

Student Outcome Intelligence Platform

An Azure and Microsoft Fabric reference design that turns fragmented university data into explainable advisor action.

FOUNDATION

Unify student history

SIS, LMS, ERP, and campus activity resolved in ADLS Gen2, Fabric OneLake, and governed history tables.

INTELLIGENCE

Score risk early

Fabric feature products, Azure ML model release, calibrated risk bands, and transparent reason codes.

ACTION

Guide advisor outreach

A Power BI/Fabric advisor queue with row-level security, intervention context, freshness, and audit evidence.

The brief is a platform design problem, not a model-only task

University context

- 35,000 students across 8 faculties.
- Different programmes, study modes, and engagement patterns.
- Advisors need early signals before official records show risk.

Data and governance challenge

- SIS, LMS, ERP, and campus systems do not share identifiers or update rhythms.
- Student status changes over time, so current-state tables create bad history.
- The platform must be accurate, explainable, fair, auditable, and durable after launch.

The hard part is trustworthy history, not the model

Build the foundation first

- Canonical student identity across all systems.
- Type-2 history for mutable student status and programme data.
- Data quality contracts, Purview lineage, and schema-drift detection.
- Point-in-time features that only use data known at scoring time.

Then make risk useful

- Weekly scoring during the semester.
- Risk bands selected by advisor capacity and released with the Azure ML model version.
- Concrete reason codes, not black-box flags.
- Fairness monitoring and complete access audit.

Scope the target so the model can be defended

Outcome

Predict non-continuation next term, excluding graduation, exchange completion, and approved leave.

Decision

Advisor prioritization only. No automated punitive, academic, or financial decision.

Timing

Score in weeks 4, 6, 8, and 10 so support can happen before grades reveal the issue.

Privacy

DPIA, purpose limitation, data minimization, student transparency, and strict access control.

Each feed needs a contract, not just an extract

SYSTEM	SIGNAL	INTEGRATION DECISION	MAIN CONTROL
SIS	Status, programme, faculty, grades, leave, withdrawal, graduation	Data Factory/Fabric extract into ADLS Gen2 bronze	Type-2 history; never train from current-state status alone
LMS	Logins, course views, submissions, forums, video, quizzes	Event Hubs or incremental pulls normalized to daily student-course facts	Normalize by course, week, programme, and study mode
ERP	Financial aid, payments, scholarships, balances, overdue days	Daily vendor snapshot with Key Vault-backed credential	Effective-dated finance snapshots and source freshness checks
Campus	Library, WiFi, building entry	Event Hubs Capture or aggregated privacy-preserving daily extract	Do not treat low campus presence as uniformly risky

Every record carries source_system, source_record_id, event_time, available_at, ingested_at, pipeline_run_id, schema version, and Purview asset reference.

Azure lakehouse plus governed ML workflow

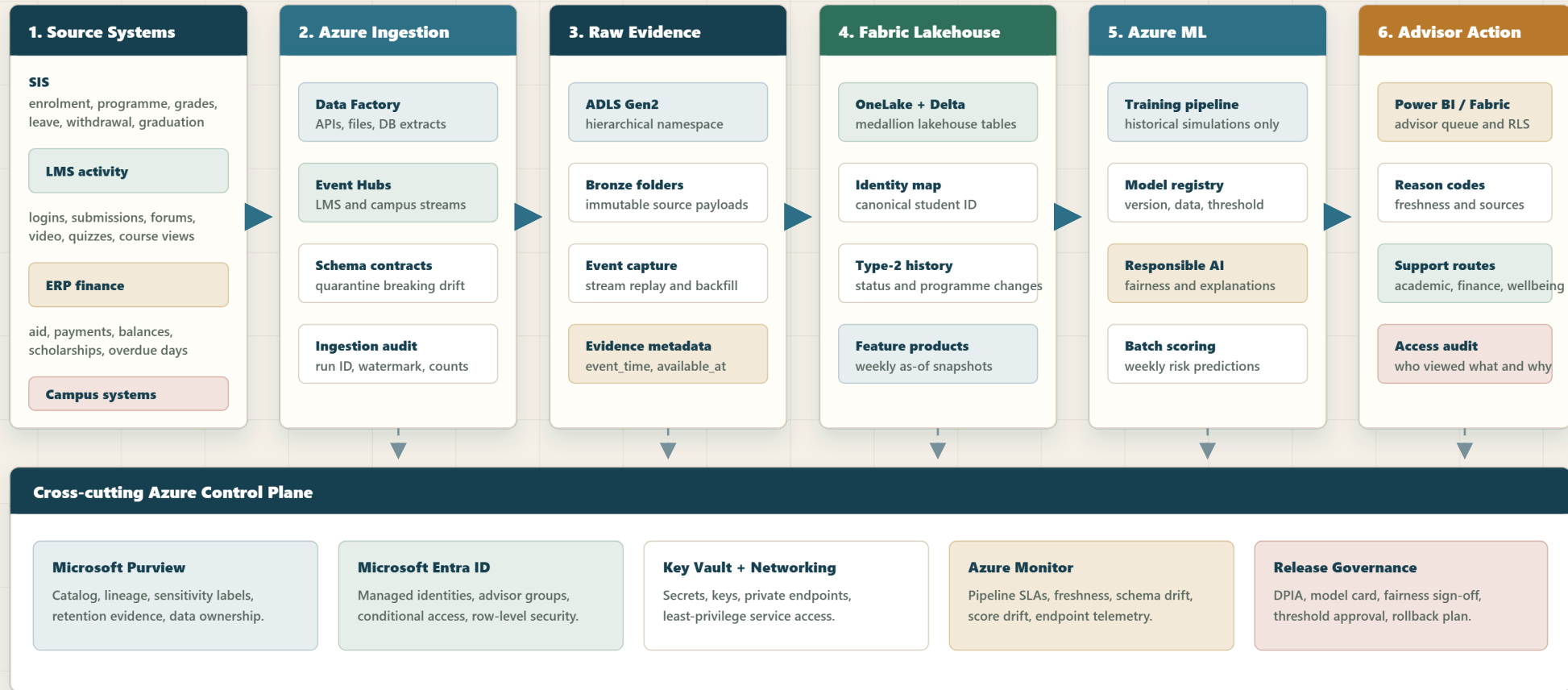
AZURE + MICROSOFT FABRIC TARGET ARCHITECTURE

Student Outcome Intelligence Platform

A governed data spine for point-in-time risk scoring, fairness review, and auditable advisor action.

Design constraints from the brief

No future data | mutable student status | diverse cohorts
Explainable red flags | complete lineage and access audit



POINT-IN-TIME LOCK

$usable(record, t) = 1[event_time \leq t \text{ and } available_at \leq t]$

Azure ML scoring reads only the Fabric feature snapshot built for the prediction as-of date.

Service choices are tied to the case risks

Data Factory + Event Hubs

One ingestion control plane for files, APIs, database extracts, and high-frequency activity streams.

ADLS Gen2 + OneLake

Raw replayable evidence, medallion lakehouse layers, Delta tables, and SQL access for feature products.

Microsoft Purview

Catalog, lineage, sensitivity labels, retention evidence, and data-product ownership.

Azure Machine Learning

Training pipelines, registry, model cards, calibration checks, Responsible AI review, and endpoint release.

Columns that make the solution work

TABLE	EXAMPLE COLUMNS	PURPOSE
identity_map	canonical_student_id, source_system, source_person_id, valid_from, valid_to, match_confidence	Joins independent systems without trusting inconsistent IDs.
ingestion_audit	pipeline_run_id, source_uri, schema_version, record_count, watermark, purview_asset_id	Links features and predictions back to Azure pipeline evidence.
student_status_history	status, faculty, programme, valid_from, valid_to, change_recorded_at	Stores historical truth when students switch, leave, withdraw, or return.
lms_activity_daily	login_count, course_views, submissions, late_submission_count, video_minutes	Captures digital engagement and assessment progress.
finance_snapshots	tuition_paid, outstanding_balance_nok, payment_overdue_days, aid_status	Signals financial friction while preserving source timestamp.
feature_student_week	as_of_date, lms_logins_14d, missing_assignments_to_date, campus_days_14d, payment_overdue_days	Model input generated as a point-in-time snapshot.
risk_prediction	model_version, risk_score, risk_band, top_reasons, feature_snapshot_hash	Reproducible advisor-facing output.

Every row has an as-of date

Feature rule

Use a source record only when:

```
event_time    <= as_of_date
available_at  <= as_of_date
valid_from    <= as_of_date
valid_to      >  as_of_date
```

Why it matters

- Late-arriving records cannot sneak into historical training.
- Future withdrawals cannot leak through current SIS status.
- Programme changes are joined as they were known then.
- Predictions can be reproduced for a student or regulator.

Start interpretable, calibrated, and capacity-aware

Baseline model

- Calibrated logistic regression or explainable gradient boosted trees.
- Train on historical term snapshots from the same weeks used in production.
- Exclude protected attributes from default model inputs.
- Version model, features, thresholds, and training data in the Azure ML release package.

Evaluation

- Recall and precision at advisor capacity.
- Calibration by risk band and by faculty.
- Responsible AI cohort review before each release.
- Lead time before official dropout signal.
- Stability and drift across terms.

Make the prediction actionable

RED

STU-101442 | Computer Science

Reasons: 3 missed assignments | 9 days since LMS login | NOK 8,900 overdue balance

Advisor: ADV-128

AMBER

STU-100087 | Nursing

Reasons: campus activity down 5 days | prior GPA 2.24 | LMS views down 61 percent

Advisor: ADV-141

GREEN

STU-100913 | Economics

No current intervention recommended; continue normal monitoring.

Advisor: ADV-119

The advisor sees reasons, freshness, support options, and intervention history. The student is never reduced to a score alone.

Engagement patterns differ for legitimate reasons

Design choices

- Compare engagement within programme, study mode, and term week.
- Do not treat low campus presence as uniformly risky.
- Keep sensitive attributes out of reason codes.
- Review advisor feedback for bias, not just model scores.

Monitoring

- Flag rate, recall, false-positive rate, and calibration by group.
- Reason-code distribution by group.
- Intervention offer and acceptance rates.
- Term-by-term drift in population and outcomes.

Answer "why was I flagged and who saw it?"

Why

Store score, band, reason codes, feature snapshot hash, feature values, model version, and threshold policy.

Who

Audit every advisor access with viewer, role, purpose, timestamp, fields returned, and student record.

Control

Row-level permissions, encrypted data, pseudonymous modeling, retention rules, and model release approval.

DPIA

Purview lineage

Azure ML model card

Feature registry

Schema tests

Rollback plan

How the answer maps to the prompt

Data foundation

- Four independent systems: source-specific contracts through Data Factory/Fabric and Event Hubs.
- Inconsistent identifiers: canonical identity map with match confidence.
- Changing student status: type-2 history and effective-dated joins.
- Schema changes: quarantine, contract tests, profiling, and promotion gates.

Decision safety

- Early warning: weekly in-semester scoring before official dropout records.
- No leakage: event_time and available_at must both be before scoring date.
- Fairness: calibration, FPR, recall, flag-rate, and reason-code checks by group.
- Auditability: explanation, model version, threshold policy, and access log.

Pilot first, then scale

PHASE	WORK	EXIT CRITERIA
0-4 weeks	DPIA, outcome definition, Azure landing zone, access model, data contracts	Approved scope and governance
4-10 weeks	ADLS/Fabric setup, ingest SIS/LMS/ERP/campus, identity map, history tables	Trusted integrated data layer
10-16 weeks	Fabric point-in-time features, labels, validation, first Azure ML model	Auditable offline model
16-22 weeks	Power BI/Fabric advisor pilot in two faculties, RLS, feedback loop	Useful interventions and safe workflow
22-30 weeks	Responsible AI review, Azure Monitor, Purview lineage, model registry, hardening	Scale decision

Key decisions I would defend

Start daily batch with event capture

Advisor intervention does not require millisecond latency. Data Factory batch plus Event Hubs capture preserves detail without overpromising real-time action.

Store raw and expose aggregates

Raw data supports audit and backfill, while aggregated features reduce privacy exposure in the advisor workflow.

Prefer explainability over small accuracy gains

An unsupported black-box score is less valuable than a slightly weaker model advisors and regulators can understand.

Use protected attributes for audit

You cannot prove fairness while refusing to measure outcomes across groups.

Dummy data and prototype outputs

Run locally

```
python .\student_outcome_platform_demo.py --out outputs --students 1200 --seed 42
```

Generated files

- source_samples/*.csv
- feature_student_week_and_predictions.csv
- advisor_risk_list.csv
- fairness_report.csv
- access_audit_sample.csv
- risk_distribution.png