
1EDTECH
Learning Impact
Accelerating Innovation
2025 WE'RE ON THE RIGHT TRACK!
Welcome to Indianapolis!

Rostering Pre-Conference Working Group

1EdTech and Ed-Fi Alliance

Agenda

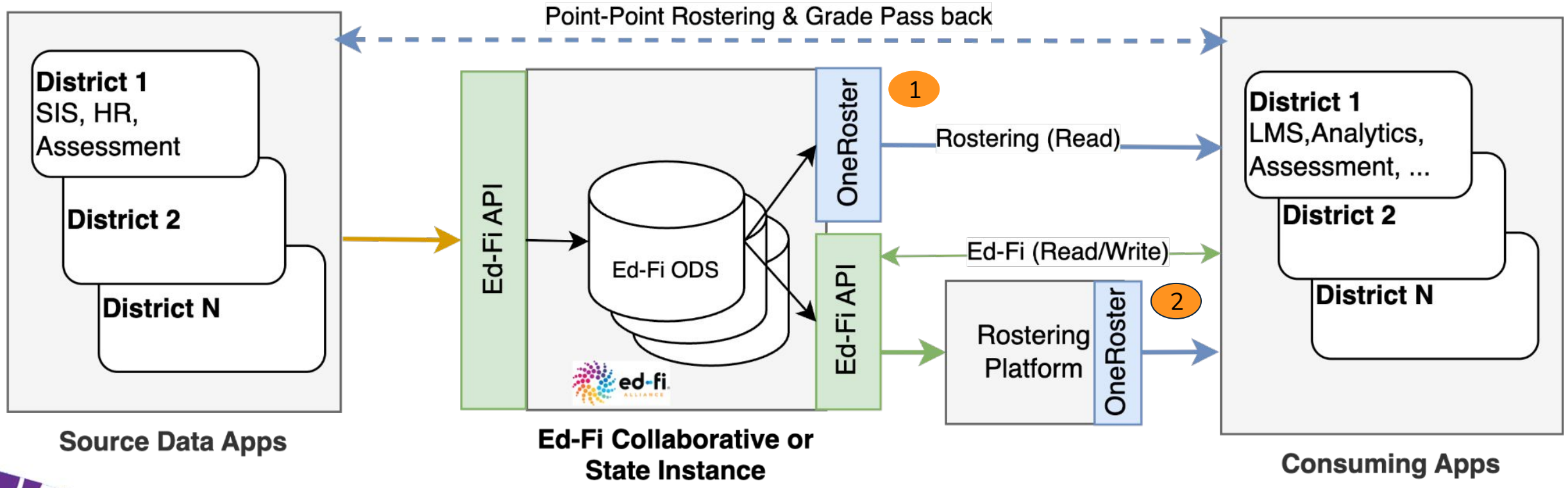
1. Summarize activities and focus coming out of Ed-Fi Tech Congress (5 mins)
2. Update and discussion on establishing field pilots (~60 mins)
   - Review and discussion of “pilot questions to answer”
   - Share details and status of planned pilot in SC
   - Discuss/determine if additional pilots are needed at this stage
3. Looking Forward - Expansion Planning (~45 mins)
4. Wrap up (10 mins)

Update and discussion on establishing field pilots

1. Review and discussion of “pilot questions to answer”
2. Share details and status of planned pilot in SC
3. Discuss/determine if additional pilots are needed at this stage

Proposed Architecture

The proposed architecture comprises a combination of services from both standards



Initial Pilots - What questions are we trying to answer?

From May 13th Pilot Coordination meeting with Steering Commitee MSP's

- Performance related to implementation strategy
- Handling custom metadata
  - Classlink has list of 36+ most commonly used metadata and can share with group
  - CEDS vocabulary possibility?
- Authorization strategy
- Feasibility of standard implementation within Ed-Fi stack
- What UI would needed to be in order to manage solutions from these answers (e.g., authorization, claimsets)
- Informing the design of a OneRoster profile for Ed-Fi
- Feedback on differences between OneRoster 1.1 and 1.2

- How identities are handled across Ed-Fi and OneRoster
  - status and dateModified are required in OneRoster spec that Ed-Fi does not provide
- Ordering of results
- 'Day 0' rostering requirement

- Pilot Agencies
  - Before & After: What did it take before vs current roster
- Consuming apps
  - Before and After: what did it take before vs after to roster

Initial Pilots - What questions are we trying to answer?

Next Steps:

- Refine the list of pilot questions into well-structured, answerable formats.
- Conduct 1:1 check-ins with each MSP technical lead to clarify what is already known, identify open questions, and avoid duplicative efforts across pilots.
- Key focus areas include: authorization strategies, handling of identities, feasibility of metadata mapping, and Day 0 rostering needs

Field Implementations - Detailed QA

OneRoster & Ed-Fi Implementation Comparison (1/2)

Criteria	Michigan Data Hub	ClassLink (Delaware)	EdGraph	Education Analytics
General Information				
OneRoster Version	1.1	1.1, 1.2 *(Clients choose; Connector outputs 1.1)*	1.1	1.1
Implementation Type	REST API Provider *(+ Flex Exporter for CSV)*	CSV Export & REST API	CSV Export & REST API *(+ PUT support)*	CSV Export only
1EdTech Certification	✓ REST Provider 1.1	✓ REST Provider 1.1, 1.2	✓ REST Provider 1.1 ✓ CSV Export 1.1	✗ No certification
Ed-Fi Version	v6.2 (DS 4.0)	v6.2 (DS 4.0)	v6.2 (DS 4.0) v7.2 (DS 5.1)	v7.1 (DS 5.0)
Production Use	376 active districts	~40 districts	100's of districts indirectly *(vendor-owned districts)*	*TBD*
Vendor Adoption	CSV: 77 vendors API: 106 vendors	CSV: ~ 10% of vendors API: ~ 90% of vendors	<5 vendors each (CSV & API)	*TBD*
Architecture				
Integration Pattern	Ed-Fi as central hub *(Direct from ODS)*	Parallel implementations with middleware translation	Parallel implementations with middleware translation	Ed-Fi as central hub + downstream data store *(ELT approach)*
Multi-tenancy	Separate Ed-Fi ODS per tenant	Tenant isolation at downstream DB	Tenant isolation at downstream DB*(Single API router)*	Tenant isolation at downstream DB

OneRoster & Ed-Fi Implementation Comparison (2/2)

Criteria	Michigan Data Hub	ClassLink (Delaware)	EdGraph	Education Analytics
Data Mapping				
Identifier Mapping	Ed-Fi IDs as source	Configurable Mapping layer (with UX)	Configurable Mapping layer	Hash of Ed-Fi natural keys
Ed-Fi Descriptor Mapping	Custom interop.OperationalContextSupport_Ext table + hard coding	*TBD*	*TBD*	Static crosswalk tables
Status Field	Statically mapped to 'active'	Dynamic based on dates *(active/tobedeleted)*	Statically mapped to 'active'	Not provided
Date Modified Field	'LastModifiedDate' from Ed-Fi tables	Last RosterServer update *(Not used by clients)*	*TBD*	Not provided
Gaps & Future Needs				
Key Gaps	• OneRoster v1.2 support	• Ed-Fi change queries • User (OR) & Attendance (Ed-Fi) alignment	• OR v1.2 (low demand)	• REST API support • Real-time capabilities
Priority needs from 1EdTech/Ed-Fi collaboration	OneRoster API support in Ed-Fi DMS stack to replace current solution	• Alignment between OR users and Ed-Fi attendance data	• Common OAuth2/OIDC • Higher Ed expansion • URL refs to Ed-Fi resources	*TBD*

Key questions to answer/inform in pilot(s)

1. Segmentation of OneRoster data
2. Mapping Ed-Fi descriptors to OneRoster enumerations
3. Identifier generation
4. Mapping to Academic Sessions
5. OneRoster extensions
6. Authorization alignment

Pilot questions to answer/inform
Segmentation of OneRoster Data?

Roster platforms surface separate OR end-points for consumption by districts

Understand why - to determine if this pattern is needed to inform an ‘Ed-Fi Core’ implementation

Pilot questions to answer/inform **Mapping Ed-Fi descriptors to OneRoster enumerations**

- Propose adopting Ed-Fi DescriptorMapping as the expected way to map Ed-Fi Descriptors to OneRoster enumerations
- See Ed-Fi Descriptor Mappings as Data → new in Ed-Fi DS 4.0

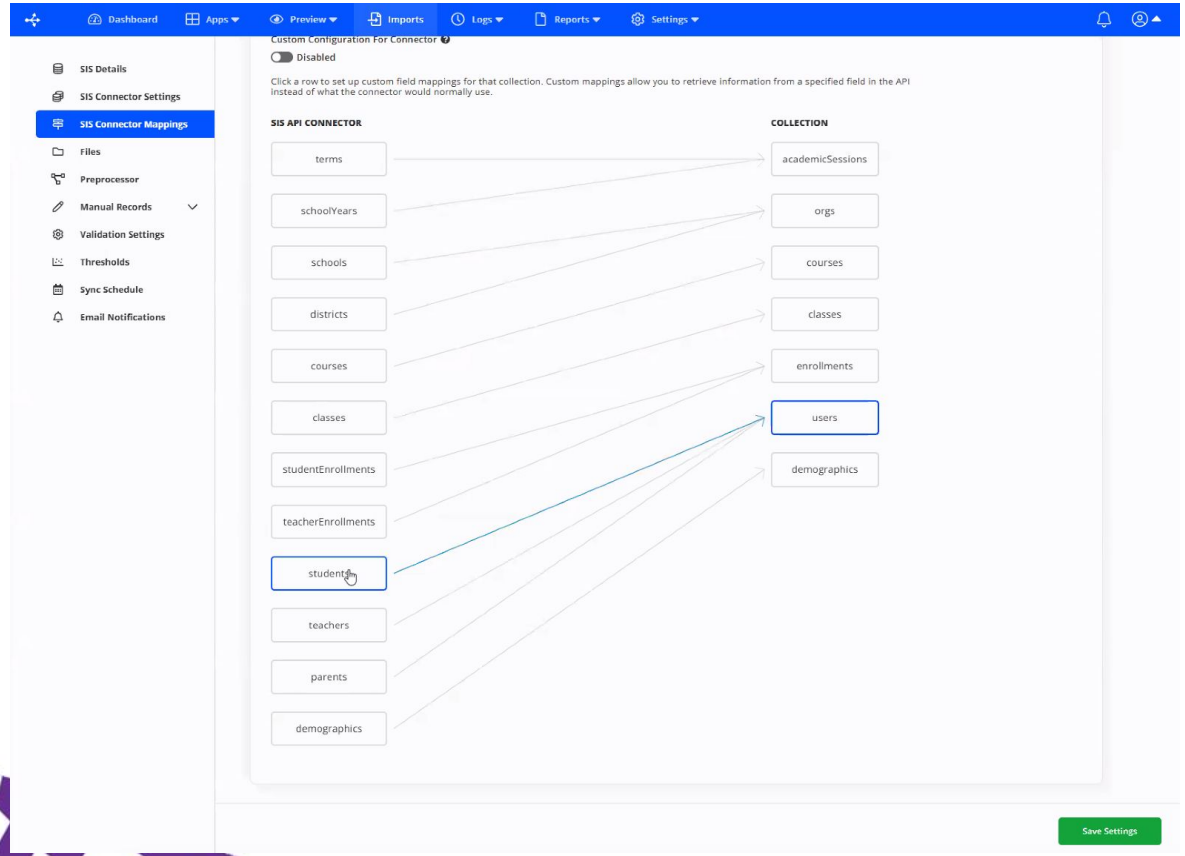
Pilot questions to answer/inform

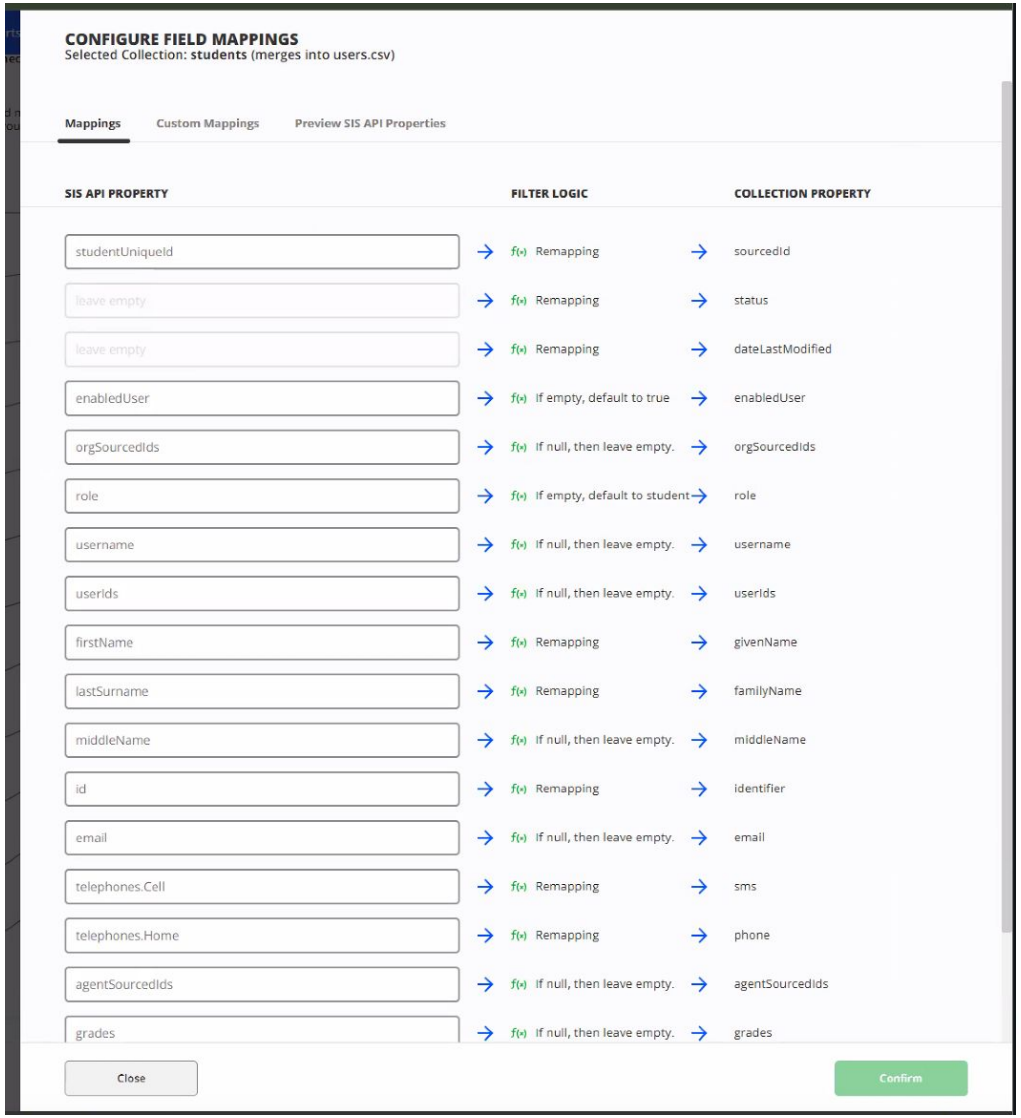
Identifier Generation

Current implementations adopt different approaches to map to SourcedId. What approach should a canonical Ed-Fi implementation adopt?

MI DataHub	ClassLink & EdGraph	EA CSV Export
For People (Students, Staff, Parents): ● Respective UniqueId fields **For Other Entities:** ● Classes/Courses: Ed-Fi GUIDs ● Organizations: EducationOrganizationId ● Enrollments: Association table GUIDs ● Academic Sessions: Simple sessions use Session GUIDs; grading periods use composite IDs (Session GUID + Grading Period GUID)	Map Ed-Fi identifiers to sourcedId by default, but can configure/override during synchronization Requires configuration/mapping UX	Uses MD5 hashes of natural keys to generate consistent IDs across different implementations Examples: ● class: local_course_code + school_id + section_id + session_name ● student enrollment: student_unique_id + local_course_code + school_id + section_id + session_name + begin_date The original natural key string is preserved as an extension called **metadata.edu.natural_key**

Field Mapping UX (Class Link)





Pilot questions to answer/inform

Mapping to Academic Sessions

Current implementations adopt different approaches. What approach should a canonical Ed-Fi implementation adopt?

MI DataHub	ClassLink	EdGraph	EA CSV Export
Transforms Ed-Fi's academic calendar data into OneRoster's hierarchical session structure.	TBD	TBD	Maps school-specific Ed-Fi sessions to district-wide OneRoster sessions

Pilot questions to answer/inform **OneRoster Extensions**

- Links back to initial community working session - see Slide 27: Three Options
- Informed by extension alignment work done by ClassLink

Possible directions:

- Educate client applications to call Ed-Fi API's for extended data found in the Ed-Fi model

  OR

- Coordinate OneRoster extensions, and hydrate these from Ed-Fi model

  OR

- Do nothing in this phase of work

Pilot questions to answer/inform **Authorization Alignment**

- See discussion slides from Tech Congress session

Proposed Rostering Pilot in SC

Proposed scope: Implement a lightweight, OneRoster-compliant API on top of materialized views in the Ed-Fi ODS (informed/inspired by MI DataHub solution)

This effort would include*:

- Development of materialized views in the Ed-Fi ODS (PostgreSQL) aligned to the OneRoster v1.2 structure
- Implementation of refresh logic (scheduled or trigger-based) for those views
- A minimal OneRoster 1.2 REST API supporting filtering, field selection, ordering, and pagination
- Pilot testing against a certified OneRoster client (e.g., Schoology)

Assumptions

- The approach assumes the Ed-Fi platform host will populate the necessary Descriptor Mappings for OneRoster support. (See Ed-Fi Descriptor Mappings as Data → new in Ed-Fi DS 4.0)
- Ed-Fi ODS environment (v7.1 / DS 5.0)

This phase would not include:

- Support for MSSQL
- Delta rostering
- *Defining or reengineering API-level authorization to align with Ed-Fi's security model would require additional design effort.

Discuss/determine if additional pilots are needed at this stage

Proposed Pilot Timeline

May - Prep	June - Kickoff, testing	July - Testing and review	August - Wrap and report
Confirm pilot sites, participating MSPs, agency leads; finalize scope and success goals Technical setup and architecture alignment	Share out at Learning Impact (initial integration, even partial) Early pilot build demos	Active pilot data exchanges running between OneRoster and Ed-Fi Iterative feedback + issue fixing Check in to share lessons learned, flag risks, adjust approach	Finalize results, draft summary, prepare report-out Expansion recommendations

Looking Forward – Expansion Planning

Roadmap Exercise

Now (2)

Read-only OneRoster endpoint on Ed-Fi

Impact ●●●●●

Effort ●●●○○

Based on pilots, establish recommended pattern(s) for OneRoster integrations

Impact ●●●●●

Effort ●●●○○

Next (5)

Build recommended pattern(s) into core Ed-Fi specs / practices

Impact ●●●●●

Effort ●●●●○

Deprecate the Ed-Fi Rostering API

Impact ●●●●○

Effort ●●●●○

Build a OneRoster Profile for Ed-Fi

Impact ●●●●○

Effort ●●●●○

Add async bindings to OneRoster in Ed-Fi

Implement GRPC-based event-driven rostering updates via the OneRoster ednpoint

Impact ●●●●●

Effort ●●●○○

Implement patterns for "Writeable" OneRoster Gradebook service

Impact ●●●●●

Effort ●●●●○

Later (4)

Implement oath-based security framework across OneRoster and Ed-Fi

Impact ●●●●●

Effort ●●●●●

Translate Ed-Fi / OneRoster "meta-specification" into 1EdTech model processing system

Impact ●●●●○

Effort ●●●●●

Generate diagnostic and certification tools for Ed-Fi / OneRoster implementations

Impact ●●●●●

Effort ●●●●○

UniformID implementation for Ed-Fi (Users? Courses? Competencies? What else?)

Impact ●●●●○

Effort ●●●●○

Much Later (4)

Integration of Edu-API into EdFi

Impact ●●●○○

Effort ●●●●●

Assessment registration for Ed-Fi with OneRoster

Impact ●●●●●

Effort ●●●●○

Generation of Comprehensive Learner Records (CLR) from Ed-Fi

Impact ●●●●●

Effort ●●●●●

Integrate with CASE standards to associate with courses / classes / assessments in Ed-Fi

Impact ●●●●●

Effort ●●●●●

Major Milestones and Deliverables

Plan Jan-Feb	Pilot/Field Prep Mar-May	Field Validation Jun-Aug	Expansion Sep-Oct
• Detailed project plans • Governance groups formed • Gap analysis • Architecture flow diagrams • Working group established	• Spec. analysis • Element mapping documents • Sample data sets • Auth and test strategy • MVP reference architecture • Field validation begins	• Pilots & field analysis • Technical results report • Expansion planning • Secure commitments from vendors & agencies	• Vendor and agency implementation support • Outcomes measurement

Backup – Slides from Prior Meetings

- *May 13th Pilot Coordination Meeting with MSP's*

1EdTech & Ed-Fi Community Working Session - October 2024

Market Problem

It is unclear how OneRoster and the Ed-Fi Student APIs should work together.

In field work, we see many custom metadata fields being added to OneRoster, and these are essentially creating standards that are ad hoc, vendor-specific, and at worst bilateral (agreements between 1 vendor and 1 agency). This creates extra work for vendors (i.e., supporting non-standard fields) and agencies (e.g., going through configuration screens to add these). It also causes confusion and extra work for aggregators relying on OneRoster for use cases like data analysis (i.e., the need to map and understand these diverse, ad hoc semantics).

The lack of coordination also means that vendors opting to use Ed-Fi are dealing with more complexity than needed at the start of projects, when OneRoster's simplified schema is a much easier starting point.

•It also means that some vendors are likely having to use both specifications, which duplicates work that could be coordinated.

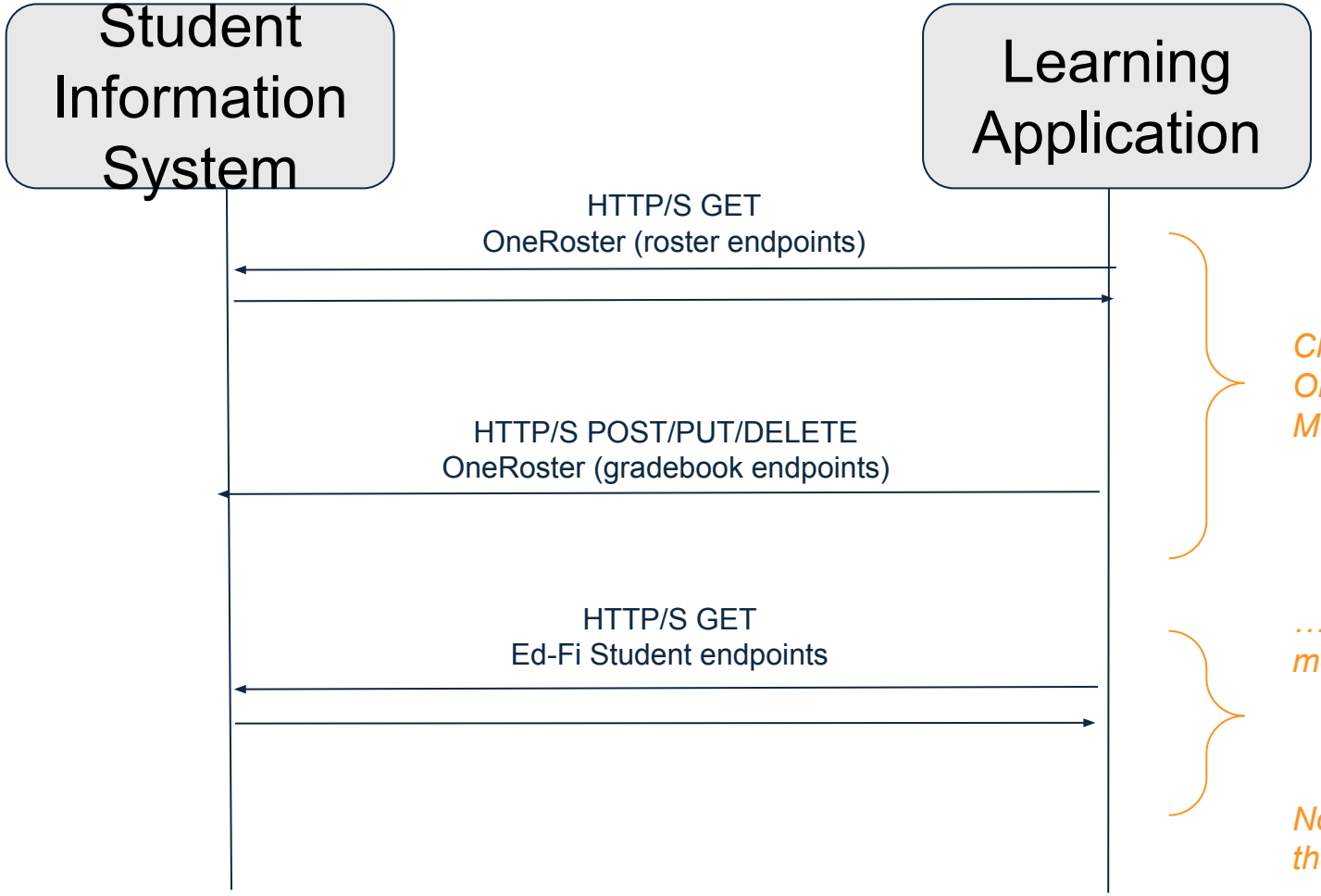
Three Options

The 1EdTech and Ed-Fi teams put our heads together on October 3 in San Antonio to get a few options on the table. We saw three, but welcome other options. These options are also not necessarily mutually exclusive.

Option	Vision	Pros	Cons
Expand OneRoster	Grow OneRoster leveraging Ed-Fi's existing elements and data dictionary	- Simple - Leverages OneRoster scale as a very widely deployed consumption API	- Will always be a partial solution to the problem, i.e., if you add 30 elements there will be a 31st, 32nd, etc. - The simplified schema of OneRoster will emerge as a key drawback at scale.
Create a spec to embed Ed-Fi elements in OneRoster	Allow OneRoster API payloads to be augmented with additional Ed-Fi data elements.	Retains semantics and elements of each spec	- APIs could be awkward - starts to break REST API conventions (e.g., how do you query for behavior data within OneRoster calls?)
Coordinate OneRoster and Ed-Fi Student APIs	Coordinate OneRoster and Ed-Fi Student APIs to develop a unified developer experience in using the APIs (e.g., common AuthN/Z, REST standards, etc.)	- The MVP can be a best practice document (i.e., no spec changes necessary) - Envisions a future state of a single API surface - Embraces REST API conventions	

Seemed like the most promising direction - further slides explore this from various stakeholder perspectives.

Student Information System Vendor View



Change nothing about what is currently working with OneRoster, except to try to reduce custom Metadata…

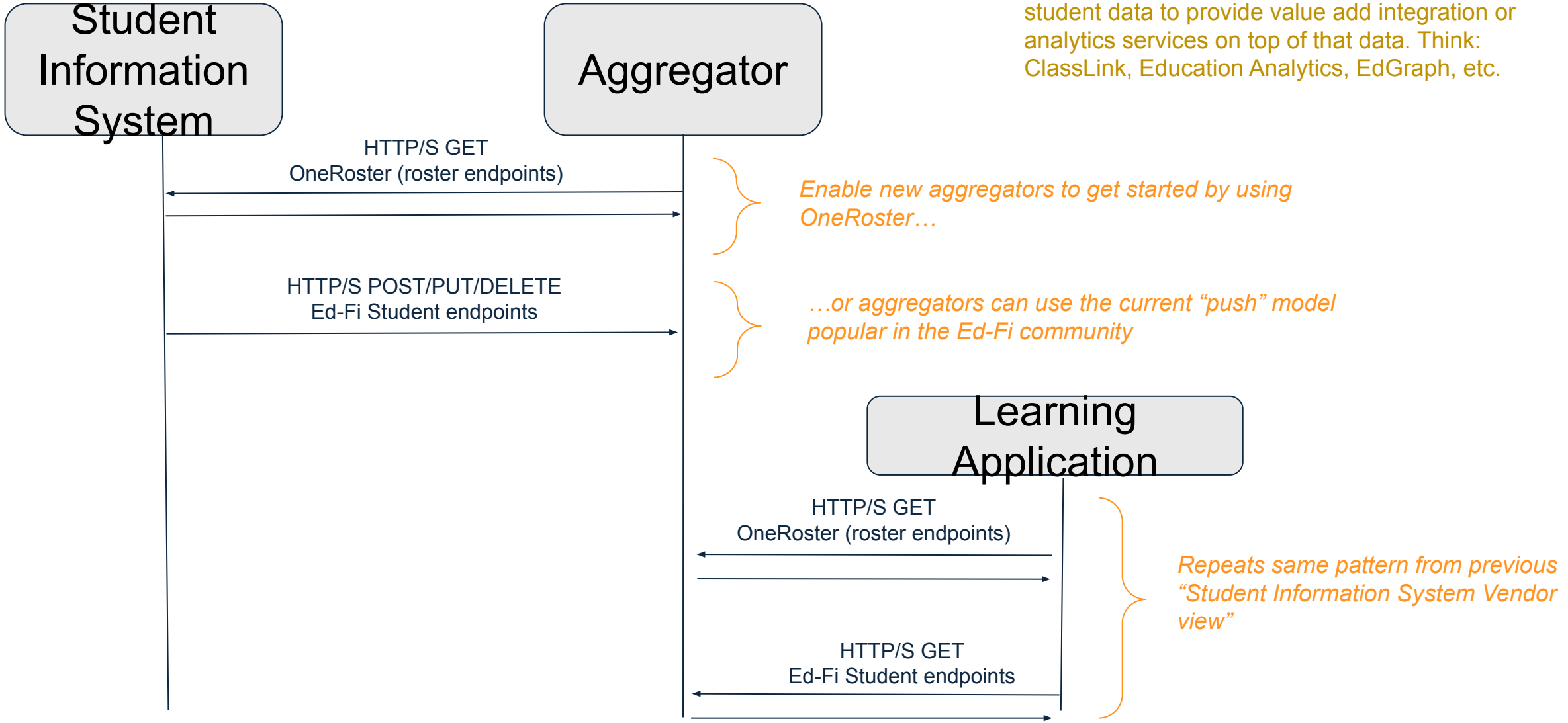
…via moving those calls to Ed-Fi's expanded data model and semantics. These could be endpoints like:

- */studentSchoolAttendanceEvents*
- */studentDisciplineIncidents*
- */studentTitleIProgramAssociations*

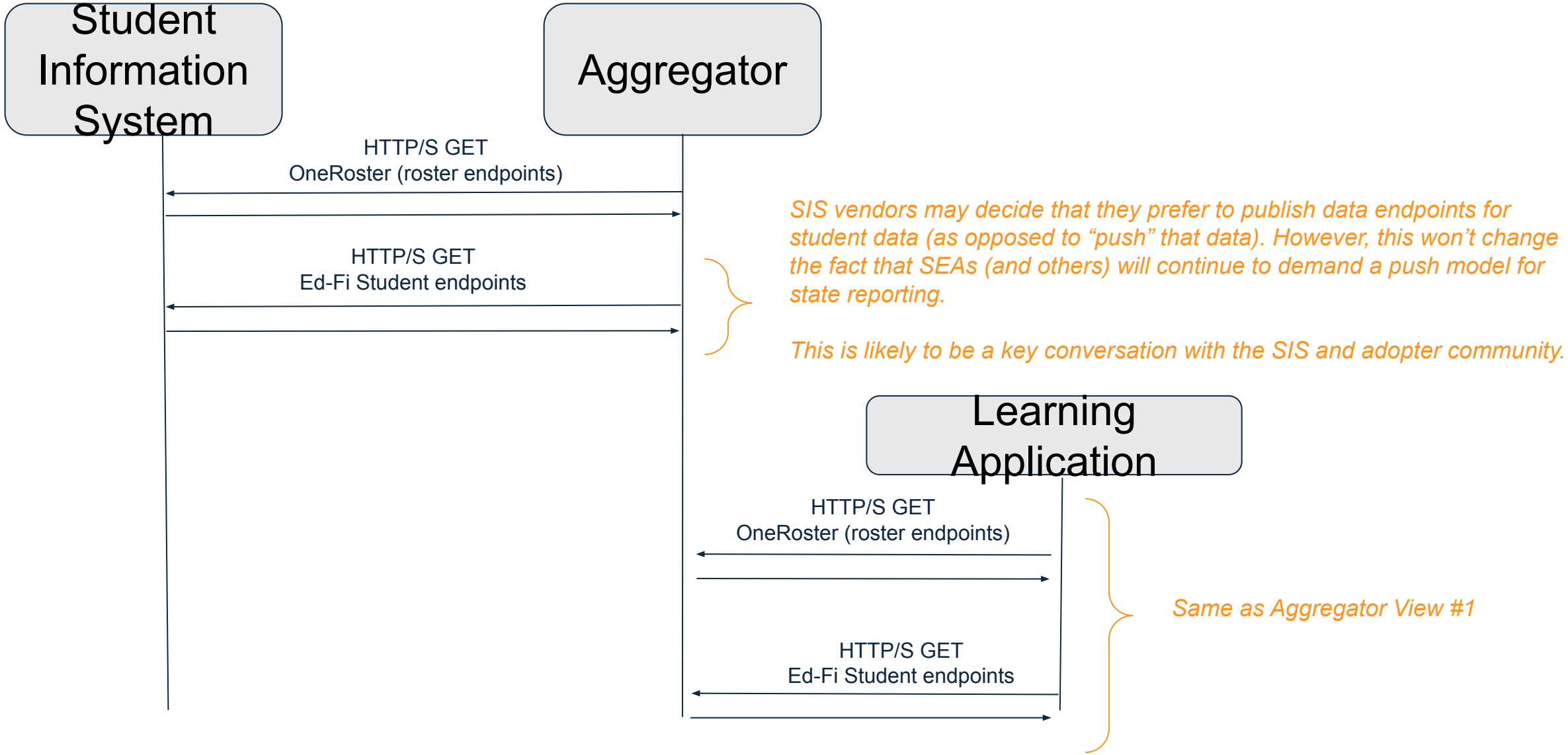
Note that SIS systems currently use a PUT/POST for these in Ed-Fi, so there is change here.

Aggregator View #1

Aggregators source various collections of student data to provide value add integration or analytics services on top of that data. Think: ClassLink, Education Analytics, EdGraph, etc.



Aggregator View #2



State Education Agency View