2015 Ed-Fi Alliance Summit

Austin Texas, October 12-14, 2015

ed-fi® ALLIANCE

It all adds up.

Ed-Fi ODS API Bootcamp

Milan Malkani
milanmalkani@gmail.com

Logistics

- Ed-Fi ODS API Bootcamp
  - 10/13 (Tuesday) from 2:30 to 5:00
  - 10/14 (Wednesday) from 9:00 to 1:00
    - Pizza will be served
- Closing Session
  - 10/14 (Wednesday) from 1:00 to 2:00
- We'll take breaks
- Ask every question you can think of

Bootcamp goals for keepers of the Ed-Fi ODS API

- Understand how to manage the security model
- Understand the how-to's of integrating with the Ed-Fi ODS API to provide assistance
- Know how to extend the data model
- **Understand the concepts

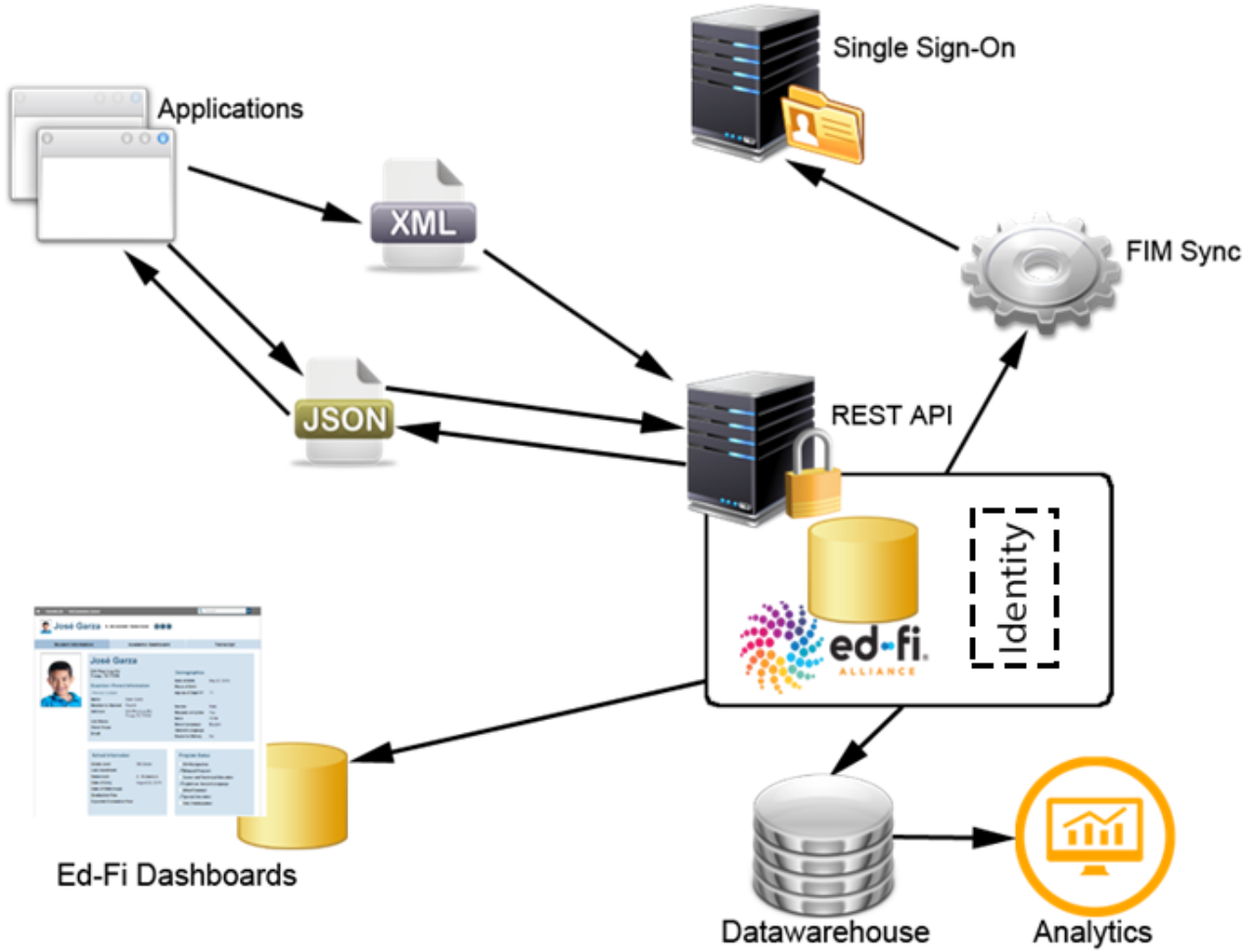
Bootcamp goals for application developers

- Understand how to navigate the security model and know which permissions to ask for
- Understand how to integrate with the Ed-Fi ODS API
- Understand the difference between core and extension data when planning multiple integrations
- **Understand the concepts

Quick survey

- Who has worked with the Ed-Fi data standard?
- Who has worked with the Ed-Fi ODS API?
- Who has intention of connecting the Ed-Fi ODS API to the Ed-Fi Dashboards?
- Who has followed the Getting Started guide to setup the Ed-Fi ODS API on a local machine?

Full ecosystem



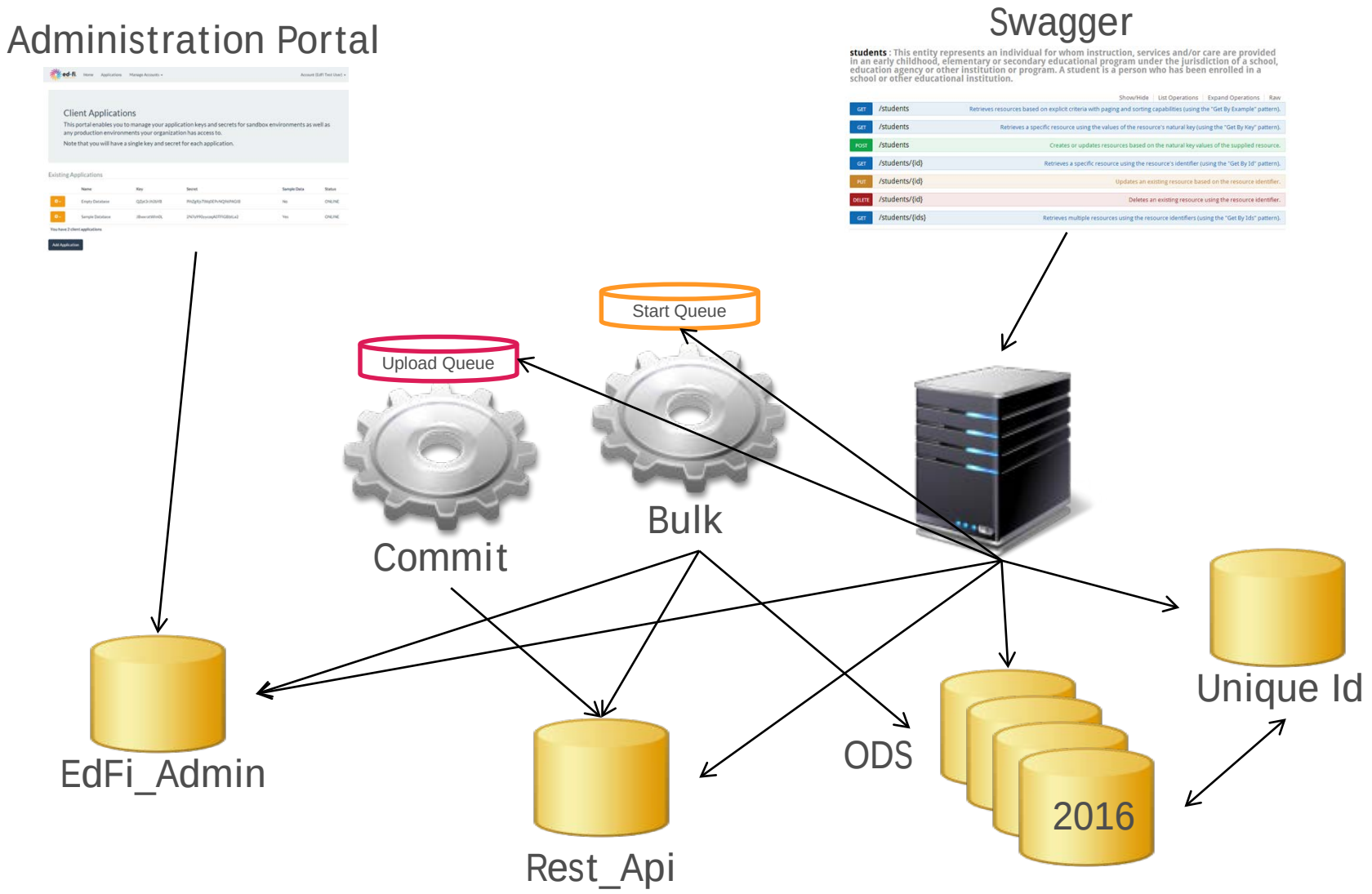
Identity interface

- Provides a common interface and models to search, get, and create Unique Ids
- Abstraction from the actual implementation
- WiseId (Wisconsin) and EduId (Idaho) have already been integrated for separate implementations
- v1 of the interface is officially available while v2 has been developed and awaiting final review

Walkthrough of the components

- API
- Admin Portal
- Swagger UI
- Commit and Bulk Worker
- Using Fiddler to monitor traffic

High level architecture



Administration portal

- We will use the Administration portal that is specifically for the sandbox environment
  - Creates one database per “sandbox”
- Separate Administration portal for “shared instance” (aka production)
  - Manages access to a shared set of databases (one per year)

Swagger UI

- Notice the metadata that is emitted by the API
- The same metadata is used by the SDK Generator
- Easy way to view and test transactions
- Great source of documentation
- Typically not deployed to production (CORS)

Fiddler

- Monitor traffic to and from the API
- Configure the host as localhost.fiddler in the filters section

LAB: Using the Admin Portal and Swagger UI

Ed-Fi ODS API SDK

- Support for Java and C#
- Uses the same metadata that powers the Swagger UI
- Handles the Oauth 2 authentication seamlessly
- Requires the RestSharp Nuget package

Generating the Ed-Fi ODS API SDK

- Documentation on Github for setup and command line prompts
- New command script being added to create the SDK
- Add in classes for Unique Id and bulk support

Types

- Core enumerations that are not anticipated to change
- Reference using the Description

Descriptors

- Flexible enumerations structure allowing for multiple variations of the same enumeration value
- Defined by the district or state but vendor-specific descriptors can be created as well
- Reference using the CodeValue

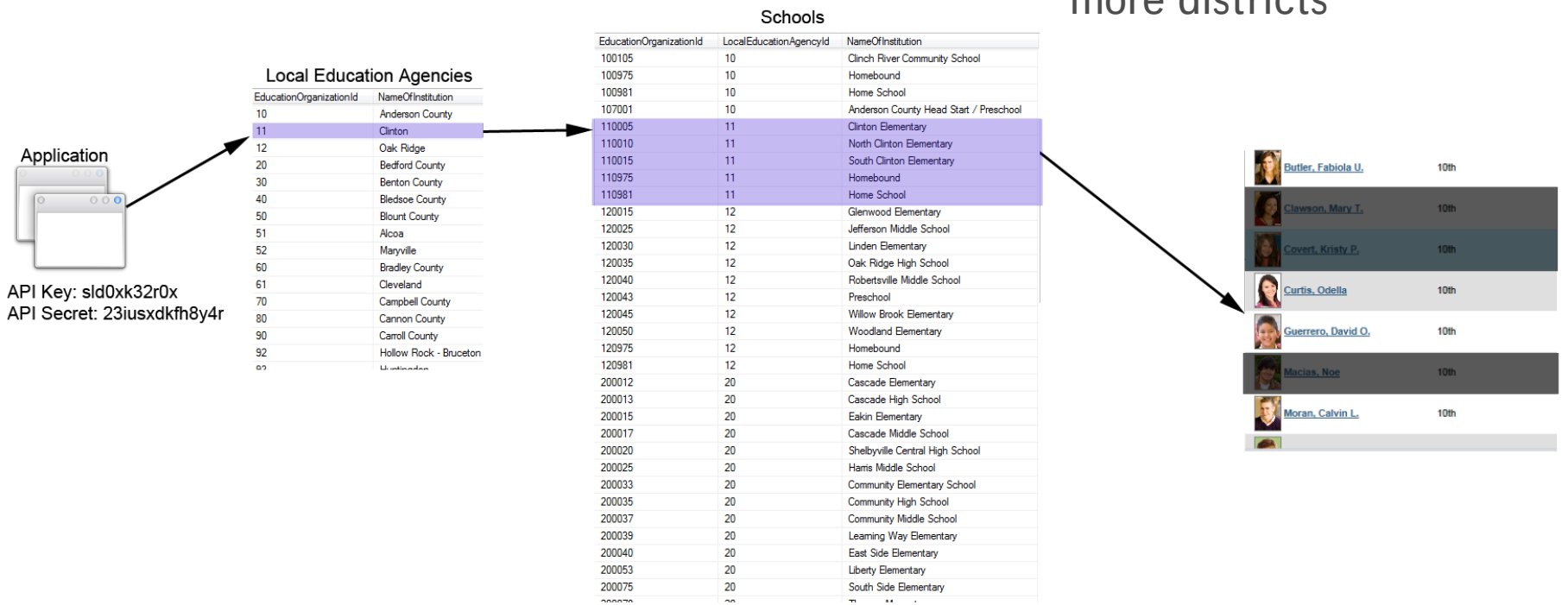
LAB: Generating New SDK Files

Security Overview

- Relationship based
- Resource + authorization strategy arranged by claim sets
- “SIS Vendor” is the default claim set for the sandbox admin portal
- Queries in techdocs will display the permissions of the claim set

Relationship based security

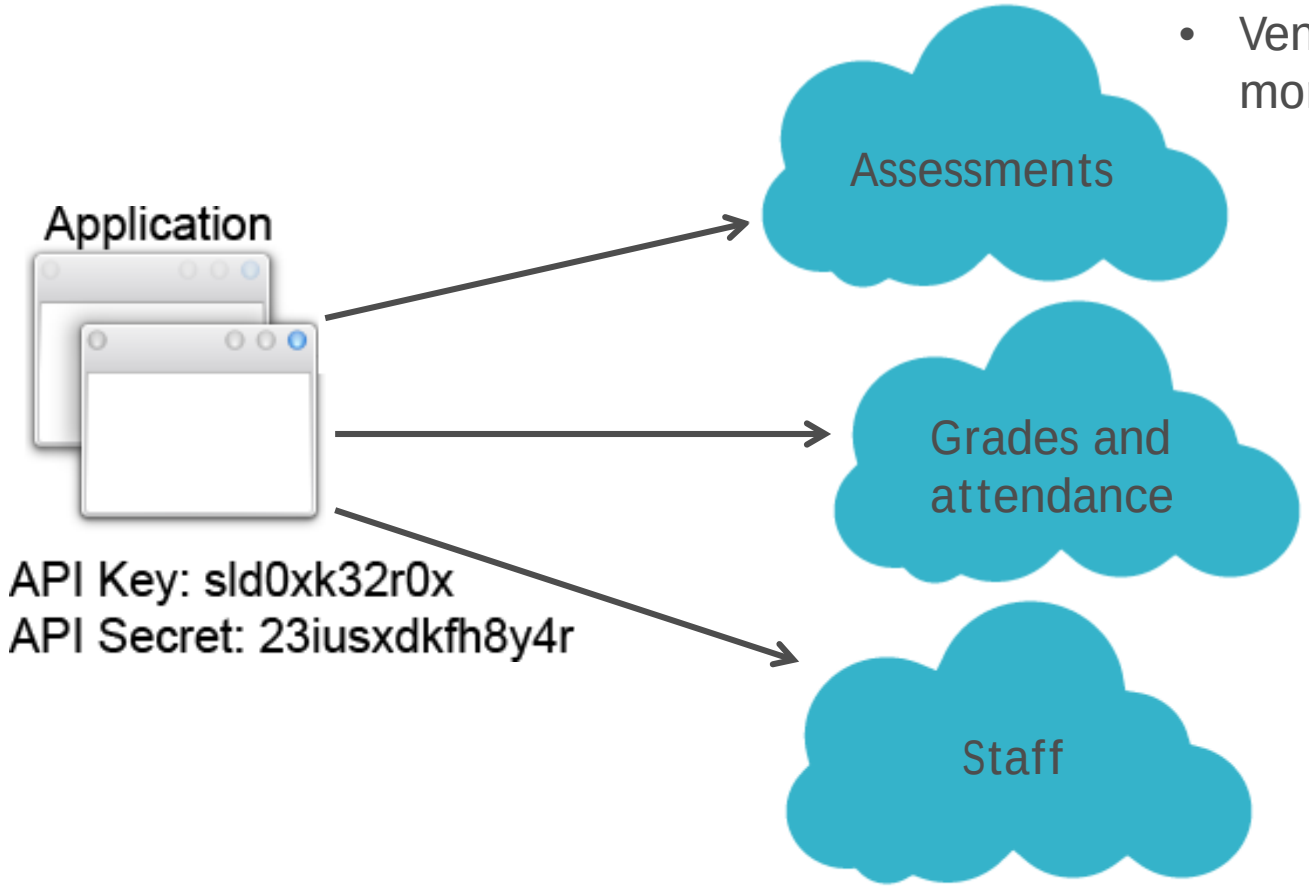
- Vendors have access to one or more districts



Bridges for relationship based security

- staffEducationOrganizationEmploymentAssociation
- staffEducationOrganizationAssignmentAssociation
- studentSchoolAssociation

Resource based security



- Vendors have access to one or more domains

Intersection of resource and relationship security

EdFi_Security database

- Holds application-agnostic security configurations through claims, claim sets, and authorization strategies
- Only used by the Ed-Fi ODS API today, but could be used by other applications moving forward

EdFi_Security answers the following

- What resources are available for a given application (Ed-Fi ODS API)?
- For a given resource and action, what authorization strategy is used?
- For a given claim set, what resources and actions are permitted?

- ** Administration tool coming soon

EdFi_Admin answers the following

- For a given vendor, what applications will they use to access the Ed-Fi ODS API?
- For a given vendor application, what claim set and education organizations are assigned?
- For a given key and secret, what application is that tied to and will that be used to access a “shared instance” or sandbox database?

- ** Shared Instance Administration Portal coming soon

SIS Vendor claim set

- During these labs we will use the SIS Vendor claim set
- Access to students, parents, staff, grades, etc under a given district

Security Use Case: Accessing student data

- Using the “SIS Vendor” claim set as an example
- Granted access to create a student
- Can modify and read only those students
- Can delete any student?!

StudentUSI vs StudentUniqueId

- 99% of the primary keys are natural keys (i.e. no surrogate/identity columns)
- Student, Parent, and Staff have USI as well as UniqueId
- Integration with the Ed-Fi ODS API is done via the UniqueId while the USI is used only as an internal identifier

LAB: Manage student data

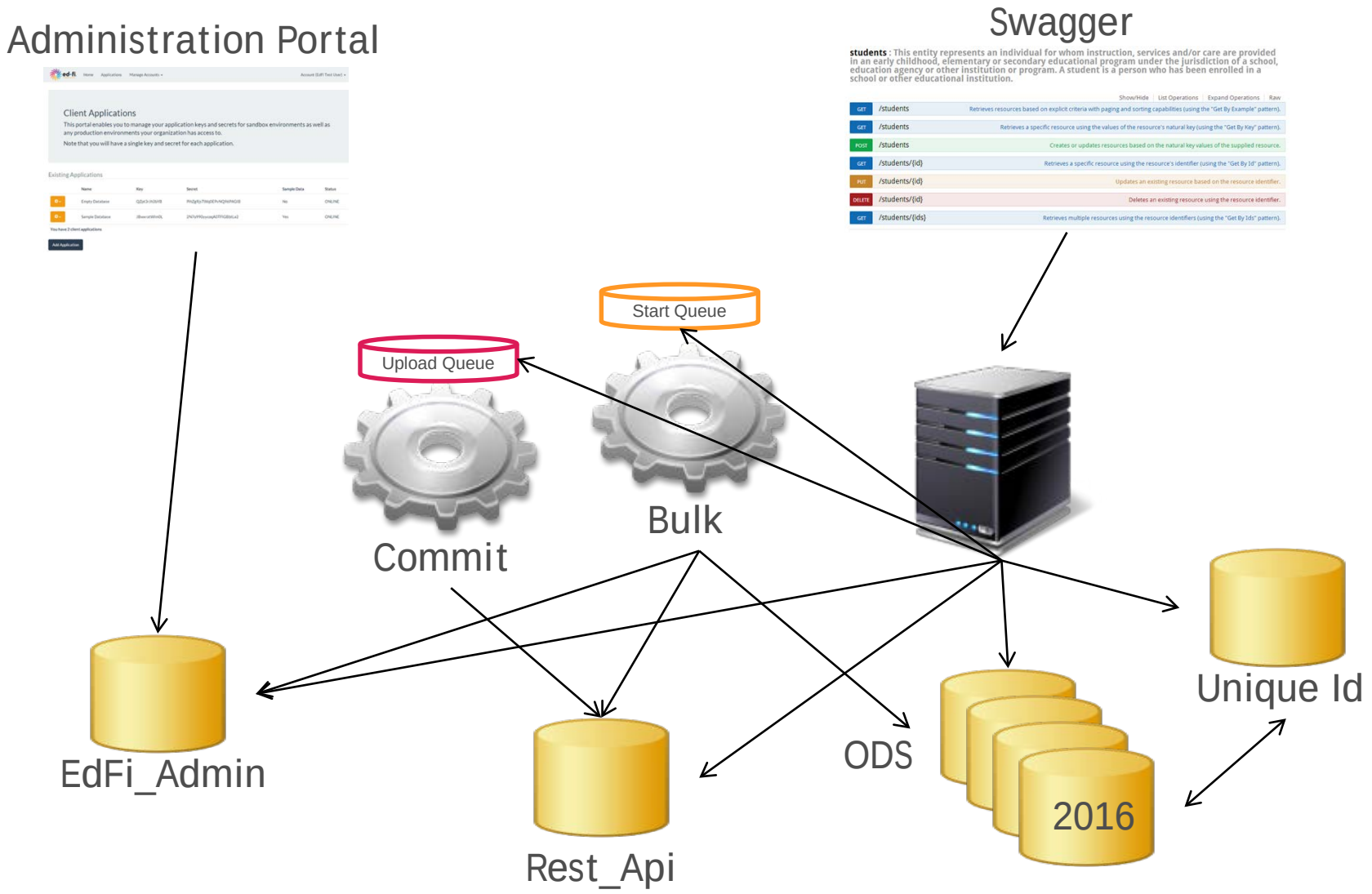
Bulk operations

- A bulk operation may include thousands of records across multiple files to insert and update
- These operations are parsed and executed as individual transactions (i.e. one failure will not fail the batch)
- The errors of the individual transactions are logged
- The error log can be inspected per bulk operation

Executing a bulk operation

- Create the operation
  - POST the bulk operation manifest to */bulkOperations*
  - Obtain an operation Id from the response
- Upload Xml file(s)
  - POST */uploads/{uploadId}/chunk?offset={}&size={}* for each “chunk”
- Commit
  - POST */uploads/{fileId}/commit*
- Check status
  - GET */bulkOperations/{id}*

Bulk operation infrastructure



Transactional vs Bulk

Transactional	Bulk
JSON	Ed-Fi Xml
Synchronous responses	Asynchronous responses
Near real-time as data is changing in the source application	For initial load or periodic refreshes
Full CRUD	Upsert only
Create and retrieve UniqueIds	No ability to create and retrieve UniqueIds

LAB: Submitting a bulk Xml file

Data extensions

- The Ed-Fi data model allows for data extensions
- Additions to existing entities, additions to aggregates, and adding brand new aggregates are all supported

Extensibility standards

- Use natural keys
- Keep the naming conventions
- Use the “extension” schema
- Keep the ODS and Xml Schema in sync
- Add references to the corresponding “edfi” table where applicable
- It’s best to make changes additive and to not add or remove elements
- Use the “extension” suffix when extending a core tables…and cascade deletes

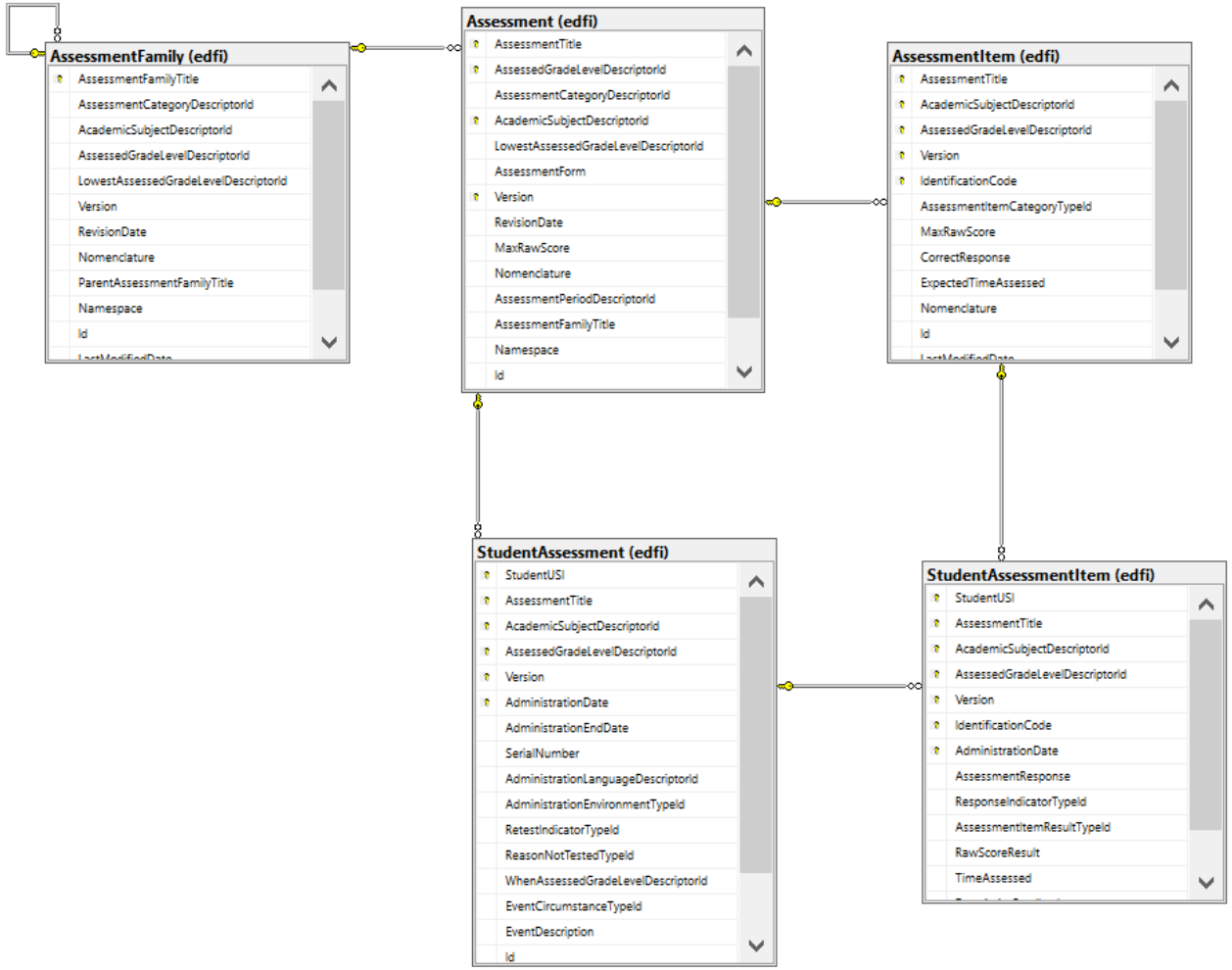
LAB: Extension by extending an existing entity

LAB: Adding a new entity to an existing aggregate

More security with namespaces

- Namespaces are assigned per vendor
- Can be used on certain types of descriptors ("managedDescriptors" in the EdFi_Security database)
- Can also be used to manage local assessment data

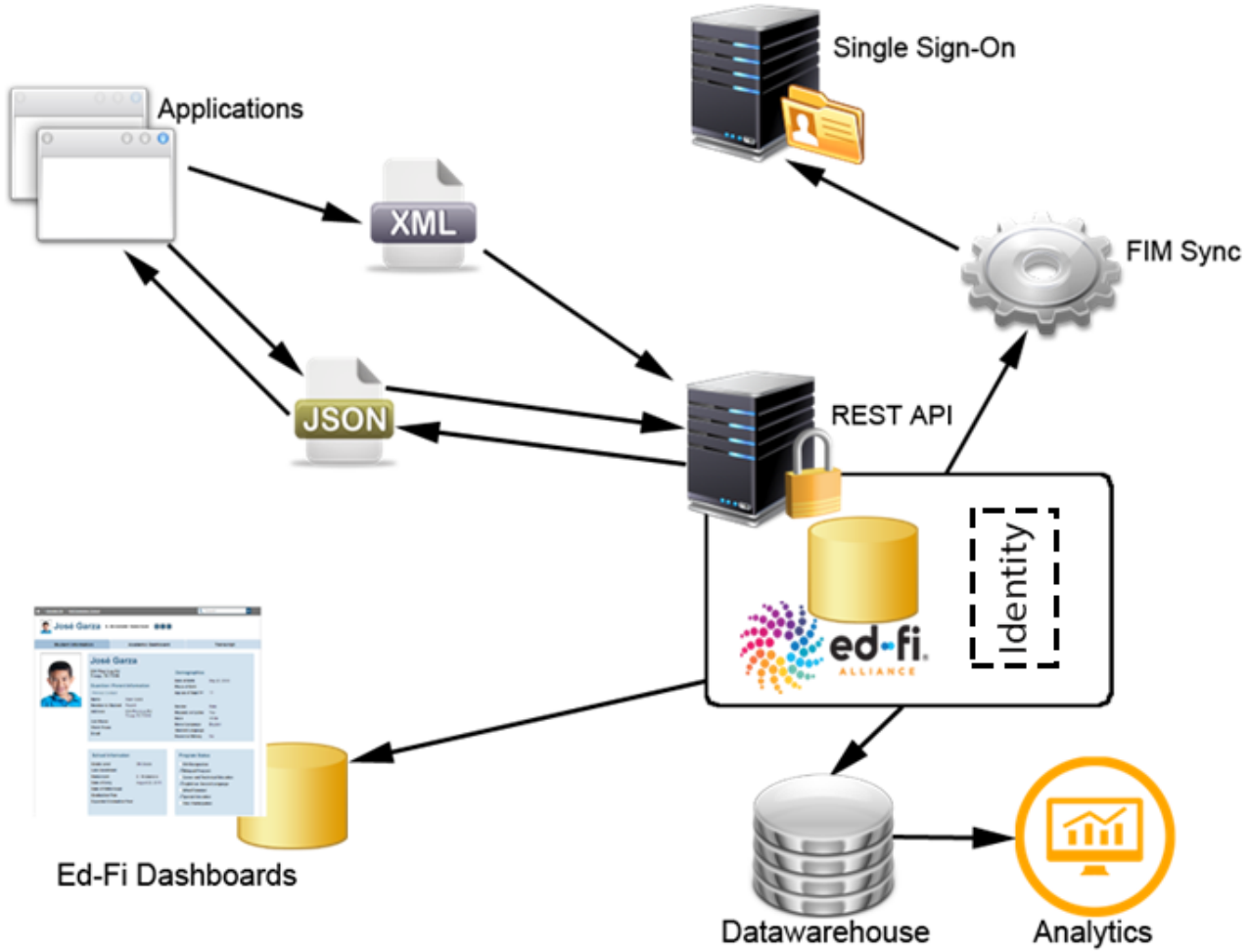
Basic assessment data



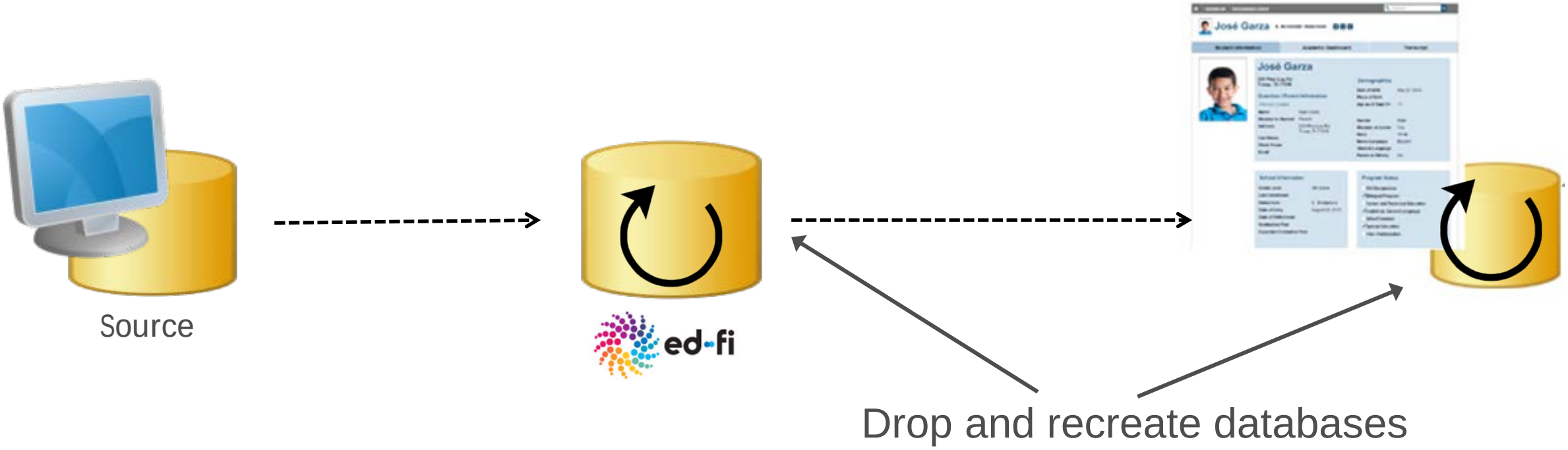
LAB: Managing descriptors

Questions and wrap-up

Full ecosystem



Current dashboard loading process



Dashboard loading process with the Ed-Fi ODS API