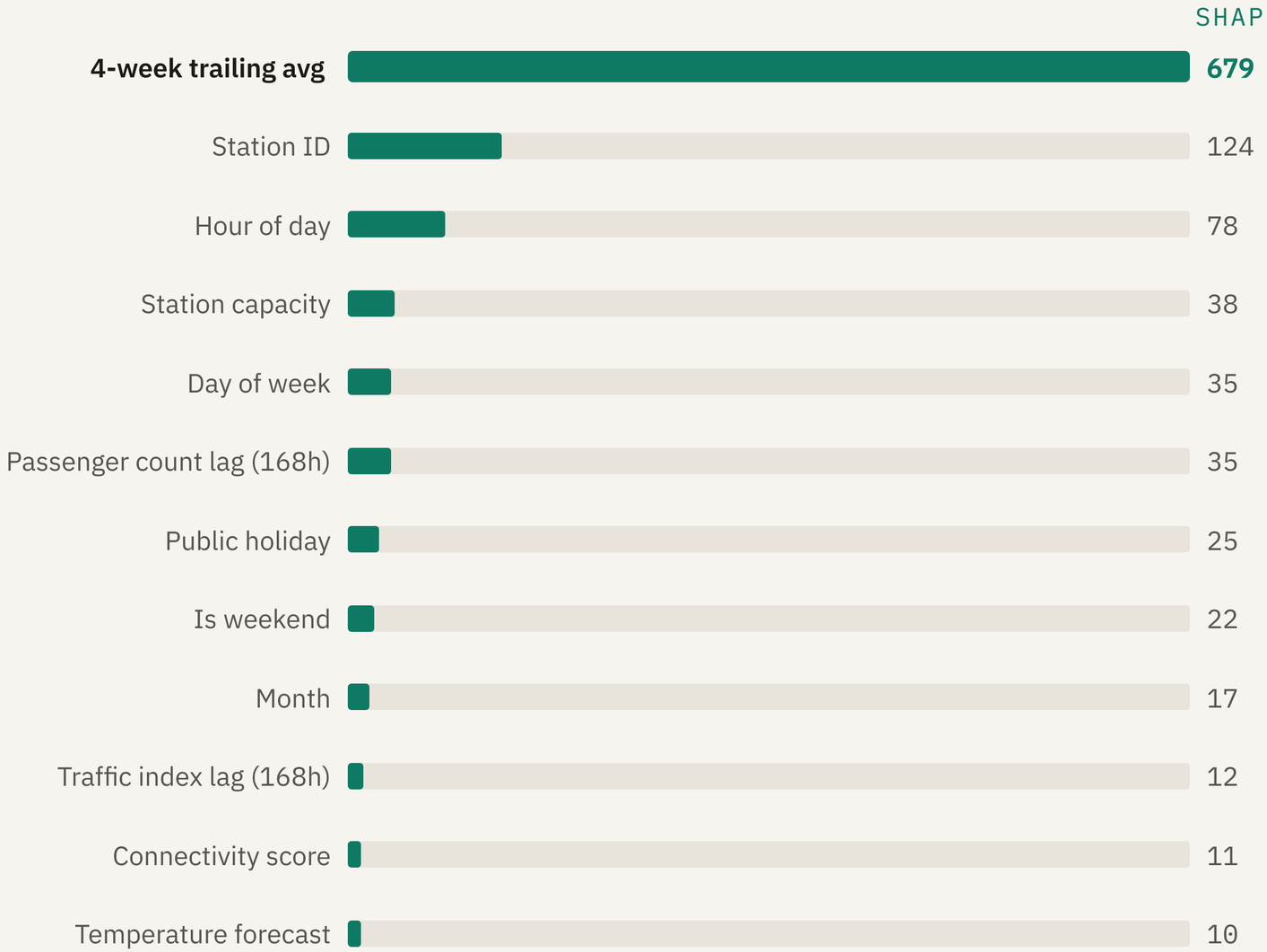
One gradient-boosted model across all stations. Industry-standard for tabular data.

`mixed features`	Handles categories, counts and continuous signals gracefully.
`interactions`	Learns effects like capacity × hour without hand-engineering.
`efficient`	Sample-efficient at this scale; trains in seconds.
`explainable`	Feature importance comes for free.

Why not the alternatives: linear is too rigid · classical time-series means 10 separate models · deep learning is data-hungry and harder to explain.

FEATURE CHOICE

What drives the model



SHAP

Measures how much each feature moves the prediction — averaged over all samples, in passenger units.

#1 4-WEEK TRAILING AVG

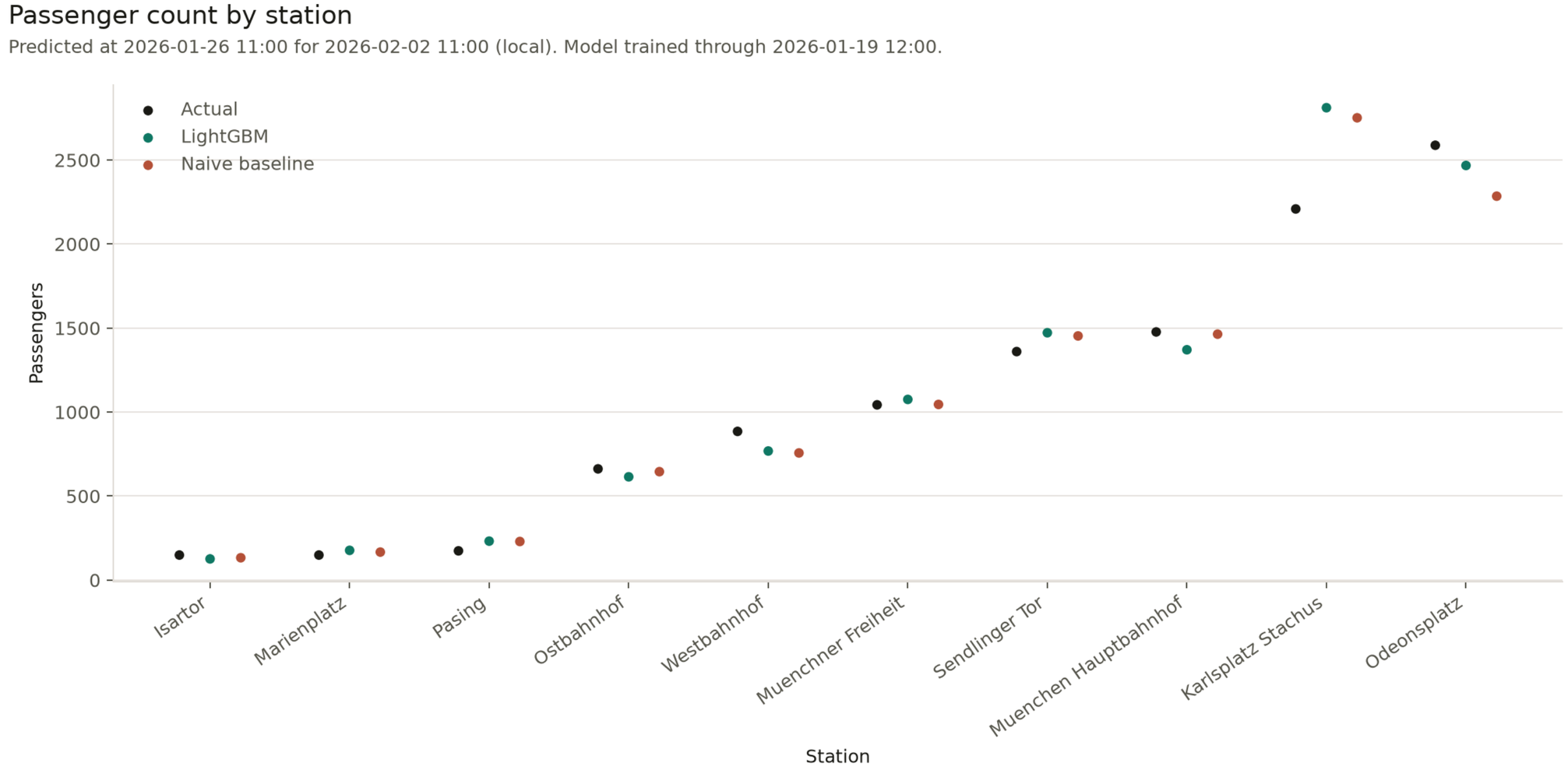
Same station, same hour, same weekday, averaged over the last 4 weeks. Compresses recent seasonal pattern.

#2 STATION

Captures per-station base rates. Persistent, station-specific structural demand differences

EVALUATION

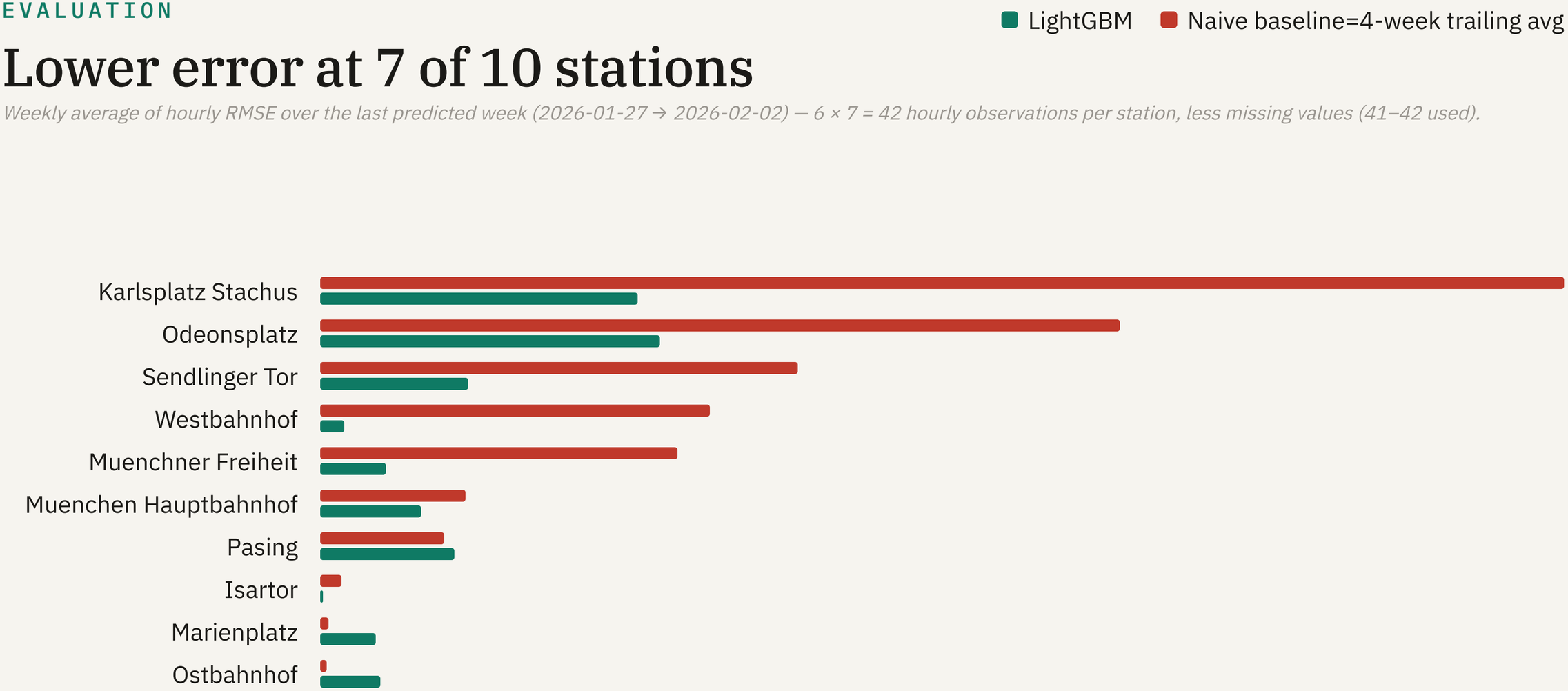
The forecast tracks actual demand



EVALUATION

Lower error at 7 of 10 stations

Weekly average of hourly RMSE over the last predicted week (2026-01-27 → 2026-02-02) — 6 × 7 = 42 hourly observations per station, less missing values (41–42 used).



LightGBM performs better at 7 of 10 stations. Mean overall RMSE is **15% lower**.

PRODUCTION

Three jobs in production

Inference

6× / day

Data Lake · APIs → LightGBM → predictions

Week-ahead forecast per station. Each run predicts exactly **T + 168h**, using the latest available features available at run time.

station_id	predicted_for	predicted_at	**passenger_count**
MUC001	2026-01-27 07:00	2026-01-20 07:00	**842**

predicted_for = predicted_at + 168h — always

Monitoring

daily

predictions · actuals → RMSE / station → ⚠ alert

Scores per station against its own baseline — overall RMSE is volume-weighted and hides drift at smaller stations.

station_id	rmse	**baseline_rmse**
Hauptbahnhof	262	**535**

⚠ alert if rmse > baseline at any station

Retrain

weekly

full history → model v_n → promote if better

Mon 06:00 — one hour before first inference. New model promoted only if it beats the current champion.

MLflow registry

ridership_lgbm / v8

RMSE 277.7 vs champion 281.2 → **promoted**

if new RMSE > champion: keep current, skip promotion

IN SUMMARY

One model, a week ahead

ACCURATE

15% lower overall RMSE than the naive baseline — lower at 7 of 10 stations.

RIGHT-SIZED

One LightGBM model across all stations; three lean jobs — inference (6×/day), monitoring (daily), retrain (weekly).

TRUSTWORTHY

Per-station error alerts, beat-baseline thresholds, and full model lineage.

Operations gets a reliable week-ahead demand signal, station by station.

That's Part A.

Next — Part B · Agentic & GenAI

CASE STUDY · PART B — AGENTIC & GENAI

From inbox to ERP order

An agentic, auditable workflow that turns unstructured order emails into ERP orders — safe, controllable, and built for human oversight.

Daniel Fridljand · danielfridljand.de

23 June 2026

email + attachments
↓
extract
↓
triage
↓
agents
↓
review
↓
ERP order ✓

THE PROBLEM TODAY

A human keys every order into the ERP by hand

A skilled operator reads each email, identifies the customer, the product and the quantity, and types it into the ERP. Accurate — but it doesn't scale.

SLOW

Reading and re-keying from scratch keeps orders sitting in the inbox.

DOESN'T SCALE

Throughput is capped by headcount; peaks create overnight backlog.

EXPENSIVE

Specialist expertise is tied up on repetitive data entry.

> When volume spikes — a seasonal peak, an acquisition — the only lever is hiring more people. That's the bottleneck we remove.

THE SOLUTION, IN ONE LINE

The agent fills in the order form. A human submits it.

UNSTRUCTURED EMAIL

From: orders@gmail.de
Subj: Re: restock

Hi — can you get us 500 of the
black conveyor belt, 500 series?
Need them for the Munich line.
Thanks, Pavel

→

ERP ORDER DRAFT

customer_id	CUST-001
product_id	CB-500
quantity	500

READY FOR REVIEW

B.A

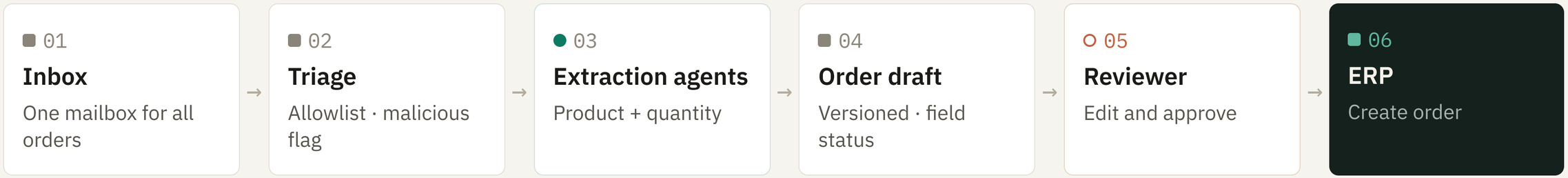
Solution Design

architecture · three fields · security · HITL · auditability · key metrics

B.A · SOLUTION DESIGN

Overall architecture

Deterministic · LLM · Human



B.A · SOLUTION DESIGN

Three fields, three mechanisms

Deterministic · LLM · Human

01 **Inbox**
One mailbox for all orders

→

02 **Triage**
Allowlist · malicious flag

→

03 **Extraction agents**
Product + quantity

→

04 **Order draft**
Versioned · field status

→

05 **Reviewer**
Edit and approve

→

06 **ERP**
Create order

`customer_id`
Sender email → customer lookup.

`quantity`

`product_id`

ZOOM IN · PRODUCT_ID

LLM
Proposes a product from the text

→

READ-ONLY TOOL
`fuzzy_match_product()`
against the ERP catalog

→

VALIDATOR
product_id ∈ catalog? else **ModelRetry**

→

RESULT
CB-500 · or null

null is an acceptable outcome — left empty for the reviewer, never guessed.

B.A · SOLUTION DESIGN

Separation of responsibilities

■ Deterministic ● LLM ○ Human

■ 01
Inbox
One mailbox for all orders

→

■ 02
Triage
Allowlist · malicious flag

- Sender allowlist
- Regex malicious-content flag

→

● 03
Extraction agents
Product + quantity

- Proposes product via catalog tool
- Proposes a quantity
- Catalog-membership check (→ also 05)

→

■ 04
Order draft
Versioned · field status

→

○ 05
Reviewer
Edit and approve

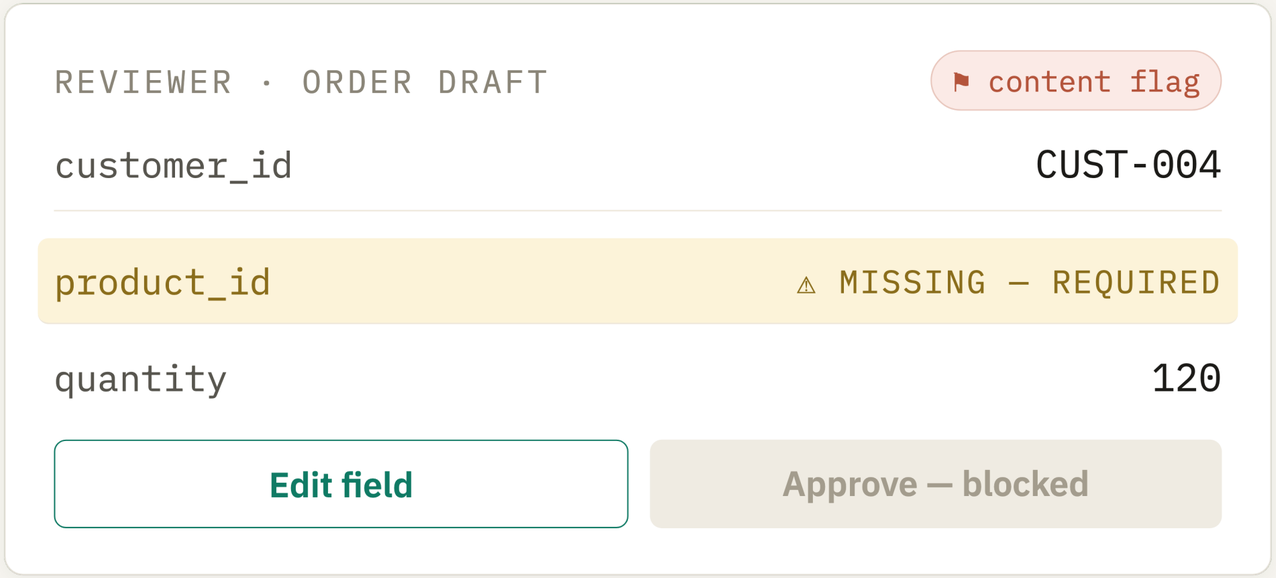
- Catalog-membership check (→ also 03)
- Completeness
- Reviews the order draft
- Edits fields
- Approves submission

→

■ 06
ERP
Create order

B.A · SOLUTION DESIGN

The AI fills the form; it cannot submit it



- Missing and flagged fields are **highlighted** for the reviewer.
- Submission is **blocked** until every required field is present.
- Every field is editable; each edit creates a **new version**.

B.A · SECURITY

The trust boundary

UNTRUSTED · TAINTED

Email body

Attachments

The LLM output

Anything that could carry an injection or a malicious instruction.

→

GUARDS

1. Sender allowlist — before the model runs
2. product_id ∈ ERP catalog — validator loop
3. Human approval — owns the decision

→

TRUSTED · THE SINK

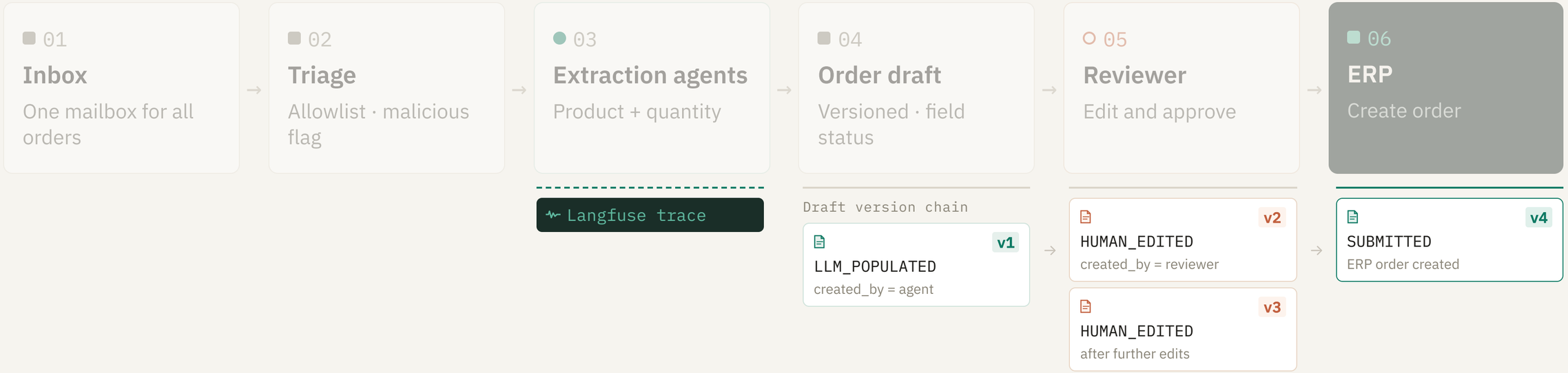
ERP create-order API

The only place an order is written. Reached only through a guard.

> Untrusted input becomes a trusted action only by passing through a guard — never because the model said so.

B.A · SOLUTION DESIGN

Auditability by design



Each field carries: `status` (ok · missing · not-in-catalog) · `populated_by` (rule · llm · human)

B.A · SOLUTION DESIGN

Key performance metrics

MOST IMPORTANT KPI

Auto-accept rate per field

	v1 agent		v4 submitted	
customer_id	CUST-004	→	CUST-004	✓ kept
product_id	— null	→	CB-500	edited
quantity	120	→	120	✓ kept

2/3 fields kept unchanged from the agent's first draft — **that is the auto-accept rate**

ALSO TRACKED

emails / hour
Throughput

missing-field rate
Draft completeness

content-flag rate
Malicious-content screening

sender rejections
A spike is a **security** signal, not just ops

review backlog
Drafts awaiting approval

time-to-approval
Order-to-cash latency

B.B

Business Case

value drivers · the KPI · why conservative = value

B.B · BUSINESS CASE

Where the value comes from

Speed

LESS TIME PER ORDER

Review a pre-filled draft instead of reading + keying from scratch

Frees expensive subject-matter-expert time for higher-value work

Faster order-to-cash and fulfilment → working capital freed, better retention

Quality

FEWER ERRORS & REWORK

Large catalog → a wrong SKU is costly: bad pricing, returns, churn

AI + human is more accurate than either alone

Error rate is monitored and improves over time

Flexibility

VOLUME ≠ HEADCOUNT

Absorb growth, seasonal peaks and M&A roll-up volume — without hiring linearly

Juniors review drafts instead of mastering the full process — wider hiring pool, faster ramp

Enables shared review teams across portfolio companies

B.B · BUSINESS CASE

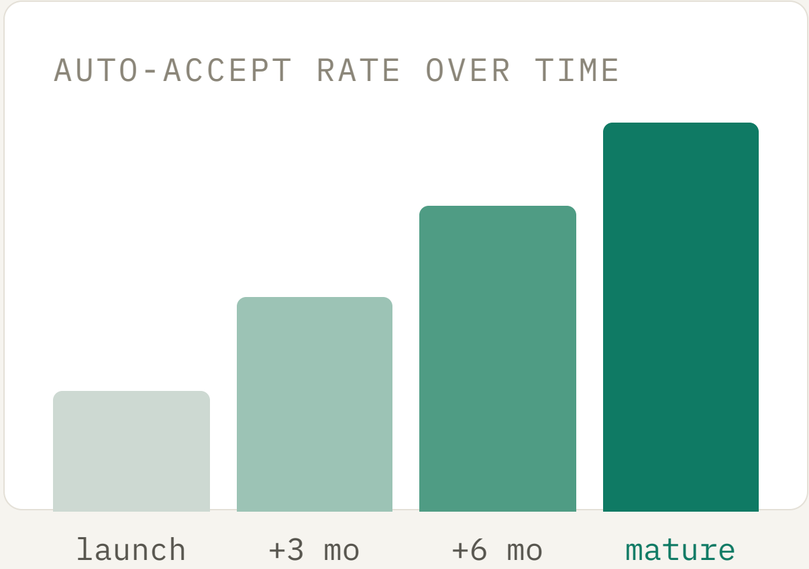
One KPI realizes the saving

> **Auto-accept rate**
> the fraction of agent-populated fields the human leaves unchanged.

It's the quality metric *and* the input to the ROI model.

Higher auto-accept → less review time → more of the saving realized.

It ramps as we monitor and improve — the saving grows, not a step change.



B.B · BUSINESS CASE

The conservative design is the value

Humans forgive human errors. They don't forgive silent software errors.

Human oversight keeps someone **accountable** — and preserves control, auditability and trust.

The worst a hostile email can do is waste a **few seconds** of a reviewer's time.

v1 doesn't need full autonomy — it needs to **cut review time** while keeping control.

IN SUMMARY

Safe, auditable, controllable — and worth more at exit

SAFE

Trust boundary + a human approves every order.

AUDITABLE

Versioned drafts, every field traced.

CONTROLLABLE

Human review gates every submission; auto-accept rate scales as trust is earned.

The notebook implements the agents, validators and audit trail end to end.

Daniel Fridljand · danielfridljand.de

Thank you — questions welcome

FOR REFERENCE

Appendix

A · model analysis & walk-forward · B · PE lens & worked example · C · technical architecture

APPENDIX · PART A

Smarter naive baseline: a closer look

three models · RMSE by station · model drivers · walk-forward stability

APPENDIX · PART A

Two models considered

BASELINE

Naive

```
ŷ = mean(y | station, hour, weekday for last 4 weeks)
```

Average of all past observations for a given station, hour of day, weekday over last weeks

KEY LIMITATIONS

Historical averages only — cannot react to weather, traffic, holidays. Sensitive to atypical weeks (e.g. holiday).

SELECTED MODEL

LightGBM

```
Gradient-boosted trees
```

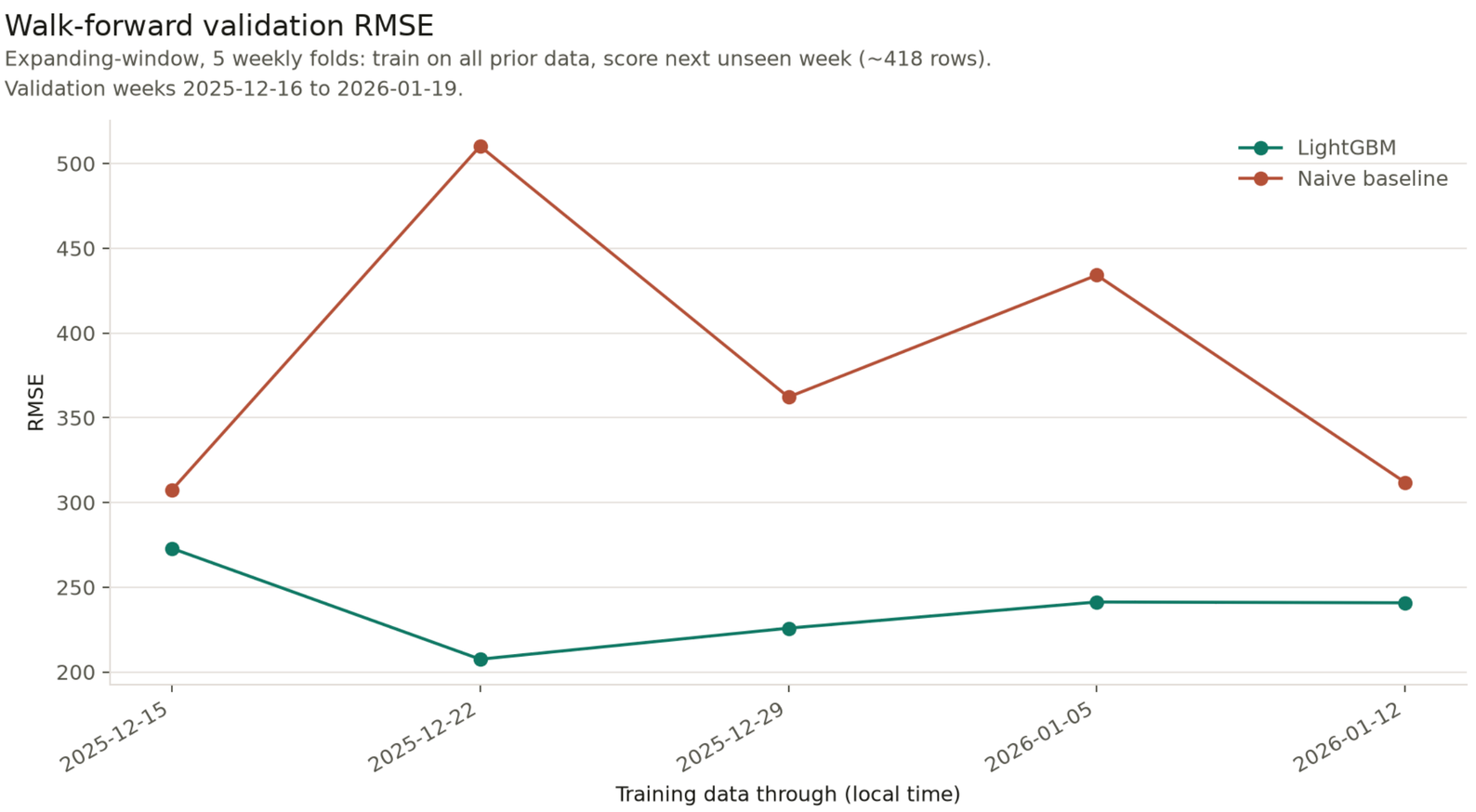
Full feature set: lagged demand, 4-week trailing avg (same station / weekday / hour), weather forecast, traffic, station metadata, holiday and Oktoberfest flags.

ADVANTAGE

Learns non-linear interactions; corrects for unusual weeks; more stable across validation folds.

APPENDIX · PART A

Why LightGBM: stability across folds



LIGHTGBM · STABLE

RMSE range: **207–278** across 5 folds. Consistently best model.

FEATURE CHOICE

The features we built

RECENT HISTORY

#1 predictive signal

same station, same hour, same weekday, averaged over the last 4 weeks

The single strongest predictor in the model.

CALENDAR

#2

hour · day-of-week · weekend · month · holiday · Oktoberfest

Carries most of the regular rhythm of demand.

STATION

#3

station id · capacity · interchange · connectivity score

Lets one model serve every station — and a new one later.

LIVE SIGNALS

#4

traffic index · weather — temperature & precipitation

Traffic: 168h lag — latest value we hold, never a future prediction.
Weather: genuine 10-day forecast, noise-calibrated in training.

APPENDIX · PART B

Illustrative exit valuation

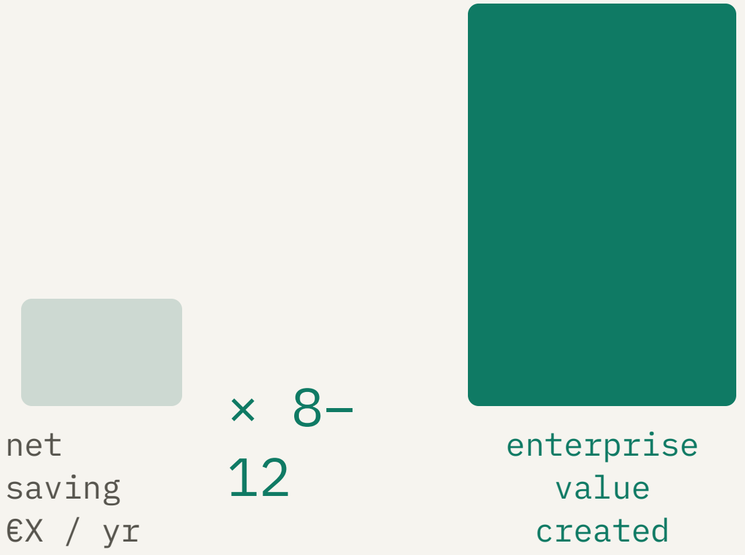
PE lens · recurring saving × exit multiple · illustrative worked example

B.B · THE PE LENS

A recurring saving is repriced at exit

In a PE-held asset, a recurring saving isn't a one-off P&L line. It lifts EBITDA — and EBITDA is **capitalized at exit**.

The saving recurs *and* gets repriced.




```
enterprise value created ≈ net annual saving × exit multiple
```

B.B · BUSINESS CASE

An illustrative worked example

ILLUSTRATIVE — PLUG IN YOUR OWN

Orders / year	60,000
Minutes saved / order	6 min
Loaded hourly cost	€60 / h
Gross – run cost	≈ €0.3M / yr net

→

NET SAVING × ~10 EXIT MULTIPLE

€3M

enterprise value created from one recurring saving

orders/yr × min saved ÷ 60 × loaded cost − run cost = net saving · × exit multiple = enterprise value

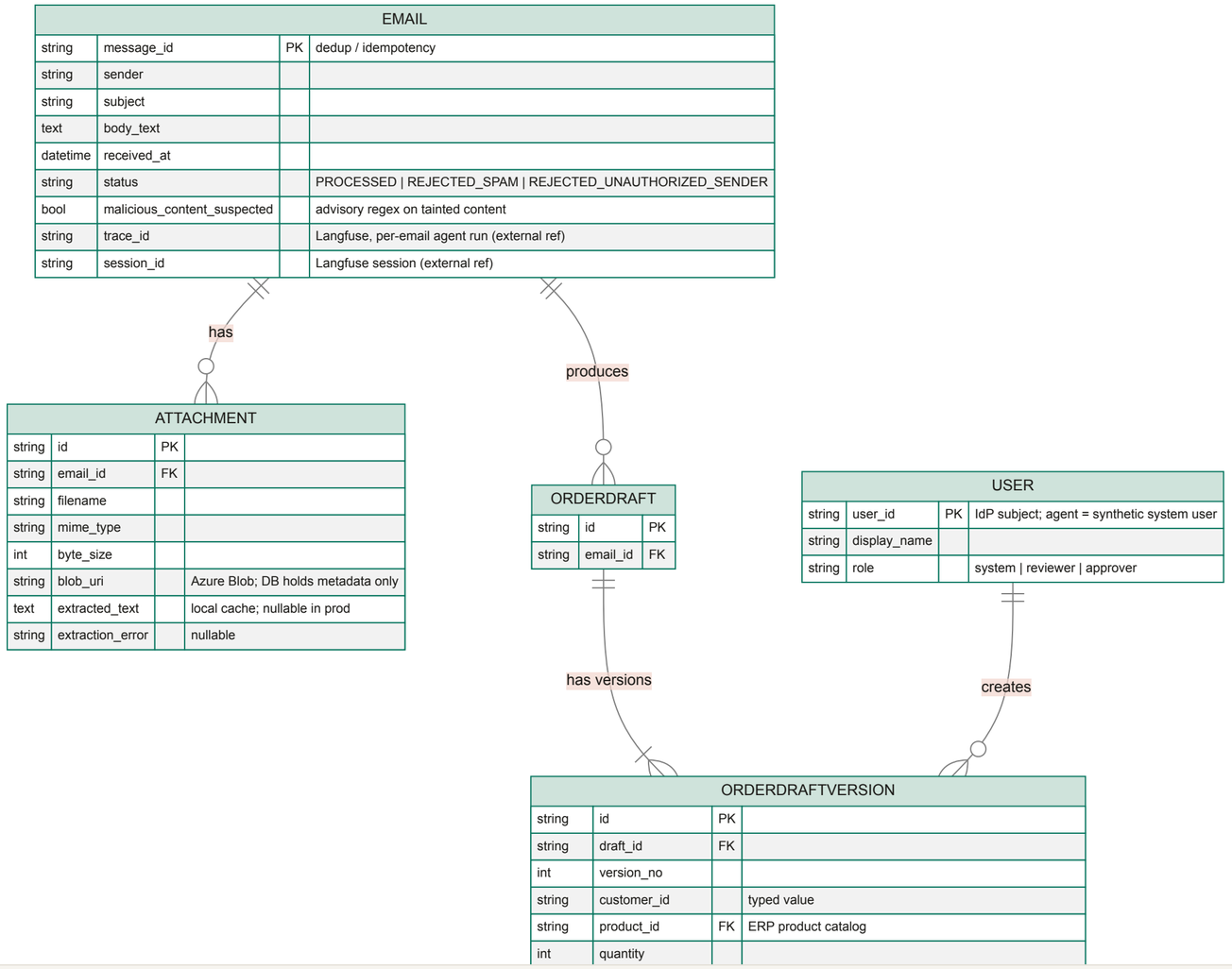
APPENDIX · PART C

Technical architecture

Challenge B · agent decision flow · database schema · deployment & monitoring

APPENDIX · PART C

Database schema



6 ENTITIES

EMAIL → ORDERDRAFT → ORDERDRAFTVERSION

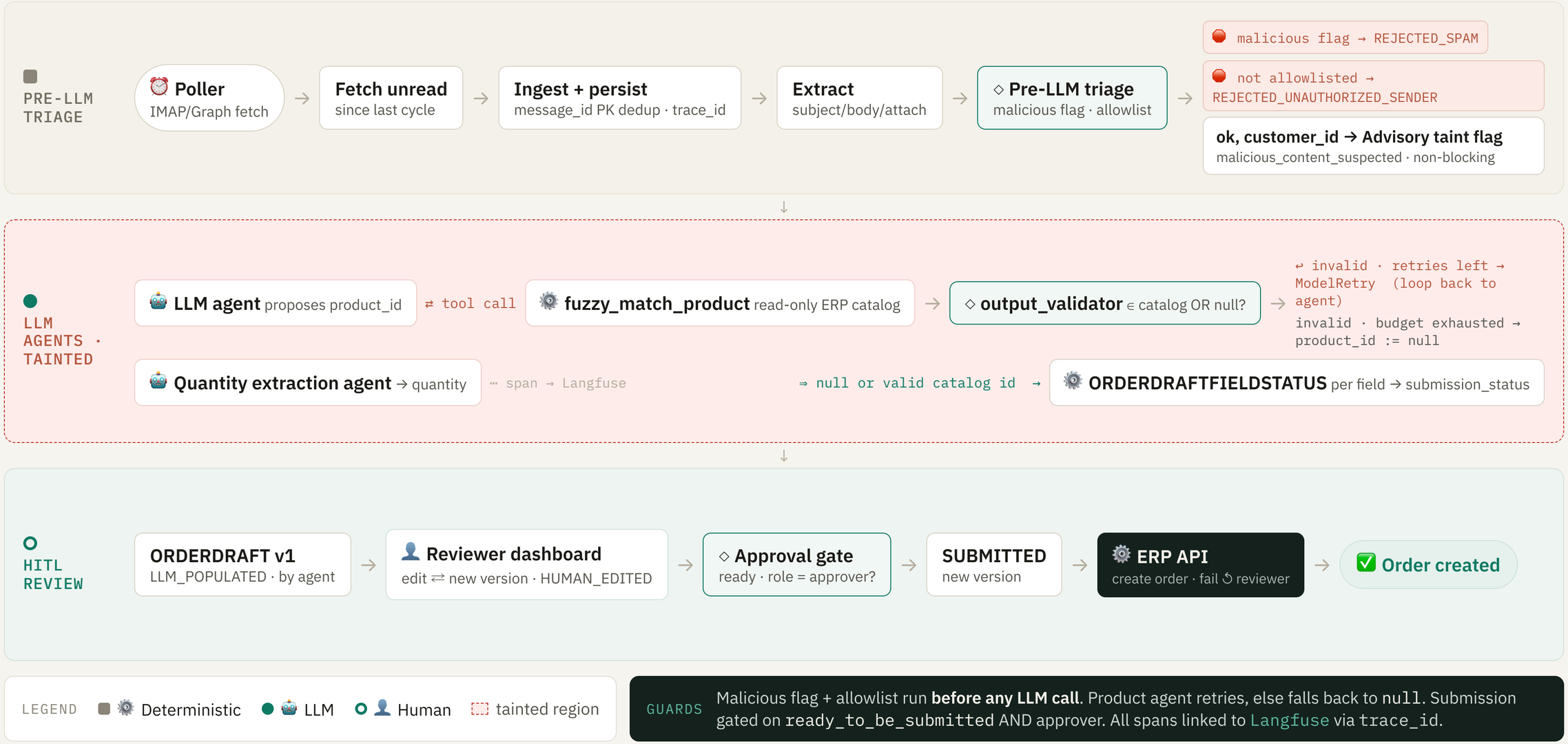
is the core write path for every processed order.

- → **ORDERDRAFTFIELDSTATUS** gives per-field audit: who set each field, how, and the Langfuse span for LLM-populated fields.
- → **ATTACHMENT** holds metadata only — binary stored in Azure Blob.
- → **USER** covers the agent (synthetic system user), reviewers, and approvers.

> Each edit creates a new `ORDERDRAFTVERSION` — the full decision trail is always preserved.

APPENDIX · PART C

Agent decision flow



LEGEND ■ Deterministic ● LLM ○ Human ⬚ tainted region

GUARDS Malicious flag + allowlist run **before any LLM call**. Product agent retries, else falls back to `null`. Submission gated on `ready_to_be_submitted` AND approver. All spans linked to Langfuse via `trace_id`.

B.A · SOLUTION DESIGN

Infrastructure & deployment

THREE DOCKER IMAGES · AZURE CONTAINER APPS

API Backend REST API

`/health · /metrics · /v1/ingest-email (internal) · /v1/order-draft/list · PATCH /{id} · POST /{id}/approve`

UI Approval UI

True API client — reviewer dashboard, field highlights, edit & approve

CRON Mailbox poller

Container App Job

IMAP / Graph API · posts to /v1/ingest-email · EMAIL.message_id deduplicates overlapping polls

PULL, NOT PUSH SPF/DKIM/malicious flag checked first · no webhook lifecycle to manage · same poller across every portfolio mail provider · 1–5 min invisible to our KPIs · idempotency free from message_id PK

AZURE DATA & SECRETS

database
PostgreSQL + SQLAlchemy ORM

attachments
Azure Blob · DB holds URI only

secrets
Key Vault · managed identity

LLM traces
Langfuse Cloud (vs. self-host)

CI / CD

1 GitHub — code & IaC
↓
2 Tekton — build & test on push
↓
3 Azure Container Registry
↓
4 Staging auto · prod manual gate

B.A · SOLUTION DESIGN

Production monitoring · three layers

LLM OBSERVABILITY

Langfuse

One trace per email (`trace_id`), one span per agent run (`observation_id`)

Per-agent: token cost · latency · `ModelRetry` count

Rising retry on product-extraction = early warning of catalog drift or model degradation — before accuracy numbers move

`ORDERDRAFTFIELDSTATUS.observation_id` links every field to the exact LLM call that produced it

SERVICE HEALTH & KPIS

Prometheus → Grafana

`/health · /metrics`

liveness / readiness · latency, error rate, queue depth

`Ops dashboard`

throughput · latency · error rate

`Business dashboard`

auto-accept trend · review backlog · time-to-approval

Two audiences, one source — ops team and business owner each get their own view

INCIDENT INVESTIGATION

Elastic

Structured per-pipeline-stage JSON shipped to Elastic

Incident query: "every event for sender X in the last 7 days"

Distinct from Langfuse (LLM-call-shaped) and Prometheus (aggregate / numeric) — the investigation and replay layer

Full replay — reconstruct every pipeline stage for any past email

ALERTING | **Ops signals:** rising latency, error rate spikes · **Business KPIs:** auto-accept rate dropping below target → page on-call