Data Mart Process

Purpose

The purpose of this document is to explain the relationship between the Database “microservices” and the “data mart” to clarify that the microservice database of any of the 3 defined for this project is the source of truth but not available for reporting, while the data mart is available for reporting, but as originally designed and configured, is not operational in a way that allows for reliable data retrieval from the microservices.

Glossary and Terms

Internal User Reports are those that are available to VAConnects participating teachers and division heads through the VAConnects user interface (UI) that are created, validated, and released to production by the VLP developer team.

External User Reports are those that are provided at regular intervals to the Virginia Department of Education (VDOE) who is the sponsor and data owner across entities of the VAConnects project.

VALLSS and PALS are the user interfaces made up as a collection of web forms to capture teacher administered subtest results. This web form data is not synched to the VALLSS database (cosmos?). This synching requires that the web form data is downloaded as csv files the reuploaded to VALLSS. For Assessment data, it is downloaded and manually reuploaded after FlexMIRT scoring is calculated for each subtest item.

FlexMIRT

<https://vpgcentral.com/software/flexmirt/>

Process Workflow

<https://myuva.sharepoint.com/sites/DataSystemIntegration/_layouts/15/stream.aspx?id=%2Fsites%2FDataSystemIntegration%2FShared%20Documents%2FGeneral%2FWorking%20Groups%2FData%20and%20Reporting%2FMeeting%20Recordings%2FVAConnects%5FDandR%5FMar15%5F2024%2Emp4&referrer=StreamWebApp%2EWeb&referrerScenario=AddressBarCopied%2Eview>

Non-relational VAConnects data (Assessment/subtests and Observational – complex, ever-changing data).

Currently 2, eventually 4 per WillowTree refinement.

A Data Mart is an “event driven” relational mini-database structured to use “micro-services” to separate and organize data from a large source to a smaller source. To start, Extract-Transform-Load (etl) triggers are built to auto populate the data mart from intermediate data sorting (“micro-services”). And from this middle step, the micro-services further refine what is needed for reporting into specifically defined Data Marts so they can efficiently return output in columns and rows across one page or more (Paginated typically built using Power BI Report Builder) and for output that returns in charts and figures (Visualizations typically build using Power BI Desktop). You can have more than 1 data mart per foundational database.

Parse of the diagram below:

User Identity = how teachers, division heads log in

Front End = the user interface where data is entered/imported into

Microservices Databases=

* SQL (relational)
* Cosmos planet icon (non-relational, BLOB)🡪 best for handling lots of frequently changing data. The Service Bus is the vehicle that delivers this data from a microservice to a data mart.
* Assessment Screener, Scoring, and Authoring

Microservices are used for these “services”. You cannot direct query them:

* User Management
* Student Management
* Location Management
* Activity Logging
* Notification
* Material Ordering

Service Bus = currently PALS platform where the daily csv user/student data uploads that imports the data into the “Student” tables.

PALS legacy is non-relational and json is used to query?

VLP does have access to COSMOS but does not use it to query data.

The microservices data is then organized into :

* **Data Lake** (non-relational source data)🡪 not yet built. Stores all events that happen on the Data Stream (aka “Service Bus”).
* **Data Warehouse** (relational source data)🡪 **not yet built**. Aggregation of data from all 3 entities. **You will be able to direct query it but bc it will contain entity data from all 3, access will be limited (don’t count on it to build VLP queries for example if you find data mart missing something).**

**!!!!!To note: there is a “data warehouse” process in PALS where Chris accesses data to provide data dump download for users. This is not the same as the Azure “Date Warehouse” process.**

* **Data Mart** (subset of Data Lake and Data Warehouse tables and views built and visible to data teams based on reporting requirements)🡪 will be broken out by entity, so there will be 3 in total for VLP, VKRP, LB5

A screenshot of a computer

Description automatically generated

Think of this concept as how you might organize a community STEM learning lab within a school division (the “large data source”) based on these requirements:

The student pool comes from multiple middle schools into a single Learning Lab.

The Learning Lab is something a child is “invited” to attend, so not all students participate.

The ETL is the student invitation, some sort of pre-req formula that selects (“triggers”) which students from the “school pool” can attend the Learning Lab.

This makes the “school pool” the micro-service (intermediate data sorting) that transports invited student via a “Service Bus” to either a Data Lake or Data Warehouse.

Within the Learning Lab, there are STEM subspecialities such as Animation, Robotics, and 3D printing to which students are assigned.

The data for these students per STEM specialty is further refined into data marts from the “Service Bus”. In this case, there are 3 data marts, Animation, Robotics, 3D printing that all contain data the division uses to track participation, learning measures, and budget.

This is “skinny” reporting; the reporting data pool (Data Mart) is smaller than the source data (Data Lake and/or Data Warehouse) because each data mart contains only what is absolutely necessary for reporting, nothing more, nothing less.

!!!Important: Any one data mart will never contain all the Learning Lab data, yet there might be data outside of Robotics, for example, from the foundational database required to create reports. Therefore, it is critical that report specifications are clearly defined and agreed. Otherwise, a data mart and triggers that populate data to it from the micro services using the service bus will be missing and block reporting needs. If this happens, reports will be unreliable and data mart(s) rebuild(s) based on revised specifications will be required (costly and time consuming).

Data Ingestion Process

<https://myuva.sharepoint.com/:w:/r/sites/DataSystemIntegration/_layouts/15/Doc.aspx?sourcedoc=%7B21046966-3526-4950-88B9-FA13E6A9B8E0%7D&file=User%20creation.docx&action=default&mobileredirect=true>

External csv files uploaded nightly to PALS (Cosmos DB) for these domains:

* + um - user management
  + lm - location management
  + sm - student management
  + as – assessment screener
  + aa – assessment authoring

and each domain has 3 BLOB flow storage options (tst, dev, local) and separate api calls that once the file(s) are uploaded starts the etl moving the data from BLOB storage to Cosmos:

uva-as-api-prod-eus2-cdb is a Cosmos storage location

uva-as-sb-prod-eus2-fa is the Service Bus “transfer station”

uva-as-ccf-prod-eus2-fa  stores the etl trigger(s) such as PublishStudentAssessments

For example, any file with the required data type and structure placed in a BLOB flow storage folder will be read by the corresponding API for processing.

* User Management

Classroon Users: User ID,Email,Classroom ID,Classroom Role,Last Modified

Divisions: User ID,Email,VDOE Division Number,Division Role,Last Modified

Sites: User ID,Email,VDOE Site Number,Site Role,Last Modified,VDOE Division Number

Users: User ID,First Name,Last Name,Email,Last Modified

* Student Management

<term>: administration method, belowbench, classroomid, completescores, dateofbirth ethnicity, gender, grade, identified, Last Modified,race,school year, sti, student id studentfirstname, studentlastname, studentmiddlename, teacher email, term timeofday, vdoe division number, vdoe site number, vdoe\_primaryfundingcode, PreSchool, NativeEnglish, DualLanguage, Service\_ESL, Service\_TitleI, Service\_Tutor, Service\_None, Service\_Other, Service\_OtherName, ServiceEIRI

* Location Management

Classroom scaled: VDOE Division Number, VDOE Site Number, Teacher, Grade, TimeOfDay, Last Modified, Classroom ID, Name

Divisions: VDOE Division Number, Division Name

Sites: VDOE Division Number, Division Name, Site Name, Division Id, VDOE School Number

Process History

<document all final versions, revisions and reason for version and date>

|  |  |  |
| --- | --- | --- |
| Name/  Title | <signature of author, reviewers, PI approver(s)> | Initial version <create date> |
| Name/  Title |  |  |