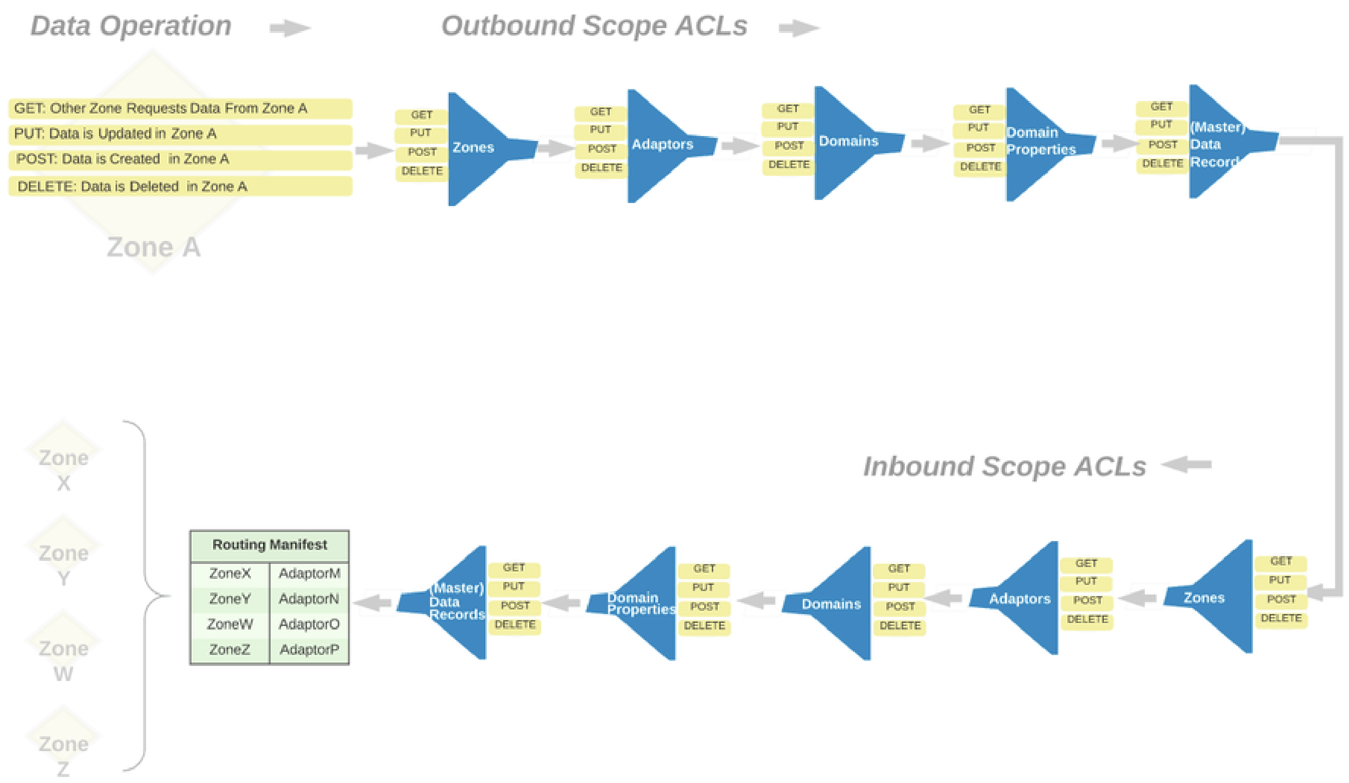
Governance



Outbound ACLs

Outbound ACLs are what provide data record visibility between zones & adaptors.

ACLs are a key component of MDM and are part of what is often referred to as the router. Outbound ACLs can be thought of as permissions on out-bound data. Access is controlled at various levels:

Source of Data	Destination	Priority
Zone[1]	Zone[2]	1
Zone[1].Adaptor[x]	Zone[2]	2
Zone[1].DR[i]	Zone[2]	3
Zone[1].DR[i].DRproperty[X]	Zone[2]	4
Zone[1].Adaptor[x].DRproperty[X]	Zone[2]	5

* At the highest level a Zone[1] can shutoff all outbound data record changes to Zone[2] and at the lowest level, Zone[1] can shutoff sharing a single attribute on a single adaptor (that it owns) to Zone[2],

* Sharing precedence is based on the priority e.g. If Zone[1] has turned off access to Zone[2] (Priority 1), then all other sharing actions are null.

* Permissions for each element are based on REST operations GET, PATCH, POST and DELETE. An additional operation is added for PUSH -- where a zone allows another to receive real-time changes however it may be determined that GET will scope will include PUSH.

Inbound ACLs

Background

Generally speaking, metadata is mostly to do (but not limited to) considerations regarding inbound data in a federated data domain.

Types of Metadata

Metadata includes settings for the following:

Incoming Filters

A zone or adaptor has the capability of filtering out changes it has scope to.

- "forbid" zone - Don't GET or accept any updates from a zone
- "forbid" adaptor - Don't GET or accept any updates from an adaptor

Classes

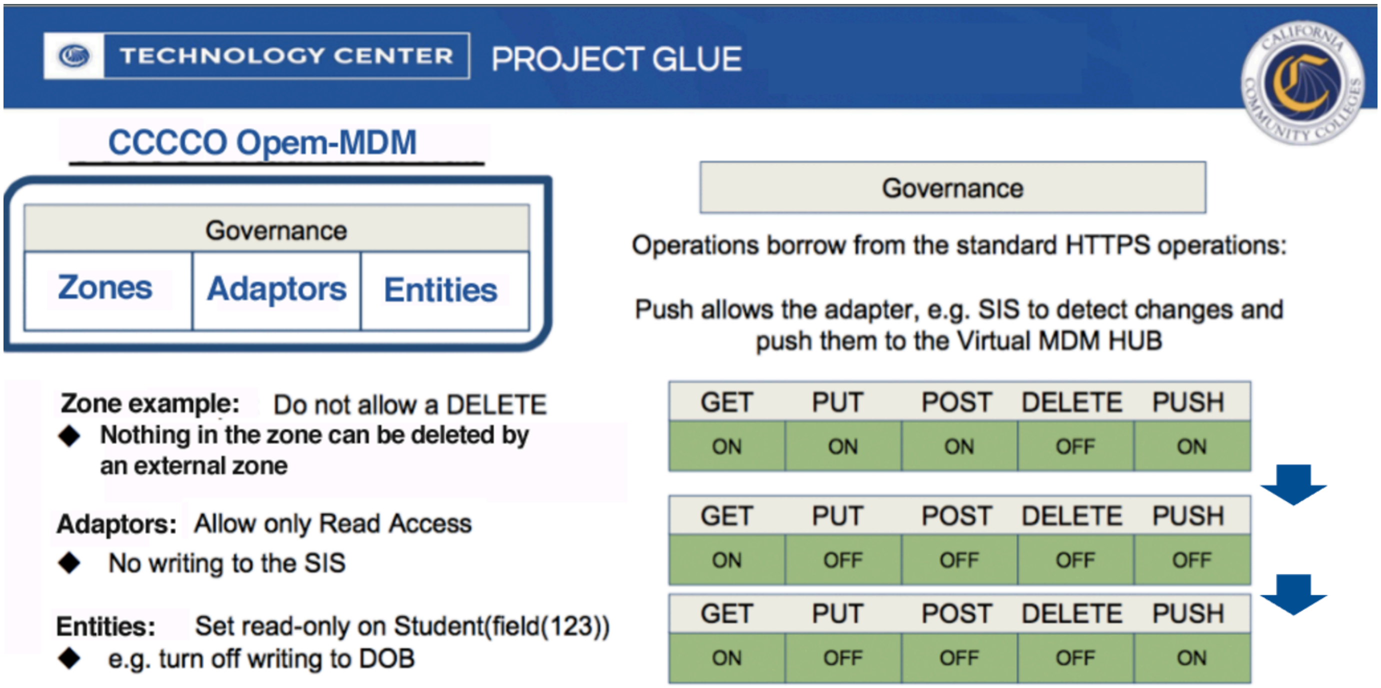
- Adaptor classes: 1, 2, 3: Allow a zone or an adaptor in a zone to set a class level on adaptors that are sharing data with them. 1 is highest and 3 is lowest. For example, if a GET yields three adaptors with the same domain property, and one adaptor is a class 1 and the other are class 2, then the data from the class 1 adaptor is returned in the GET.

Timestamps

- key/map of change timestamps and hashes
    - if a GET yields two adaptors with the same property and both are the same level, we can take the one with the latest timestamp.

Latency (post pilot)

- If a GET request is issued with a reduced-latency parameter, the request will query only the adaptors that are in PLAY or PLAY_RO with the lowest latency times.



TECHNOLOGY CENTER PROJECT GLUE

CCCCO Opem-MDM

Governance

Zones Adaptors Entities

Zone example: Do not allow a DELETE

- **Nothing in the zone can be deleted by an external zone**

Adaptors: Allow only Read Access

- No writing to the SIS

Entities: Set read-only on Student(field(123))

- e.g. turn off writing to DOB

Governance

Operations borrow from the standard HTTPS operations:

Push allows the adapter, e.g. SIS to detect changes and push them to the Virtual MDM HUB

GET	PUT	POST	DELETE	PUSH
ON	ON	ON	OFF	ON

GET	PUT	POST	DELETE	PUSH
ON	OFF	OFF	OFF	OFF

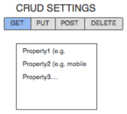
GET	PUT	POST	DELETE	PUSH
ON	OFF	OFF	OFF	ON

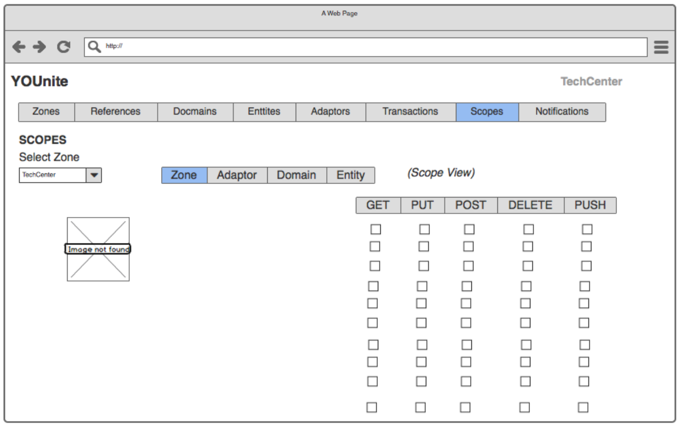
SCOPES UI

The Core of Scopes

ZONE	ADAPTOR	DOMAIN.PROPERTY[ENTITIES]

plus metadata e.g. last modfied, weight (how valuable), crud/zone permission settings





GET	PUT	POST	DELETE	PUSH
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□

FOR HTTP METHODS (GET, PUT, POST, DELETE, PUSH):

If child is turned on or off, then parents are unaffected

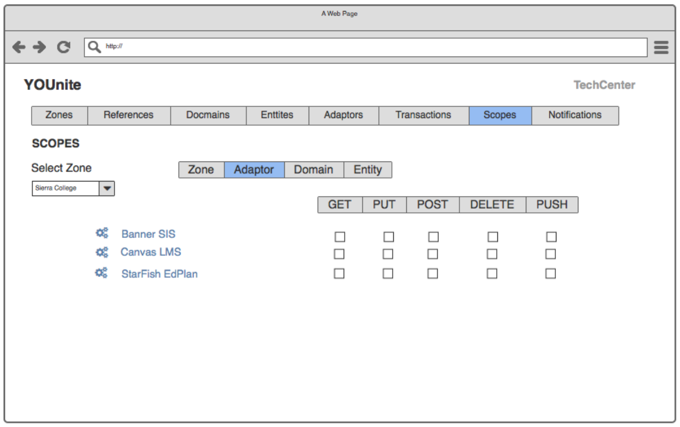
If parent is turned on, then children are unaffected however,MILIND can we have a three-way switch that says?

- turn on (me only)
- turn off
- turn on and all of my children on (and only the children that have granted the parent permission are turned on)

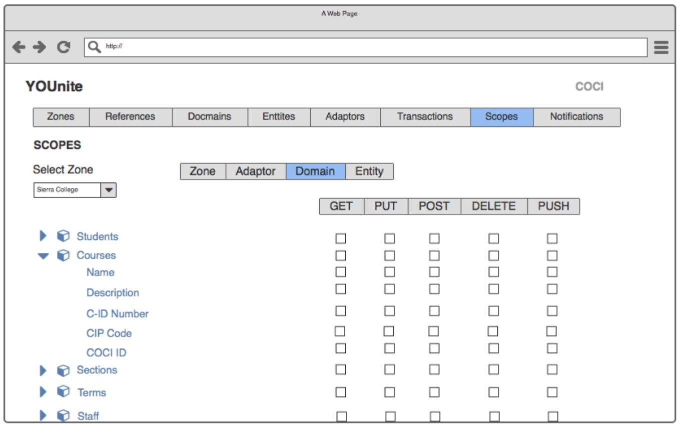
ALL
Three way for that too?

- turn all HTTP Methods on (GET, PUT, POST, DELETE, PUSH)
- turn all off
- turn all on and all children on

WE WILL need to add a scope management control option to a child zone in the "Zones" page. This would prevent a parent zone from setting controls on a child zone. SCOPE MANAGER.



	GET	PUT	POST	DELETE	PUSH
Banner SIS	□	□	□	□	□
Canvas LMS	□	□	□	□	□
StarFish EdPlan	□	□	□	□	□



	GET	PUT	POST	DELETE	PUSH
Students	□	□	□	□	□
Courses	□	□	□	□	□
Name	□	□	□	□	□
Description	□	□	□	□	□
C-ID Number	□	□	□	□	□
CIP Code	□	□	□	□	□
COCI ID	□	□	□	□	□
Sections	□	□	□	□	□
Terms	□	□	□	□	□
Staff	□	□	□	□	□

SCOPES TABLES

The BIG question for starters is... do we store the scopes in our own DB or do we store these in oAuth ?

Zones

scopes_zone_publish

ZONE_UUID	SUBSCRIBING_ZONE_UUID	GET	PUT	POST	DELETE
AAAA-AAAA	FFFF-FFFF	true	false	false	false

scopes_zone_subscribe

ZONE_UUID	PUBLISHING_ZONE_UUID	GET	PUT	POST	DELETE
FFFF-FFFF	AAAA-AAAA	true	false	false	false

GET - Subscribing zone can retrieve zone data and metadata (e.g. zone name, description, parent, scope settings, etc).

PUT - Subscribing zone can update zone data (e.g. name, description, move the zone, etc.)

POST - Subscribing zone can create a new zone, adaptor or reference in the publishing zone.

DELETE - Subscribing zone can delete the zone.

Domains

scopes_references_publish

ZONE_UUID	REFERENCE_UUID	SUBSCRIBING_ZONE_UUID	GET	PUT	POST	DELETE
AAAA-AAAA	BBBB-BBBB	FFFF-FFFF	true	false	false	false

scopes_references_subscribe

ZONE_UUID	REFERENCE_UUID	PUBLISHING_ZONE_UUID	GET	PUT	POST	DELETE
FFFF-FFFF	BBBB-BBBB	AAAA-AAAA	true	false	false	false

GET - Subscribing zone can retrieve reference data and metadata.

PUT - Subscribing zone can update the reference (e.g. name, description, and modify entries in the reference).

POST - Subscriber can add new references entries to the reference (e.g. add a state to the state reference.

DELETE - Subscribing zone can delete reference entries (e.g. delete a state from the state reference).

Adaptors

scopes_publish

ZONE_UUID	ADAPTOR_UUID	SUBSCRIBING_ZONE_UUID	GET	PUT	POST	DELETE
AAAA-AAAA	CCCC-CCCC	FFFF-FFFF	true	false	false	false

scopes_subscribe

ZONE_UUID	ADAPTOR_UUID	PUBLISHING_ZONE_UUID	GET	PUT	POST	DELETE
FFFF-FFFF	CCCC-CCCC	AAAA-AAAA	true	false	false	false

GET - Subscribing zone can retrieve adaptor data and metadata (e.g. adaptor name, description, supported properties, scope settings, etc).

PUT - Subscribing zone can update the adaptor (e.g. name, description, etc.)

POST - *TBD*

DELETE - Subscribing zone can delete the adaptor.

scopes_publish

ZONE_UUID	REFERENCE_UUID	SUBSCRIBING_ZONE_UUID	GET	PUT	POST	DELETE
AAAA-AAAA	BBBB-BBBB	FFFF-FFFF	true	false	false	false

scopes_references_subscribe

ZONE_UUID	REFERENCE_UUID	PUBLISHING_ZONE_UUID	GET	PUT	POST	DELETE
FFFF-FFFF	BBBB-BBBB	AAAA-AAAA	true	false	false	false

Data Domains

TODO

Entities

TODO