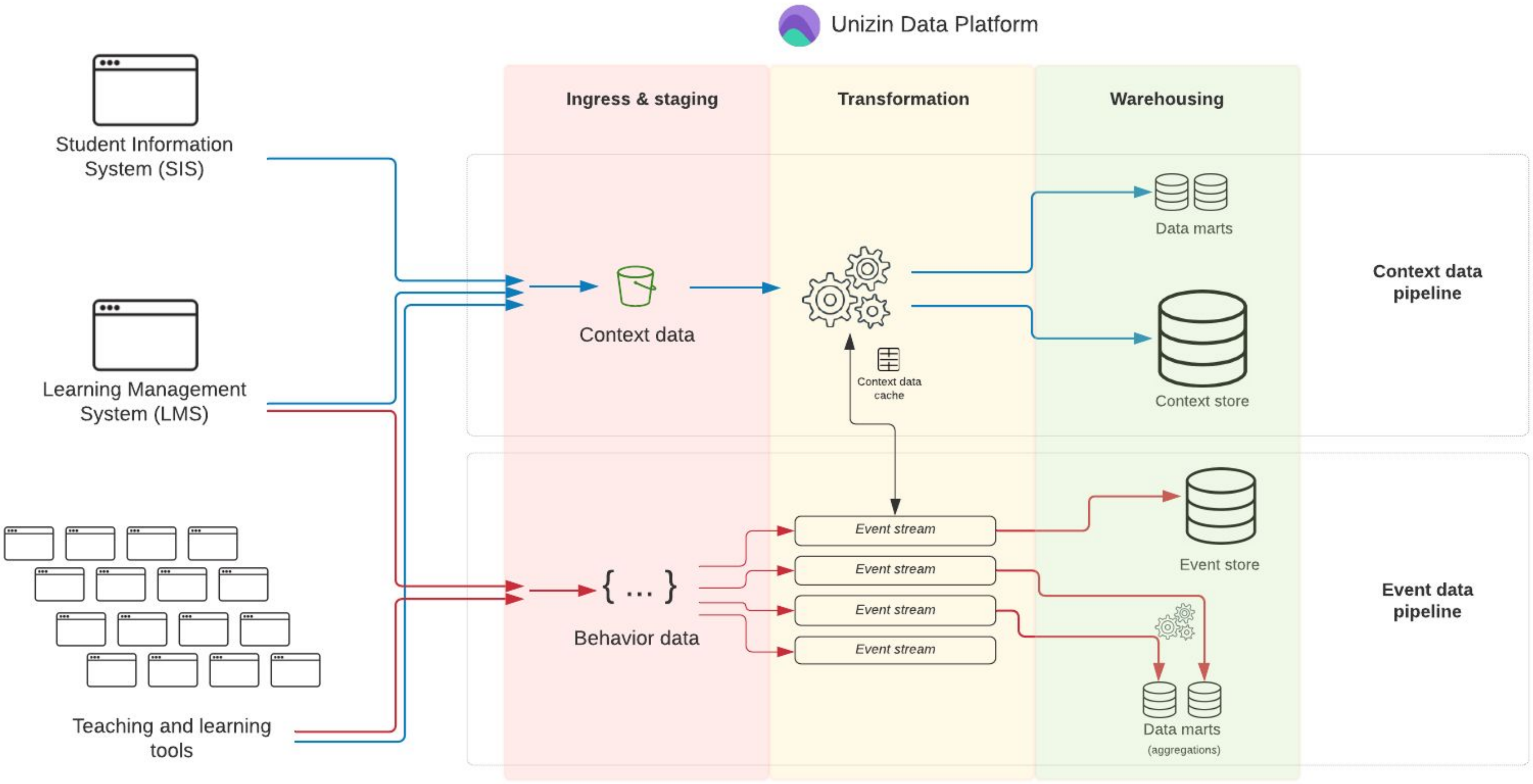
Unizin

LRS standards

January 27, 2021

Agenda

1. UDP recap
2. UDP’s mart/hub architecture
3. Demos


Unizin Data Platform
Student Information System (SIS)
Learning Management System (LMS)
Teaching and learning tools
Ingress & staging
Transformation
Warehousing
Context data
Context data cache
Data marts
Context store
Context data pipeline
Behavior data
Event stream
Event stream
Event stream
Event stream
Event store
Data marts
(aggregations)
Event data pipeline

Context data

Describe objects (e.g., learners, assignments, modules, outcomes, learning design, course catalog, degree) and relationships relevant to learners, learning environments, and overall academic experience.

- Rich in description
- Relational
- Typically from an ODS
- Suitable for batch processing

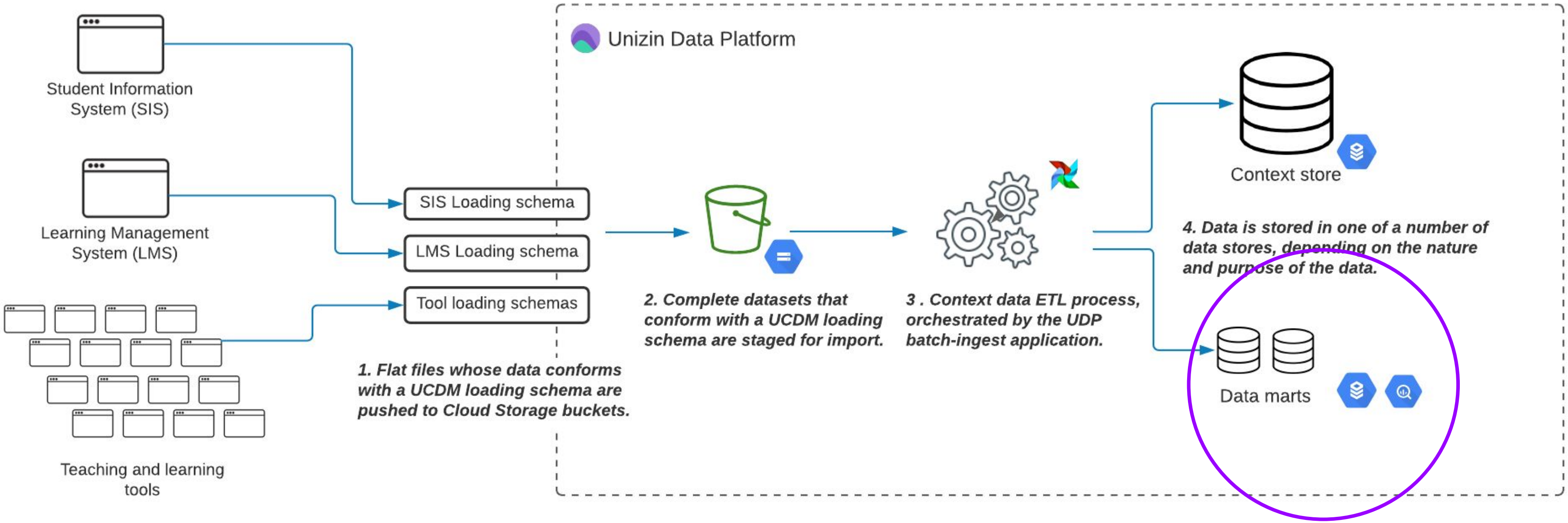
Data standard: UCDM

Behavior data

Describes the discrete actions of instructors, learners, teaching assistants, and even tools themselves in the learning environment

- Thin in description
- Event-driven
- Emitted from apps & APIs
- Suitable for event/signal processing

Data standard: IMS Global Caliper



What it does:

- Stage context data N sources
- Normalize in a single ontology
- Coalesce via surrogate IDs
- Unified presentation & derived marts

New!
Data marts of context data and derived metrics.

Context data

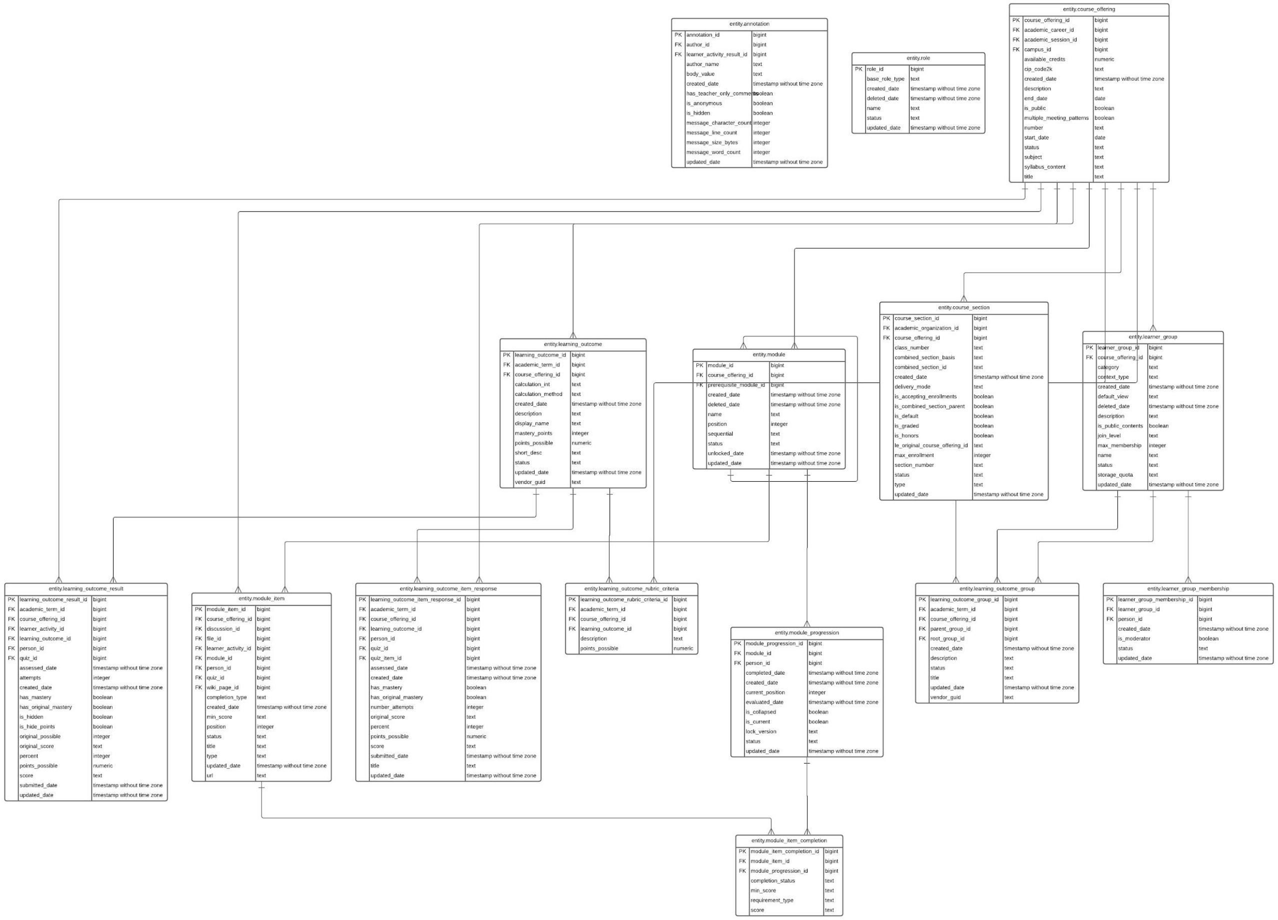
Data dictionary:

- 70 UCDM entities (primary concepts and tables).
- 800+ elements (attributes)
- Intended to model SIS, LMS, and Learning tool data in ODS

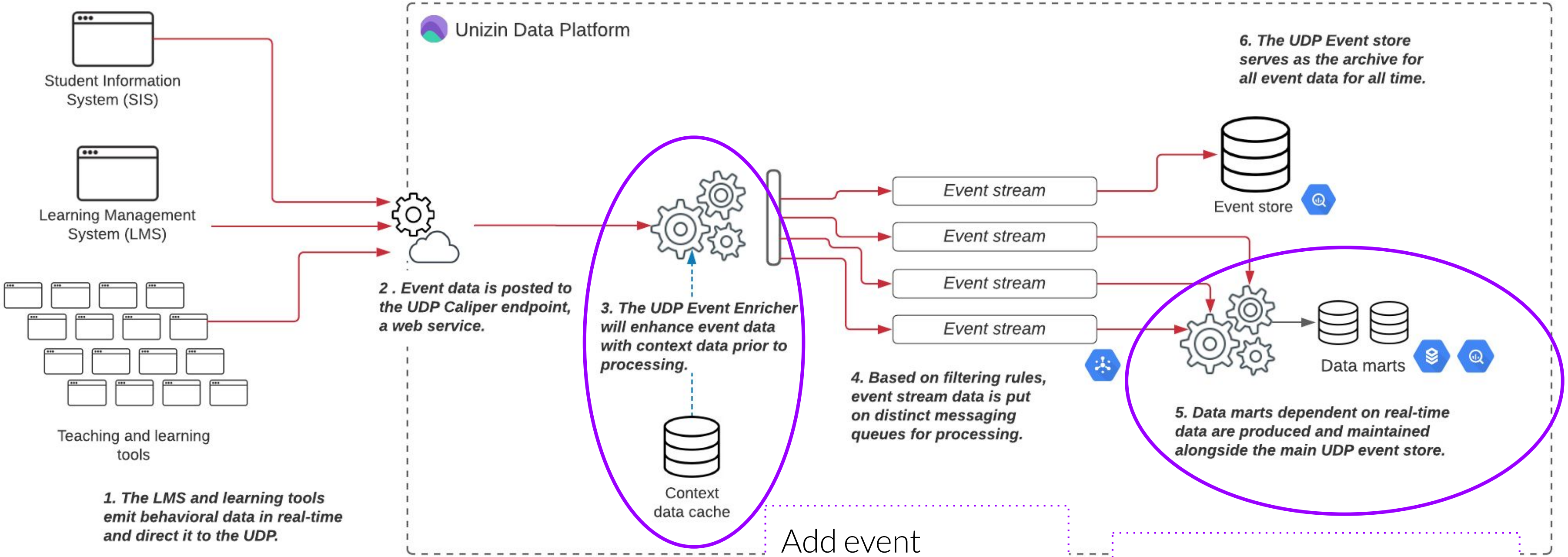
Relational model:

- All entities in a single relational model
- Enables all SIS, LMS, and learning data records to be associated

Visit: https://resources.unizin.org/display/UDP/Unizin+Common+Data+Model



Unizin



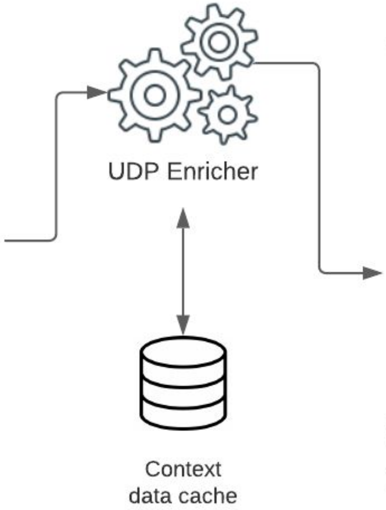
Add event attributes to enrich behavior data.

New!
Data marts of behavior data, aggregations, and metrics.

What it does:

- Accept, authenticate, validate event data
- Enrich data with context store cache
- Route events
- Process for various downstream purposes

```
{
  "@context": "http://purl.imsglobal.org/ctx/caliper/v1p1",
  "action": "Submitted",
  "actor": {
    "id": "https://learning-tool.com/caliper/user/456",
    "type": "Person"
  },
  "edApp": {
    "id": "https://learning-tool.com/",
    "type": "SoftwareApplication"
  },
  "eventTime": "2020-10-19T00:15:53.000Z",
  "id": "urn:uuid:46c46d0e-c1ae-2c3d-4a67-23ea4608db87",
  "object": {
    "id": "https://learning-tool.com/caliper/exam/1234",
    "name": "exam",
    "type": "Assessment"
  },
  "type": "AssessmentEvent"
}
```




```
{
  "@context": "http://purl.imsglobal.org/ctx/caliper/v1p1",
  "action": "Submitted",
  "actor": {
    "id": "https://learning-tool.com/caliper/user/456",
    "type": "Person"
  },
  "edApp": {
    "id": "https://learning-tool.com/",
    "type": "SoftwareApplication"
  },
  "eventTime": "2020-10-19T00:15:53.000Z",
  "id": "urn:uuid:46c46d0e-c1ae-2c3d-4a67-23ea4608db87",
  "object": {
    "id": "https://learning-tool.com/caliper/exam/1234",
    "name": "exam",
    "type": "Assessment"
  },
  "type": "AssessmentEvent"
},
{
  "udp_person_id": 1111
}
```

"Enrichment" technique enables context data to be associated with behavior data.

Data marts

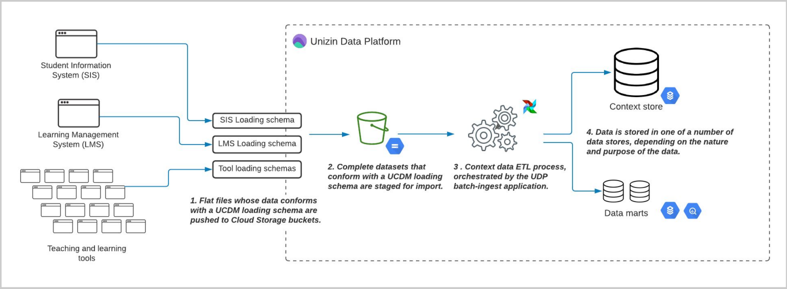
Build off the data pipeline architecture and use context & behavior processing capabilities.

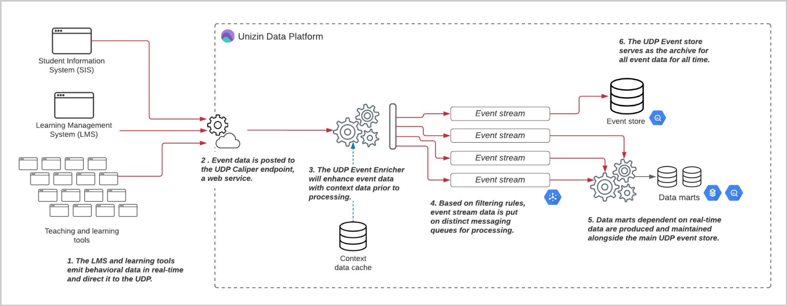
Context data:

- Expand Airflow ETL orchestration to build denormed/narrow data marts

Behavior data:

- Python scripts running in k8s clusters listening to PubSub, performing some compute, writing to a store
- Routing/processing interdependencies, esp. for aggregations
- Mix of real-time and scheduled ETL
- No special tooling yet (e.g., Kafka, Spark)





Tools we use

Pub/Sub

Events from a single fire hose are routed and then replicated along distinct messaging queues, each dedicated to a particular form of processing.

BigQuery

Strong candidate for storing and processing large volumes of data, both scheduled and ad hoc.

We use BigQuery for a variety of event-based data marts and research / ML training datasets.

Kubernetes

Almost all of our event processing logic is captured in small Python scripts that run concurrently and that are deployed in k8s clusters.

CloudSQL

Reasonable candidate for particular real-time data marts.

Airflow

Batch ETL orchestration for our context data pipeline.

Architecting for marts

Divide the broad domain of marts into a handful of dimensions:

- **Grain**: what level of detail is associated with the facts, metrics, measures in the mart?
- **Features/metrics**: what features, facts, or metrics are expressed for this grain of data?
- **Latency/periodicity**: how fresh does the mart's data need to be to satisfy its use-cases?
- **Complementarity**: can marts work together to meet use-case requirements (e.g., slowly-changing labels and real-time metrics).

Architecting for persona

Unizin institutions generally need to serve four classes of data stakeholders:

- **Ops/BI reporting staff**: produce timely, actionable reporting and analytics for decision-support.
- **Faculty, advising, CTL**: consume timely course measures and analytics to drive intervention, teaching insights.
- **Researchers**: create and/or consume prepared research datasets for regular and ad hoc inquiry.
- **ML practice**: produce datasets to train, and then use, ML models for various classifications.

Course readiness

Problem: how do we ensure that all LMS courses are published and available by the start of the term?

Grain: By course offering

Data's role: provide near real-time insight into unpublished courses that likely ought to be published.

Who: BI, IR, Ops, Faculty/instructors

Latency: < 1 hour

Maintained via context data pipeline.

Course offering design metrics		
PK	course_offering_id	bigint
FK	academic_session_id	bigint
	term_name	string
	course_title	string
	num_enrollments	bigint
	num_assignments	bigint
	num_modules	bigint
...	...	...

D enormalized mart of course design metrics. Slowly changing data updated every 24 hours via context data ETL.

Maintained via event data pipeline.

Course offering status		
PK	course_offering_id	bigint
	status	string

Simple event-based mart to represent course status. Runs every hour and maintains current course status with < 1hour latency.

Last activity

Problem: how do we identify students who may not be interacting in the LMS but ought to be?

Grain: By course offering, by student

Data's role: provide ability to query for students who are lagging, per metrics that are relevant to context, in real-time.

Who: BI, IR, Ops

Latency: < 1 minute

Maintained via context data pipeline.

User attributes, denormed		
PK	user_id	bigint
FK	name	bigint
...	...	...

Denormed table of user labels (name, email, etc) used for reporting purposes.

Maintained via event data pipeline.

Last activity		
FK	course_offering_id	bigint
FK	actor_id	bigint
	last_activity	timestamp
	last_navigation_activity	timestamp
	last_media_activity	timestamp
	last_grade_activity	timestamp
	last_assessment_activity	timestamp
	last_assignment_activity	timestamp

Table (in postgres) updated every 15 seconds (via upsert) to maintain the last time a student was activity on some particular kind of activity per the Caliper ontology.

Tool launches

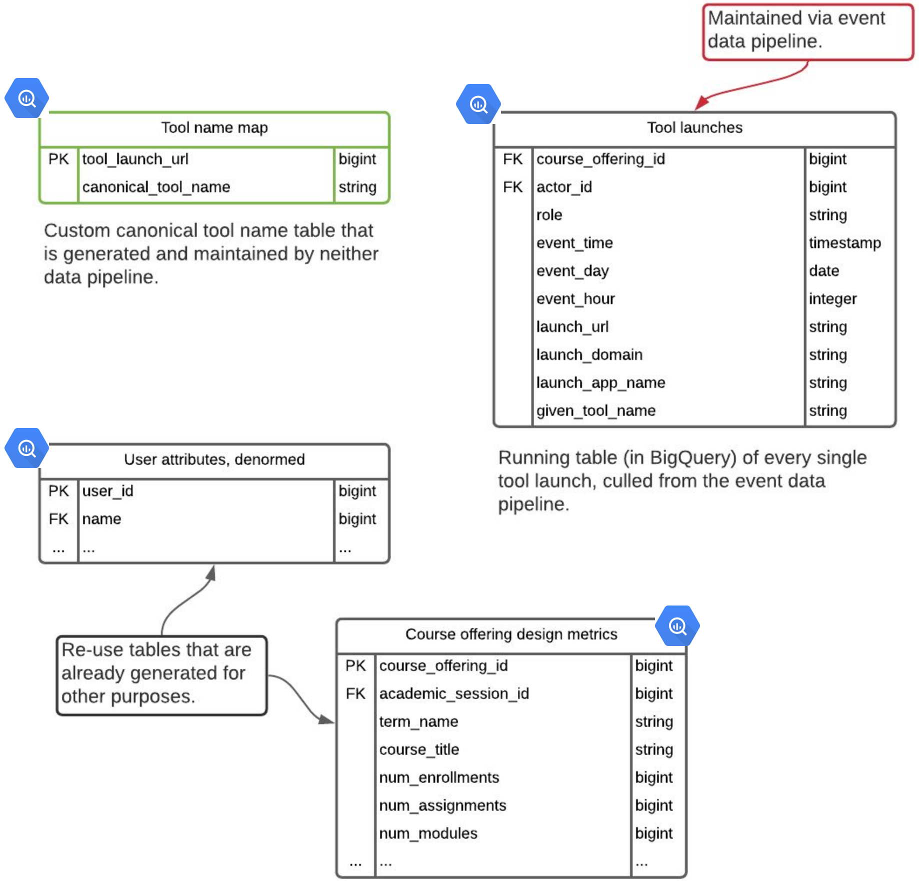
Problem: we have little insight to what LTI tools are used where, how much, and by which departments and courses.

Grain: By course offering, by user, by tool launch event

Data's role: provide near-real time insight into LTI tool usage, measured as a function of launches by users, across the LMS and the institution.

Who: BI, IR, Ops

Latency: < 1 minute



Course grade prediction training/scoring sets

Problem: classify likelihood of a failing grade; confidence must improve as term unfolds.

Grain: By course enrollment

Data's role: beyond demo/test/gpa data, leverage behavioral metrics.

Who: Research, ML practice

Latency: Weekly

Metrics in the feature set:

- By person, by course observations
- Grade/outcomes
- Demo, SES, standardized test
- Term, course, department, etc.
- Performance & behaviors, by week of term, for all activities
- Ahead/behind performance & behaviors, relative to class, for each week of term

https://gitlab.com/unizin/community/unizin-data-platform/iu-bar-cgr

Course to course conditional probability

Problem: we need to provide course recommendations to students to find an optimal pathway.

Grain: By course enrollment

Data's role: leverage historical and outcomes data to train models or provide research data.

Who: Research, ML practice

Latency: Every term

Course to course conditional probability		
	current_term_order	integer
	current_node_course	bigint
	next_term_order	integer
	next_node_course	bigint
	next_node_given_current_node_percentage	numerical
	student_transition_count	integer

```
/*
  Mart / Course offering / Course to course conditional probability

  Computes the conditional probability that if you took Course A in Term X
  you will take course B in Term Rank X + 1 (next term).

  Dependencies:
    * Base / Course offering
    * Base / Course section enrollment
*/

/*
  Every course taken by the student, decorated with the title of the course and
  the rank of the term in their academic career in which they took it.
*/
WITH student_course_term AS (
  SELECT
  co.title AS course_title
  , cse.person_id AS person_id
  , cse.rank_person_academic_term_order_asc AS rank_person_academic_term_order_asc
  FROM
  base.course_section_enrollment cse
  LEFT JOIN
  base.course_offering co USING(course_offering_id)
  WHERE
  cse.is_role_student = 1
  AND cse.credits_taken > 0
),

/*
  Computes the count of the student population represented in the previously-
  generated table of students and their courses.
*/
total_student_count AS (
  SELECT
  COUNT(distinct sct.person_id) AS total_student_count
  FROM
  student_course_term AS sct
),

/*
  Computes the count of students who took the courses by the order of their
  Academic term.
*/
```

Unizin

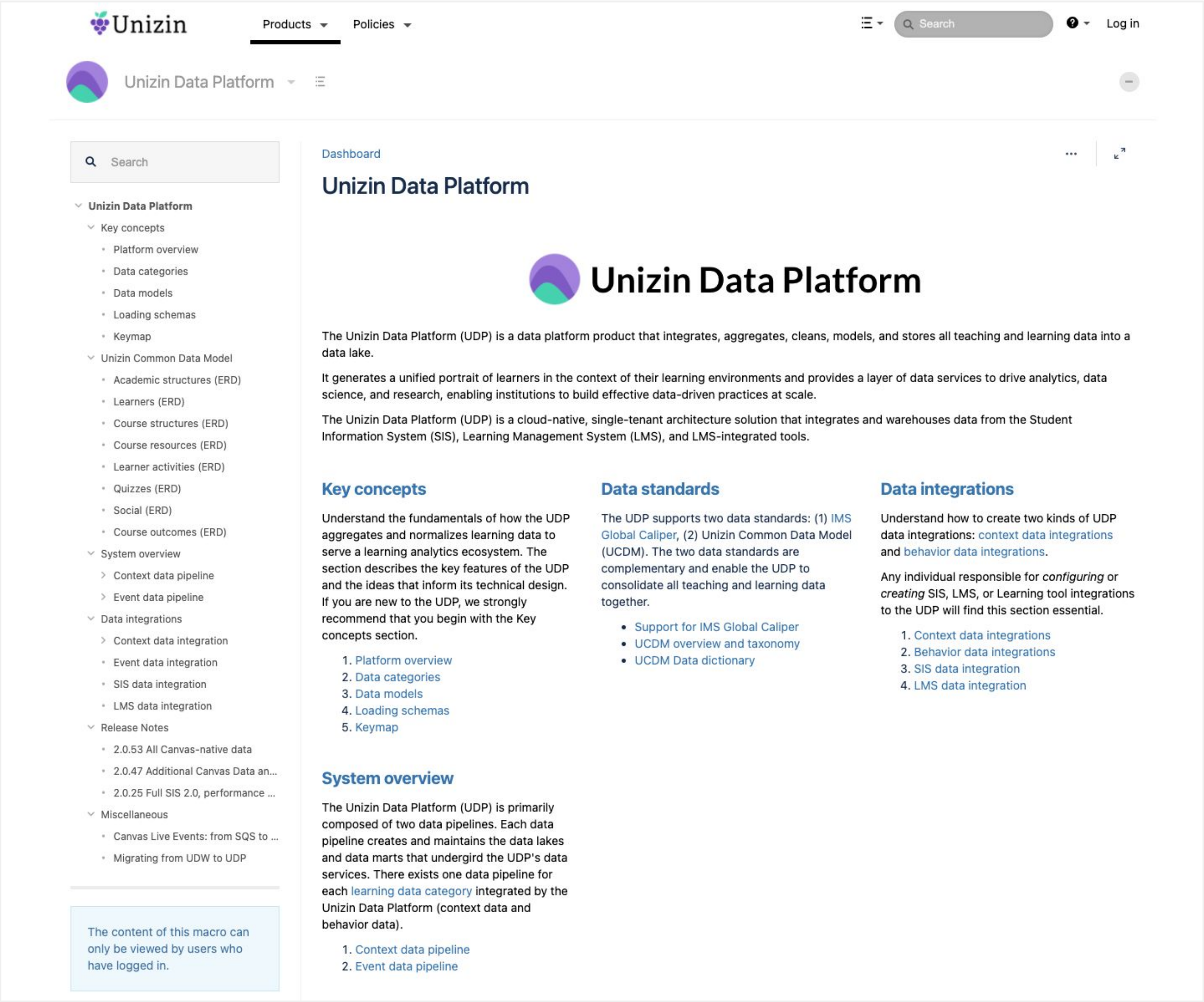
UDP Documentation

We've recently updated our UDP documentation.

Intended for a variety of audiences (technical, analytics, research, etc.).

The emerging UDP Data services layer will be documented here, too.

Visit:
https://resources.unzin.org



Unizin Products Policies Search Log in

Unizin Data Platform

Search

Unizin Data Platform
- Key concepts
  - Platform overview
  - Data categories
  - Data models
  - Loading schemas
  - Keymap
- Unizin Common Data Model
  - Academic structures (ERD)
  - Learners (ERD)
  - Course structures (ERD)
  - Course resources (ERD)
  - Learner activities (ERD)
  - Quizzes (ERD)
  - Social (ERD)
  - Course outcomes (ERD)
- System overview
  - Context data pipeline
  - Event data pipeline
- Data integrations
  - Context data integration
  - Event data integration
  - SIS data integration
  - LMS data integration
- Release Notes
  - 2.0.53 All Canvas-native data
  - 2.0.47 Additional Canvas Data an...
  - 2.0.25 Full SIS 2.0, performance ...
- Miscellaneous
  - Canvas Live Events: from SQS to ...
  - Migrating from UDW to UDP

The content of this macro can only be viewed by users who have logged in.

Dashboard

Unizin Data Platform

Unizin Data Platform

The Unizin Data Platform (UDP) is a data platform product that integrates, aggregates, cleans, models, and stores all teaching and learning data into a data lake.

It generates a unified portrait of learners in the context of their learning environments and provides a layer of data services to drive analytics, data science, and research, enabling institutions to build effective data-driven practices at scale.

The Unizin Data Platform (UDP) is a cloud-native, single-tenant architecture solution that integrates and warehouses data from the Student Information System (SIS), Learning Management System (LMS), and LMS-integrated tools.

Key concepts

Understand the fundamentals of how the UDP aggregates and normalizes learning data to serve a learning analytics ecosystem. The section describes the key features of the UDP and the ideas that inform its technical design. If you are new to the UDP, we strongly recommend that you begin with the Key concepts section.

1. Platform overview
2. Data categories
3. Data models
4. Loading schemas
5. Keymap

System overview

The Unizin Data Platform (UDP) is primarily composed of two data pipelines. Each data pipeline creates and maintains the data lakes and data marts that undergird the UDP's data services. There exists one data pipeline for each learning data category integrated by the Unizin Data Platform (context data and behavior data).

1. Context data pipeline
2. Event data pipeline

Data standards

The UDP supports two data standards: (1) IMS Global Caliper, (2) Unizin Common Data Model (UCDM). The two data standards are complementary and enable the UDP to consolidate all teaching and learning data together.

- Support for IMS Global Caliper
- UCDM overview and taxonomy
- UCDM Data dictionary

Data integrations

Understand how to create two kinds of UDP data integrations: context data integrations and behavior data integrations.

Any individual responsible for *configuring* or *creating* SIS, LMS, or Learning tool integrations to the UDP will find this section essential.

1. Context data integrations
2. Behavior data integrations
3. SIS data integration
4. LMS data integration

Thank you

unizin.org