AUTODESK
DevCon

Autodesk Construction Cloud API – Deep Dive

Mikako Harada
Senor Manager, Developer Technical Services AEC

Safe Harbor Statement

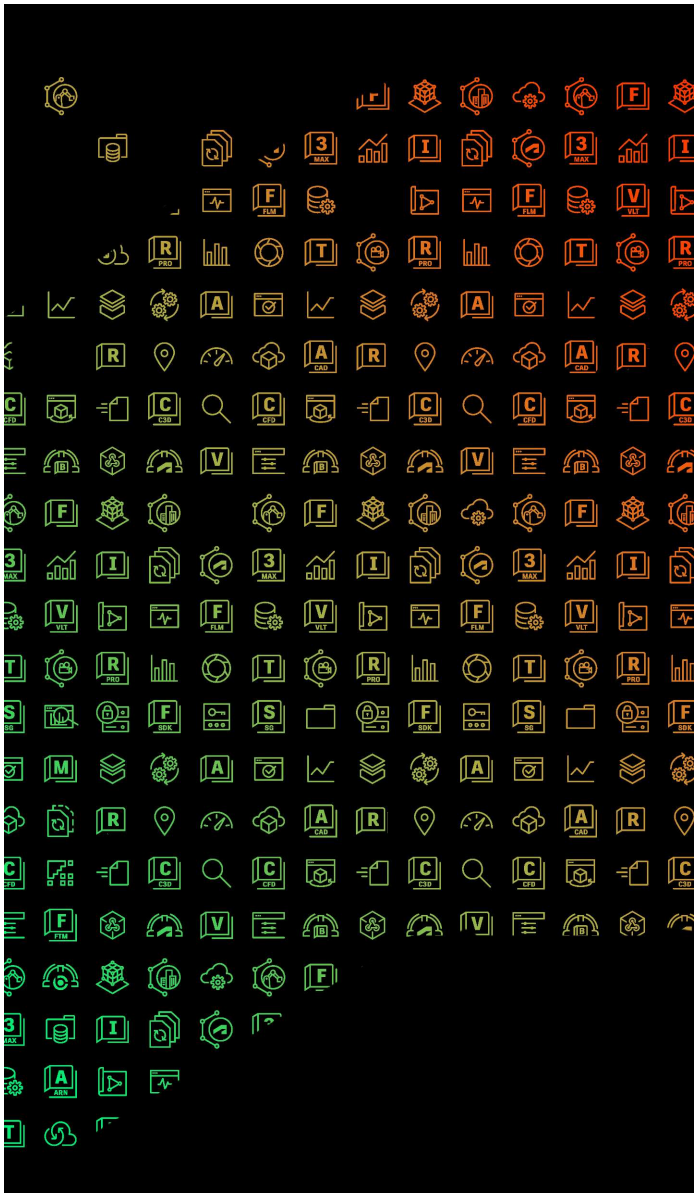
The presentations during this event may contain forward-looking statements about our outlook, future results and related assumptions, total addressable markets, acquisitions, products and product capabilities, and strategies. These statements reflect our best judgment based on currently known factors. Actual events or results could differ materially. Please refer to our SEC filings, including our most recent Form 10-K and Form 10-Q filings available at www.sec.gov, for important risks and other factors that may cause our actual results to differ from those in our forward-looking statements.

The forward-looking statements made in these presentations are being made as of the time and date of their live presentation. If these presentations are reviewed after the time and date of their live presentation, even if subsequently made available by us, on our website or otherwise, these presentations may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statements.

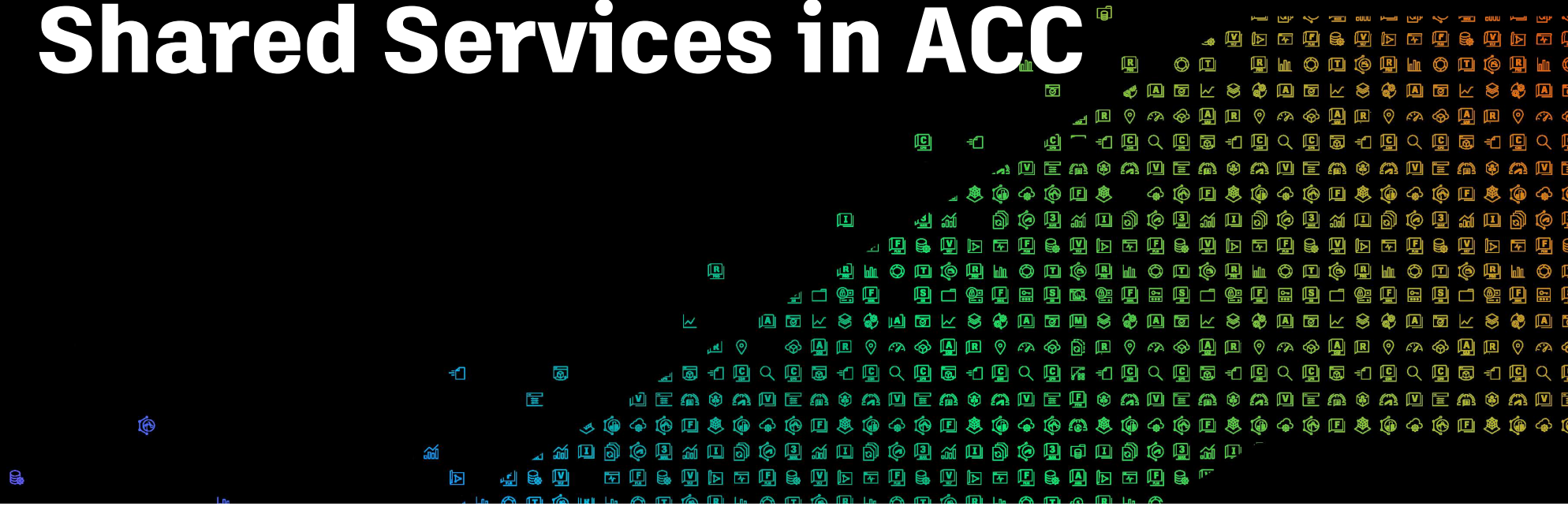
Statements regarding planned or future development efforts for our products and services are not intended to be a promise or guarantee of future availability of products, services, or features but merely reflect our current plans and based on factors currently known to us. Purchasing decisions should not be made based upon reliance on these statements.

Agenda

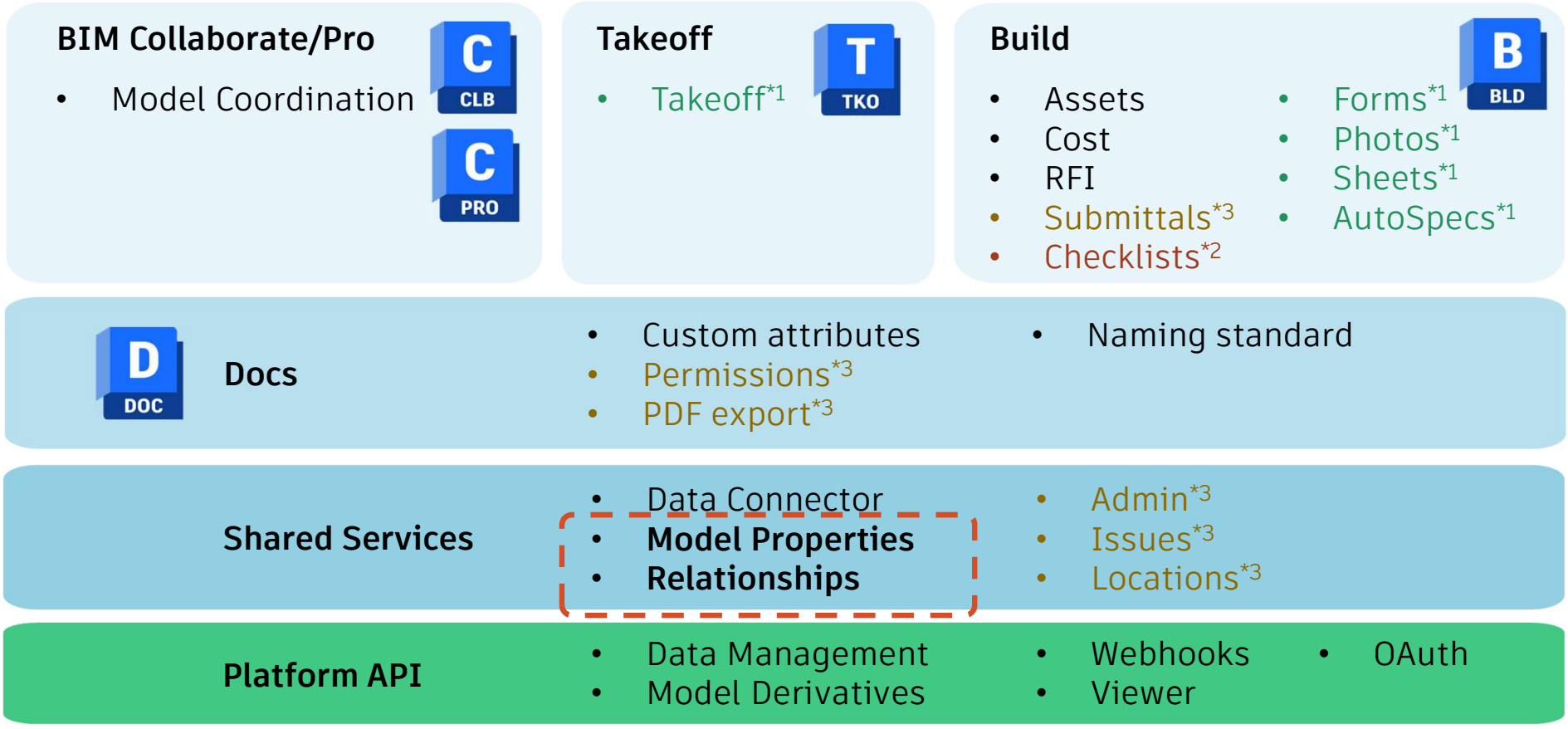
1. Introduction: Shared Services in ACC
2. Relationships API
3. Model Properties API
4. Common Questions
5. What’s Next



Shared Services in ACC

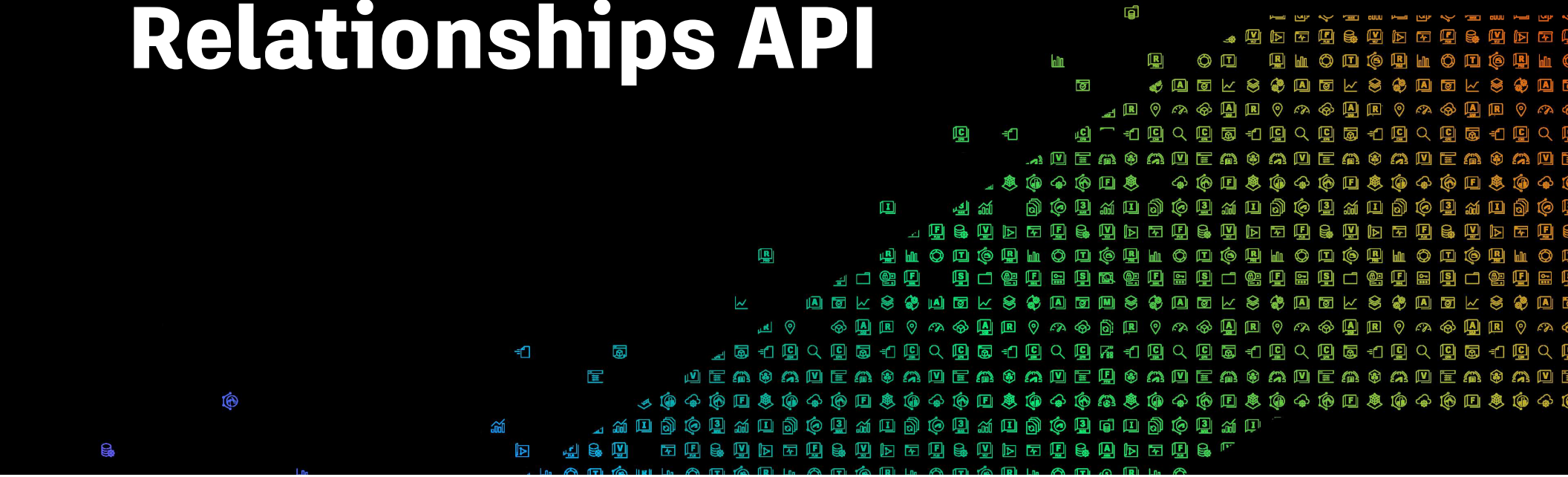


ACC/BIM 360 API Component Services & Structure



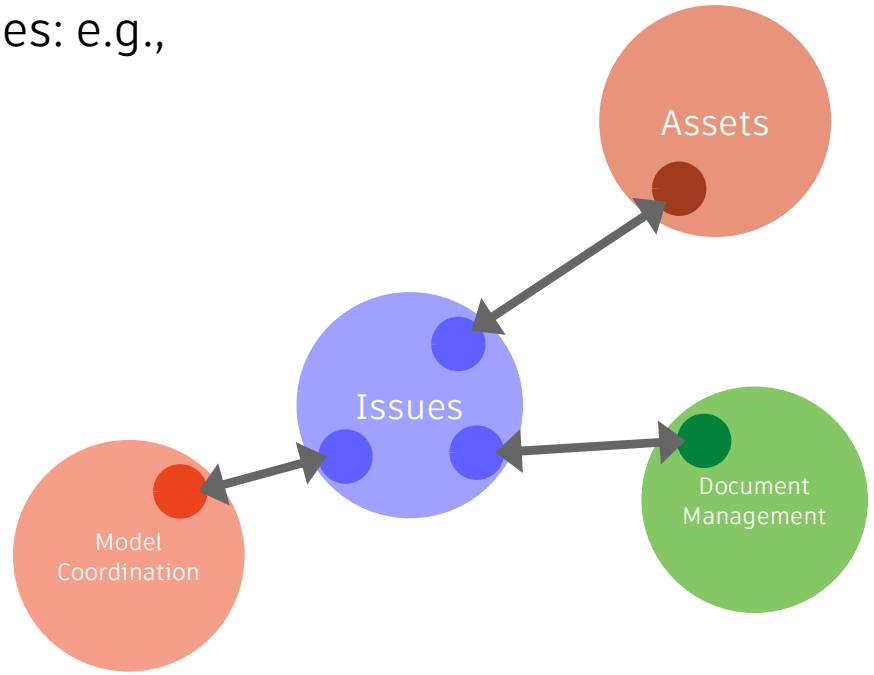
*1) ACC only, *2) BIM 360 only, *3) Not compatible or partially compatible

Relationships API

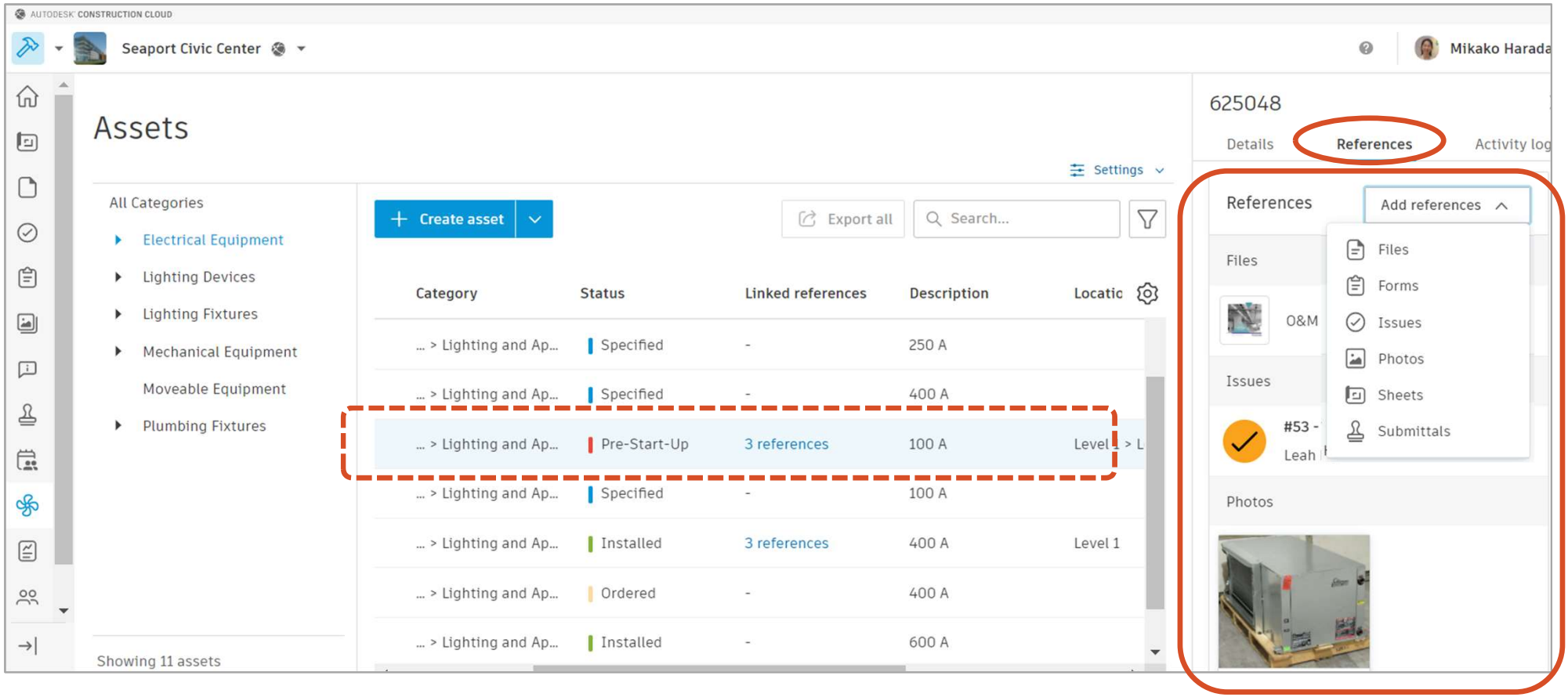


Relationships API

- Represent links between items (Entity) that reside in different functional components (Domain)
- Used in ACC and BIM 360 by various modules: e.g.,
  - Model Coordination, Assets, Forms, Photos, Cost
  - Older implementations may not be using it
- Read & Write
- Publicly available today

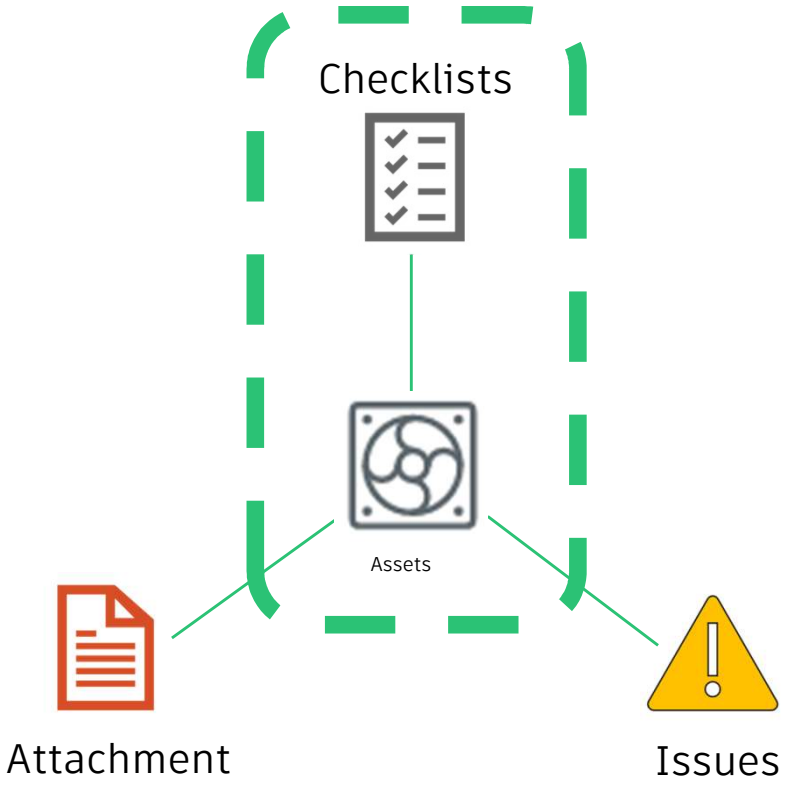


Relationships API

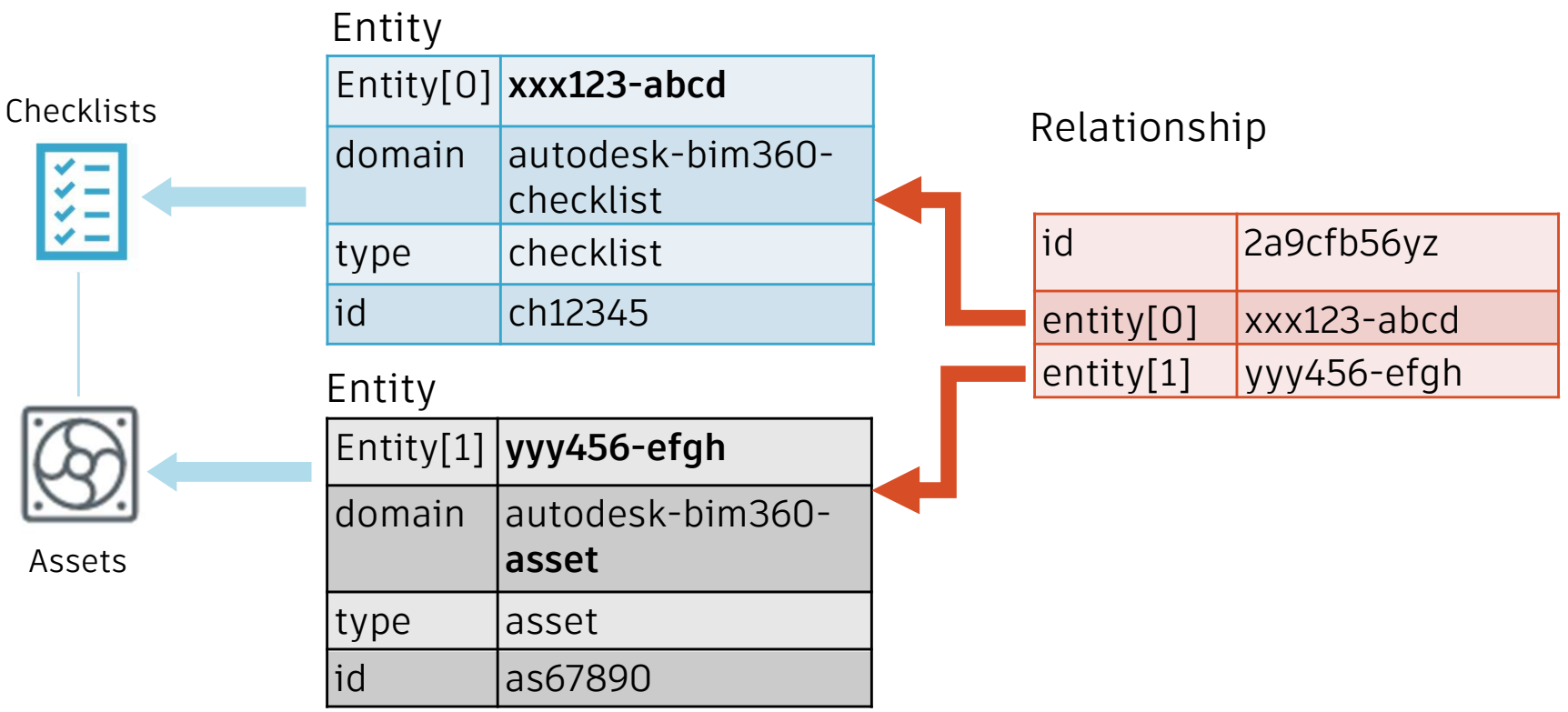


Relationships API

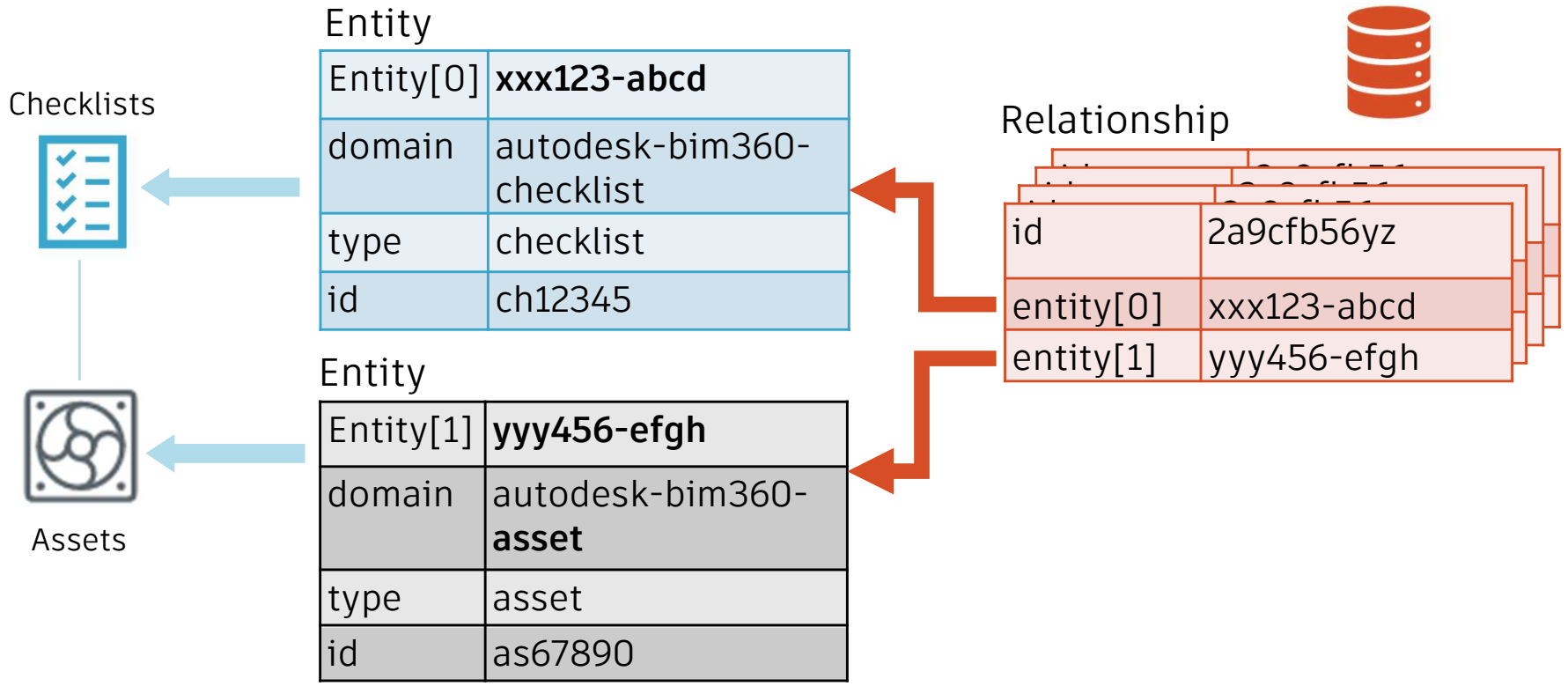
Example



Relationships API



Relationships API



Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

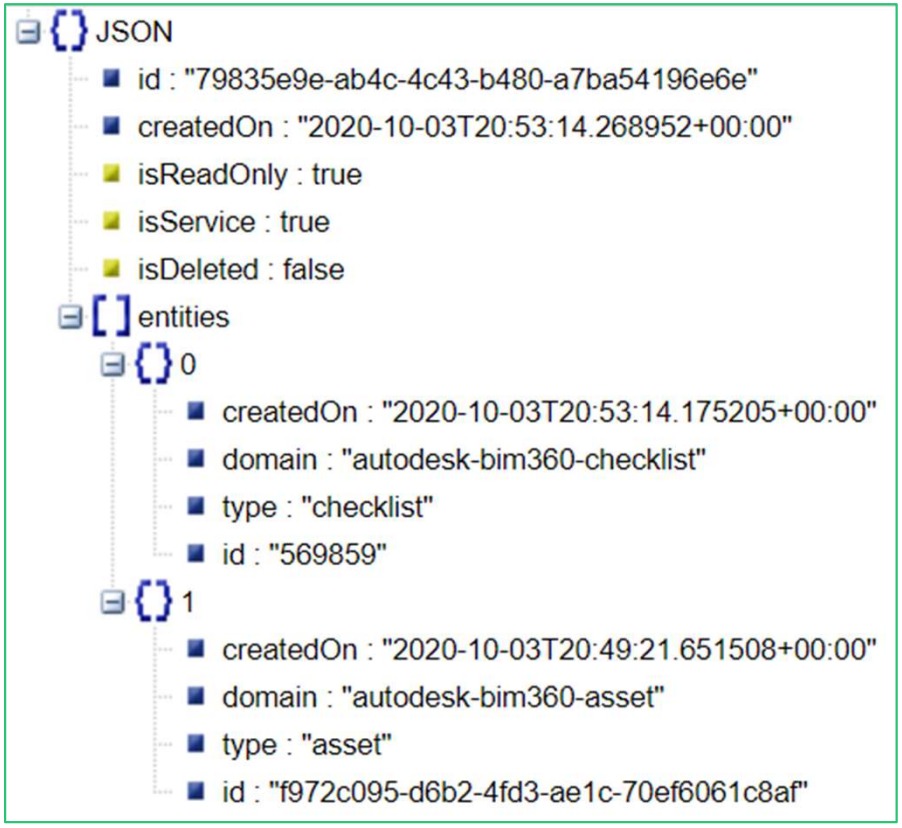
Save and search relationships independent from the implementation detail of each components

9 endpoints

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

Get relationship(s) from id(s)



Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

Search rel's that match the given criteria (domain, type, ids, date). e.g.,

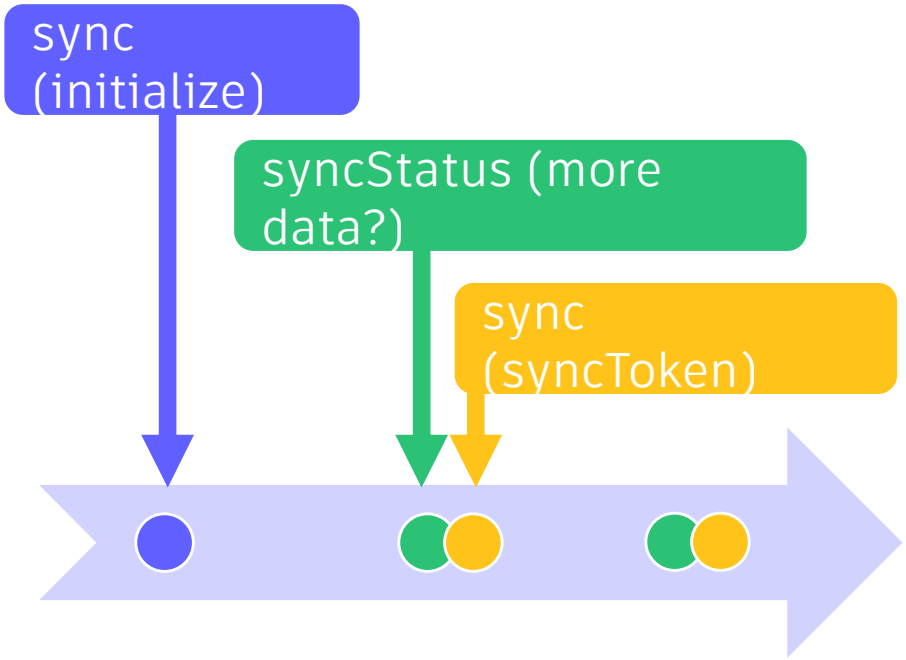
"Give me all the relationships between assets and checklists"

```
GET relationships:/search?
domain=autodesk-bim360-asset
&withDomain=autodesk-bim360-
checklist
```

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

Used to sync rel. data and external repository.



Relationships API - Additions

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

- Retrieves **compatible entity types** to create/delete relationships
- Supported domain/entity types:
  - Asset/asset category
  - Document/File lineage/version
  - Issue
  - Form/form field/form template
  - Photo
  - Etc.
- Unsupported types - may have workaround.
- Expected to be extended overtime

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

```
[
  {
    "domain": "autodesk-bim360-asset",
    "entityTypes": [
      {
        "entityType": "asset",
        "allow": [
          {
            "domain": "autodesk-bim360-issue",
            "entityTypes": [
              "issue"
            ]
          },
          {
            "domain": "autodesk-bim360-documentmanagement",
            "entityTypes": [
              "documentlineage"
            ]
          },
          {
            "domain": "autodesk-construction-photo",
            "entityTypes": [
              "photo"
            ]
          },

          […]
```

Relationships API

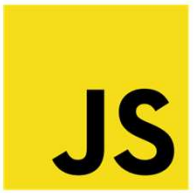
Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

- **Create** a relationship(2) between two entities.
  - Two entities are writable
  - Entities must exist
  - must have write access to both entities
- **Delete** a given relationship
  - “Soft” delete
  - must have write access to both entities

Relationships API: Developer Resources

SDK

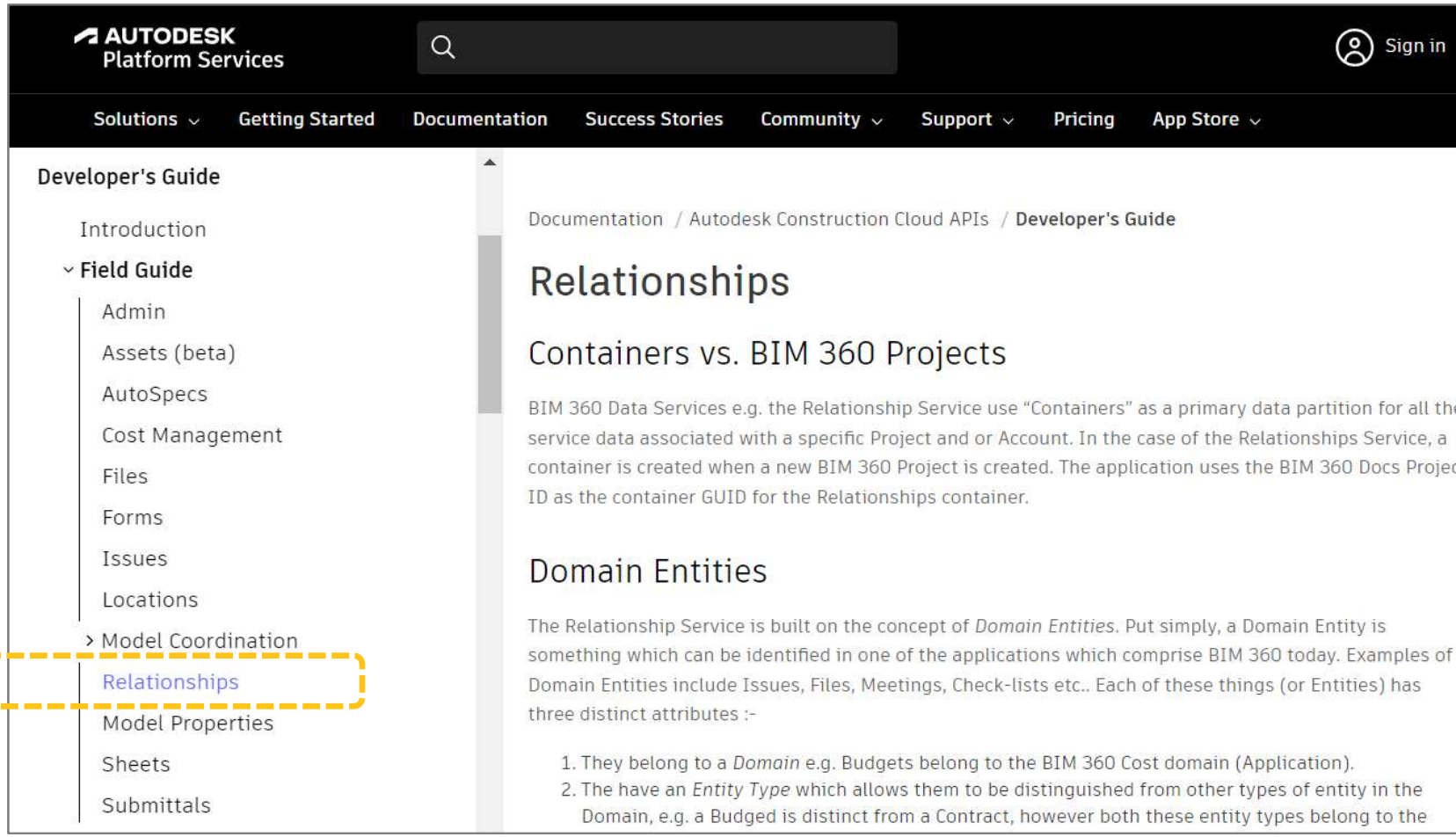
- .NET Core nuget packages
- JavaScript and Node.js



Relationships API: Developer Resources

Documentation

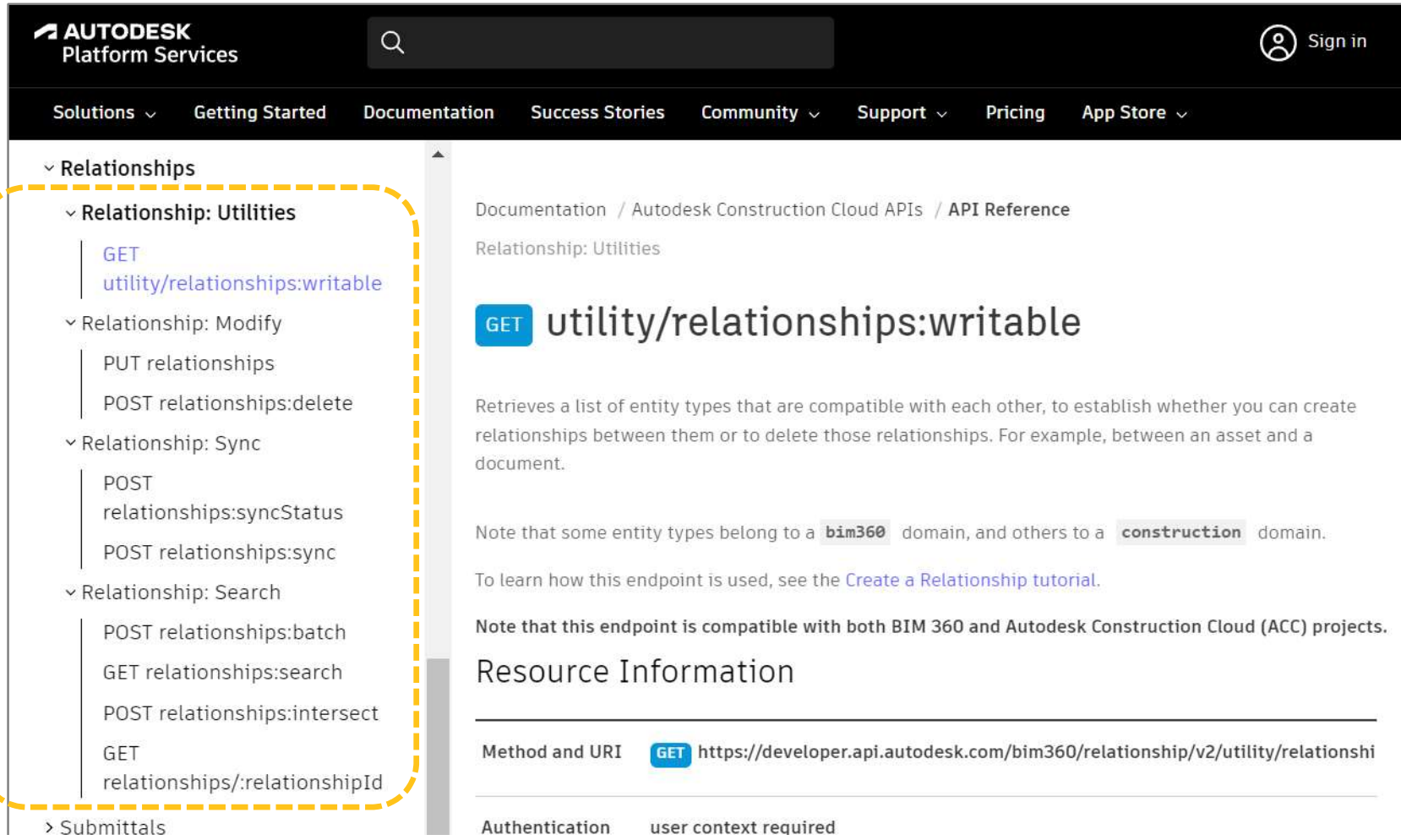
- Field Guide
- Step by Step Tutorials



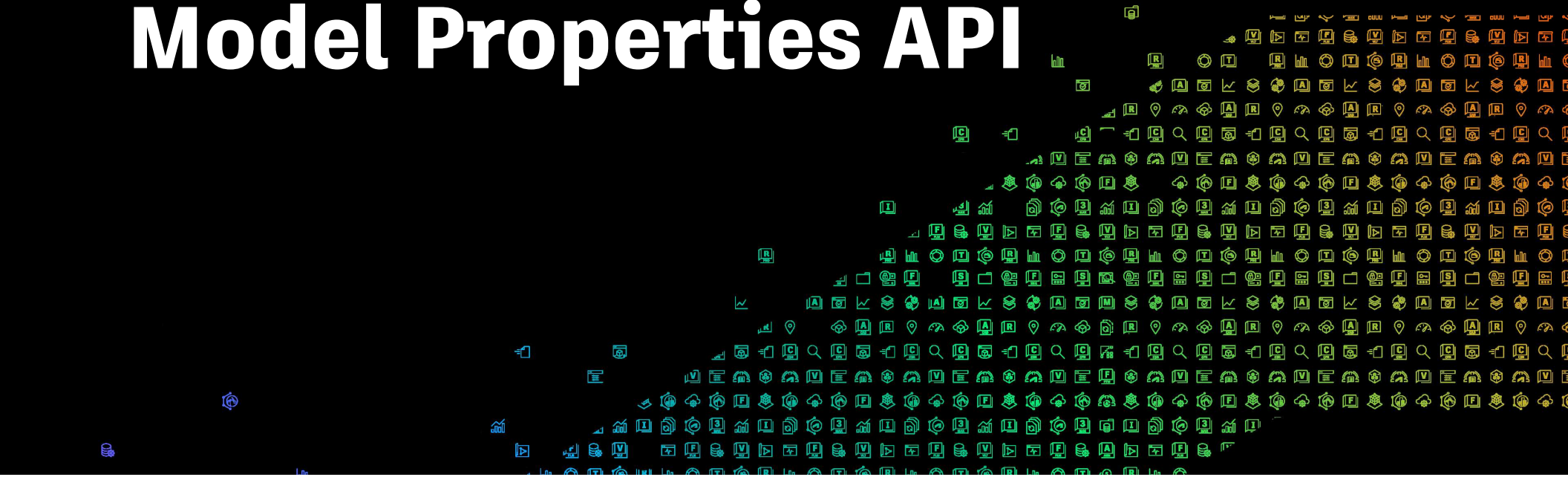
Relationships API: Developer Resources

Documentation

- Reference Guide

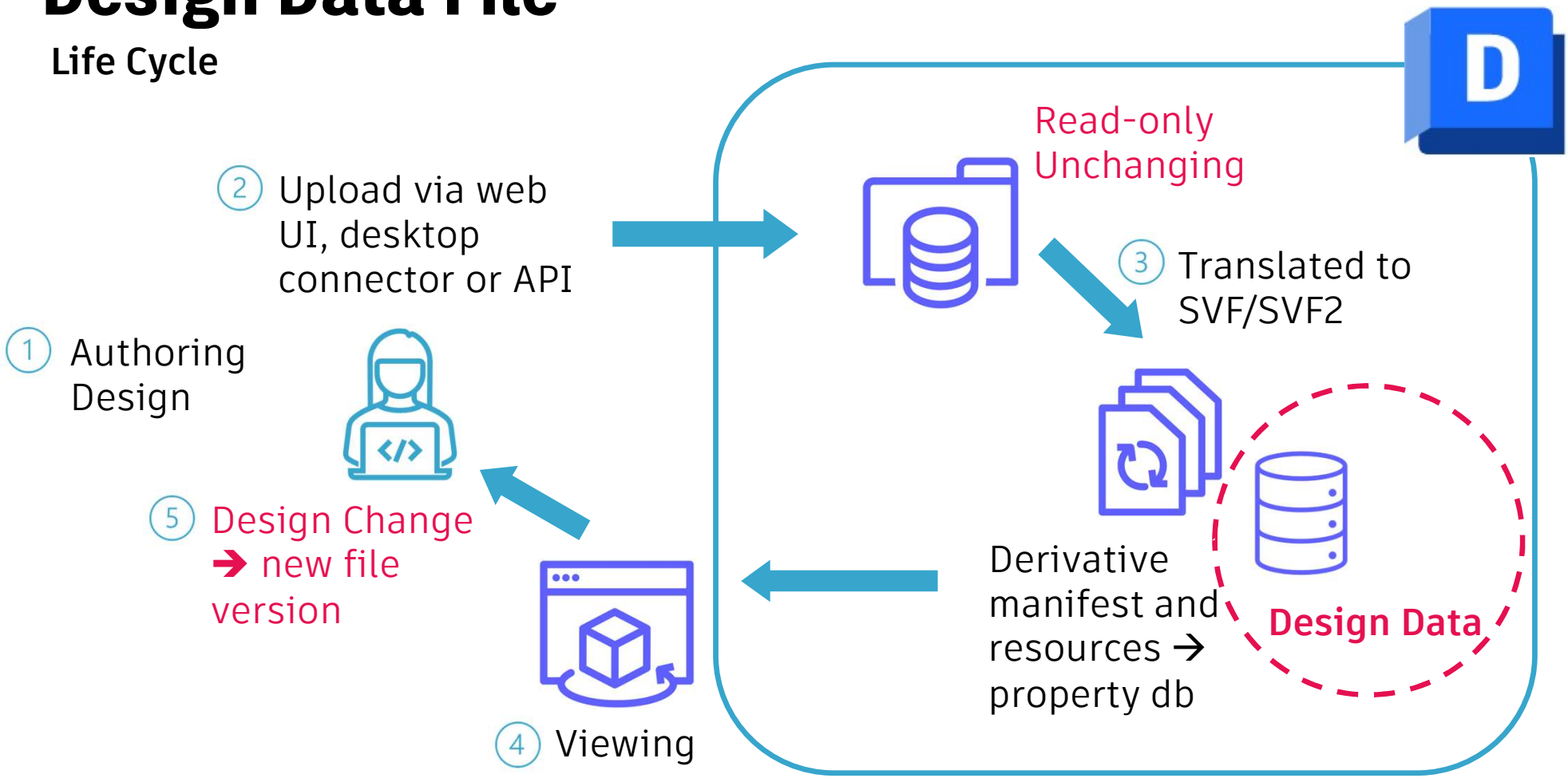


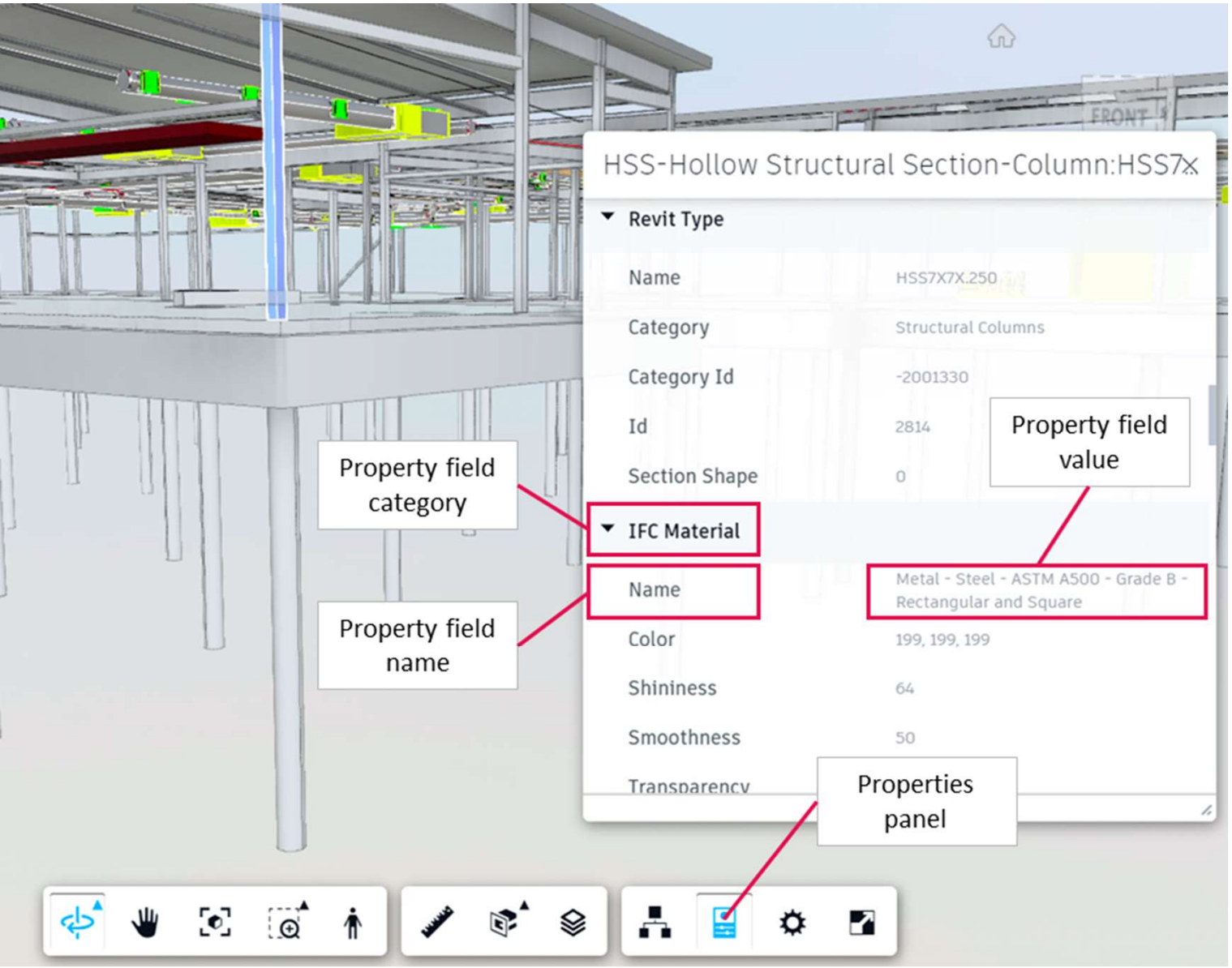
Model Properties API



Design Data File

Life Cycle



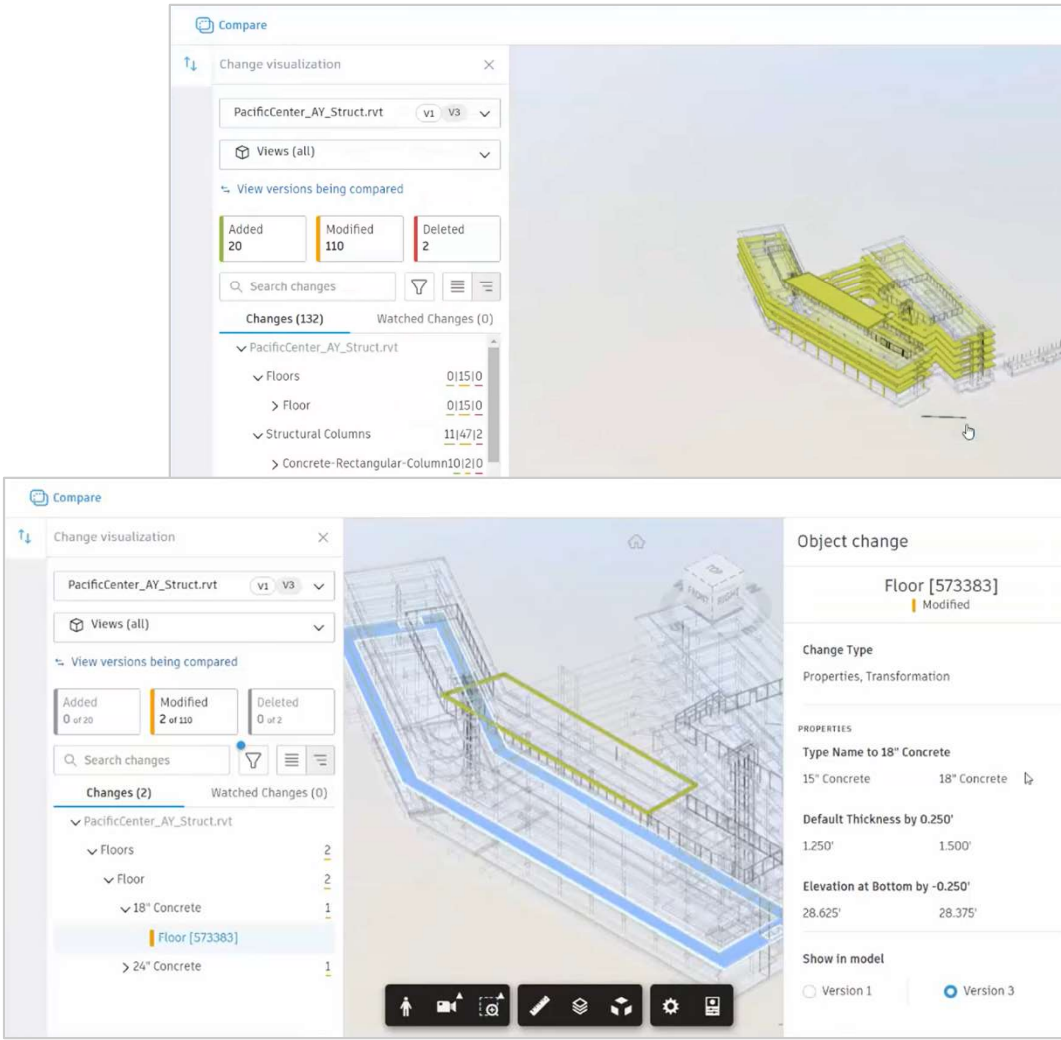


Classifying (BIM) SVF2 model properties

- Category
- Name
- Type (e.g., string, double)
- (Optional) Unit of Measure

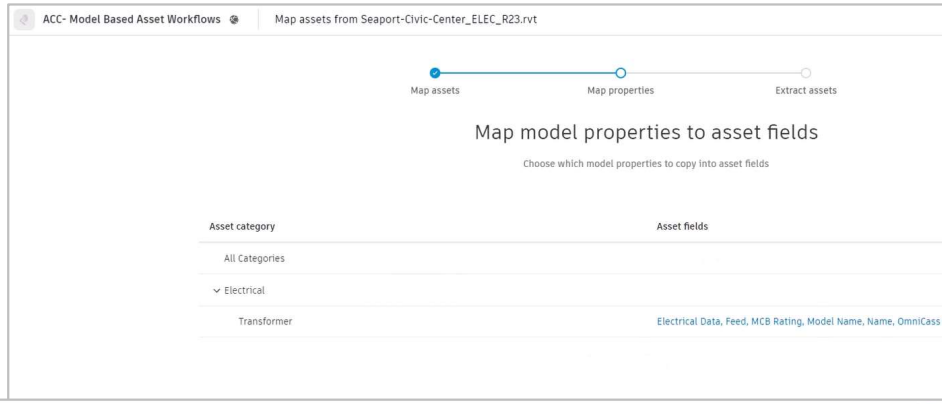
Model Properties API

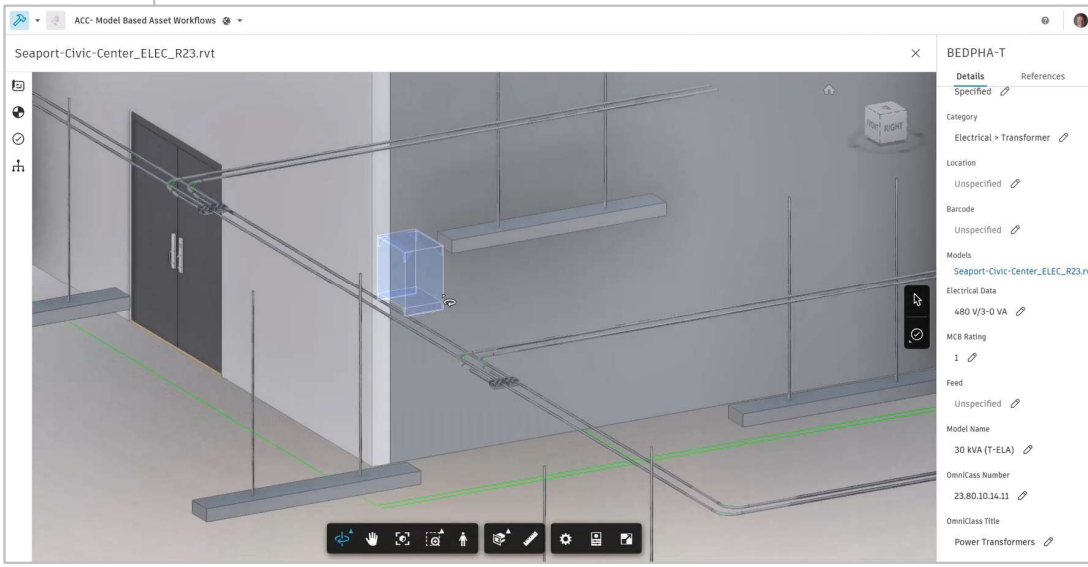
- Released Feb 2022
- Autodesk/BIM 360 Docs based products
- Built on PropertyDb from Derivative Services (svf translation)
  - **Index** (Base) – query, filter properties of svf2 objects + extra (viewable bbox)
  - **Diff** – Index + compare two versions
- Used in product
  - Design Collaboration - Change analysis
  - Model Coordination
    - Model property breakdown / advanced filtering
    - Publishing views to Docs
    - Object exclusion
  - Assets – integrating with models



Model Properties API

- Released Feb 2022
- Autodesk/BIM 360 Docs based products. US and EMEA
- Built on PropertyDb from Derivative Services (svf translation)
  - **Index** (Base) – query, filter properties of svf2 objects + extra (viewable bbox)
  - **Diff** – Index + compare two versions
- Used in product
  - Design Collaboration - Change analysis
  - Model Coordination
    - Model property breakdown / advanced filtering
    - Publishing views to Docs
    - Object exclusion
  - Assets – integrating with models

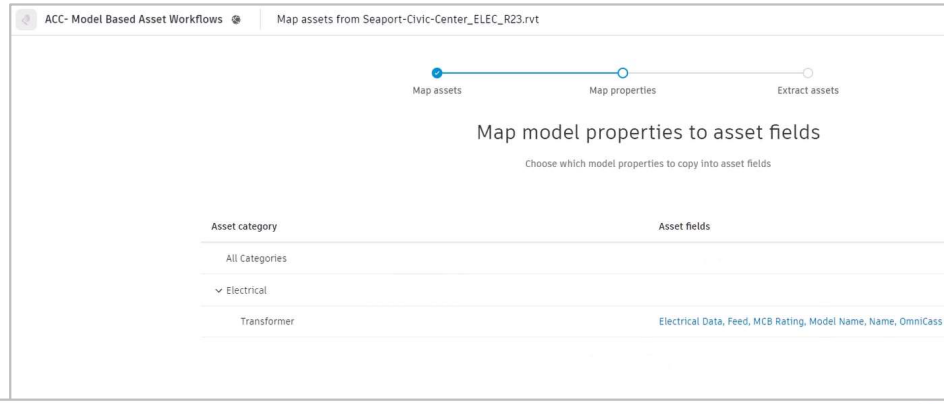


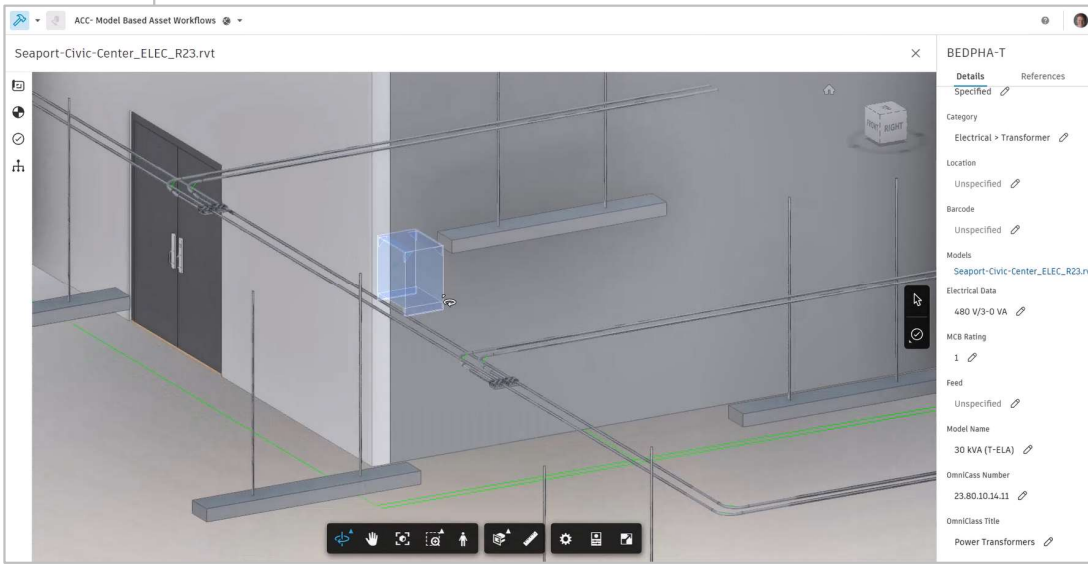


Model Properties API

Supported Files

- **Index** – any files supported Derivative Service /SVF2
- **Diff** - element id needs to be stable (uniquely identifiable)
  - **RVT**
  - **DWG**
  - **NWC** exported from:
    - Revit and
    - AutoCAD verticals
  - **IFC** exported from:
    - AutoCAD Architecture, MEP, Civil 3D 2018+
    - ARCHCAD
    - Revit
    - MigiCAD for Revit
    - Tekla Structures





Sample Applications

Model Properties API

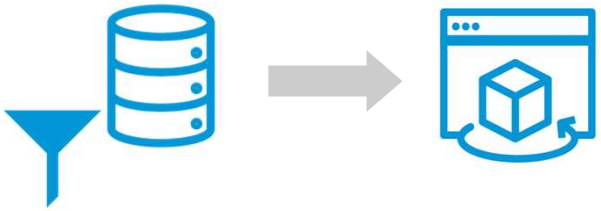
Filter Elements & Partial Model Load

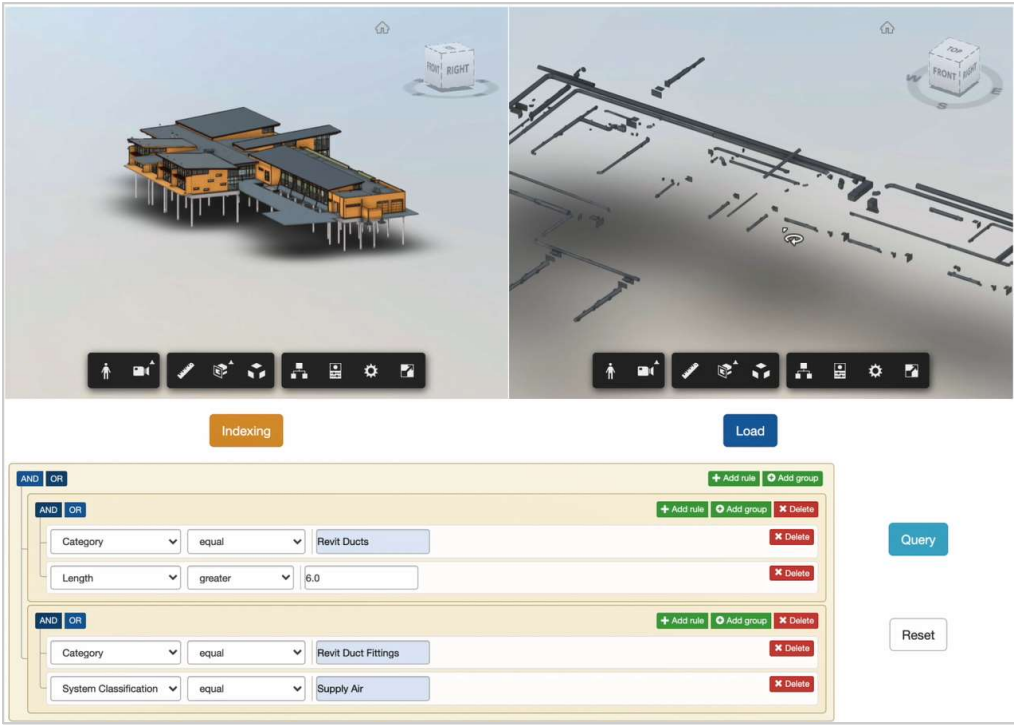
Index

What it does: Filters elements by properties, which can be geometric properties such as length and height. A filter condition is defined as a binary expression form and can be combined by AND/OR. The results are visualized in the Forge viewer.

Code:
https://github.com/autodesk-platform-services/aps-model.properties-elements.filtering

Author: Xiaodong Liang, Autodesk





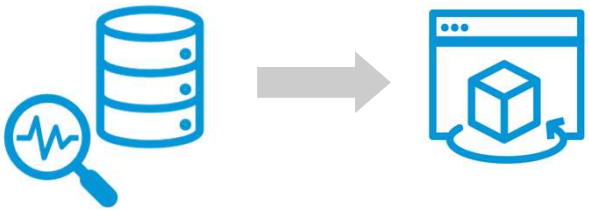
Compare Two Versions

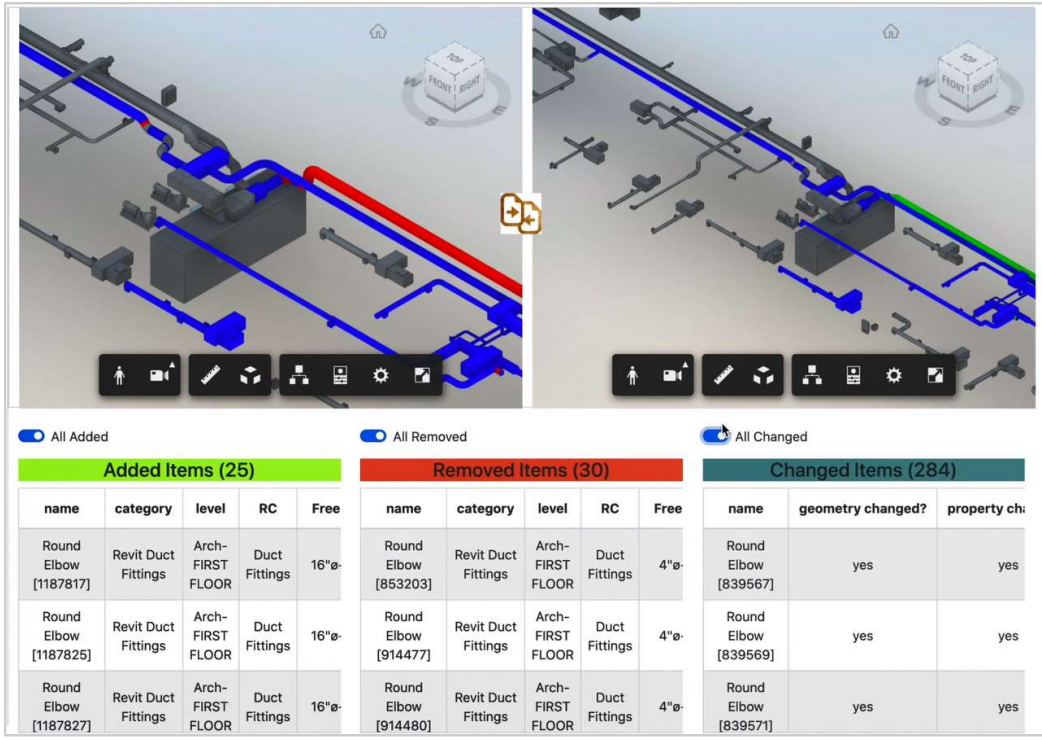
Diff

What it does: Compares two versions of a model and visualizes the differences found in the Forge Viewer. Differences can be in geometries and properties. Elements may be added, modified and removed.

Code:
https://github.com/autodesk-platform-services/aps-model.properties-versions.difference

Author: Xiaodong Liang, Autodesk





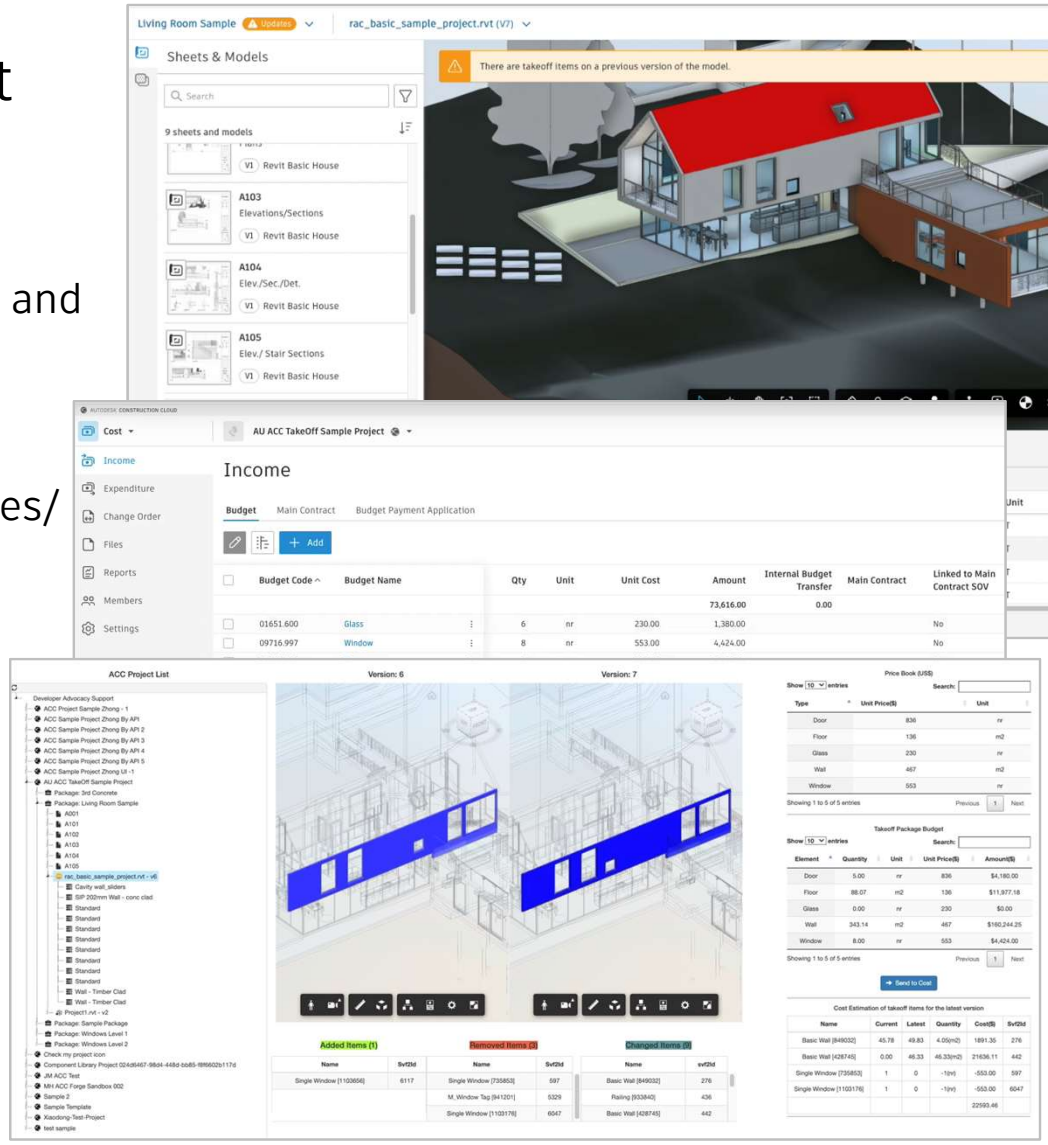
Change Analysis in Takeoff and Cost

Application of **Diff** in Estimate

What it does: Identifies changes in takeoff items, compares current and previous versions of a model and visualizes the differences in the Viewer. Update the budgets data in Cost module.

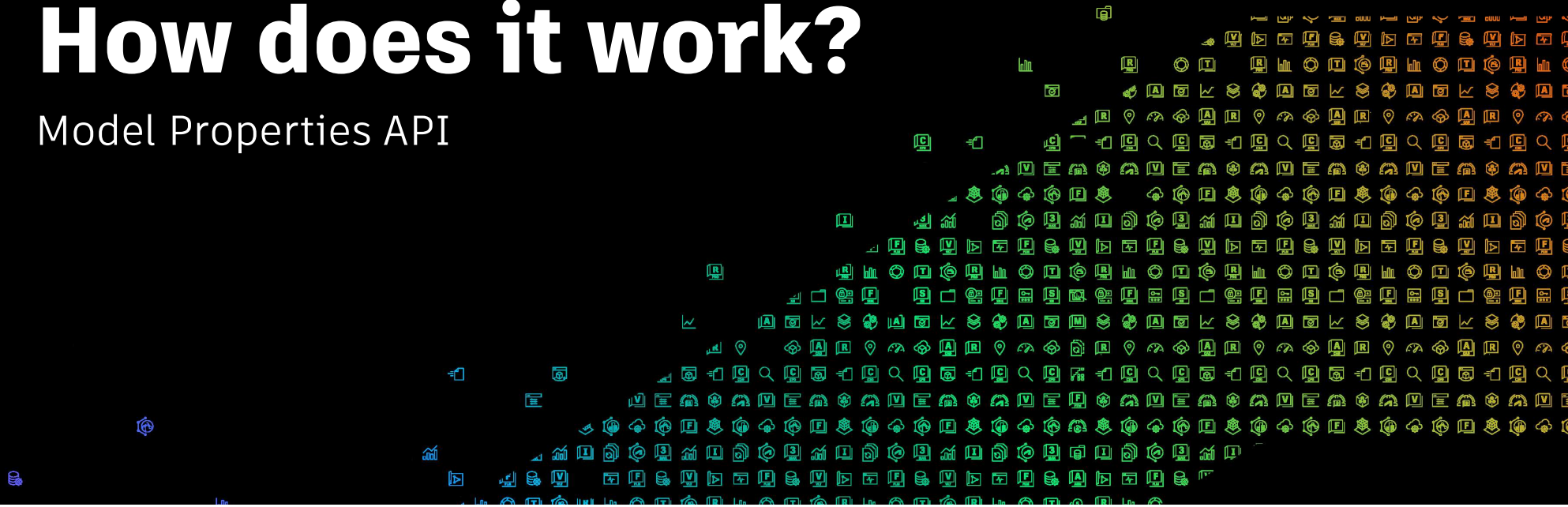
Code: https://github.com/autodesk-platform-services/aps-acc-takeoff-versions-compare-cost

Author: Zhong Wu, Autodesk



How does it work?

Model Properties API



How does it work?

Basic Workflow

Model Derivative

DM/Docs

Files uploaded to Docs via UI, Desktop Connector and API

Model Derivative

Files are translated to svf/svf2. In Docs, translation is triggered automatically.

Design Data PropertyDb

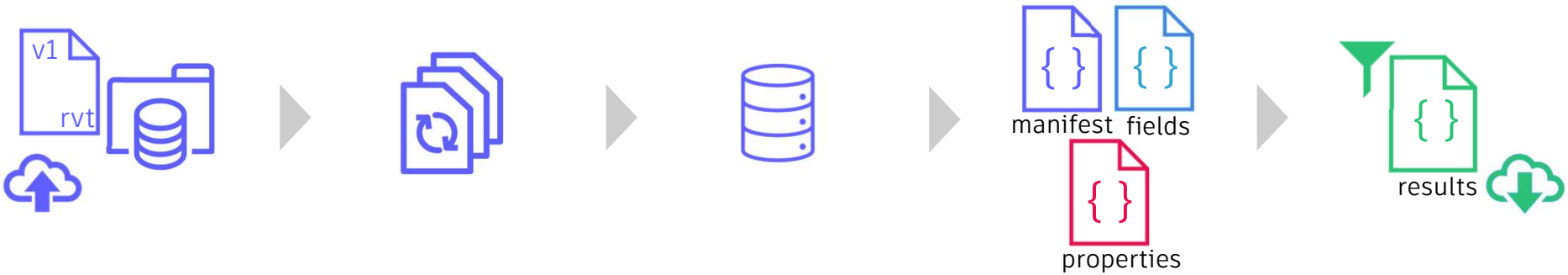
Property database, manifest, resources.

How does it work?

Basic Workflow

Model Derivative

Model Properties



DM/Docs

Files uploaded to Docs via UI, Desktop Connector and API

Model Derivative

Files are translated to svf/svf2. In Docs, translation is triggered automatically.

Design Data

PropertyDb

Property database, manifest, resources.

Index

Basic index generates three downloadable JSON resources:

- manifest
- fields
- properties

Query

results: JSON files matching the query filter.

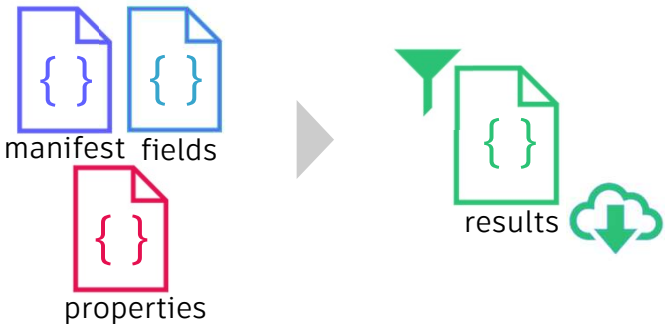
How does it work?

Resources Generated

Resource	Type	Description
manifest	JSON	Manifest for index or query detailing the seed files, svf2 propertyDb used to generate index rows
fields	NDJSON	The set of unique fields (property types) extracted for an index or query
properties	NDJSON	The raw objects property values for the index.
results	NDJSON	The object property results from a query executed.

NDJSON = new-line delimited JSON

Model Properties



Index

Basic index generates three downloadable JSON resources:

- manifest
- fields
- properties

Query

results: JSON files matching the query filter.

How does it work?

Index Endpoints

	Endpoints	
Index	POST	**indexes:batch-status**
	GET	indexes/**:indexId**
	GET	indexes/:indexId/**manifest** 
	GET	indexes/:indexId/**fields** 
	GET	indexes/:indexId/**properties** 
Query	POST	indexes/:indexId/**queries**
	GET	indexes/:indexId/queries/**:queryId**
	GET	indexes/:indexId/queries/:queryId/**properties** 

- 8 endpoints

How does it work?

Index Endpoints for Creation

	Endpoints	
Index	POST	**indexes:batch-status**
	GET	indexes/**:indexId**
	GET	indexes/:indexId/**manifest**
	GET	indexes/:indexId/**fields**
	GET	indexes/:indexId/**properties**
Query	POST	indexes/:indexId/**queries**
	GET	indexes/:indexId/queries/**:queryId**
	GET	indexes/:indexId/queries/:queryId/**properties**




- Create basic index – “lazy”
  - First time – start the indexing job and cache the results
  - Once executed – use the cache
  - Cached 30 days since the last used
- Poll for progress
  - state: PROCESSING, FINISHED, FAILED
- Response JSON is identical
- stats: objects (# of object)
- Create 3 downloadable json.gz resources
  - Manifest, fields, properties

Ex. Create basic index – POST indexes:batch-status

```
curl --request POST 'https://developer.api.autodesk.com/construction/index/v2/projects/f83c      ...      /indexes:batch-status' \
     --header 'Authorization: Bearer ****' \
     --header 'Content-Type: application/json' \
     --data-raw '{
        "versions": [
            {
                "versionUrn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4"
            }
        ]
    }'
```

request

response

```
[
    {
        "projectId": "f83cef12-deef-4771-9feb-4f85643e3c46",
        "indexId": "qTmPiKJZ7siqxkTNpWGANw",
        "type": "INDEX",
        "state": "PROCESSING",
        "selfUrl": "https://developer.api.autodesk.com/construction/index/v2/projects/f83cef12-deef-4771-9feb-4f8564
        "versionUrns": [
            "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4"
        ],
        "updatedAt": "2021-08-19T08:21:13.8771187+00:00",
        "retryAt": "2021-08-27T14:28:28.8382067+00:00",
        "stats": null,
        "manifestUrl": null,
        "fieldsUrl": null,
        "propertiesUrl": null
    }
]
```

Ex. Polling for Progress - GET indexes/:indexId

```
curl --request GET 'https://developer.api.autodesk.com/construction/index/v2/projects/  ···  /indexes/qTmPiKJZ7siqxkTNpWGANw' \
     --header 'Authorization: Bearer ****'
```

request

response


```
{
    "projectId": "f83cef12-deef-4771-9feb-4f85643e3c46",
    "indexId": "qTmPiKJZ7siqxkTNpWGANw",
    "type": "INDEX",
    "state": "FINISHED",
    "selfUrl": "https://developer.api.autodesk.com/construction/index/v2/proje ··· /indexes/qTmPiKJZ7siqxkTNpWGANw",
    "versionUrns": [
        "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4"
    ],
    "updatedAt": "2021-08-19T08:21:13.8771187+00:00",
    "retryAt": "2021-08-27T14:31:55.1444684+00:00",
    "stats": {
        "objects": 33097
    },
    "manifestUrl": "https://developer.api.autodesk.com/construction/index/v2/pro ··· 46/indexes/qTmPiKJZ7siqxkTNpWGANw/manifest",
    "fieldsUrl": "https://developer.api.autodesk.com/construction/index/v2/proje ··· /indexes/qTmPiKJZ7siqxkTNpWGANw/fields",
    "propertiesUrl": "https://developer.api.autodesk.com/construction/index/v2/p ··· 3c46/indexes/qTmPiKJZ7siqxkTNpWGANw/properties"
}
```

How does it work?

Index Endpoints for Download

	Endpoints	
Index	POST	**indexes:batch-status**
	GET	indexes/**:indexId**
	GET	indexes/:indexId/**manifest**
	GET	indexes/:indexId/**fields**
	GET	indexes/:indexId/**properties**
Query	POST	indexes/:indexId/**queries**
	GET	indexes/:indexId/queries/**:queryId**
	GET	indexes/:indexId/queries/:queryId/**properties**

- (Optional) download
  - manifest
  - fields
  - properties

Manifest (.json)


```
{
    "schema": "2.0.0",
    "projectId": "f83cef12-deef-4771-9feb-4f85643e3c46",
    "status": "Succeeded",
    "createdAt": "2021-07-23T08:56:07.0868303+00:00",
    "seedFiles": [
        {
            "lineageId": "a19f7db",
            "lineageUrn": "urn:adsk.wipprod:dm.lineage:DyTWutcvTcOLUNUARxcTzQ",
            "versionUrn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4",
            "databases": [
                {
                    "id": "3747dccf",
                    "offsets": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2l  ...  yc2lvbj04/output/Resource/objects_offs.json.gz",
                    "attributes": "urn:adsk.viewing:fs.file:dXJuOmFkc2su  ...  dmVyc2lvbj04/output/Resource/objects_attrs.json.gz",
                    "values": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2l  ...  c2lvbj04/output/Resource/objects_vals.json.gz",
                    "mapping": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2l  ...  yc2lvbj04/output/Resource/objects_avs.json.gz",
                    "ids": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2lwcHJ  ...  vbj04/output/Resource/objects_ids.json.gz"
                }
            ],
            "views": [
                {
                    "id": "e7fda9d5",
                    "urn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4",
                    "is3d": true,
                    "viewableName": "{3D}",
                    "viewableId": "0935d8b2-149b-4a0d-b816-863f0d595a20-000bcd64",
                    "viewableGuid": "00cd2da3-fbfa-44a9-7a33-cad0bc4720cb"
                },
                {
                    "id": "12fcb372",
                    "urn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4",
                    "is3d": true,
                    "viewableName": "New Construction",
                    "viewableId": "c884ae1b-61e7-4f9d-0001-719e20b22d0b-00120bb2",
                    "viewableGuid": "4a966c2a-ead6-65c3-4f98-273dd7543047"
                }
            ]
        },
    ],
    "errors": [],
    "stats": {
        "objects": 33097,
        "contentLength": 1881318
    }
}
```


Lineage & Version

SVF2 Prop DB Resource URNs

Viewables

Index object count and byte size

Index Fields (json.gz)


```
{"key":"p153cb174","category":"__name__","type":"String","name":"name","uom":null}
{"key":"p74a9a490","category":"__document__","type":"String","name":"schema_name","uom":null}
{"key":"p137c14f2","category":"__document__","type":"String","name":"schema_version","uom":null}
{"key":"p1490bcea","category":"__document__","type":"Boolean","name":"is_doc_property","uom":null}
{"key":"p5eddc473","category":"__category__","type":"String","name":"Category","uom":null}
{"key":"p00723fa6","category":"Identity Data","type":"String","name":"Design Option","uom":null}
{"key":"pe8094f29","category":"Other","type":"String","name":"Project Issue Date","uom":null}
{"key":"p50756a0d","category":"Other","type":"String","name":"Client Name","uom":null}
{"key":"p32791eb0","category":"Other","type":"String","name":"Project Address","uom":null}
{"key":"pbf75ced9","category":"Other","type":"String","name":"Project Name","uom":null}
{"key":"p8213f1ad","category":"Other","type":"String","name":"Project Number","uom":null}
{"key":"pa7275c45","category":"__categoryId__","type":"Integer","name":"CategoryId","uom":null}
{"key":"p93e93af5","category":"__parent__","type":"DbKey","name":"parent","uom":null}
{"key":"p1d45bc4f","category":"Dimensions","type":"Double","name":"Computation Height","uom":"ft"}
{"key":"pe01bd7ef","category":"Extents","type":"String","name":"Scope Box","uom":null}
{"key":"p9fffb245","category":"Materials and Finishes","type":"Integer","name":"Color","uom":null}
{"key":"p1b3b6224","category":"Materials and Finishes","type":"String","name":"Transparency","uom":null}
{"key":"pd9fcab30","category":"Materials and Finishes","type":"Boolean","name":"Glow","uom":null}
{"key":"pf62e5a3c","category":"Structural","type":"Double","name":"Structural Framing Length Roundoff","uom":"ft"}
```


Field Key → SQL column name	Category	Type	Name	UOM (Unit of Measurement)

Basic Properties (json.gz)

Object IDs, keys for index manifest JSON. (Type and ChangeType IF diff index)

Index field property values

Property hash + properties ignored when calculating the hash

SVF2 geometry hash

Bounding box min/max for viewable object

Viewables containing the object (manifest keys to viewable)

How does it work?

Index Endpoints for Query

	Endpoints	
Index	POST	**indexes:batch-status**
	GET	indexes/**:indexId**
	GET	indexes/:indexId/**manifest**
	GET	indexes/:indexId/**fields**
	GET	indexes/:indexId/**properties**
Query	POST	indexes/:indexId/**queries**
	GET	indexes/:indexId/queries/**:queryId**
	GET	indexes/:indexId/queries/:queryId/**properties**




- Build and run query.
  - Index queries are described using custom JSON schema, (which is converted to a filter expression. AWS S3 Select)
  - Columns can be restricted. Can use alias (have different header)
- Poll for progress.
  - state: PROCESSING, FINISHED, FAILED

Index Fields

```
// Forge viewer element display name field
{"key":"p153cb174","category":"__name__","type":"String","name":"name","uom":null}
// Revit category name field
{"key":"p20d8441e","category":"__category__","type":"String","name":"_RC","uom":null}
// Revit family name field
{"key":"p30db51f9","category":"__category__","type":"String","name":"_RFN","uom":null}
// Revit type name field
{"key":"p13b6b3a0","category":"__category__","type":"String","name":"_RFT","uom":null}
```

Sample Query:
Get Revit Classification with Column Transform

Query

```
{
    "query": {
        "$and": [
            { "$notnull": "s.props.p20d8441e" },
            { "$notnull": "s.props.p30db51f9" },
            { "$notnull": "s.props.p13b6b3a0" },
            { "$gt": [{ "$count": "s.views" }, 0] }
        ]
    },
    "columns": {
        "s.svf2Id": true,
        "lmvName": "s.props.p153cb174",
        "revitCategory": "s.props.p20d8441e",
        "revitFamily": "s.props.p30db51f9",
        "revitType": "s.props.p13b6b3a0",
        "s.views": true
    }
}
```

Row has Revit classification

Views array has count more than 0

Columns define alias

Equivalent in S3 SQL

```
select
    s.svf2Id,
    s.props.p153cb174 as lmvName,
    s.props.p20d8441e as revitCategory,
    s.props.p30db51f9 as revitFamily,
    s.props.p13b6b3a0 as revitType,
    s.views
from S3Object[*] s
where
    s.props.p20d8441e is not null and
    s.props.p30db51f9 is not null and
    s.props.p13b6b3a0 is not null and
    count(s.views) > 0
```

How does it work?

Index Endpoint to Download Query Results

	Endpoints	
Index	POST	**indexes:batch-status**
	GET	indexes/**:indexId**
	GET	indexes/:indexId/**manifest**
	GET	indexes/:indexId/**fields**
	GET	indexes/:indexId/**properties**
Query	POST	indexes/:indexId/**queries**
	GET	indexes/:indexId/queries/**:queryId**
	GET	indexes/:indexId/queries/:queryId/**properties**

- Download the query results
  - use the queryResultsUrl in query call or query id to download the index rows which match the submitted query expression.
  - Result: line delimited JSON
  - a sub-set of the property index rows
  - Format is the exactly the same as properties we saw earlier.

How does it work?

Diff Endpoints

	Endpoints	
Diff	POST	diffs:batch-status
	GET	diffs/:diffId
	GET	diffs/:diffId/manifest
	GET	diffs/:diffId/fields
	GET	diffs/:diffId/properties
Query	POST	diffs/:diffId/queries
	GET	diffs/:diffId/queries/:queryId
	GET	diffs/:diffId/queries/:queryId/properties

- Diff - The steps are the same as Index
- Specify two version urn's to compare:

```
{
  "diffs": [
    {
      "prevVersionUrn": "urn:adsk.w
      "curVersionUrn": "urn:adsk.wi
    }
  ]
}
```

- stats: add, removed, modified

Type if diff index

Previous (prev) object embedded in current row. Lineage manifest key & viewer id.

Array of property keys which have values different to current

Previous bounding boxes, hashes and viewable keys in manifest

Basic Index Row vs. Diff Index Row

Description	Current version	Previous version
IDs	s.svf2Id s.**externalId**	
Change type, previous vs. current		s.**type** s.**changeType**
lineage version info, SVF2 database URNs	s.lmvId s.lineageId s.databaseId	s.**prev**.lmvId s.prev.lineageId s.prev.databaseId
Property values	s.props.* s.propsHash s.propsIgnored.*	s.prev.props.* s.prev.propsHash s.prev.propsIgnored.*
Geometry hash and bounding box values IF viewable	s.geomHash s.bboxMin.x s.bboxMin.y s.bboxMin.z s.bboxMax.x s.bboxMax.y s.bboxMax.z	s.prev.geomHash s.prev.bboxMin.x s.prev.bboxMin.y s.prev.bboxMin.z s.prev.bboxMax.x s.prev.bboxMax.y s.prev.bboxMax.z
Viewable keys IF viewable	s.views s.views[*i*]	s.prev.views s.prev.views[*i*]

JSON Abstract Syntax Tree → S3 Select(AWS)

```
$not
$and
$or
$gt
$lt
$eq
$le
$ge
```

```
$like
$between
$in
$contains
$isnull
$notnull
$add
$sub
$mul
$div
```

```
$cat
$coalesce
$mod
$cast
$nullif
$date_add
$date_diff
$extract
$substring
$to_string
$to_timestamp
```

```
$char_length
$lower
$upper
$count
$sum
$avg
$min
$max
$trim
$utcnow
$case
```

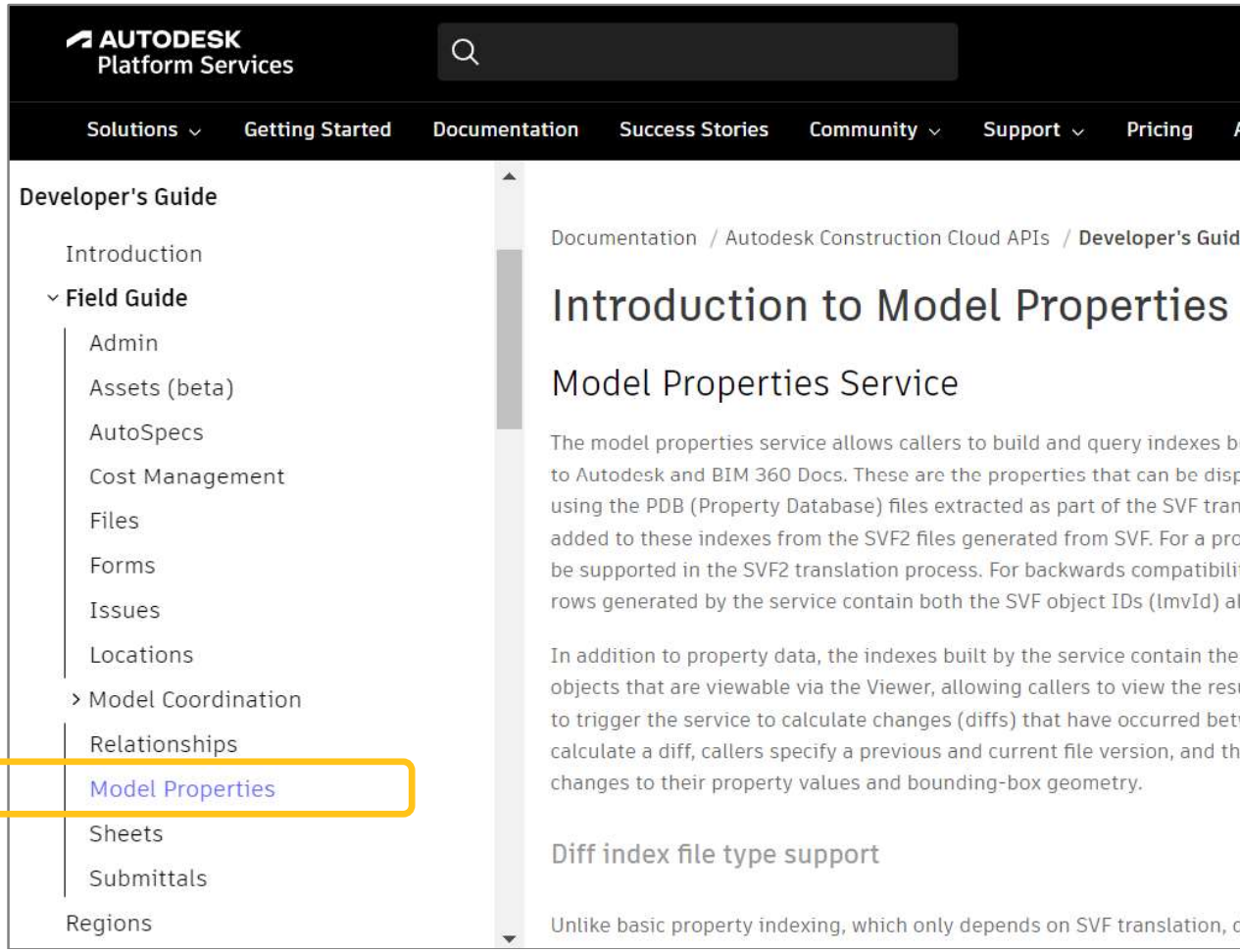
Developer Resources

Model Properties API

Developer Resources

Documentation

- Field Guide
  - Introduction to Model Properties
- Step-by-Step Tutorials
  - Index Querying
  - Tracking Changes
  - Query Language Reference
- Reference Guide
  - Index
  - Diff



Developer Resources

- Code Samples on GitHub
  - Postman Collection (correspond to three Step-by-Step tutorials)
  - Model Properties API Walkthrough in PowerShell Core (scripting to explore query language)
  - Element Filtering and Partial Model Load (Integration with Viewer)
  - Compare Two Versions (Integration with Viewer)

- Blog Post
  - “BIM 360/ACC Model Properties API”
    https://aps.autodesk.com/blog/bim-360acc-model-properties-api
    includes links to the resources
  - Search more with “Model Properties”, e.g.,
    https://aps.autodesk.com/blog/model-properties-api-vs-model-derivative-api

Common Questions

Methods to Access Design Data

Model Derivative	Model Properties	Design Automation	Data Exchange	AEC Data Model
				
Platform	ACC/BIM 360	Platform	Autodesk Docs	Autodesk Docs[*2]
File based	File based	File based/RCM	Cloud hosted	Cloud hosted
Entire model	Entire model	Entire model Revit	Partial model Revit	Entire model Revit
Light weight query	Full query/filter	Control same as desktop add-in	Flexible query (GraphQL)	Flexible query (GraphQL)
Read	Read	Read/write	Read & evolving limited write	Read & longer term evolving limited write
Today	Today	Today	Public beta/Future	Public beta/Future

ACC API Roadmap

~2024 Q1 | 2024 Q2 | 2024 Q3/Q4 | 2024 Q4~

Added ‘24

- Submittals read GA
- Docs file description read
- Automatic region routing for all ACC specific API
- Data Connector schema
- Parameters sharing collections
- Cost batch budgets-contracts link
- BuildingConnected write
- TradeTapp read

Currently Active

- Submittals write
- Optional region header change from APAC to AUS
- BuildingConnected (BC) Webhooks for opportunities and bids
- BC opportunity comments
- BC adding bidders to bid packages

Near Future

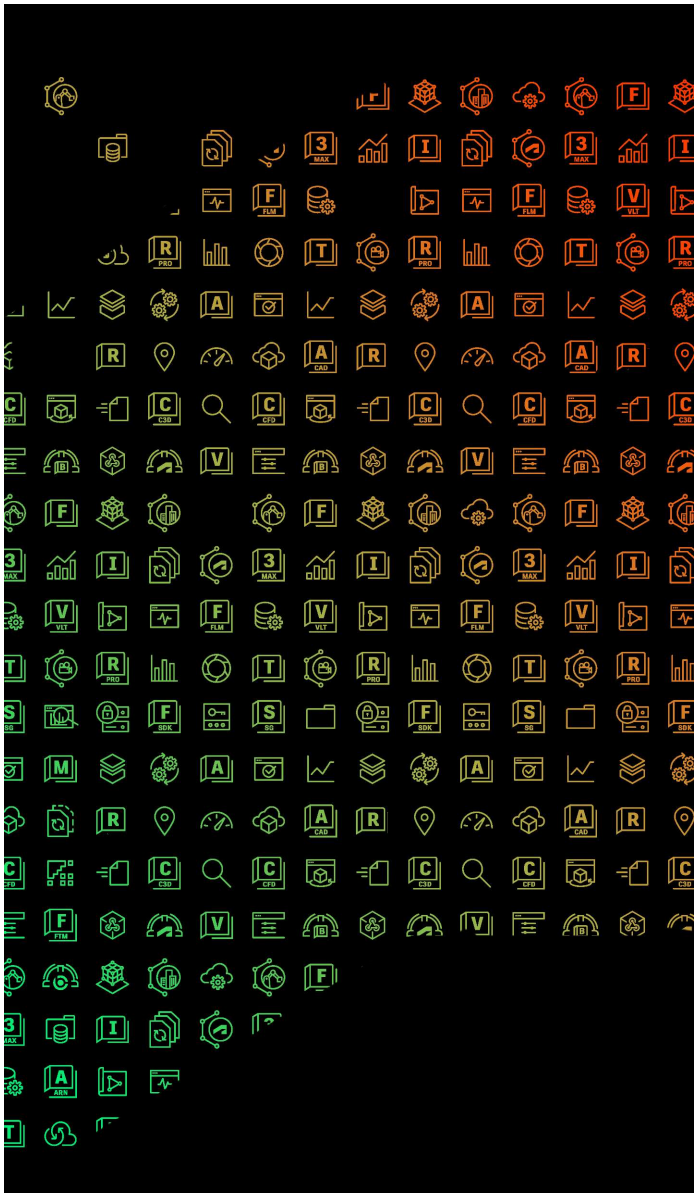
- Review public (currently private beta)
- Markup a.k.a. Issues pushpin
- Model Aggregate
- Project level custom attributes
- Data Connector-multi projects
- Cost - cost item sub items, tax, workflow

Beyond

- Secure Service Account
- Alignment
- Design Collaboration packages/settings
- Cost – budget transfers, documents, payment, activities

Summary

1. Introduction: Shared Services in ACC
2. Relationships API
3. Model Properties API
4. Common Questions
5. What’s Next



Make Anything

https://aps.autodesk.com/get-help
aps.help@autodesk.com