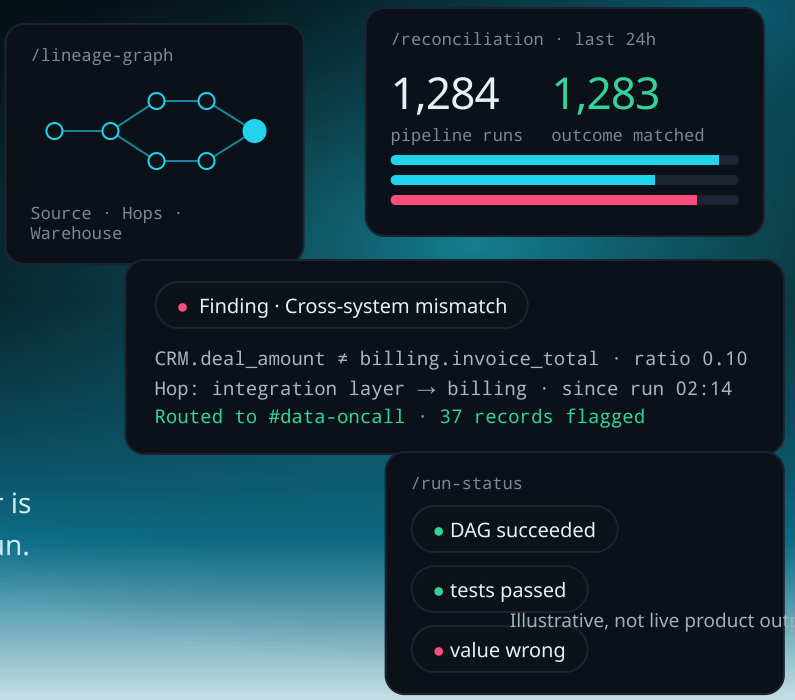
mmune

/datasheet

Data Integrity for Data Pipelines

mmune validates what your pipelines do to the business, not just whether they ran. When a number is wrong, you get the hop, the field and the first bad run.



A pipeline that completes is not a pipeline that is correct.

Orchestrators confirm the jobs ran. Tests confirm the rules you wrote. Observability confirms the tables are fresh. None of them confirms the record left the CRM and reached billing at the right amount.

So the failures that cost the most never fire an alert. The business notices first, days later. Most of the incident is spent finding the problem, not fixing it.

mmune learns how data should behave between systems, compares each hop with the one before, and flags the break within minutes. Read-only. No code in the pipeline.

A $42,000 deal went out as a $4,200 invoice.

One decimal, one field mapping. Every job was green.

- DAG succeeded
- All tests passed
- Warehouse fresh
- 0 alerts fired

Illustrative, not a customer figure.

01

Your pipeline, as it runs today

Every task ran, on time, in order. The value is wrong and nothing errors.

02

mmune validates every hop

Compares what arrived with what left. No rules to write; the baseline is the rule.

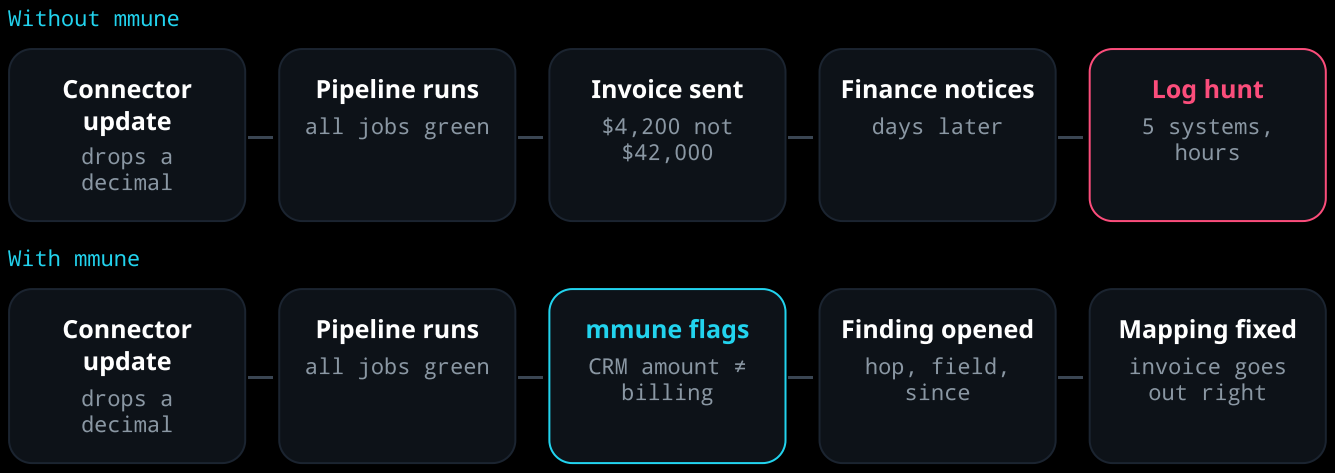
03

The finding, not the log hunt

One finding per root cause, with the evidence, routed to the team that owns the hop.

/one incident, with and without mmune

Same failure, two paths: the finding replaces the search



Same update, same green pipeline. One search starts from a symptom; the other from the hop that changed.

Illustrative.

/failure modes

The failures your tests were never written for

VALUE MUTATION

Right schema, wrong number

Type checks pass. Row counts match. The amount is 10× off.

SILENT FIELD DROP

Loads succeed, totals drift

A field goes empty upstream. Nothing fails. Sums stop adding up.

DUPLICATE · STALE RESEND

Counted twice, dated today

Yesterday's batch, today's timestamp. Revenue is doubled.

CROSS-SYSTEM DESYNC

Two systems, two truths

CRM says closed, ERP says open. Both are consistent. The process is not.

/on-call

What changes for the on-call engineer

Before	With mmune
Hears from the business, days later	Finding within minutes of the bad run
Starts from "the Q3 number looks off"	Starts from the hop, the field, the first bad run
Pulls logs from five systems by hand	Correlation arrives done, evidence attached
Guesses what to re-run	Affected tables, jobs and records listed
Writes one more test	The baseline already covers the next variant

/capabilities

For the pipeline, and for the people who run it

Check the outcome, not the run

- **Lineage graph**
  Systems, tables, queues, and who owns each hop.
- **Learned baseline**
  Ranges, ratios, cadence, cross-system correspondence. No rules to write.
- **Reconciliation**
  Same amount, count, status, keys across systems. Checked every 2–5 minutes.

Replace the log hunt

- **Pinpointed finding**
  Hop, field, deviation, first bad run, and the change it coincides with.
- **Blast radius**
  Which tables, jobs and records consumed the bad data.
- **Logs stay where they are**
  mmune tells you which one to open, and usually that you do not need to.

Calmer on-call

- **One finding per root cause**
  Ten broken dashboards, one finding.
- **Severity by business effect**
  Revenue, compliance, operations, integrity.
- **Guided next step**
  What to hold, what to re-run. mmune notifies; your team acts.

How it sits in your stack

- **Read-only observer**
  Samples landing tables and queues on a schedule. Never in the data path.
- **Metadata and aggregates**
  By default, mmune reads metadata and aggregates, not message payloads. On that default, PII stays inside your perimeter.
- **Keeps what you have**
  Orchestrator, dbt tests, observability, APM all stay.

Every finding arrives with the questions already answered.

01 What broke?

02 Which hop, which field?

03 Since when, what changed?

04 What did it hit?

Opened in Jira, ServiceNow, Slack or PagerDuty, routed to the hop owner.

Start with one flow that matters. Read-only, your data stays put.

mmune.io

Deal-to-invoice, order-to-cash, or the pipeline behind your most-watched dashboard. mmune reads the landing points for a short period and reports every cross-system deviation it found.

Works alongside the stack you already run

Airflow · Dagster | dbt · Spark | Kafka | Snowflake · Databricks | Jira · ServiceNow | Slack · PagerDuty