



BootCamp – ODS/API 201

ODS/API Agenda

- **Development Environment (1 hour)**
 - Getting Development Environment up and running (30 mins)
 - Walkthrough solution – (30 mins)
 - Admin apps
 - Swagger / API
 - Bulk - Ben
 - Error Logging - Ben
 - Production Deployment – Best Practices (Ben)

Break – 20 mins
- **Security – (1 hour)**
 - Overview
 - Hands-on focused on Claim Sets and Visualization Tool
 - Security Configuration Tool vs Sandbox Administration Tool
 - Profiles + Lab

Break – 20 mins
- **Customization – 30 mins (Ben)**
 - Composites + Lab
 - Identities API – reference MI, WI
 - Other Customization hooks
- **Ed-Fi Implementation (Lessons Learned/Best Practices) – Michael Taylor (Director of Education Technology Services/Indiana University) – 30mins**

Connect to VM - Instructions

Remote Desktop Connection to VM

Instructions for connecting to a Boot Camp Virtual Machine (VM) on Azure

- i. Run your Remote Desktop Protocol (RDP) client. If you are running a Mac, please install the RDP client from your App Store.
 - ii. In the Computer Name text box type in the appropriate computer name based on the computer that is assigned to you.
 - iii. You should be typing the fully qualified name along with the port number in the text box. Below is a list of computer names.
- b. Login Information:
- i. Login: train
 - ii. Password: train01!
 - iii. Domain: Your Computer (Train01 or Train02....)

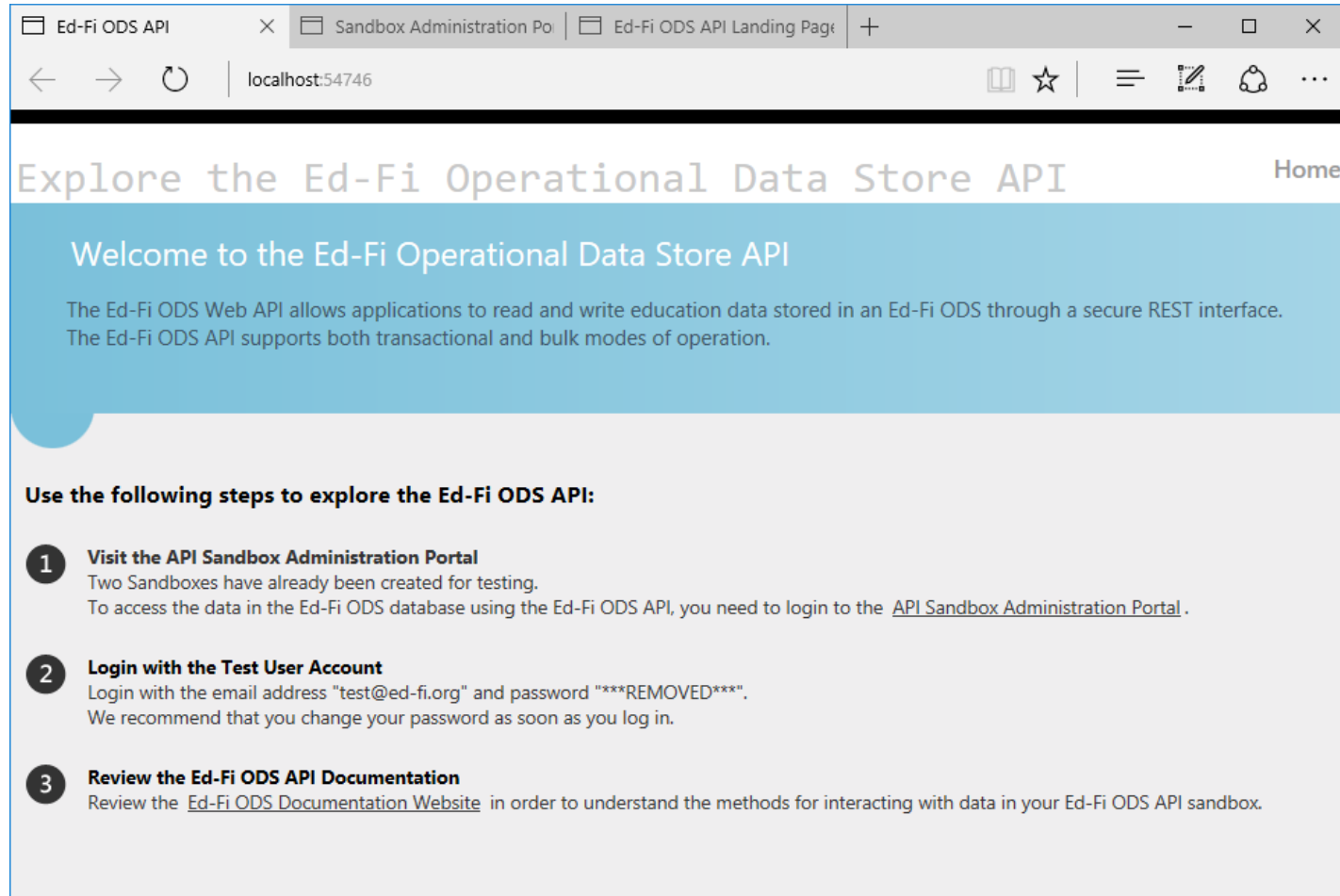
Computer	Computer Name
Train01	summit2017bclab2071283676000.southcentralus.cloudapp.azure.com:59698
Train02	summit2017bclab2071283676000.southcentralus.cloudapp.azure.com:53392
Train03	summit2017bclab2071283676000.southcentralus.cloudapp.azure.com:49318
Train04	summit2017bclab2071283676000.southcentralus.cloudapp.azure.com:58871
Train05	summit2017bclab2071283676000.southcentralus.cloudapp.azure.com:64976
Train06	summit2017bclab2071283676000.southcentralus.cloudapp.azure.com:50151
Train07	summit2017bclab2071283676000.southcentralus.cloudapp.azure.com:63427

Ed-Fi ODS/API – Build/Run Solution


- Start up projects:

Website	Project	URL
Sandbox Administration	EdFi.Ods.Admin.Web	http://localhost:38928/
Ed-Fi ODS API Documentation	EdFi.Ods.SwaggerUI	http://localhost:56641/
Ed-Fi ODS API	EdFi.Ods.WebApi	http://localhost:54746/

ODS/API Home Page





Sandbox Administration Portal

 Home Manage Sandboxes Manage Accounts Account (Test Admin)

Sandbox Administration Portal for the Ed-Fi ODS API

This portal enables you to manage sandboxes for the Ed-Fi ODS API, including the keys and secrets for accessing these sandboxes.

Existing Sandboxes

	Application	Name	Key	Secret	Sample Data	Status
	Default Sandbox Application Sample	Populated Demonstration Sandbox	populatedSandbox	populatedSandboxSecret	Yes	ONLINE
	Default Sandbox Application Minimal	Minimal Demonstration Sandbox	minimalSandbox	minimalSandboxSecret	No	ONLINE

You have 2 sandboxes

Add Sandbox

Sandbox Administration Tool – Lab

- Create a sandbox with or without data
- Change assigned secret
- Get an access token

Swagger UI – Online Documentation

API Section: Resources

Authorize

Available authorizations

Ed-Fi ODS Authorization

Please provide your api key and secret and press authorize to retrieve a session token.

api key: populatedSandbox

api secret: populatedSandboxSecret

Authorize

Cancel

Swagger UI - Individual Resource View

students

This entity represents an individual for whom instruction, services, and/or care are provided in an early childhood, elementary, or secondary educational program under the jurisdiction of a school, education agency or other institution or program. A student is a person who has been enrolled in a school or other educational institution.

GET	/students	Retrieves resources based with paging capabilities (using the "Get All" pattern).
GET	/students	Retrieves resources matching values of an example resource (using the "Get By Example" pattern).
GET	/students	Retrieves a specific resource using the values of the resource's natural key (using the "Get By Key" pattern).
POST	/students	Creates or updates resources based on the natural key values of the supplied resource.
DELETE	/students/{id}	Deletes an existing resource using the resource identifier.
GET	/students/{id}	Retrieves a specific resource using the resource's identifier (using the "Get By Id" pattern).
PUT	/students/{id}	Updates or creates a resource based on the resource identifier.

Swagger UI – Model vs Example

students

This entity represents an individual for whom instruction, services, and/or care are provided in an early childhood, elementary, or secondary educational program under the jurisdiction of a school, education agency or other institution or program. A student is a person who has been enrolled in a school or other educational institution.

GET

/students

Retrieves resources based with paging capabilities (using the "Get All" pattern).

Implementation Notes

This GET operation provides access to resources using the "Get All" pattern. In this version of the API there is support for paging.

Response Class (Status 200)

The matching resource(s) were successfully retrieved. If no instances are found will return an empty collection.

Model

Example Value

```
student {
  id (string): The unique identifier of the resource.,
  studentUniqueId (string): A unique alphanumeric code assigned to a student.,
  personalTitlePrefix (string, optional): A prefix used to denote the title, degree, position, or seniority of the person.,
  firstName (string): A name given to an individual at birth, baptism, or during another naming ceremony, or through legal change.,
  middleName (string, optional): A secondary name given to an individual at birth, baptism, or during another naming ceremony.,
  lastName (string): The name borne in common by members of a family.,
  generationCodeSuffix (string, optional): An appendage, if any, used to denote an individual's generation in his family (e.g., Jr., Sr., III).,
  maidenName (string, optional): The person's maiden name.,
  sexType (string): A person's gender.,
  birthDate (date-time): The month, day, and year on which an individual was born.,
  birthCity (string, optional): The set of elements that capture relevant data regarding a person's birth, including birth date and place of birth.,
  birthStateAbbreviationType (string, optional): The abbreviation for the name of the state (within the United States) or extra-state jurisdiction in which an individual was born.,
  dateEnteredUS (date-time, optional): For students born outside of the U.S., the date the student entered the U.S.,
  multipleBirthStatus (boolean, optional): Indicator of whether the student was born with other siblings (i.e., twins, triplets, etc.),
  profileThumbnail (string, optional): ProfileThumbnail
```

Performing Read and Write – Lab

- GET: Students GetAll
- GET: Students GetByKey
- POST: Student from GetByKey – Change values.
- Error: Retrieve a student with no StudentSchoolAssociation

Bulk

Production Deployment - Best Practices

- Ben to add his slides

Error Logging

ODS/API Agenda

✓ Development Environment (1 hour)

- Getting Development Environment up and running (30 mins)
- Walkthrough solution – (30 mins)
 - Admin apps
 - Swagger / API
 - Bulk - Ben
 - Error Logging - Ben
- Production Deployment – Best Practices (Ben)



Break – 20 mins

• Security – (1 hour)

- Overview
- Hands-on focused on Claim Sets and Visualization Tool
- Security Configuration Tool vs Sandbox Administration Tool
- Profiles + Lab
- Composites + Lab

Break – 20 mins

• Customization – 30 mins (Ben)

- Identities API – reference MI, WI
- Other Customization hooks

• Ed-Fi Implementation (Lessons Learned/Best Practices) – Michael Taylor (Director of Education Technology Services/Indiana University) – 30mins

Security Concepts

- Security: Major concern for all organization. Security in Ed-Fi ODS/API consists of 3 components:
 - Authentication – Identifying who the API caller is. Authentication is granted to client applications.
 - Authorization – Establishing whether a particular user has rights to work with the ODS/API
 - Enforcing Authorization Policies
- Authorization Concepts
 - Resources, Taxonomy, Claim Sets, Authorization Strategies
- Profiles

Authentication: OAuth2.0 – Abstract Protocol Flow

Step A: Client requests authorization from the Resource Owner

Step B: Client receives Authorization grant. This could be one of 4 grant types.

Step C & D : Client presents the grant authenticating itself and gets the access token.

Step E: Submits the access token on every request.

Step F: Resource server validates the access token and serves the request.

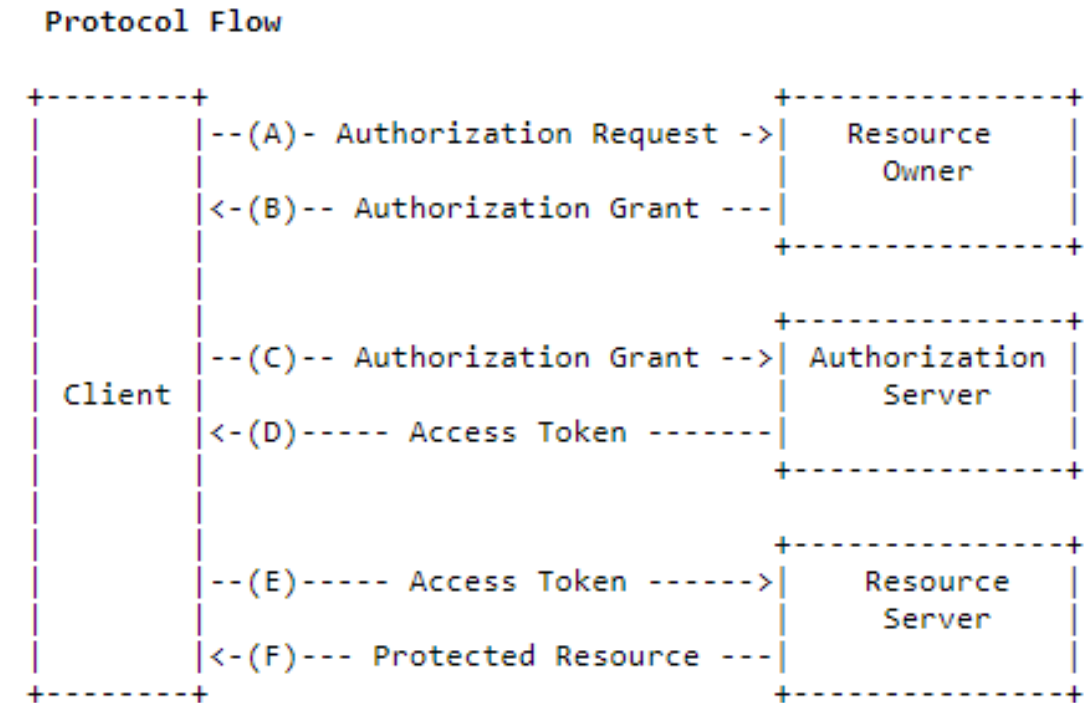


Figure 1: Abstract Protocol Flow

Authentication

- The ODS / API uses 2-legged OAuth2 for authentication
 - Before accessing the resources in an ODS / API platform, client applications need to obtain an access token from the API platform
 - This access token is validated on every call made to the API as a representation of the client's application key and secret.
 - Clients and platforms talk to each other over HTTPS



Authentication

- Note: The implementation of OAuth2 is changing in Ed-Fi ODS/API v3.0
- Attend “205 – Ed-Fi ODS/API – New Capabilities” on Weds morning for details

Security Configuration Tool – Lab

Purpose: Manages access rights to vendor applications.

Security Admin vs Security Configuration

Security Admin: Used for issuing credentials for Sandbox environment

Security Configuration Tool: Used for issuing credentials for Production environment

1. Change the connection string in Security Configuration Tool - Web.config to point to the correct Ed-Fi-ODS-XXX database.
1. Create a folder smtp under c:\temp\
 2. Add Vendor, Edit Vendor, Delete Vendor
 3. Add Application, Edit Application, Delete Application
 4. Generate Application Key and Secret
 5. Key Retrieval Tool

Authorization

- Ed-Fi ODS/API allows only authorized users/applications to access data
- Capable of giving access rights to individual resources
- Two-pronged authorization approach
 - Relationship based – Vendors are associated with one or more local education agencies
 - Resource based – Vendors are given right to perform actions against specific resources.

Example:

SIS can be authorized to read and write against students and staff data associated with a LEA that they serve.

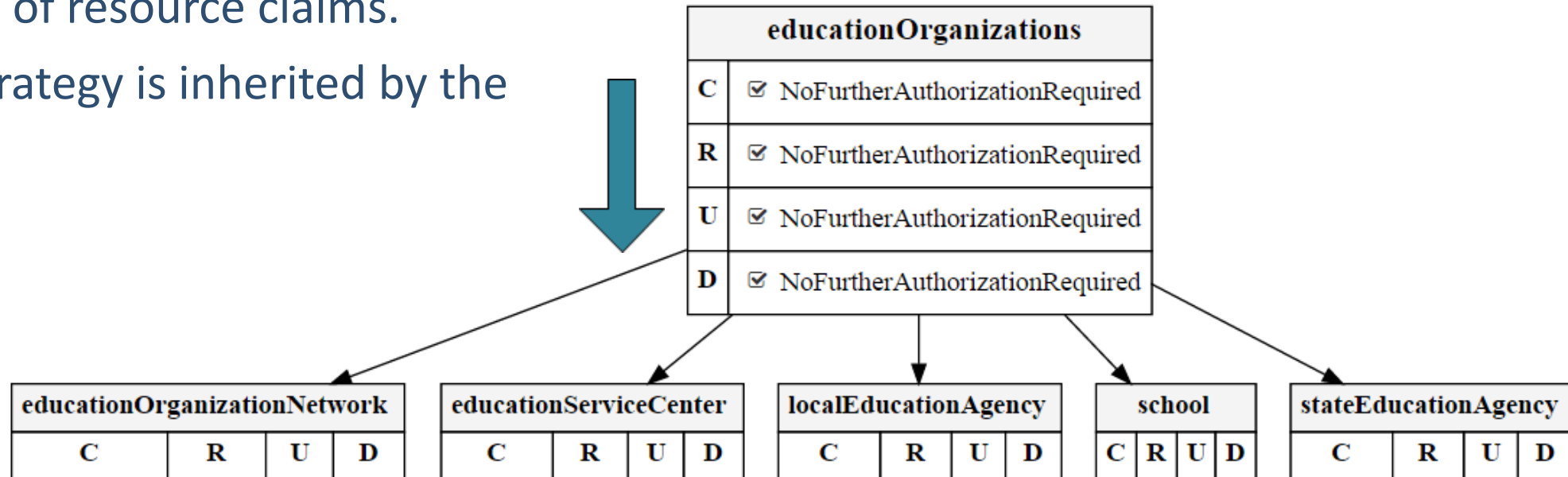
Assessment application is authorized to read and write assessment metadata and assessments

Authorization - Concepts

- Resources – The endpoints exposed by the API (this is a domain aggregate with a root element exposed as an API resource)
- Taxonomy – The organization of Resources into a hierarchy
- Claim Sets – A named set of claims given to an API consumer (aka Role)
- Authorization Strategies – Domain-specific approaches for authorizing requests for specific Resources.

Authorization - Taxonomy

- The surface area of the API is large
 - 82 “Domain” Resources
 - 181 Types/Descriptors Resources
- Logical grouping of resource claims.
- Authorization strategy is inherited by the children.



Authorization - Taxonomy

- Out-of-the-Box Higher Order Resource Claims
 - people – Students, Staff, Parents
 - educationOrganizations – SEAs, LEAs, ESCs, Schools, etc.
 - assessmentMetadata – Assessments, Assessment Families, etc.
 - educationStandards – Learning Objectives and Standards
 - relationshipBasedData – Other API Resources with relationships to EdOrgs or People
 - primaryRelationships – Association Resources that must be established before access to relationshipBasedData can be granted
 - systemDescriptors – Descriptors that are managed by the API host
 - managedDescriptors – Descriptors that are managed by the API consumers
 - types – All standard Ed-Fi “enumeration” Resources (corresponding to the XML Schema)

Authorization – Claim Sets

- Claim – Authorization decisions are claims based. Each consumer application's key will be associated with a claim.

Example:

SIS vendor token will be given claim to read Ed-Org

System Administrator will be able to read/write/delete Ed-Org

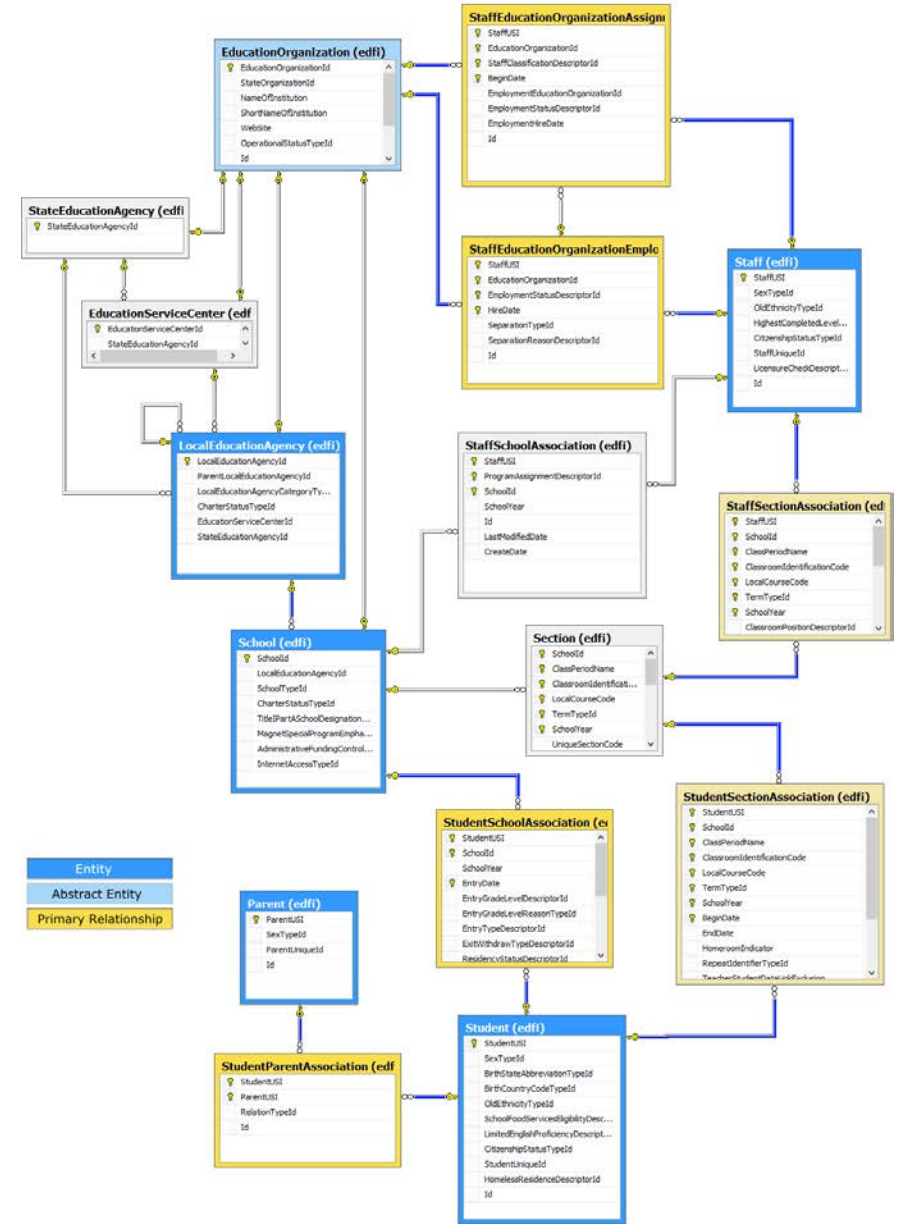
- Claim Set – A named collection of Claims assigned to a caller (aka “Role”)
- Resource Claims in the Ed-Fi ODS API are logical, not physical
 - Claims aren't issued and transmitted between parties (e.g. SAML, JWT)
 - Claims are “issued” based on the EdFi_Security database and the caller's access token
 - Cached locally in each Ed-Fi ODS API host process until token expires

Authorization – Strategies

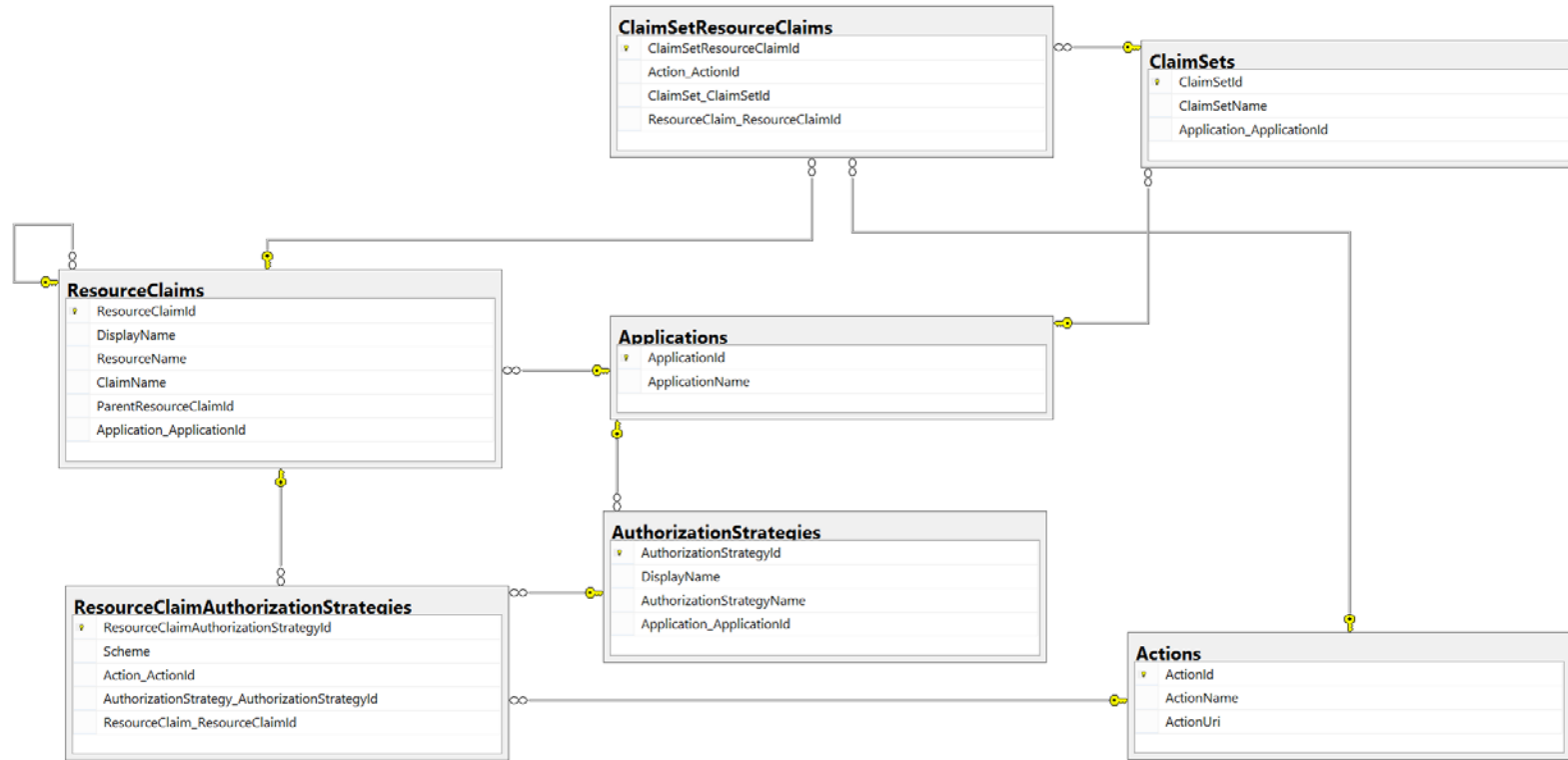
- Enable domain-specific logic to be applied to authorization decisions once simple CRUD checks have passed.
 - NoFurtherAuthorizationRequired – No additional logic is applied
 - NamespaceBased – Authorization is based on ownership via Namespace
 - AssessmentMetadata – Authorization is based on ownership via Namespace of the related Assessment or Assessment Family
 - RelationshipsWithEdOrgAndPeople – Authorization is based on relationships the caller has with People and Education Organizations
 - Primary relationship tables in Ed-Fi ODS:

Authorization - Strategies

- RelationshipsWithEdOrgsAndPeople
- Example: SIS Vendor accessing a Student
 - Relationships from the SIS Vendor's associated LEA to the Student are verified in the Ed-Fi ODS
 - In other words...
LEA → School → StudentSchoolAssociation → Student



Authorization – Ed-Fi Security Database



Configure Claim Sets - Lab

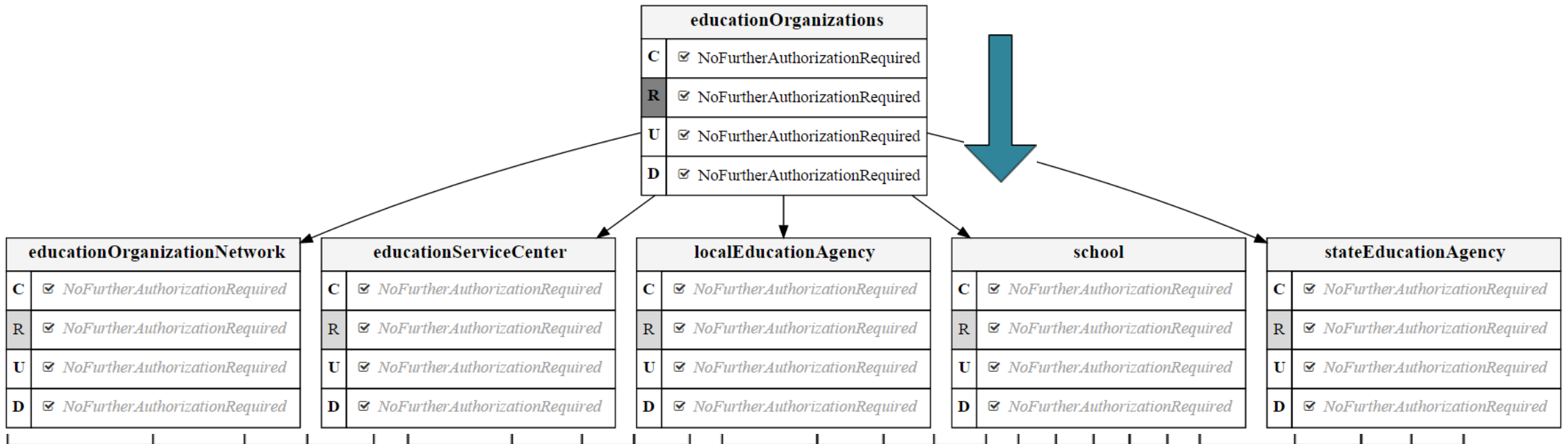
- Add a Resource and Action to a Claim Set - SQL - Hands on
- Add a Resource and Action to an Authorization Strategy – SQL Hands on
- View Security Configuration – SQL - Hands on

Security Visualization Tool – Hands on

- Build Solution \Ed-Fi-ODS\Utilities\GenerateSecurityGraphs\GenerateSecurityGraphs\bin\Debug
- Run GenerateSecurityGraphs.exe – View Parameters
- GenerateSecurityGraphs.exe -o "C:\graphs" -f
- Review Output – Education Organization
 - Ed-Fi Sandbox ClaimSet
 - SIS Vendor ClaimSet
- Review Output – Managed Descriptors , System Descriptors for SIS Vendor ClaimSet

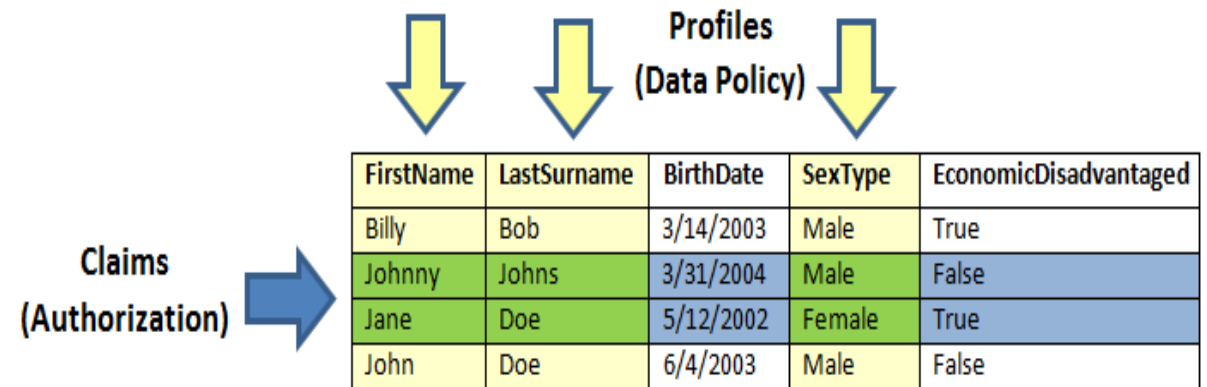
Authorization – View Claim Set – SIS Vendor

- Claim Sets enable the assignment of CRUD permissions to Resources
- Example: Education Organizations
- Claim Set: SIS Vendor (Read-only)



Profiles - Definition

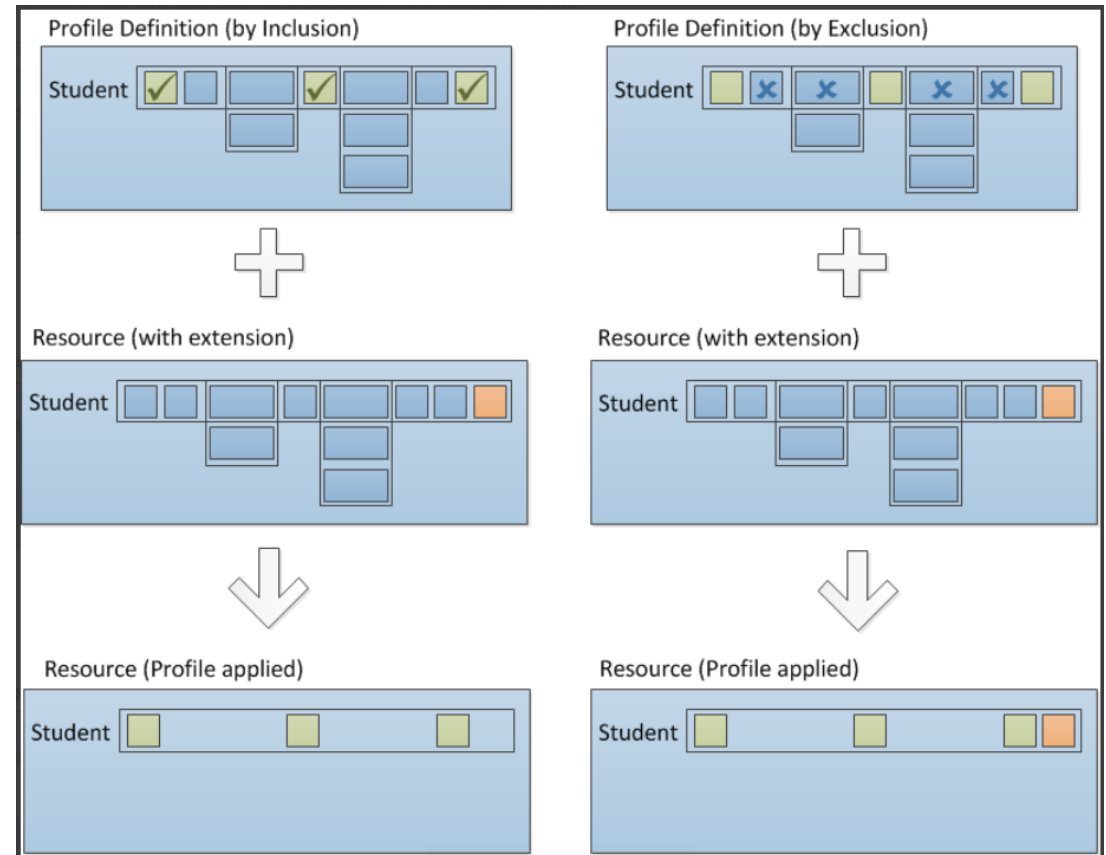
- Define what data elements are available for reading and writing.
- Is a data policy for a particular resource.
- Policy is set for all resources(resources/types/properties) based on inclusion or exclusion method
- Assigned to client applications



Profiles – Include only vs Exclude only

Include only: Process by inclusion. If a model is extended or if a data standard is upgraded, the data elements will NOT be included.

Exclude only: Process by exclusion. If a model is extended, then the new model will be included in the profile definition.



Profiles - Example

- Profile definition is expressed in XML and is used by the code generation process in VS to generate the API artifacts.
- Multiple Resources Example (School and Student)

```
<!-- Multiple resources -->
<Profile name="Test-Profile-Student-and-School-Include-All">
  <Resource name="School">
    <ReadContentType memberSelection="IncludeAll" />
    <WriteContentType memberSelection="IncludeAll" />
  </Resource>
  <Resource name="Student">
    <ReadContentType memberSelection="IncludeAll" />
    <WriteContentType memberSelection="IncludeAll" />
  </Resource>
</Profile>
```

Profiles – Hands on

- Download the template and run the template

ODS/API Agenda

✓ Development Environment (1 hour)

- Getting Development Environment up and running (30 mins)
- Walkthrough solution – (30 mins)
 - Admin apps
 - Swagger / API
 - Bulk - Ben
 - Error Logging - Ben
- Production Deployment – Best Practices (Ben)

Break – 20 mins

✓ Security – (1 hour)

- Overview
- Hands-on focused on Claim Sets and Visualization Tool
- Security Configuration Tool vs Sandbox Administration Tool
- Profiles + Lab
- Composites + Lab

Break – 20 mins

• Customization – 30 mins (Ben)

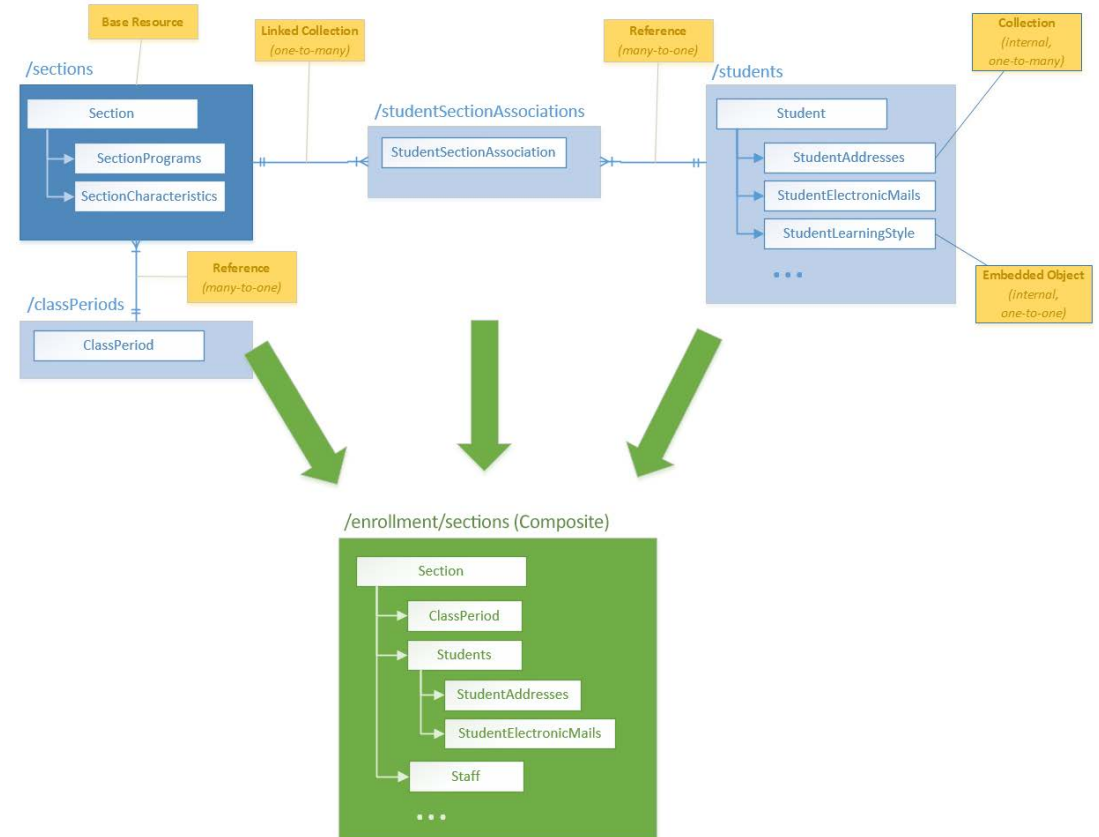
- Identities API – reference MI, WI
- Other Customization hooks

• Ed-Fi Implementation (Lessons Learned/Best Practices) – Michael Taylor (Director of Education Technology Services/Indiana University) – 30mins



Composites

- Compose multiple data elements across multiple discrete entities in the Ed-Fi data model.
- Transactional synchronization vs on demand transfer of data
- Target system initiating transfer of data from a source system and controls the data flow
- Base resource is identified first that will serve the source of data



Composites – Hands on

Identities API - Ben

Customization Hooks - Ben

Customization Hooks

Ed-Fi Implementation – Guest Speaker