Ed-Fi Temporal ODS

Chris Moffatt

Director of Technology, Ed-Fi Alliance

Agenda

- Review Background and Context
- Temporal ODS Project
  - Project Overview
  - Design and Architecture
  - Demo – Temporal Snapshot & Query Proof of Concept
- Discussion

Background - Ed-Fi “Founding Era”

- Ed-Fi ODS did not need a strong time or history component to fulfill its original use cases:
  1. Collect and centralize education accountability data (LEA → SEA)
     - w/ down stream systems for district fund allocation, public accountability, etc.
     - Batch collection windows & offline, manual processes for data corrections
  2. SEA-hosted student performance dashboards for educators*
- These use cases de-emphasized the importance of storing multiple years of data in a single repository

*Ed-Fi Dashboards display historical data, but source of historical data is not the ODS…dashboard-specific data created from daily snapshots of the dashboard metrics

Background - Ed-Fi Data Warehouse

- Field implementations of an Ed-Fi Data Warehouse
  - Dimensional representation of the Ed-Fi data model, uses Ed-Fi ODS as source
  - Initial implementations in DSST and TN
  - Subsequent implementations in Shelby, others?
- Findings:
  - Traditional dimensional data warehouse models were (a) highly complex to use and (b) often not a complete solution for certain kinds of analysis
  - Significant investment would be needed to ingest the data warehouse artifacts, and keep it aligned to the Ed-Fi ODS as the Ed-Fi data standard and ODS evolved – cost to both Alliance (core) and implementers (extensions)
- Alliance chose to publish the data warehouse artifacts to Ed-Fi Exchange (vs. incorporating into core)

Background – 2015 - 2016

- Several Ed-Fi licensees have raised use cases related to historical data and historical data snapshots
  - LEA's, regional centers have begun adopting Ed-Fi technology
    - Core use case: Centralized, standards-based hub for data integration of K12 data
    - Additional use cases - related to reporting and analytics - are unfulfilled because the ODS/API lacks a time-based aspect to maintain historical, longitudinal data
  - SEA's with ODS/API implementations would like to support data collection where historical data is accessible and correctable by source systems (i.e. available in the ODS)
- Work was undertaken in 2016 within the Alliance and a community Special Interest Group – with two distinct – but related – threads of work:
  - **Project 1: Adding date support to the Ed-Fi Unifying Data Model**
  - **Project 2: Adding temporal support in the Ed-Fi ODS**
- Summary presented at Ed-Fi Summit 2016 (link)

Background – 2016 - "UDM Dates"

Project 1: UDM Dates

Problem Statement

- The Ed-Fi Unifying Data Model defines the basic structure of the XSD. SQL database. API endpoint
- The Ed-Fi UDM
  - Association
  - Student At
  - Assessmer
  - Etc.
- However
  1. There are
     - E.g. Stu
  2. There are longitudi
     - E.g. Address, a student Proj

Project 1: UDM Dates

Solution

1. Identify ar and attrib
   - Have date
   - Are expec
   - Are used
2. Identify ar storage

Project 1: UDM Dates

Next Steps

Ed-Fi Summit - 2016

- Current Status
  - We are not confident that we have an "obvious" direction – that meets the complex set of constraints
  - We don't have a field implementation underway or teed up, to learn from
  - Given this, adding additional UDM dates into the data standard in the v2.1 timeframe (RFC in Q4/16) is too risky
- Instead
  - Recruit proof-of-concept in Q1, 2017 that explores ecosystem ability to manage selective historical data
  - Flesh out correct approach, for inclusion in data standard v2.2

Background – 2016 - "Temporal ODS"

Project 2: Temporal ODS

Problem Statement

- The Ed-Fi ODS / for entities in th
- However, it is nc
  1. Storing multip
  2. Maintaining p attributes tha
  3. Loading or co

Project 2: Temporal ODS

Solution

Develop a Te

1. Storing ar Current-Y
2. Take a Sn
3. Bulk Load
4. View and Endpoint

Project 2: Temporal ODS

Next Steps

Ed-Fi Summit - 2016

- Move forward with RFP to develop "Core Temporal ODS"
  - Defer Use Case 4: "View and edit Temporal Data in the ODS through API Endpoints"
  - Design solution to be "data standard agnostic" – i.e. feature can be added to a version of the ODS/API that supports the data standard v2.0 or v2.1
- Timeline

Activity	Estimated Timeline
Publish RFP	November 2016
Review responses and select vendor	January 2017
Begin Implementation	February 2017
Complete Implementation	May 2017

Ed-Fi Temporal ODS Project - 2017

Temporal ODS Project - 2017

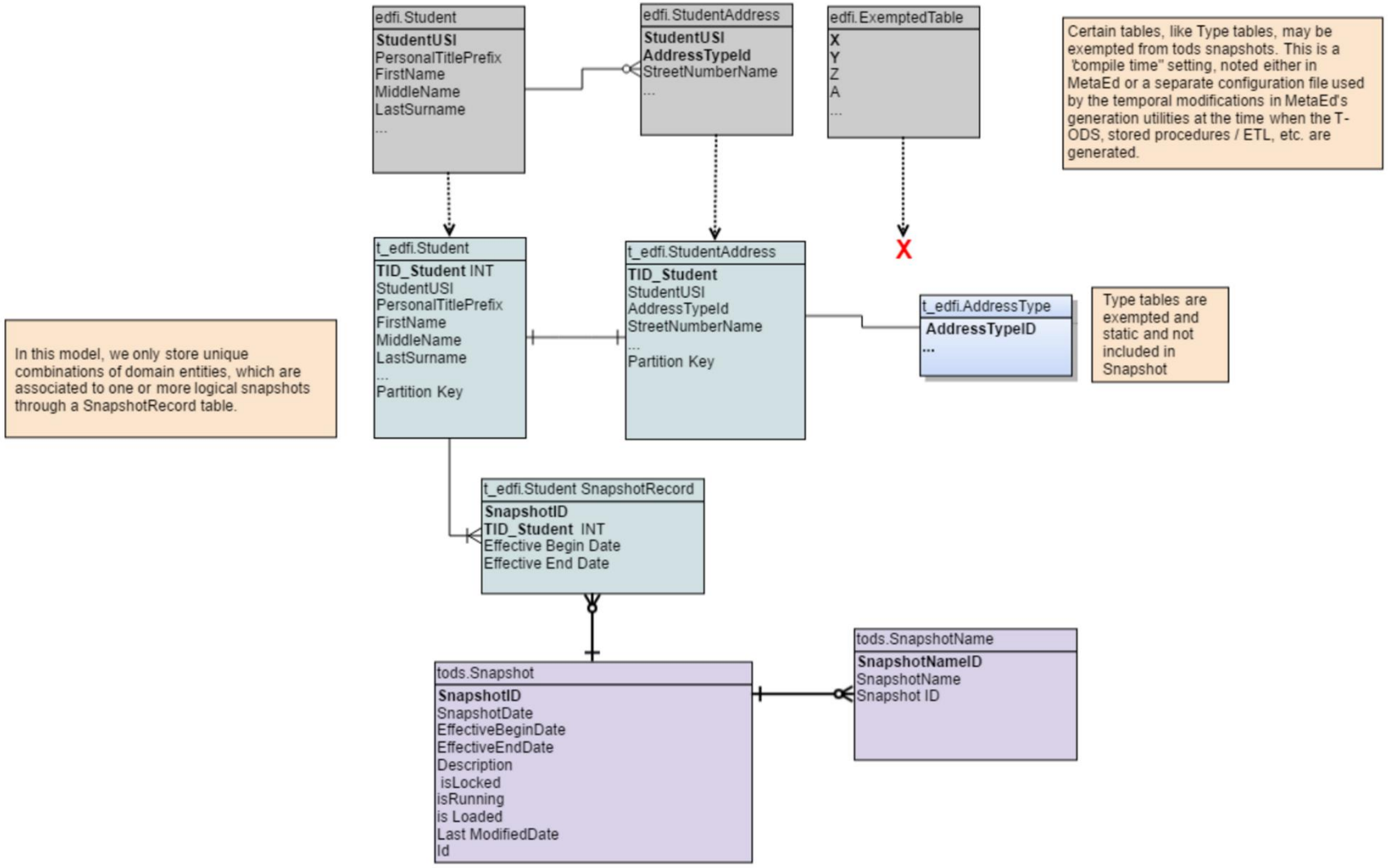
- Published an RFP in November 2016 – for full Temporal ODS capability

Use Cases
UC1 – Store and Query Temporal Data
UC2 – Take a Snapshot of Current ODS Data
UC3 - Bulk Load Prior-Year and Snapshot Data
UC4 – View and Edit Temporal Data Through API Endpoints

- Selected proposal from Double Line Partners and started project in Mid-February

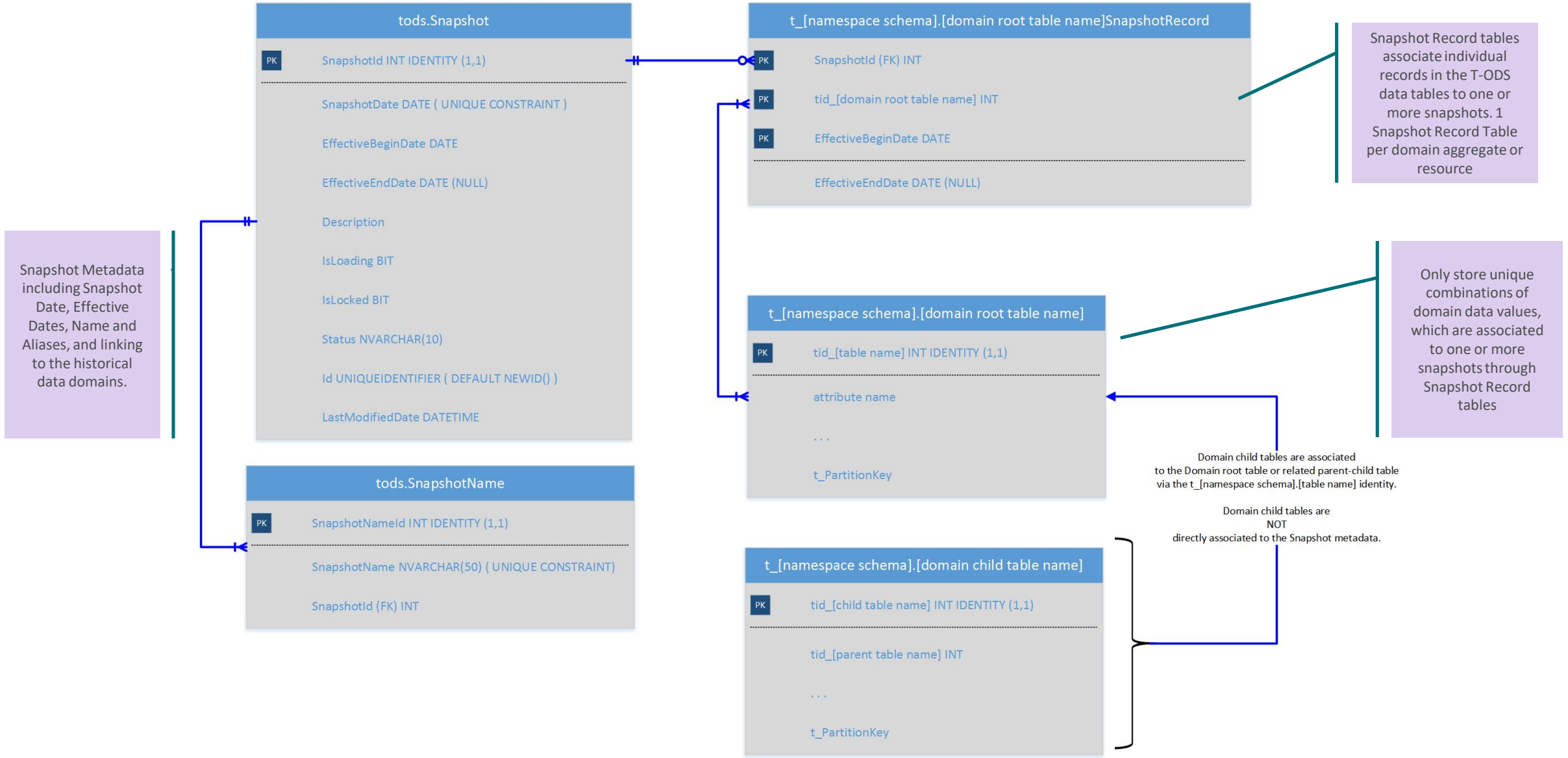
Design and Architecture

Design and Architecture – Temporal Schema

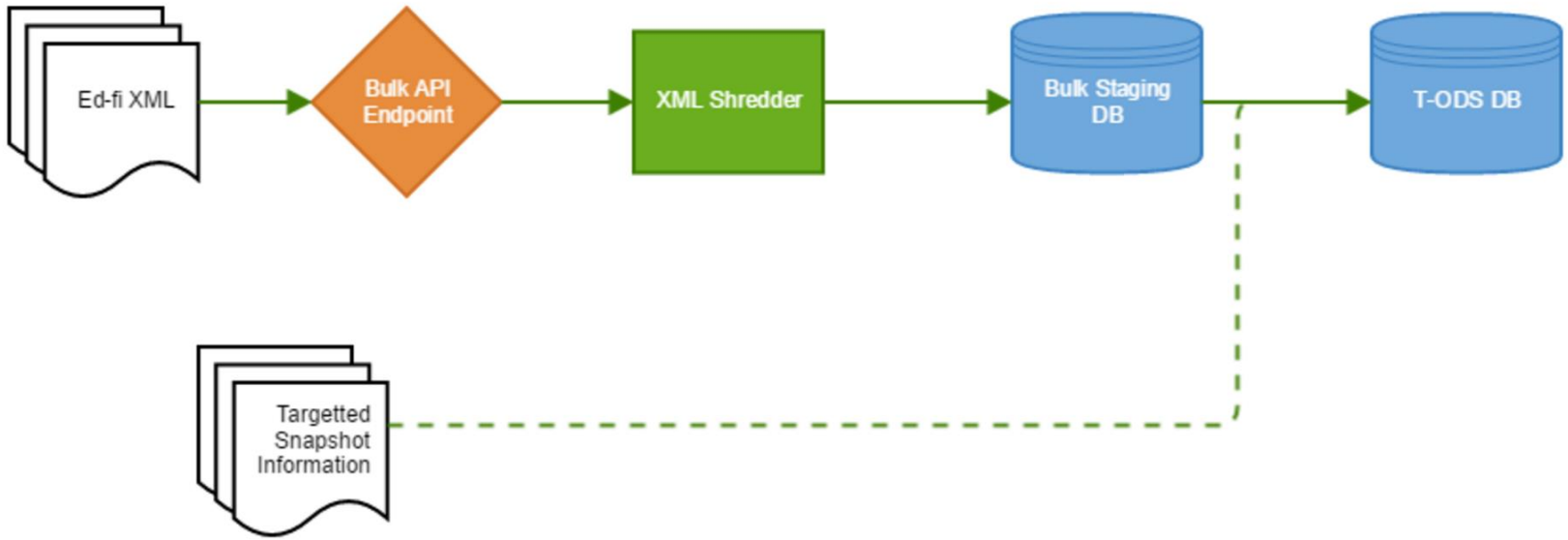


- Each ODS table has a T-ODS counterpart following a predictable pattern and generated from MetaEd
- T-ODS tables are populated from ODS tables at time of snapshot via T-ODS stored procedures
- T-ODS tables store only *unique combinations of attribute values by domain entity* to avoid data explosion
- Domain child entities (i.e. StudentAddress) are not directly associated to the snapshot metadata.
- T-ODS data can be migrated between Ed-Fi versions in the same way as the ODS
- No modifications to the ODS tables

Design and Architecture – Temporal Schema

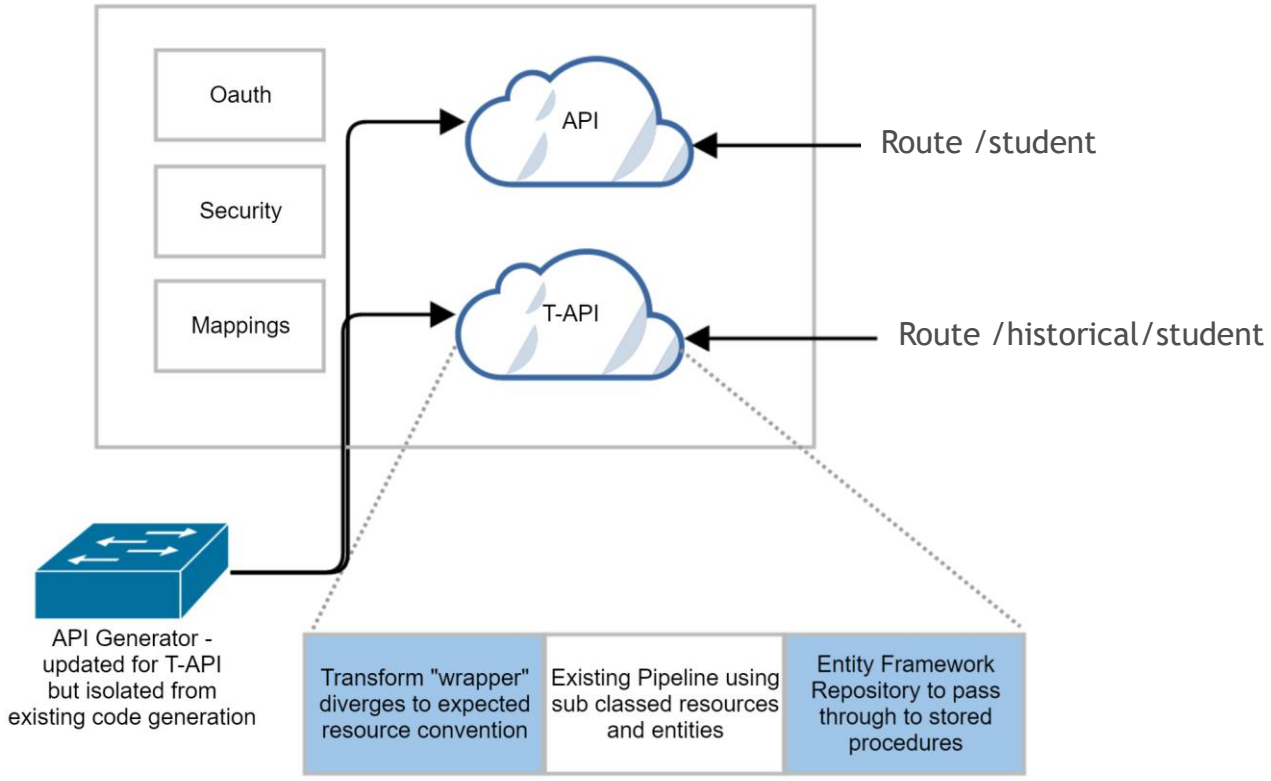


Design and Architecture – Temporal Bulk Load



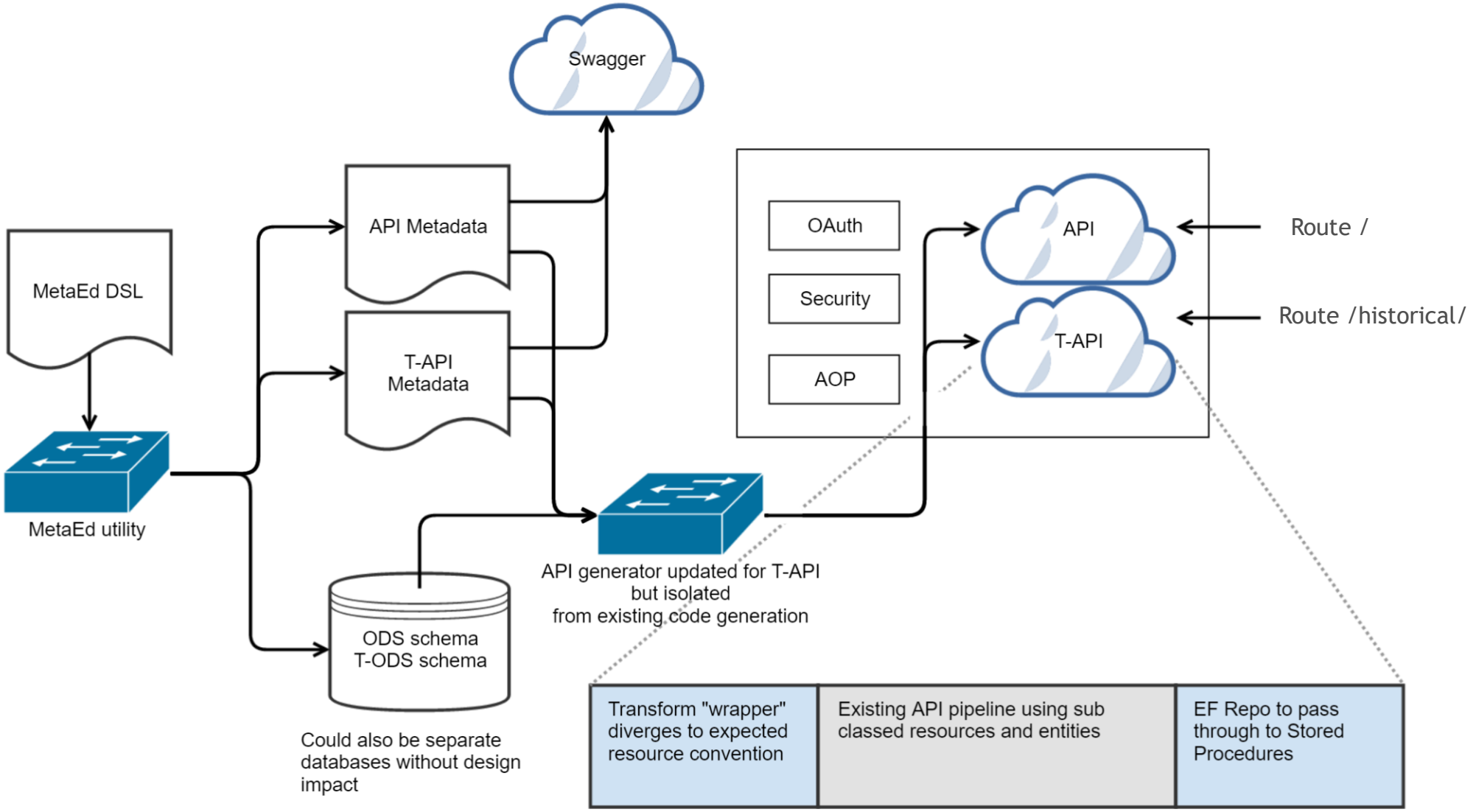
- Temporal bulk re-uses as much of the existing Bulk API pipeline as possible (circa bulk enhancements in ODS/API v2.2)
- Which snapshot to target provided either through a wrapper or separate parameters
- Bulk Pipeline steps dealing with the ingestion of bulk XML files, processing by the XML shredder and population of the bulk staging database will not be modified
- A new process to move the snapshot data from the Bulk Staging DB and merge into the T-ODS - This process will enforce relational integrity
- Process creates, updates and cleans up any snapshot metadata records impacted by the changes submitted

Design and Architecture – Temporal API



- The Temporal API generated in a similar manner as the Ed-Fi ODS API
- Leverages existing API pipeline using sub-classed resources and entities
- Where possible, duplication of logic/code & benefits from use of existing cross-cutting concerns in the API code base
- Entity Framework repository for ORM integrated with Stored Procedures

Code Generation and Extensibility



Demo – T-ODS Proof of Concept

Design Approach - Summary

- Minimize data explosion
- Easy to use, and – for licensees who don't need it – easy to ignore
- Leverage MetaEd, the Ed-Fi ODS/API, Swagger, and other existing components
- Enable the Ed-Fi Alliance & community to control this new functionality in the same manner as existing functionality
- Avoid introducing unnecessary architectural complexity
- Follow established architectural, documentation, and other conventions
- Minimize the increase in maintenance and support costs
- Future-proof

Schedule

- T-ODS Technical Preview : October 30th
- Public Release of T-ODS – part of Ed-Fi ODS/API v3.1 (Q1, 2018)

Discussion Questions

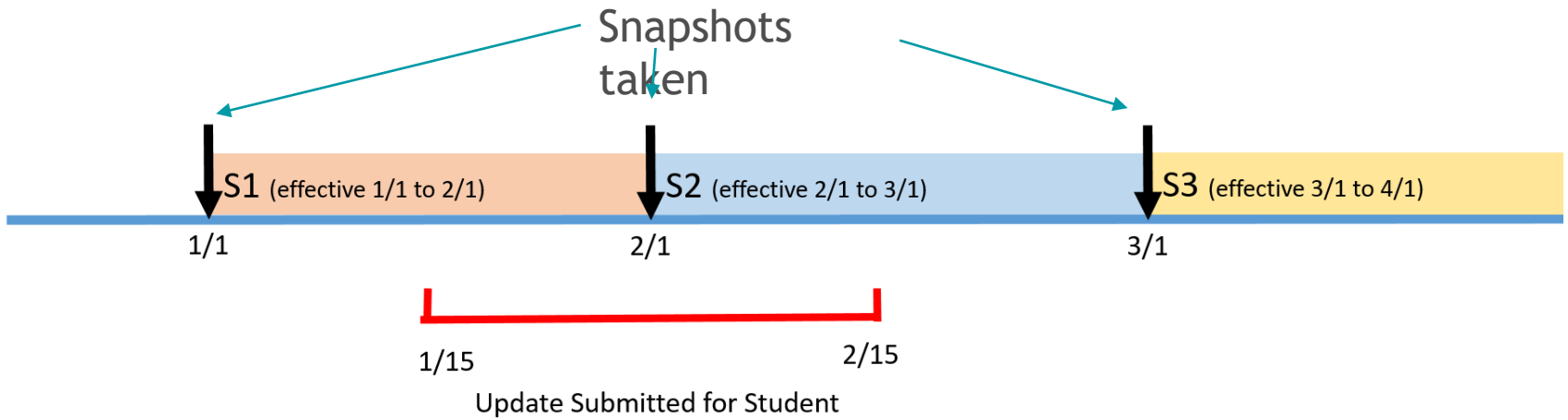
- What do you see as the tradeoffs of having a single set of API end points for current and historical vs. having a separate set of API end points for historical?
- Should non-data-steward access (e.g. SIS) to a historical resource in a snapshot be determined by access to the equivalent resource (based on relationship auth or other auth strategy) in the current ODS
- For bulk use cases:
  - Which do you expect:
    - Specifically target one or more snapshots

    OR

    - Bulk load data that is true for a date range, regardless of alignment with snapshots?
  - Are you expecting to use XML for incremental updates such as: (1) a single student record, (2) grades for a single grading period or will bulk upload represent the districts full and complete file?
- Are there use cases where you would anticipate wanting to not include specific ODS tables in the snapshot?

Supporting Arbitrary Effective Date Updates

- Snapshots will contain effective begin and end dates, and the data records within each snapshot will inherit those effective dates by default.
- The T-API will support updates to history for effective date ranges that do not align with the snapshot date ranges.
- This allows a user to indicate that a change in data was effective as of a date that does not align with a snapshot range – improving accuracy of as of date queries - when needed.



In the above scenario - A query for the student as of 1/10 will return the student that existed at the time of snapshot 1. A query on 1/20 will return the student submitted in the arbitrary effective update.