energyADE

CityGML Energy ADE 3.0 Workshop 2026

TU Delft 3Dgeoinfo

UrbäNRG

TU Delft UEI URBAN ENERGY INSTITUTE

A deeper look at the Energy ADE 3.0

(beta 8, as of June 2026)

Giorgio Agugiaro

Delft, 23 June 2026

DUT Driving Urban Transitions

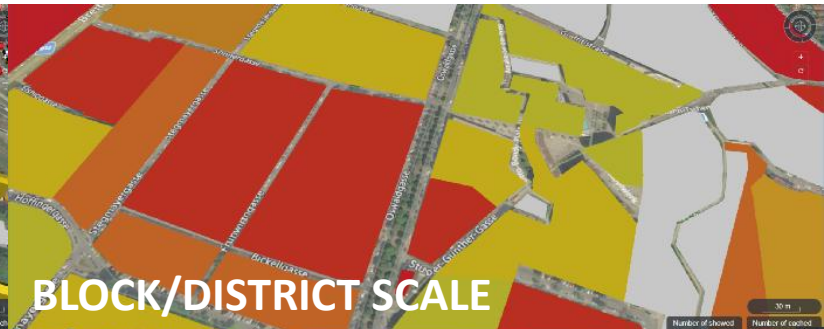
Co-funded by the European Union

NWO

License

Energy ADE

- Ease interoperability for energy data
  - Between different software tools
  - Between different stakeholders
- Define (and store) energy-related data in a standard, open, urban data model
  - Allow for multi-scale energy modelling
- Based on multidisciplinary cooperation between digital twin users, data producers, software vendors, …, and scientists



Energy ADE: What has happened so far...

- 2012: **CityGML 2.0**
  - Still the most used version world-wide
- 2014-2018: **Energy ADE**
  - Used in several national and international projects since its release
- 2019: **Energy ADE 2.0**
  - Minor improvements and semplifications
- 2021: **CityGML 3.0**
  - Major changes and improvements
- From 2024 onwards: **Energy ADE 3.0**
  - Integrate ideas and lessons learned since 2018
  - Developed for CityGML 2.0, but already looking at CityGML 3.0

Agugiaro, G., Benner, J., Cipriano, P., Nouvel, R., 2018. **The Energy Application Domain Extension for CityGML: Enhancing interoperability for urban energy simulations.** *Open Geospatial Data, Software and Standards*, 2018 3:2. Link

Agugiaro, G., Padsala, R., 2025. **A proposal to update and enhance the CityGML Energy Application Domain Extension**. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., 20th 3D GeoInfo Conference, Japan*. X-4/W6-2025, pp. 1-8. Link

Energy ADE 3.0: Main characteristics

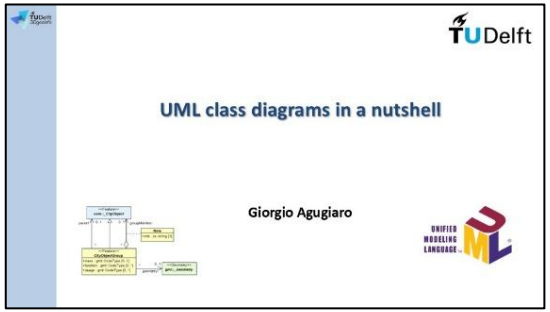
- Keep the overall modular structure of the previous versions

- Updates based on accumulated past experiences
  - Updates, improvements, etc.
  - Backport some concepts/ideas from CityGML 3.0
  - Better alignment with the overall modelling "logic" of CityGML

- Designed for CityGML **2.0**, but keeping in mind the future mapping to CityGML **3.0**

- Avoid past pitfalls!!!
  - Provide good documentation
  - Provide modelling rules
  - Provide sufficient example data
  - Implement it! 😊

CityGML 2.0 "memory refresher"



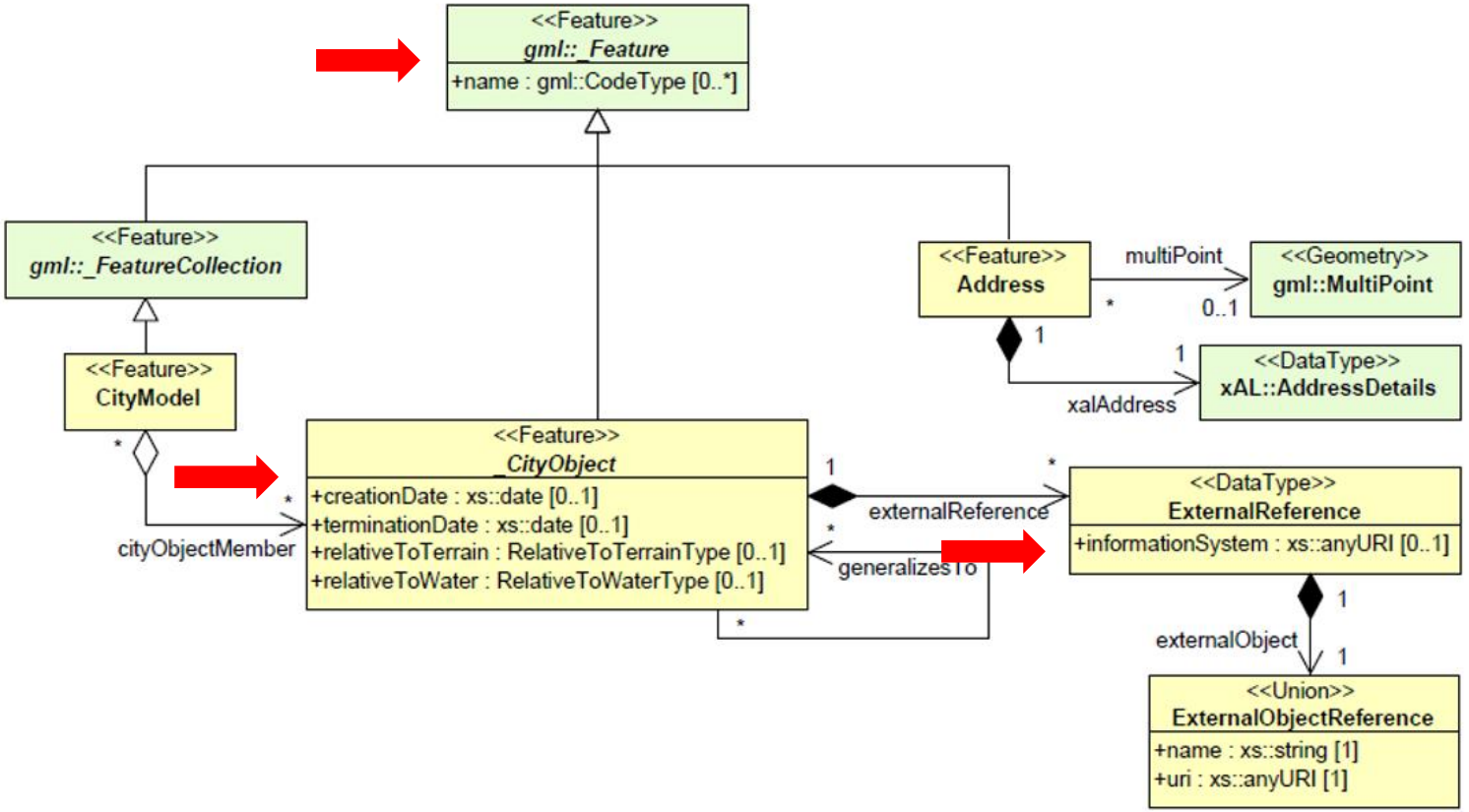
- **WARNING!** No chance you can "survive" if you can't read and understand UML. 😉
  Check free tutorial here

- In GML, ***(Abstract)Feature*** class is the starting point for nearly all subclasses. It comes with *id*, *name*, *description* and *extents* properties
  - One feature object can have multiple geometrical representations at the same time

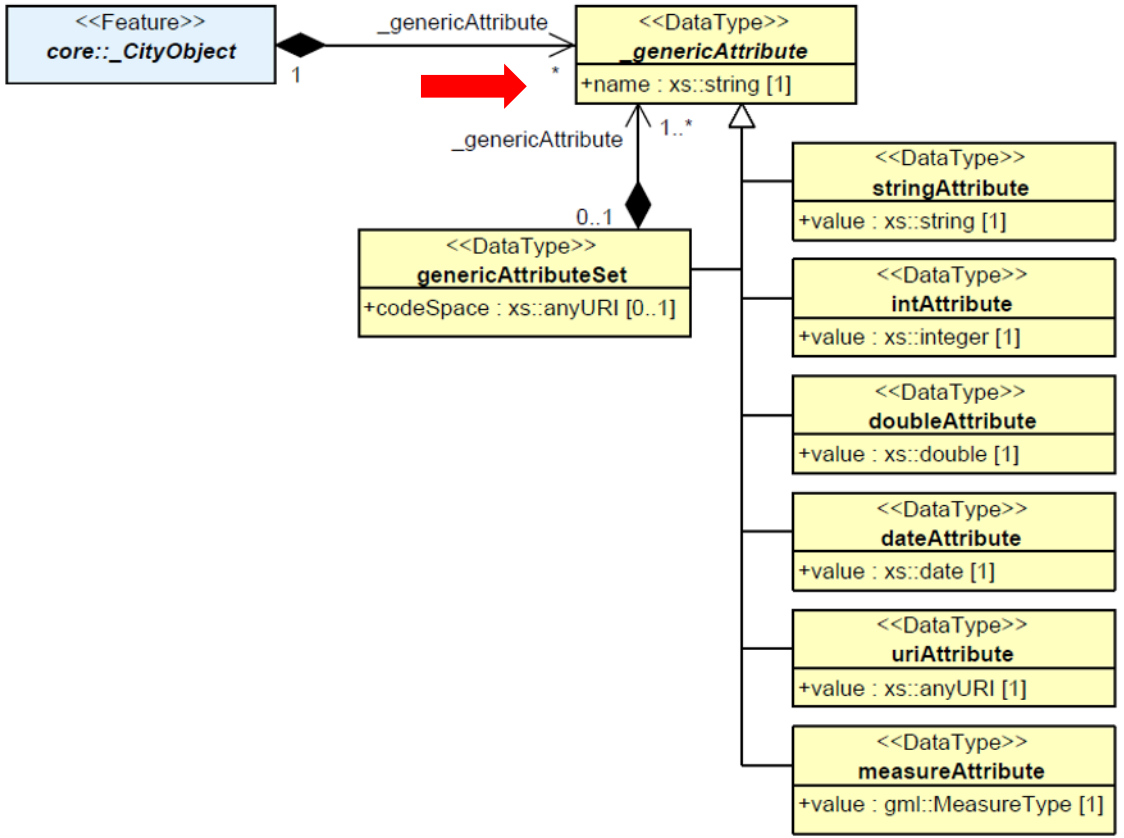
- In CityGML 2.0, abstract ***_CityObject*** is subclass of ***(Abstract)Feature*** and has, among the rest, 0..* **ExternalReference** objects and 0..* **generic attributes**.

- Nearly all classes in CityGML 2.0 are derived from ***_CityObject***

CityGML 2.0 "memory refresher"

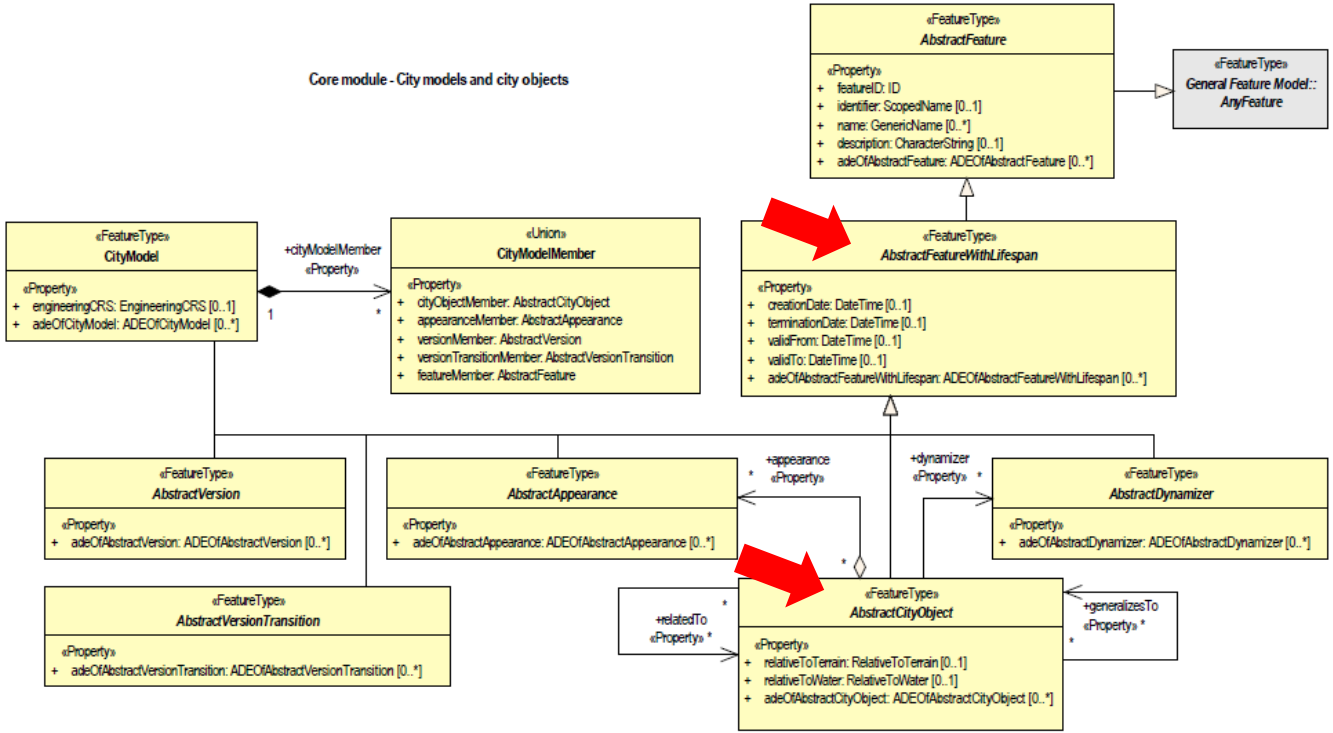


CityGML 2.0 "memory refresher"



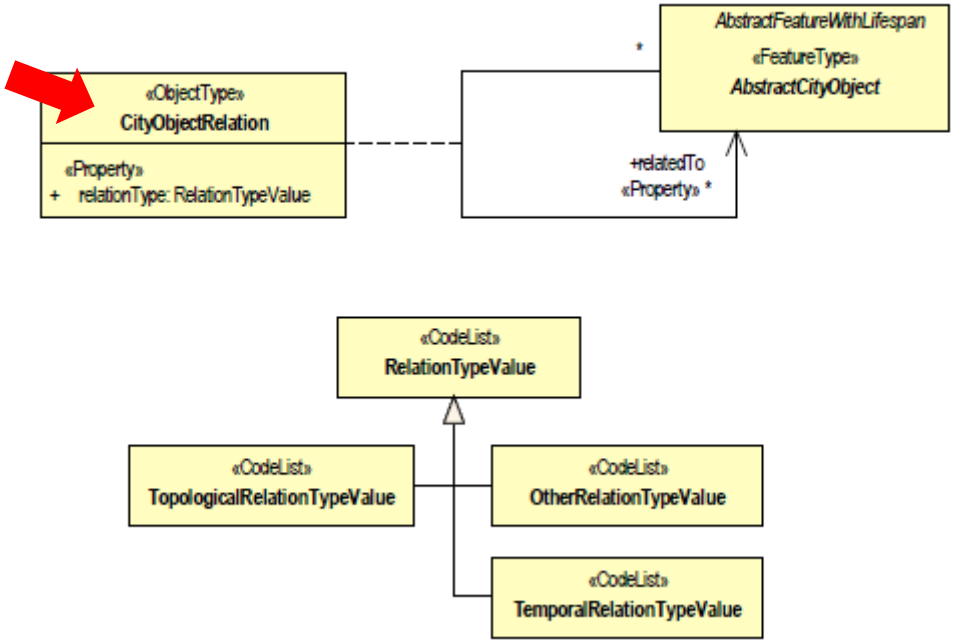
CityGML 3.0 "memory refresher"

- In CityGML 3.0, a few more classes and attributes are added
  - ***AbstractFeature*** has the *identifier* as stable, long-term id (unlike gml/featureID)
  - ***AbstractFeatureWithLifespan*** adds the concept of *validity* (from, to)



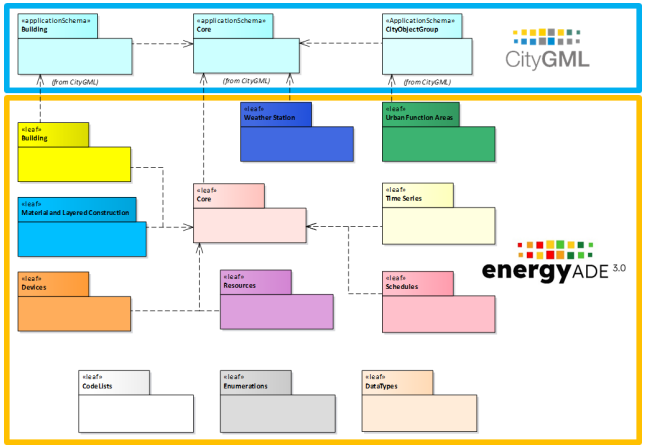
CityGML 3.0 "memory refresher"

- In CityGML 3.0, a few more classes and attributes are added
  - An ***AbstractCityObject*** can be related to any other ***AbstractCityObject***(s) by means of the **CityObjectRelation** class

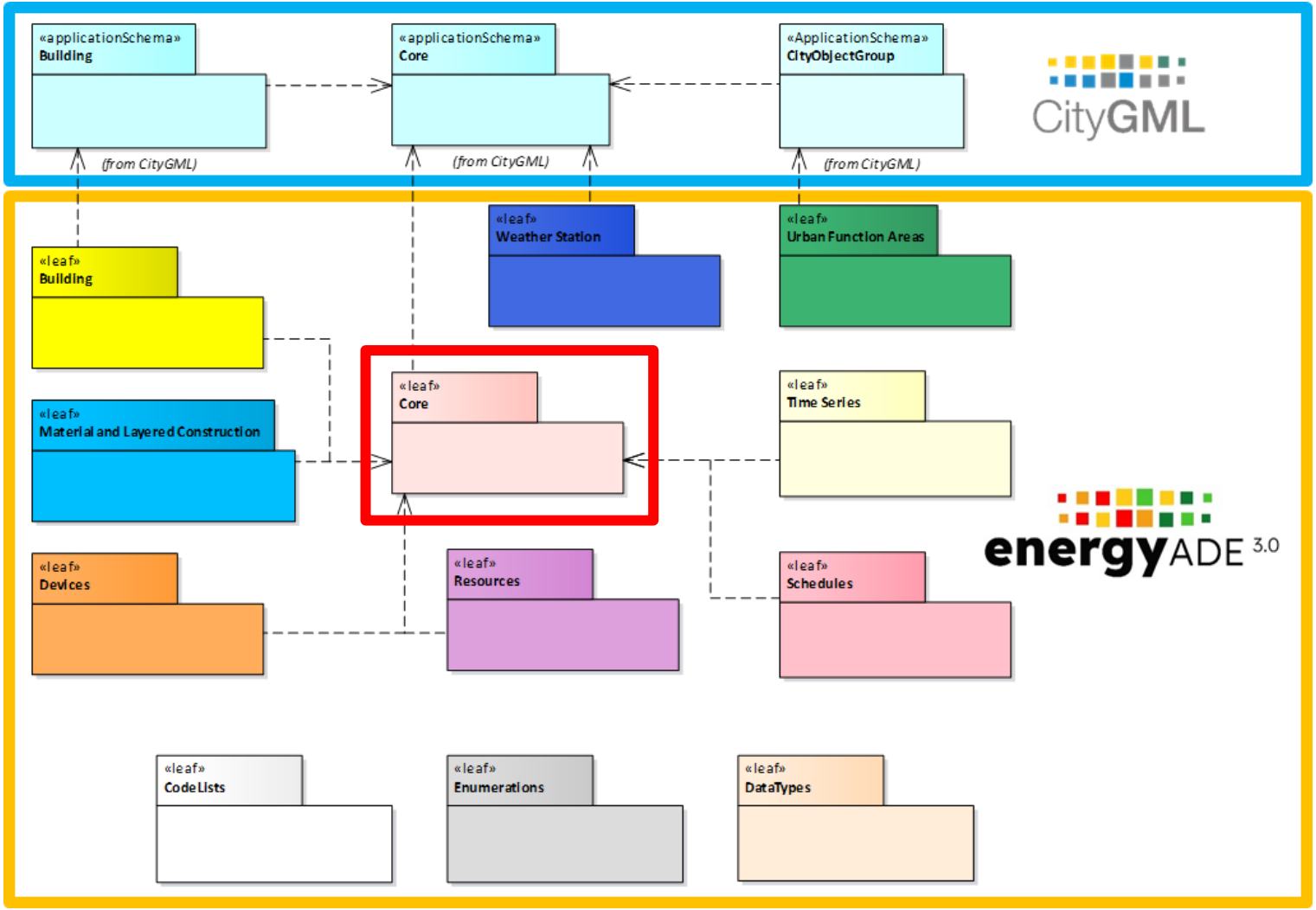


Energy ADE 3.0: Modules

- **Core** module
  - Shared classes (metadata, refurbishments, indicators, network connectivity, ...)
- **Building** module
  - Zones, zone boundaries and openings, occupants
  - Building units, EPCs
- **Devices** module
  - Energy conversion (PV, …), storage, distribution
- **Resources** module
  - Energy, water, and other resources



- **Layered construction** module
  - Simple modelling (e.g. only U-value or g-values) or multi-layered constructions with physical properties of materials
- **Urban function areas** modules
- **Supporting** modules
  - Schedules, time series, weather data, codelists, enumerations

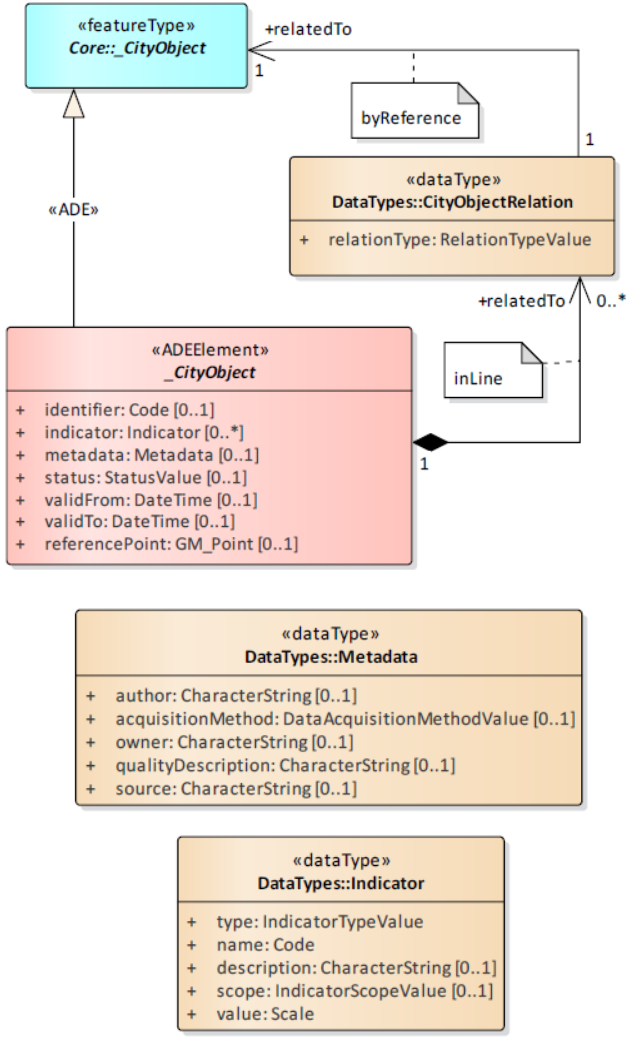


Core module 1/3

- Backports from CityGML 3.0 for ***_CityObject***:
  - Class **CityObjectRelation** for arbitrary relations between cityobjects
  - Identifier, validity interval
  - All city objects can be represented with a **point geometry**

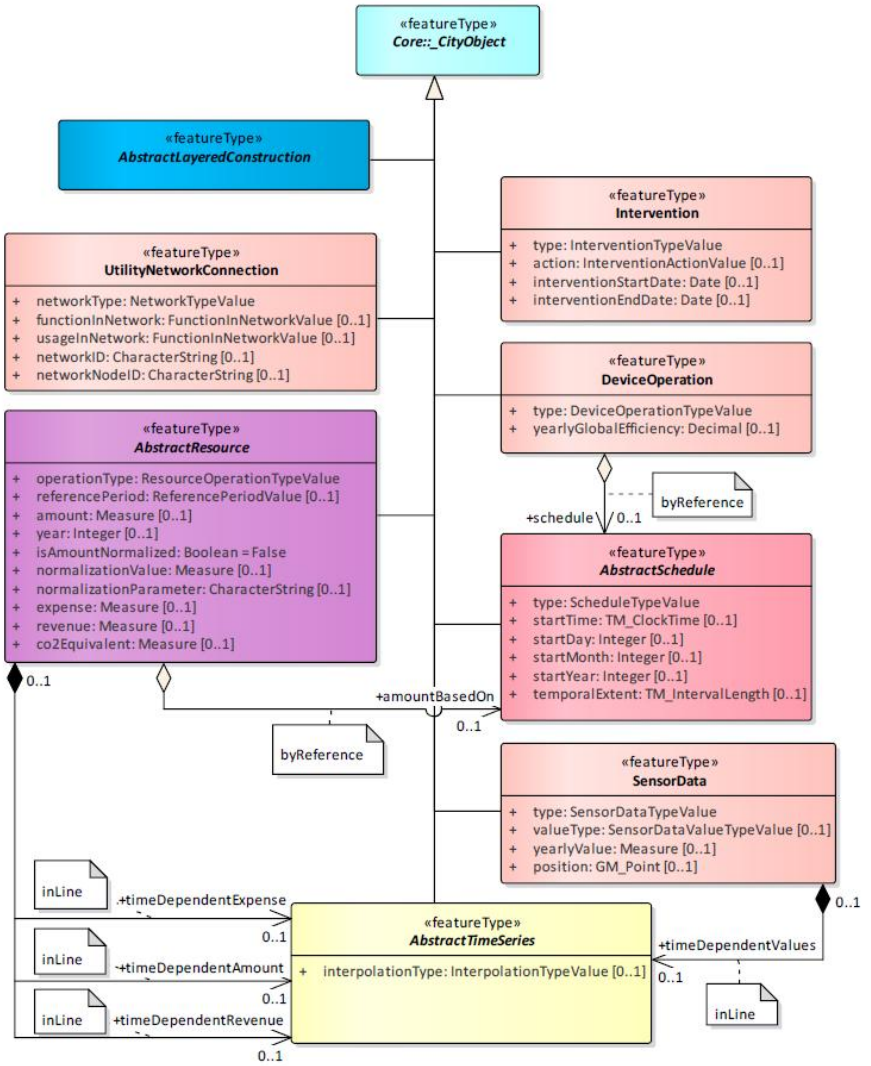
- New properties of ***_CityObject***
  - First, simple, support for *medatata*
  - Possibility to define customized *indicators*
  - Connection to other shared core classes (see later)

- Minor adjustments for future mapping to CityGML 3.0
  - <<Type>> stereotype replaced with <<featureType>>



Core module 2/3

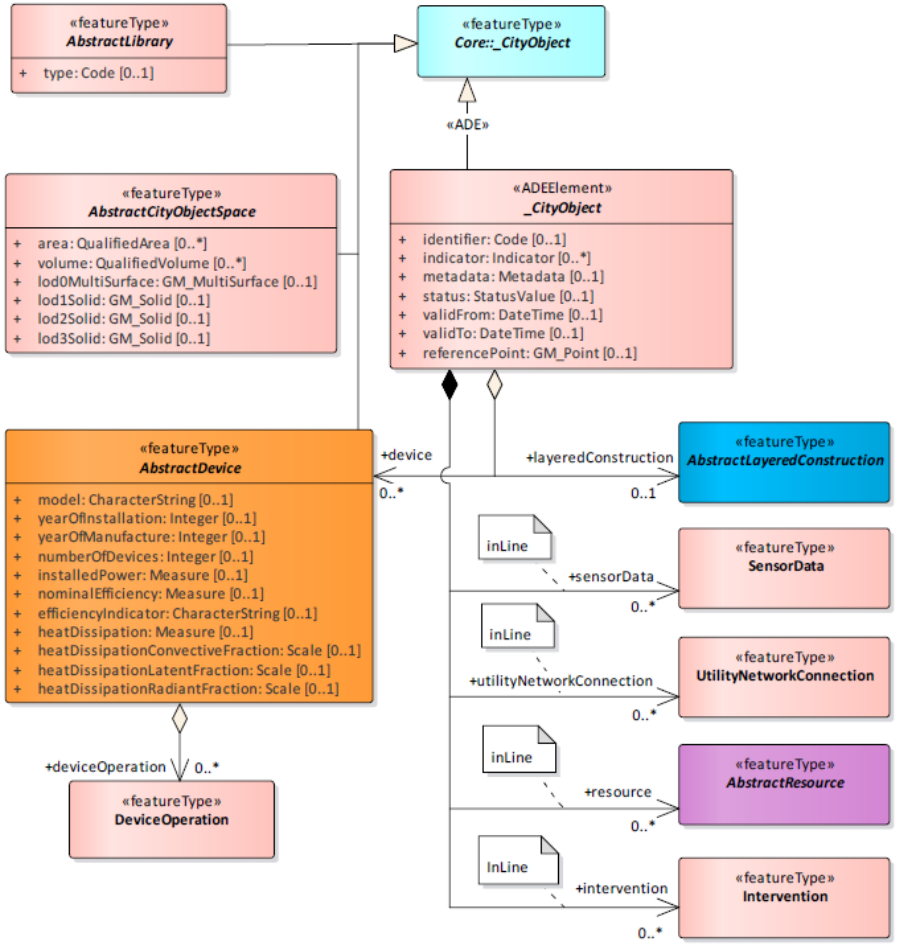
- Contains additional shared "utility" classes, also derived from ***_CityObject***
  - **Intervention** (e.g. Maintenance, Inspection, Refurbishment)
  - **UtilityNetworkConnection**
  - **SensorData**
  - …
- Contains abstract base classes for other modules
  - ***AbstractResource***
  - ***AbstractLayeredConstruction***
  - ***AbstractTimeSeries***
  - ***AbstractSchedule***

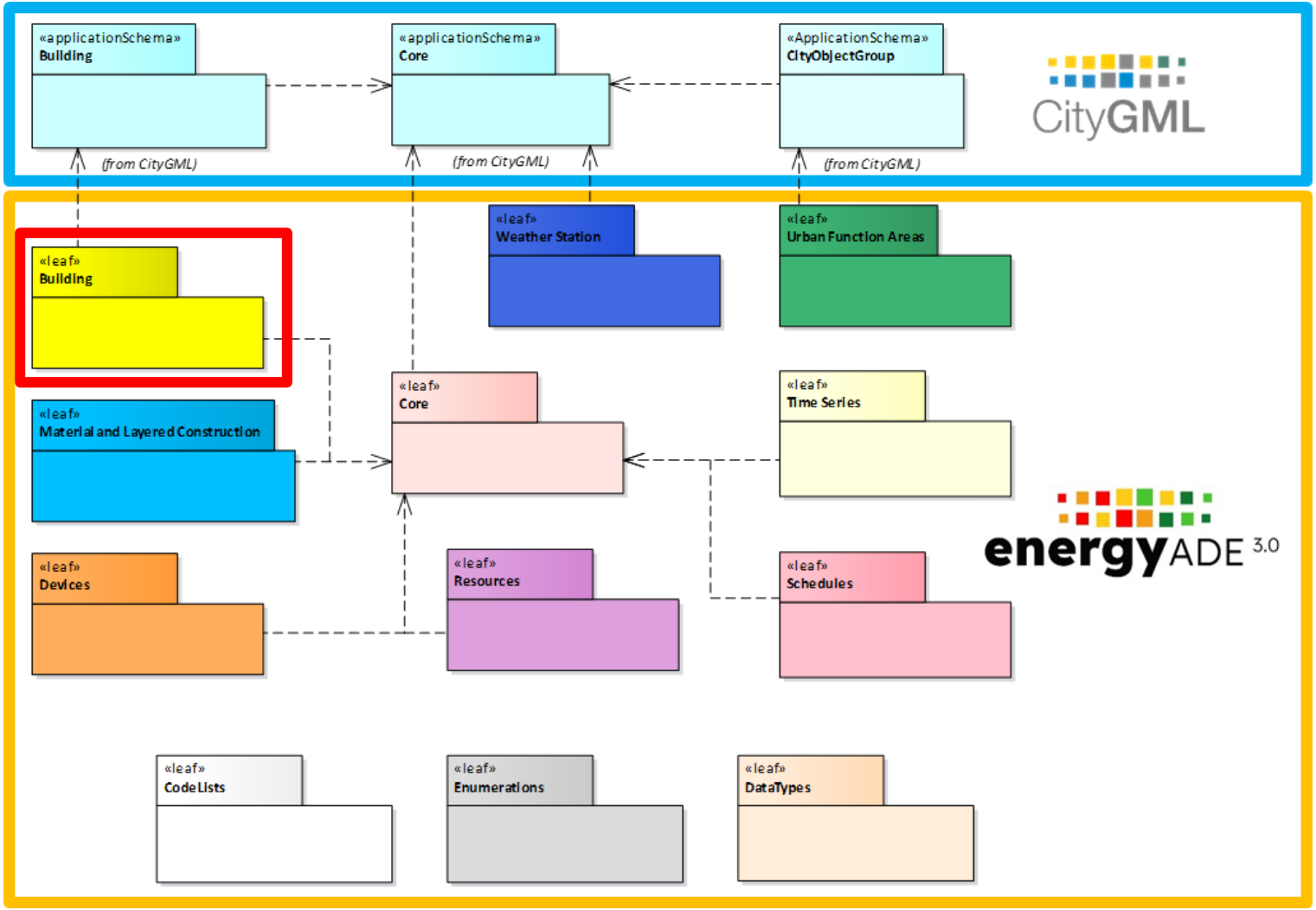


Core module 3/3

- Contains abstract base classes for other modules (ctd)
  - ***AbstractLibrary***
  - ***AbstractCityObjectSpace***
  - ***AbstractDevice***

- ***AbstractCityObjectSpace*** "mimics" the space concept of CityGML 3.0
  - LoD0 multisurface
  - LoD1-3 solids
  - Qualified attributes for area and volume

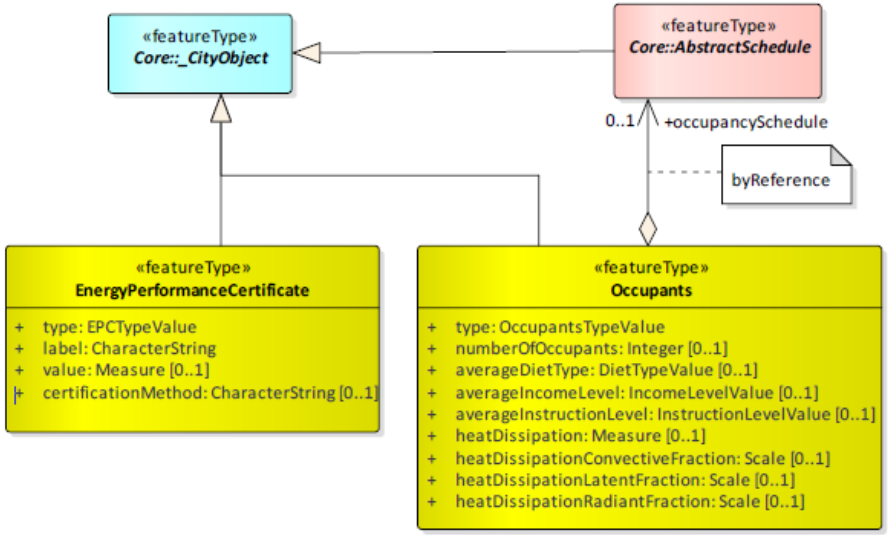




Building module 1/5

- This module allows to model the building thermal zoning and occupants' behaviour
  - Integrates and merges former *Building physics* and *Occupancy* modules from Energy ADE 1.0 and 2.0

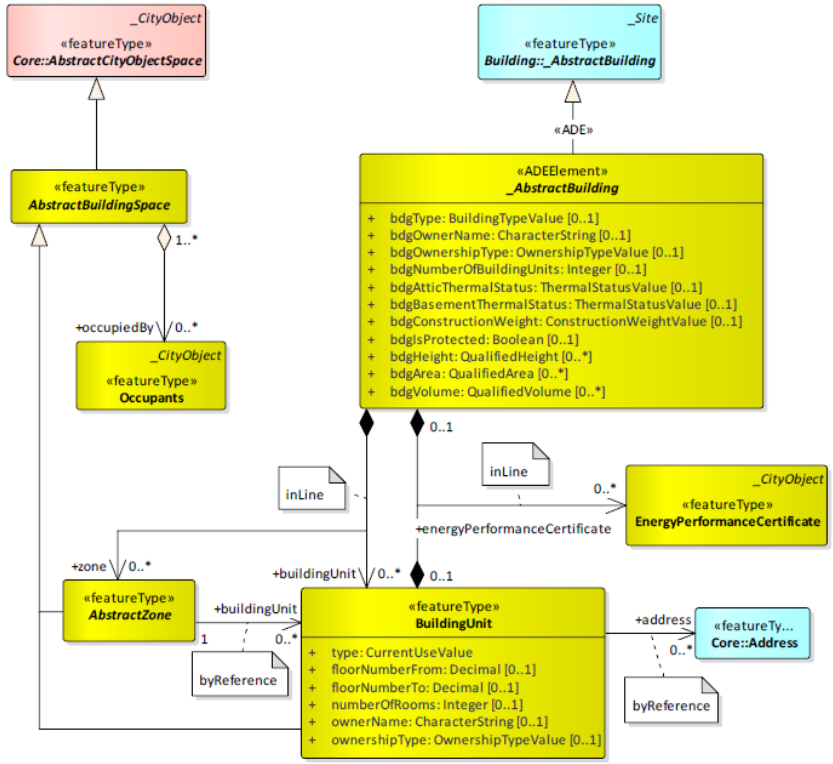
- Contains classes **EnergyPerformanceCertificate** and **Occupants**, needed by other classes in the module



Building module 2/5

- Extends class **Building**
  - *Type*, *Owner's name*, *attic and basement thermal status*, *isProtected*, *qualified attributes* for height, area, volume (netFloorArea, grossFloorArea, heatedFloorArea, etc.)

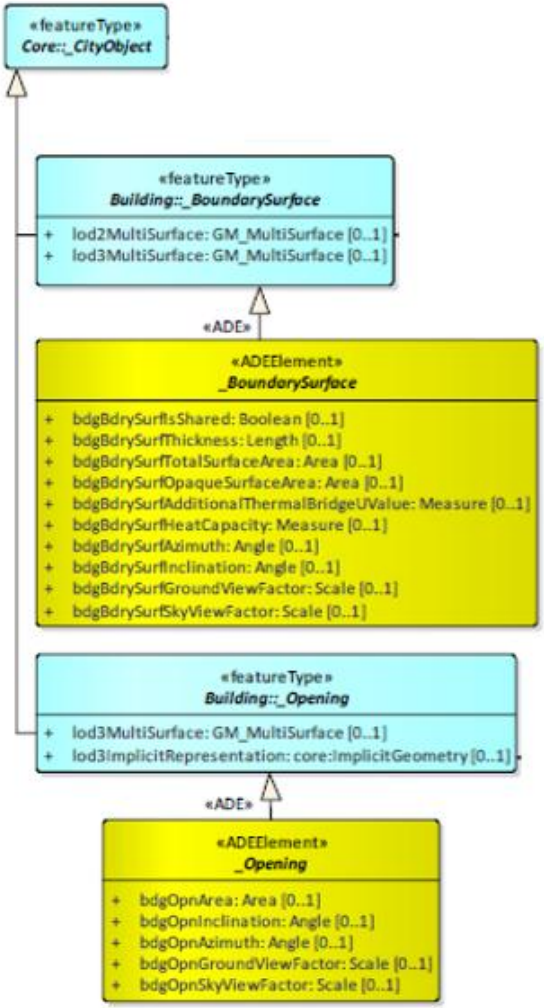
- Defines class **BuildingUnit** and ***AbstractZone***
  - From ***AbstractCityObjectSpace***
  - Both can have **Occupants**



Building module 3/5

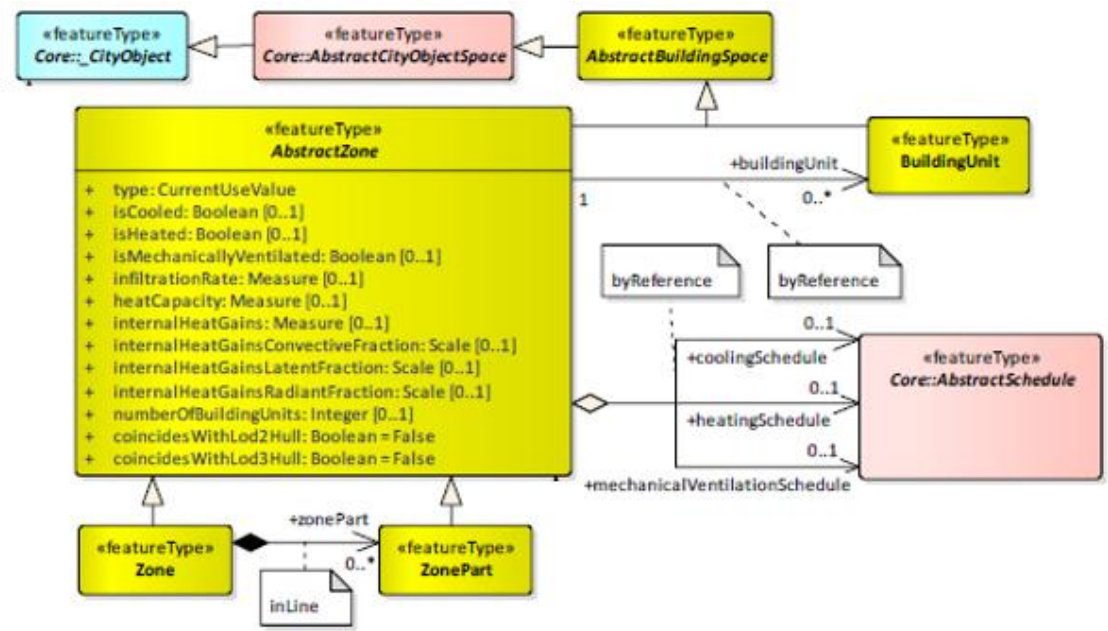
- All (Building) thematic surfaces (**WallSurface**, **RoofSurface**, etc.) receiveoptional additional attributes that can precomputed once and stored for later faster retrieval
  - *Azimuth*, *inclination*, *surface area* (total and opaque), *SVF*, *GVF*, etc.
  - *isShared* can be used to identify shared surfaces between adjacent buildings

- Similarly, class (Building) ***_Openings*** (i.e. **Window** and **Door** classes) receive additional optional attributes



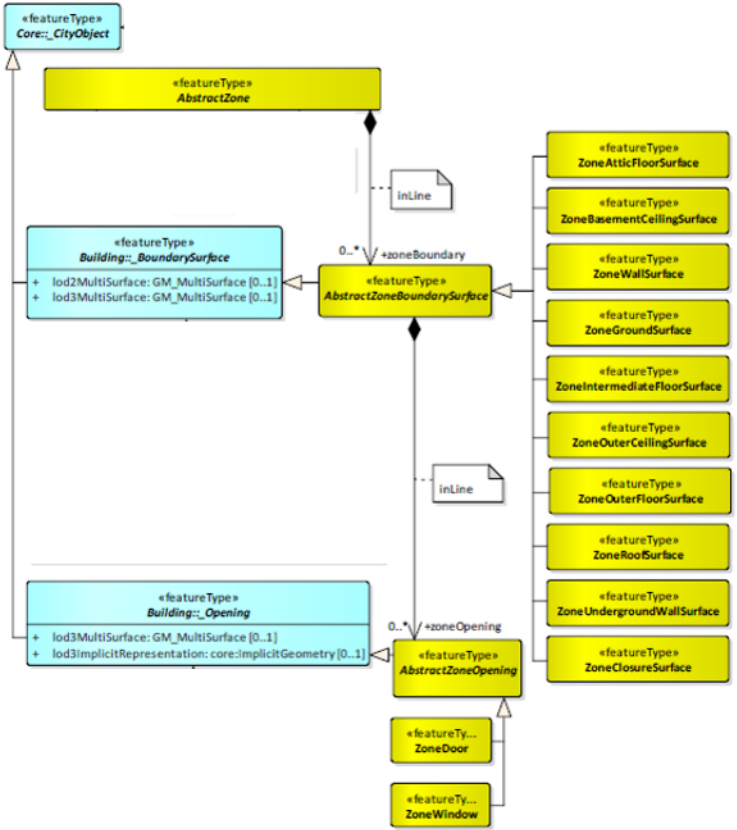
Building module 4/5

- A building volume can be partitioned into **Zones** (and **ZoneParts**), each one characterised by several attributes
  - *type*, *infiltrationRate*, *internalHeatGains*
  - *isHeated*, *isCooled*, ...
  - And the respective **Schedules** for heating, cooling, etc...



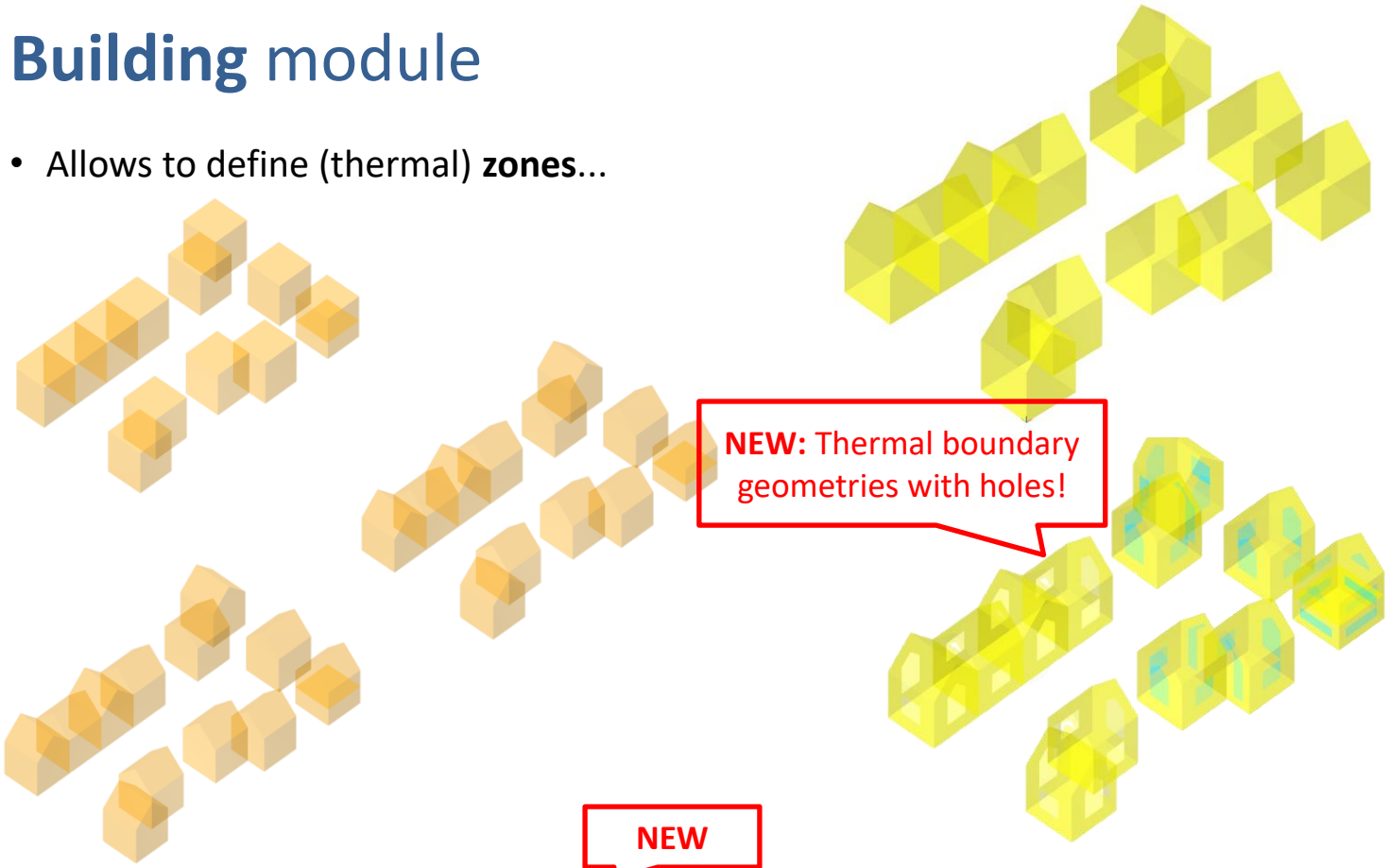
Building module 5/5

- **Zone** (and **ZonePart**) objects can be further modelled via their own thematic surfaces
  - **ZoneWallSurface**, **ZoneRoofSurface**, etc.
  - LoD2-3, no LoD4 allowed (in line with CityGML 3.0)
  - Intentional decoupling of thematic surfaces and zone thematic surfaces
- Zone thematic surfaces can have their own openings (**ZoneWindow** and **ZoneDoor**)
  - Openings must be coplanar with the parent thematic surface and perfectly "fill the hole"
  - LoD2-3, no LoD4 allowed
  - Geometries should not touch any exterior border of the parent thematic surface (i.e. be contained).
    - Please note: this is not a constraint, but a good modelling habit to guarantee compatibility with EnergyPlus and gbXML



Building module

- Allows to define (thermal) **zones**...



NEW: Thermal boundary geometries with holes!

NEW

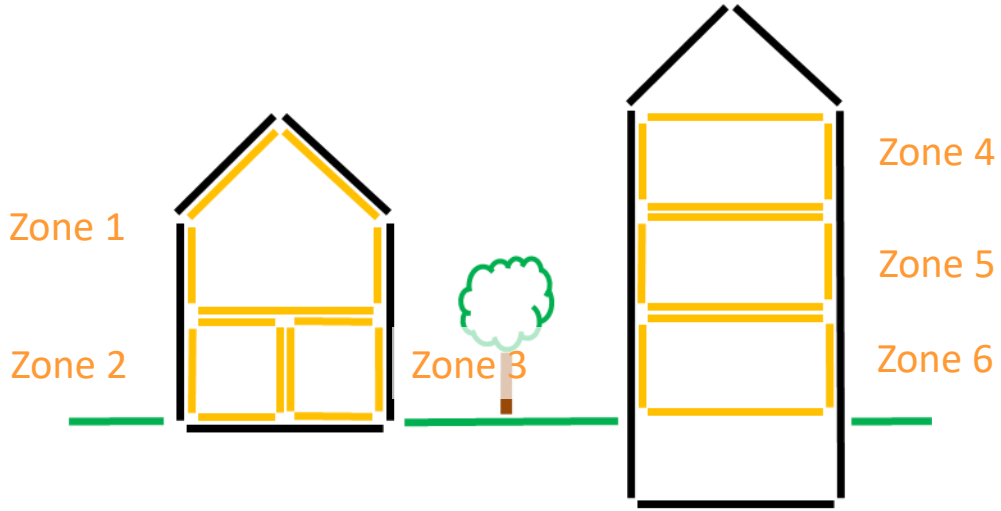
Thermal zones, **LoD1 to LoD3** (solid geometries)

NEW

Thermal boundaries (+ thermal openings), **LoD2 and LoD3**, multi-surface geometries

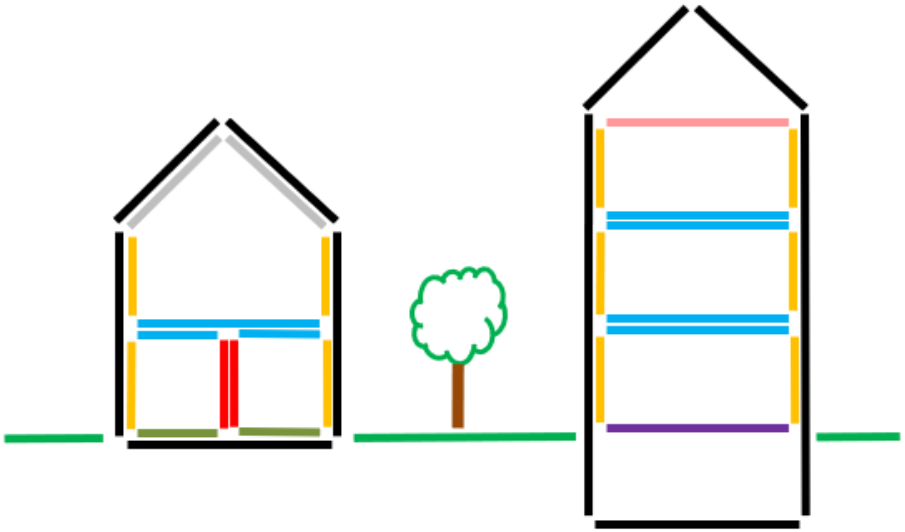
Building module

- Allows to define (thermal) **zones**...



Building module

- Allows to define (thermal) **zones** and their boundaries by means of **thematic surfaces**



ZoneRoofSurface

(exterior) ZoneWallSurface

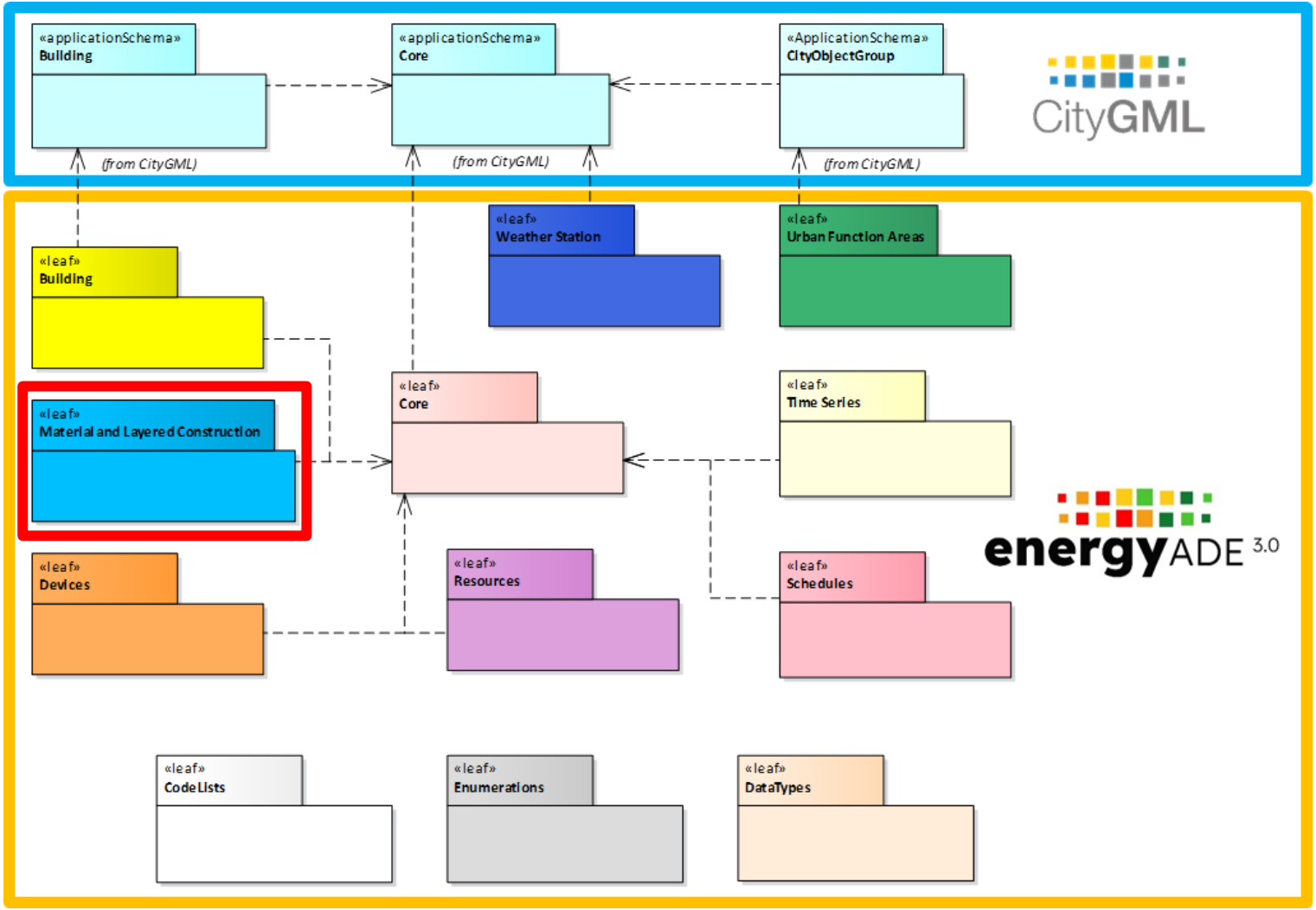
ZoneAtticFloorSurface

ZoneIntermediateFloorSurface

ZoneBasementCeilingSuirface

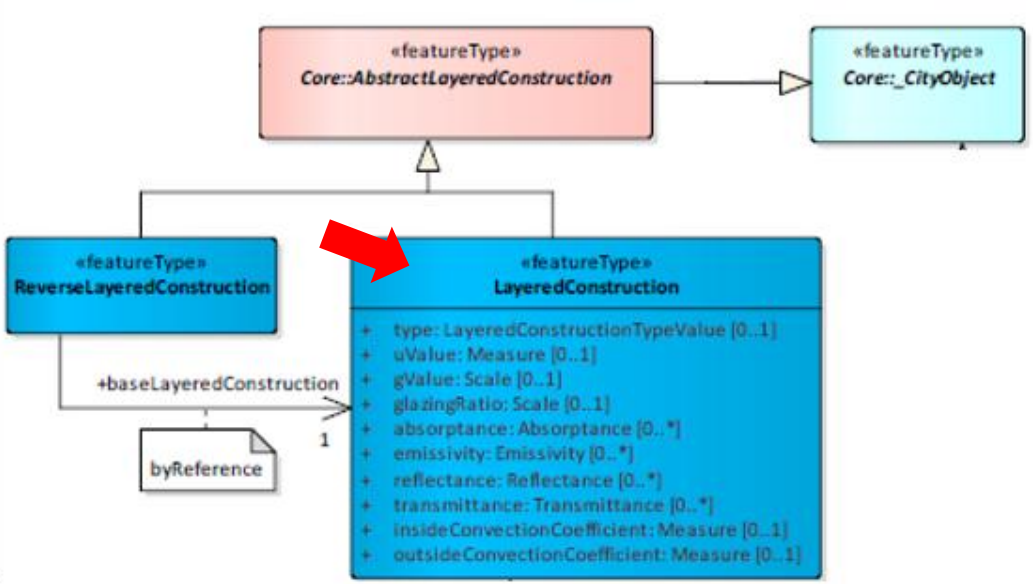
(shared) ZoneWallSurface

ZoneGroundSurface



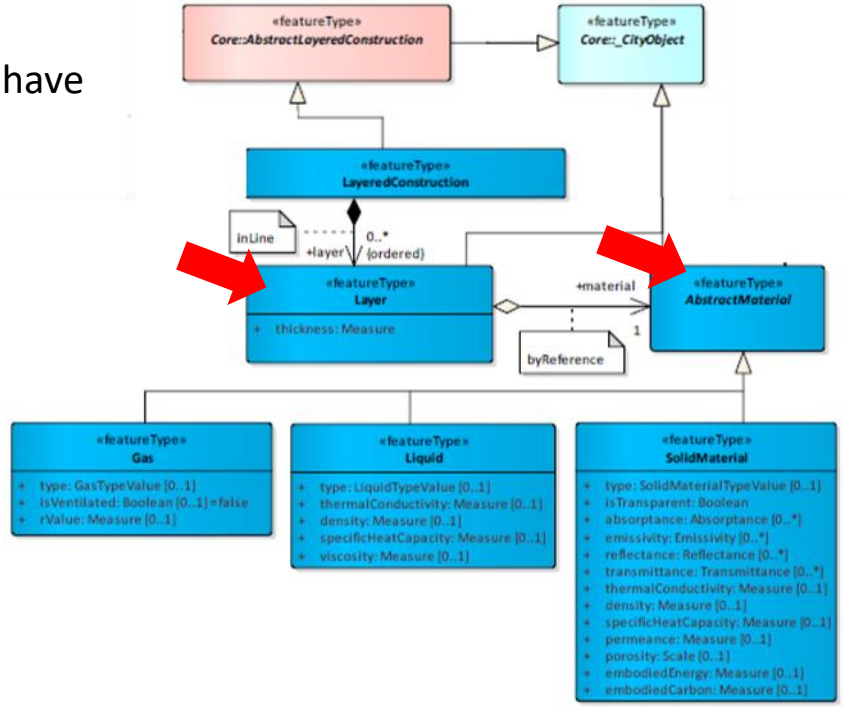
Material and Layered Construction module 1/3

- Contains the classes to model the physical characteristics of layered contructions (walls, etc.)
- Simple modelling via **LayeredConstruction**
  - Global *U-value*, *g-value*, *optical properties*, etc.
- Best used with typical approaches "á la Tabula"



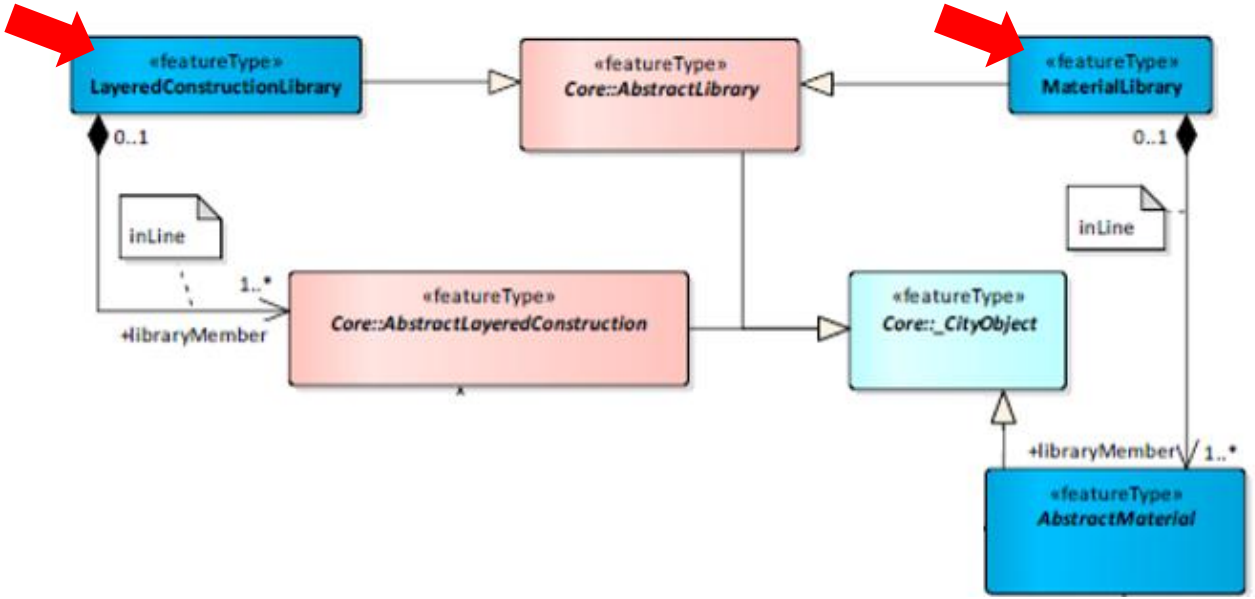
Material and Layered Construction module 2/3

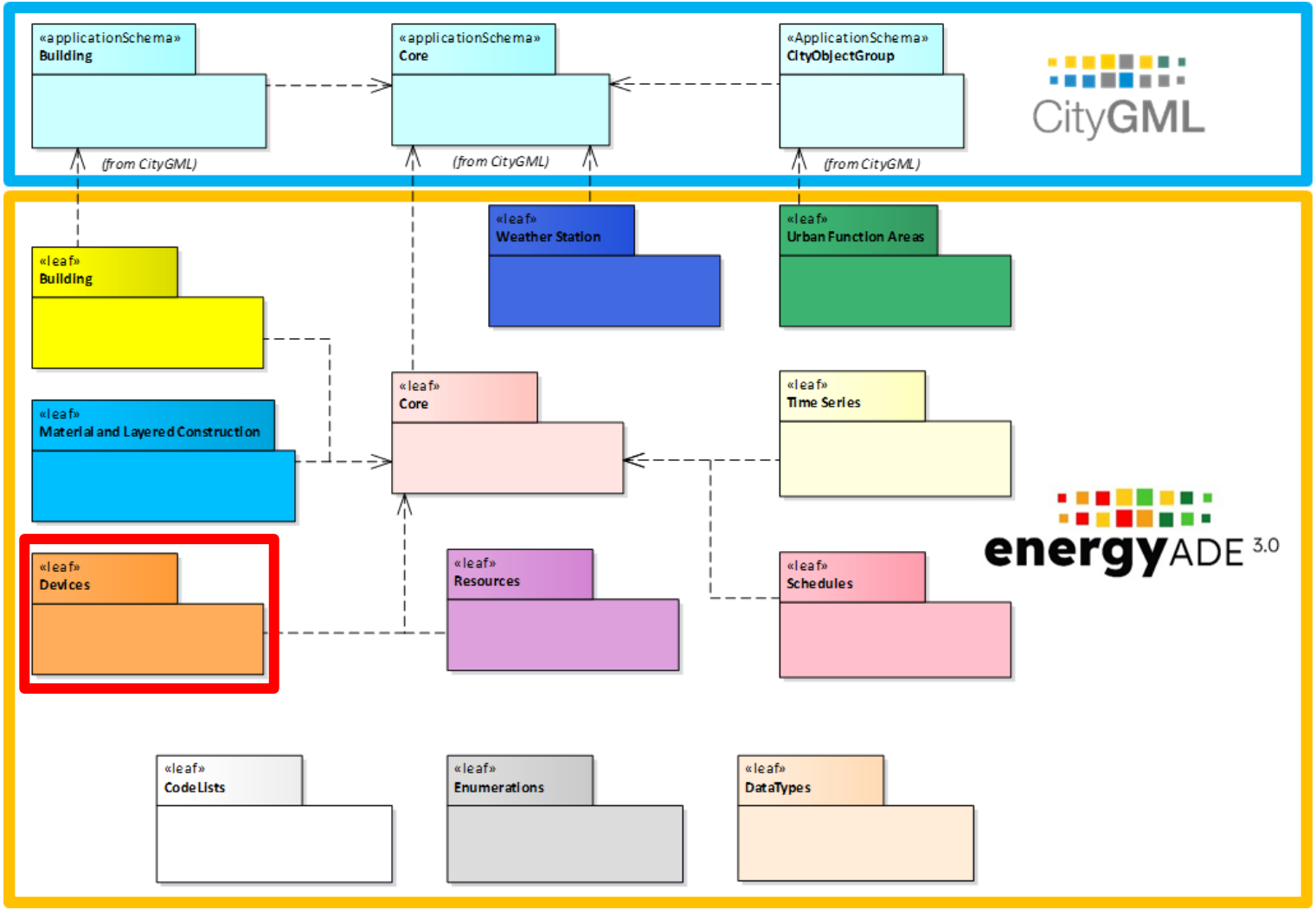
- Advanced modelling via **Layer** objects of different materials ordered from the inside to the outside of the zone boundary surface
- Materials (**SolidMaterial**, **Liquid**, **Gas**) have their own physical properties
- Best used with detailed simulations e.g. EnergyPlus, CitySim, etc.



Material and Layered Construction module 3/3

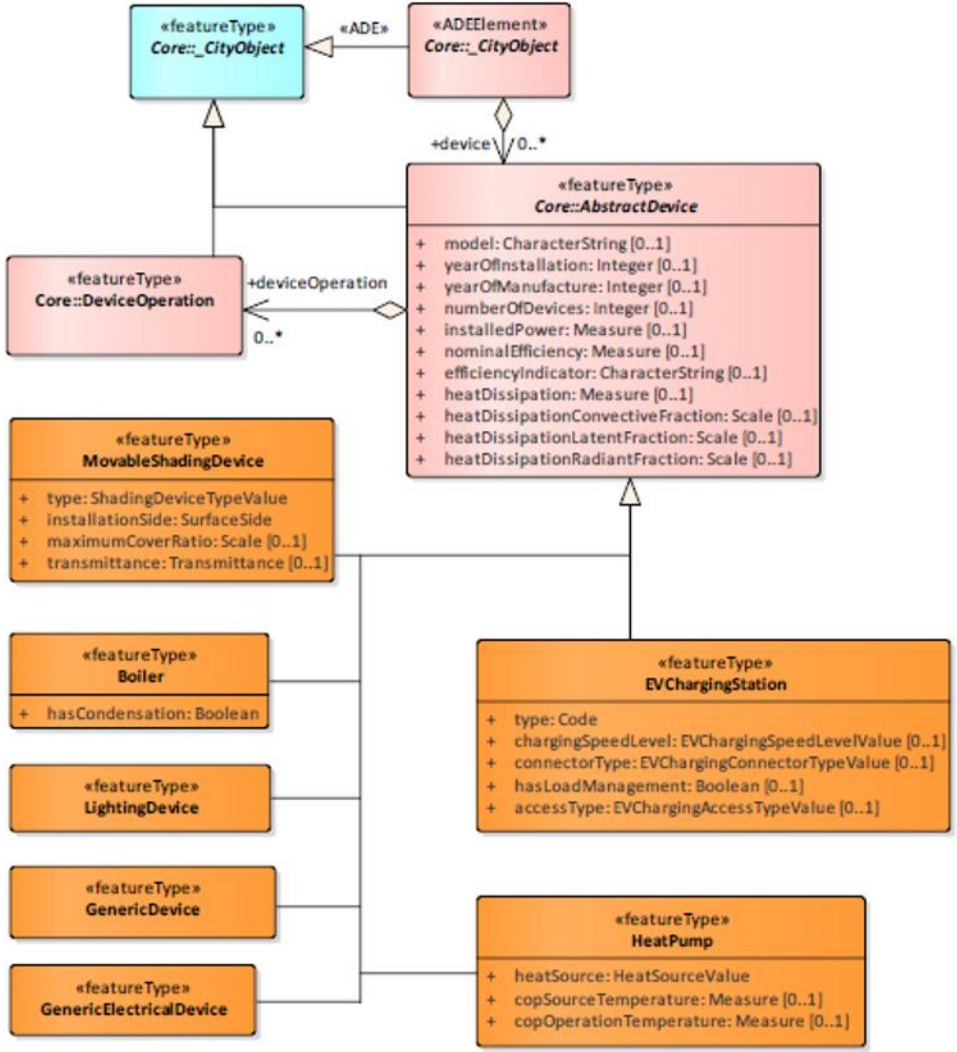
- All **LayeredConstruction** and all **Material** objects are contained in two separate libraries (containers) and only referenced (e.g. via Xlinks)
  - **Library** objects are top-level elements



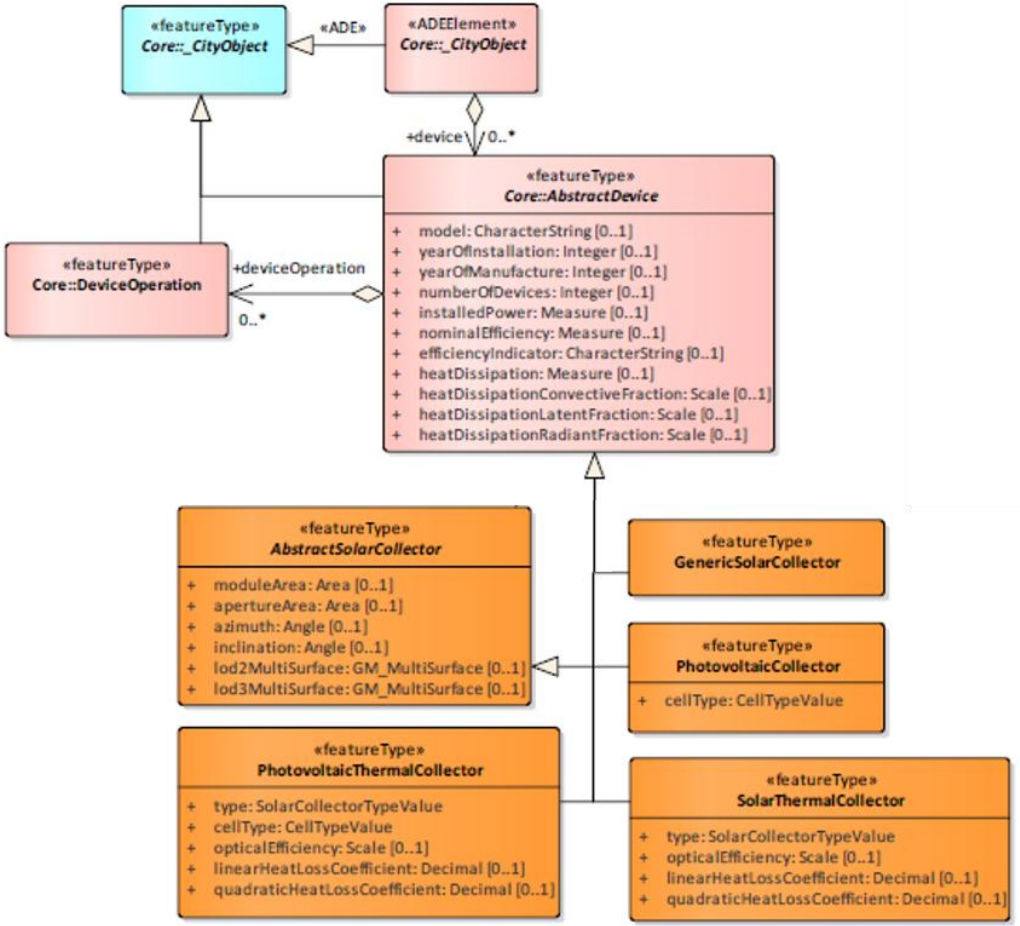


Devices module 1/3

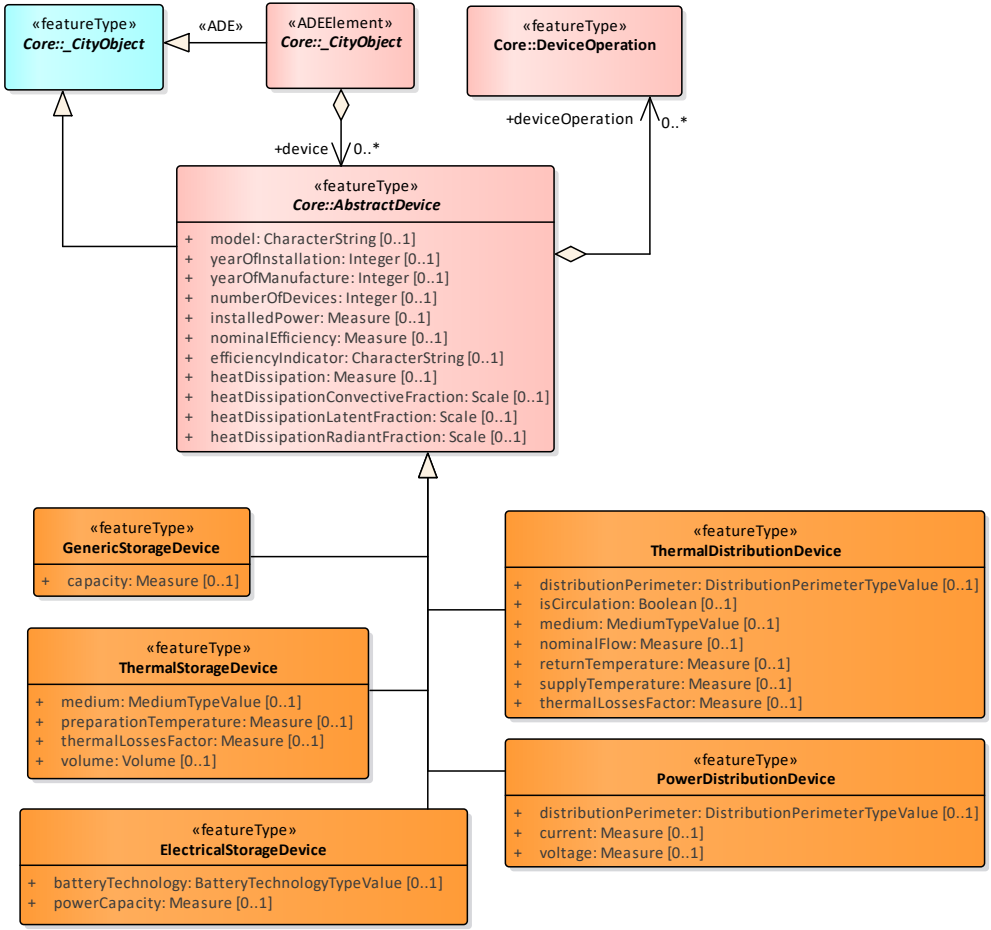
- Contains a collection of "hardware" devices that are meant to use or transform Energy
- Used to be the "Energy systems module" on Energy ADE 1.0
  - Too complex → totally dropped from Energy ADE 1.0 KIT profile and Energy ADE 2.0
- Now "just" a set of devices

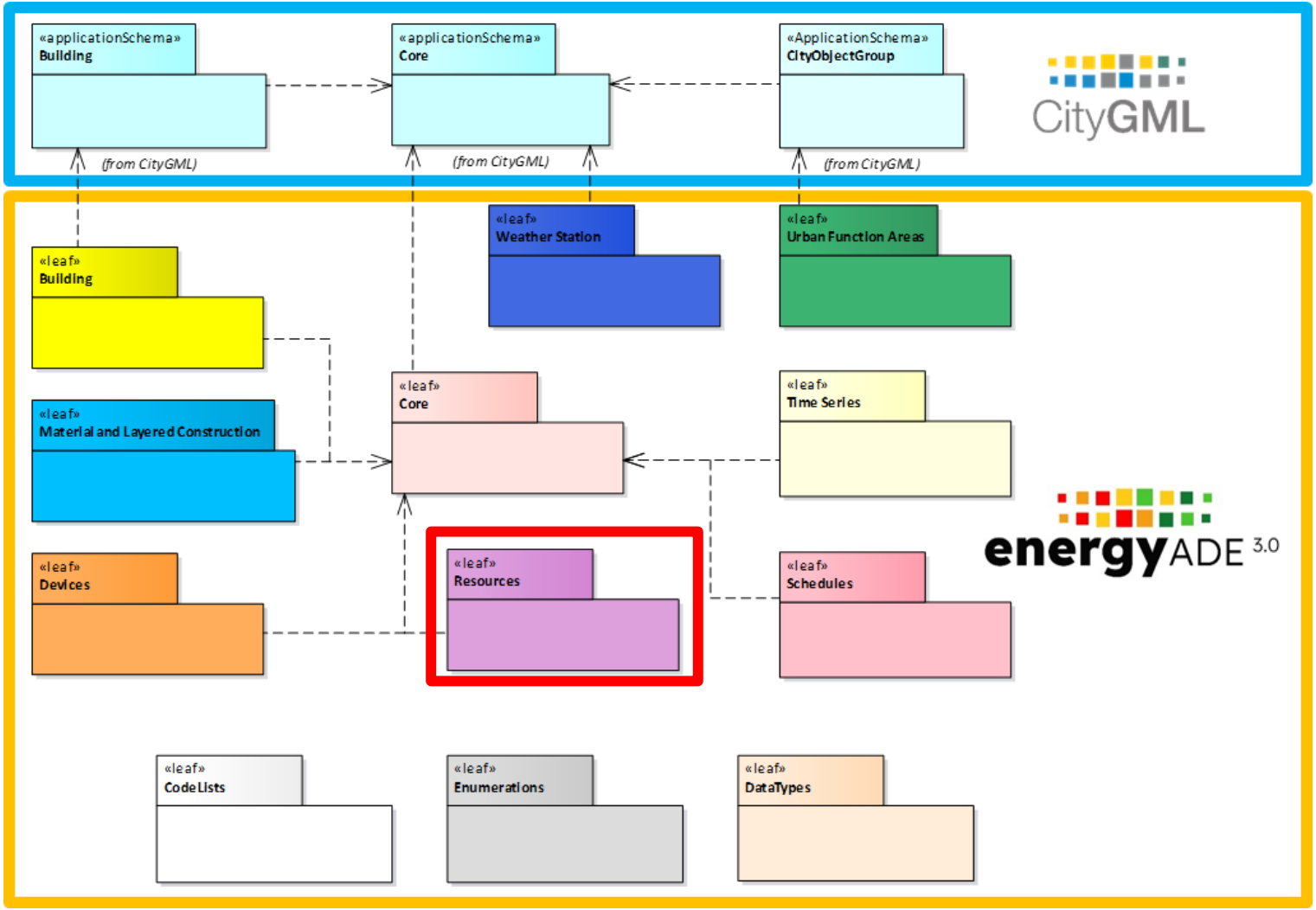


Devices module 2/3



Devices module 3/3





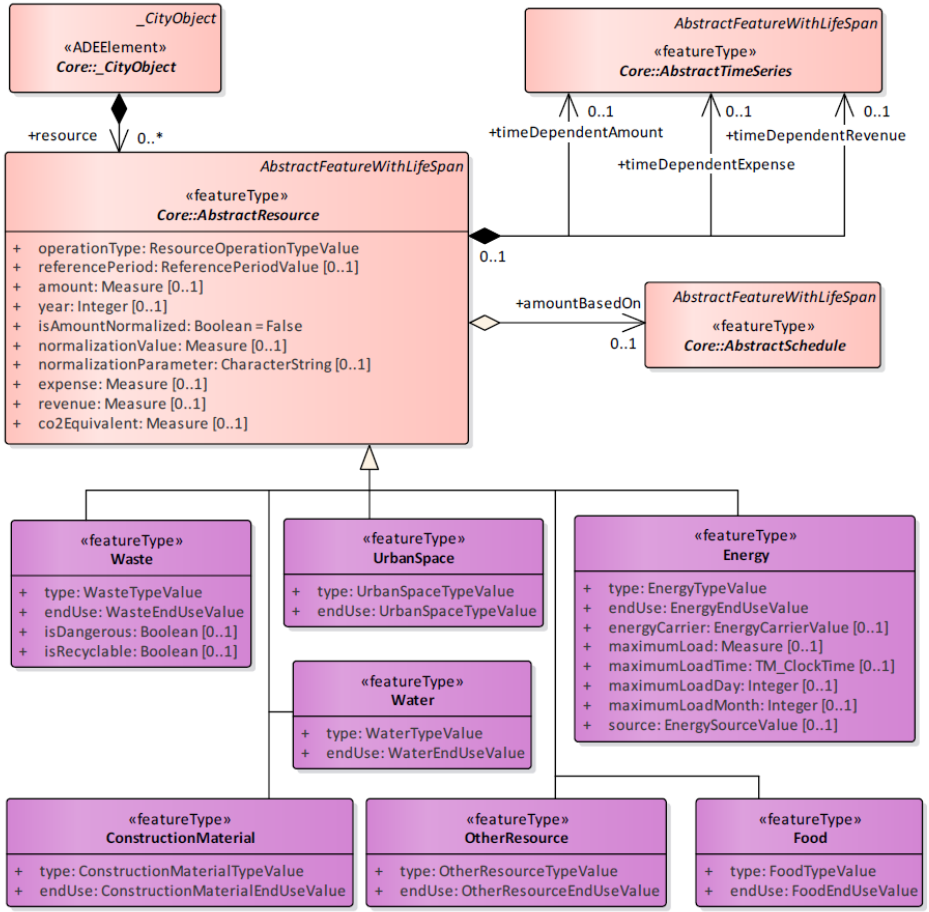
Resources module 1/2

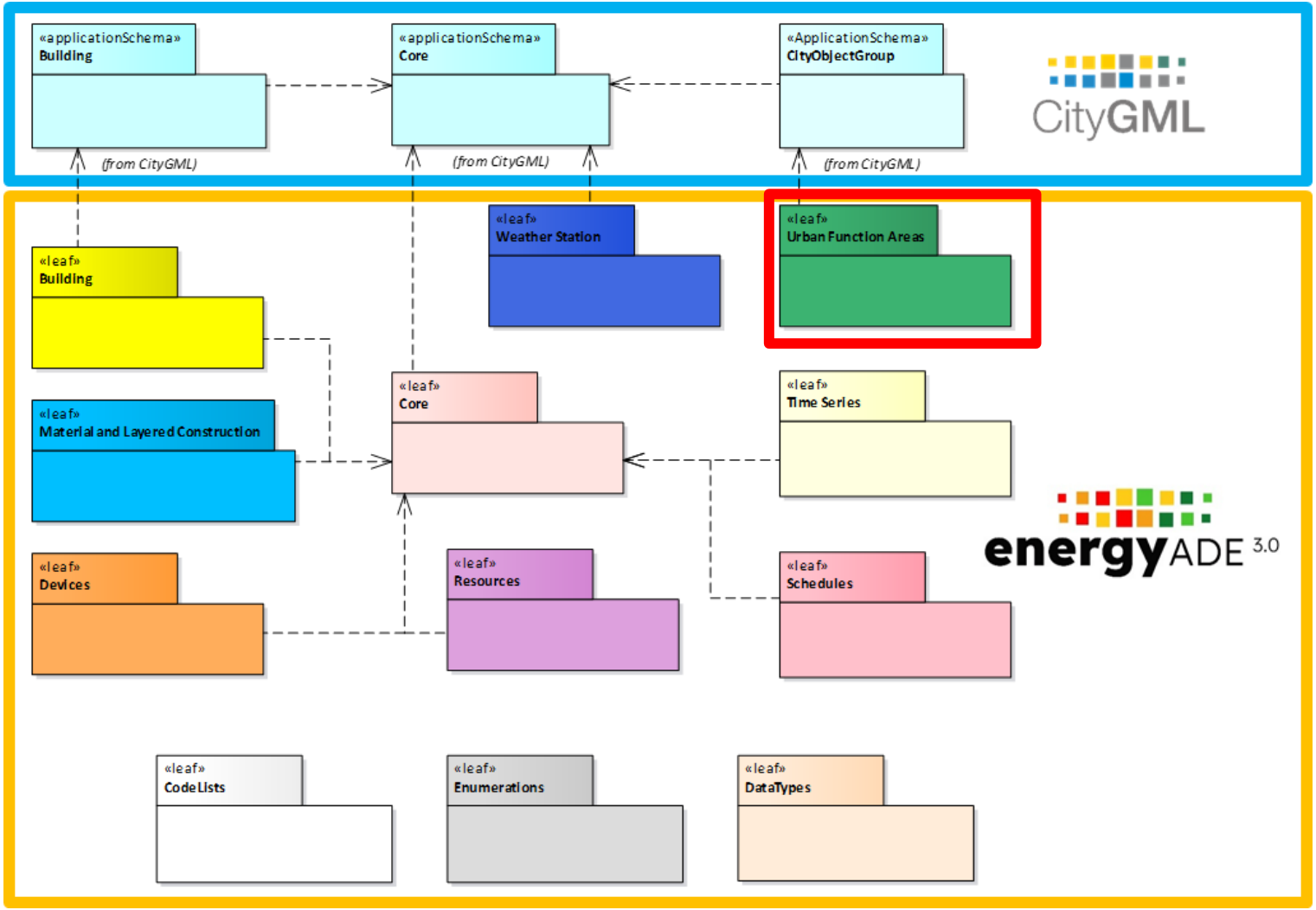
- The main idea is that *every* CityObject can...
  - ...demand/consume... produce/generate... store/accumulate...
  - ..."something" (i.e. a *resource*)

- This "something" can be **Energy**, **Water**, **Food**, **Waste**, **ConstructionMaterial**, etc.
  - The quantity can be expressed as a (yearly) amount, or as a time series (or both)
  - Values can be provided as absolute or specific values (+ normalization value)
    - Example: Energy demand expressed in kWh/year, but providing the reference area, we can obtain the specific value in kWh/m^2/year – or the other way around

- It is now possible to deal with
  - *actual* and *potential* values,
  - *simulated* and *measured* values

Resources module 2/2



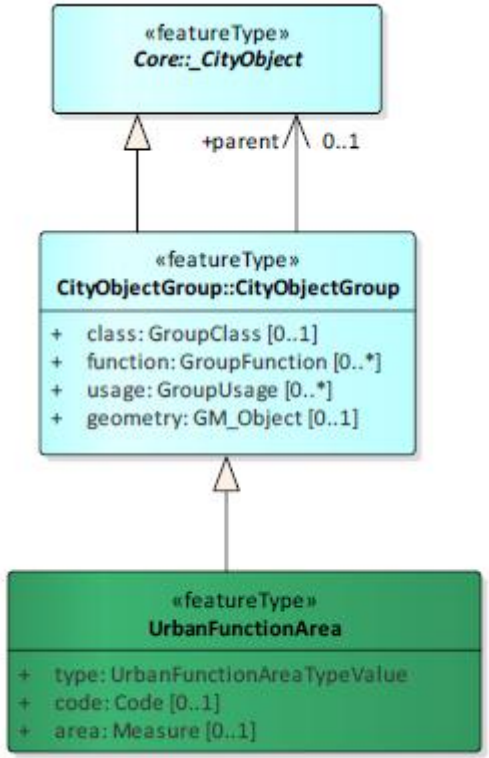


Urban function areas module 1/3

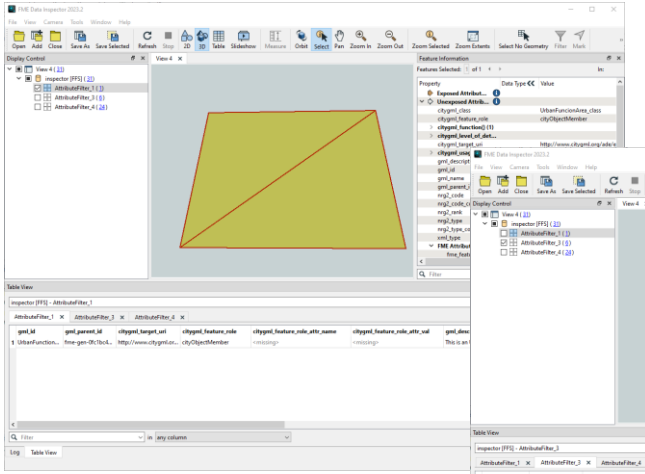
- The main idea is to have "spatial entities" that identify portions of space which do not exist physically, but that we use for grouping objects or identifying certain areas

- These "spatial entities" can be hierarchically grouped
  - They can follow administrative boundaries
    - E.g. block, neighbourhood, district, city, etc.
  - Or they can follow a regular pattern
    - E.g. grid cells (e.g. statistical/census data)

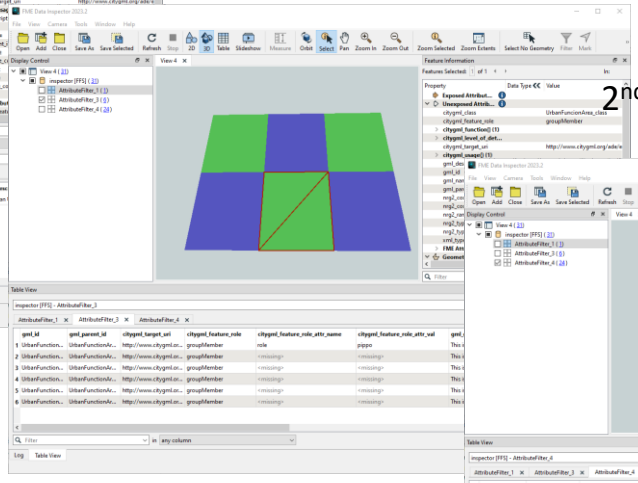
- Using this module we can present aggregated results, using different grouping strategies



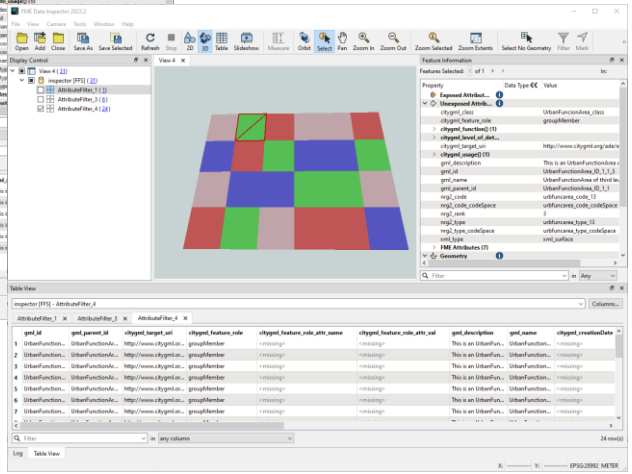
Urban function areas module 2/3



1st level: **City**



2nd level: **District**

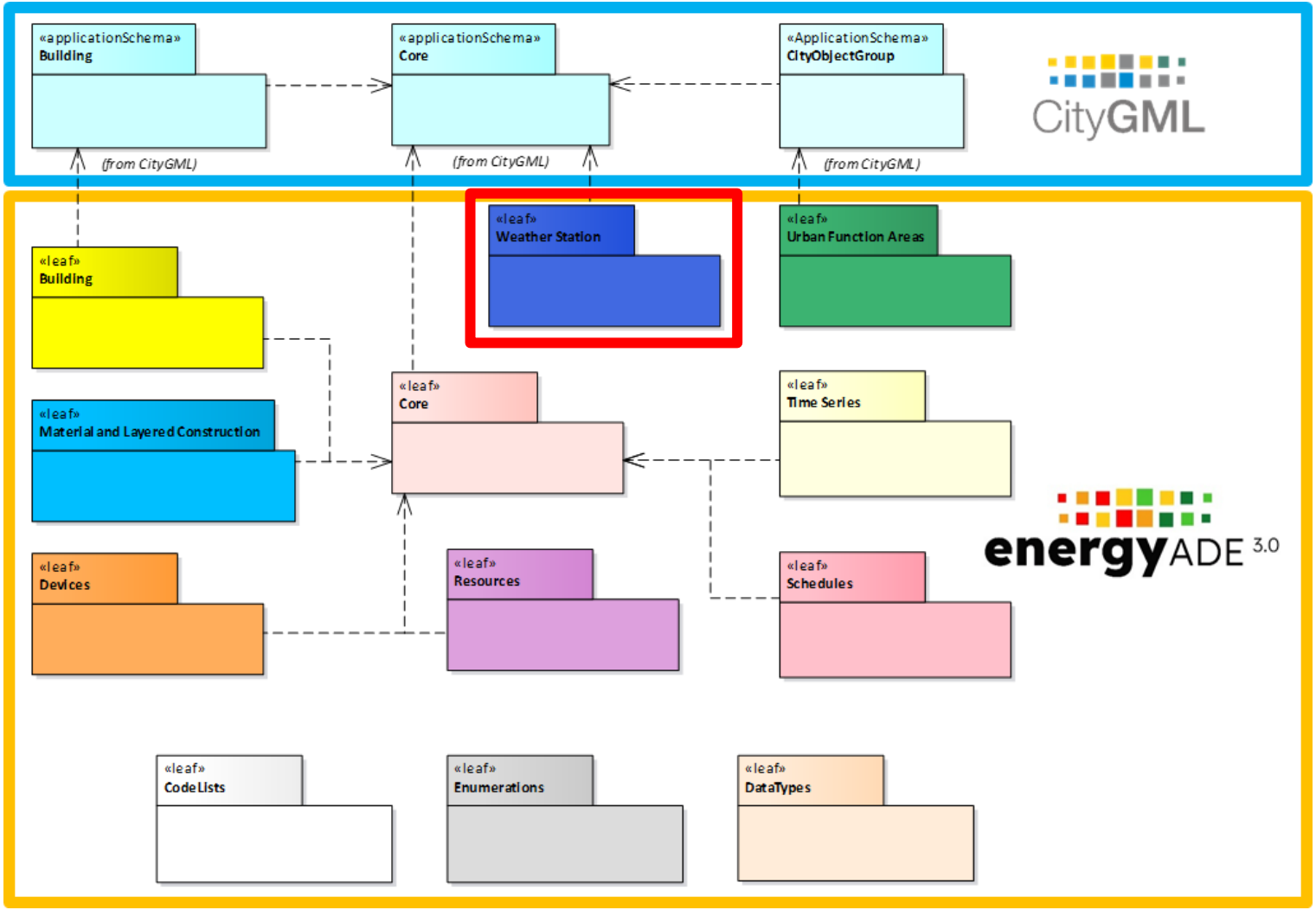


3rd level: **Neighbourhood**

...4th level: **Block**

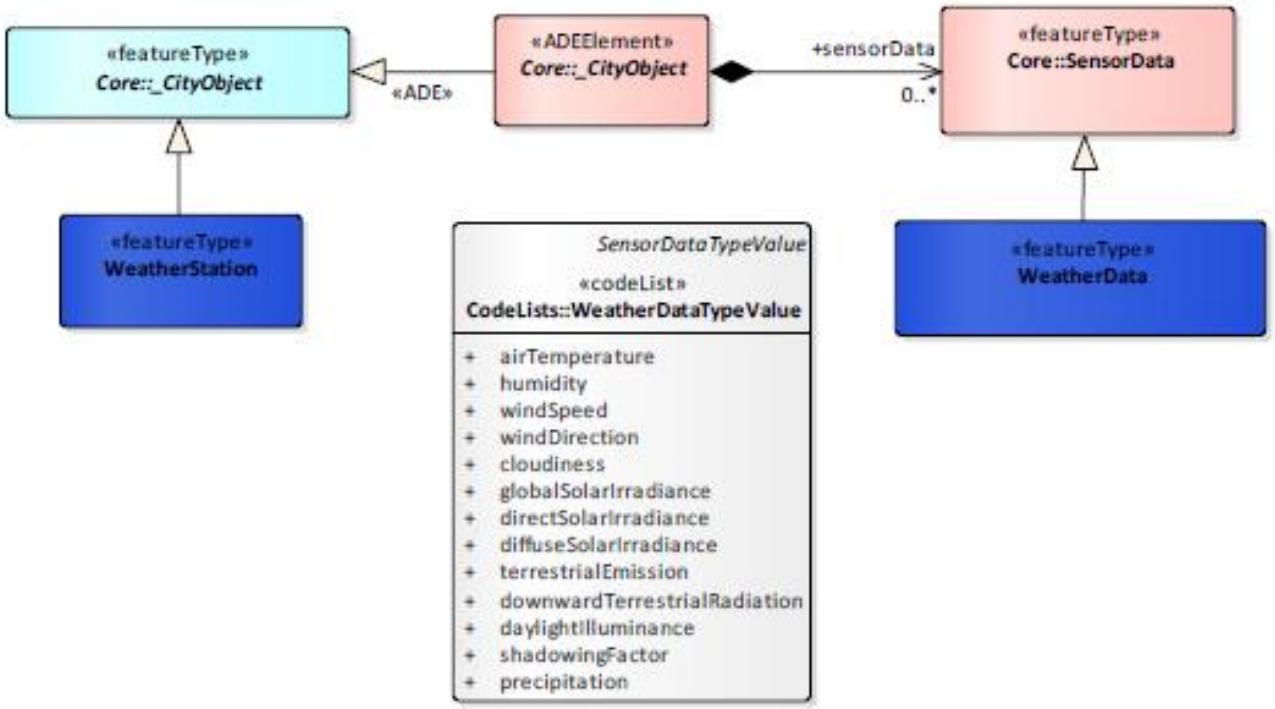
Urban function areas module 3/3

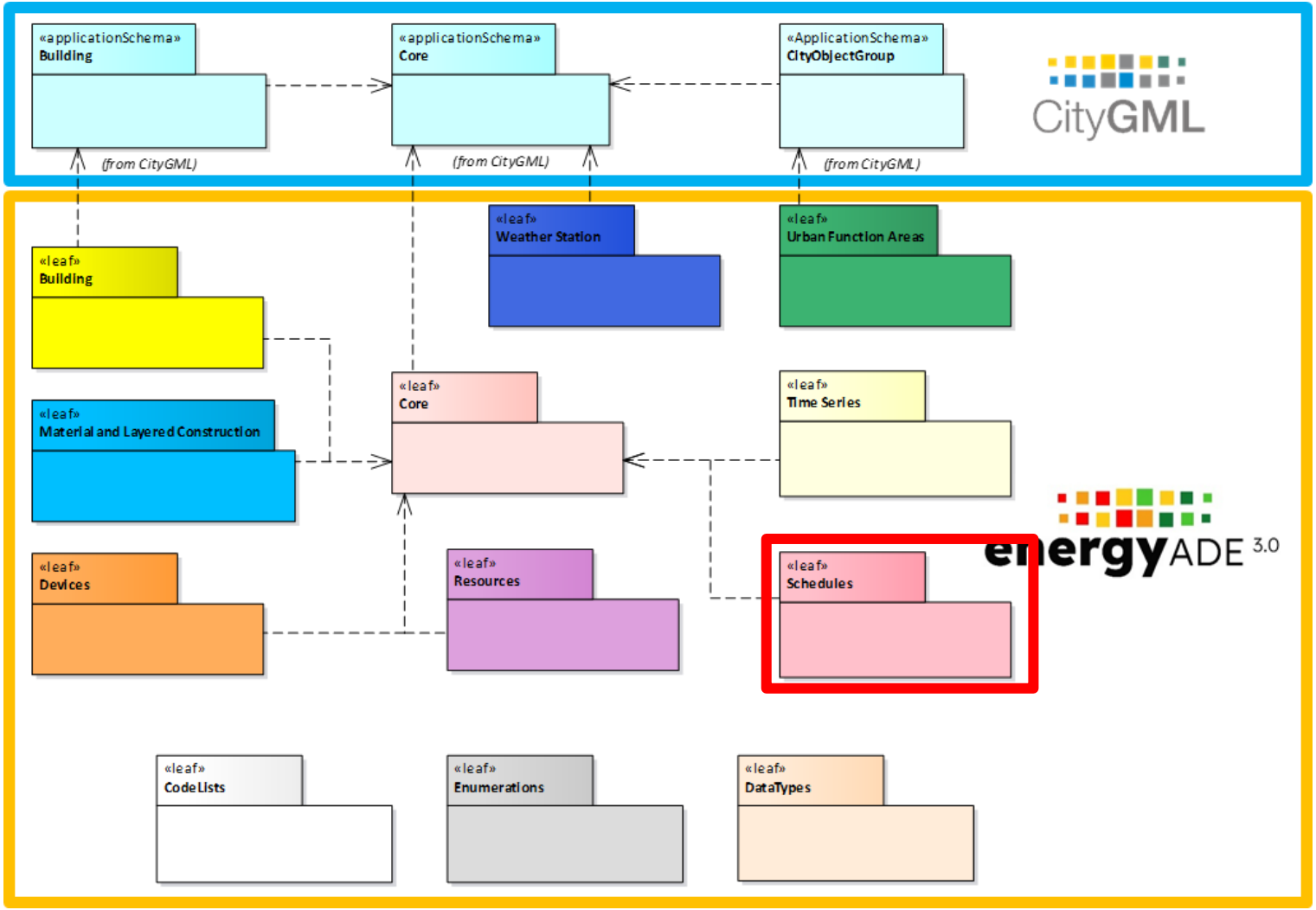
Cooling energy demand in Rotterdam at **building** and at **neighbourhood level**



Weather station module

- Allows to model weather-related data



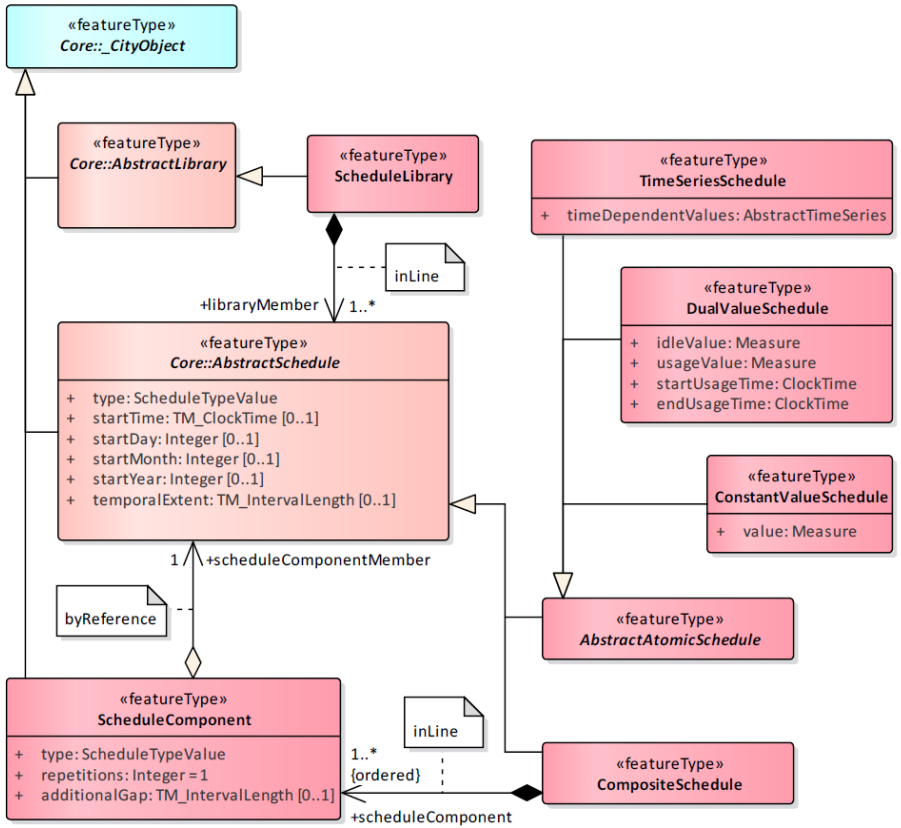


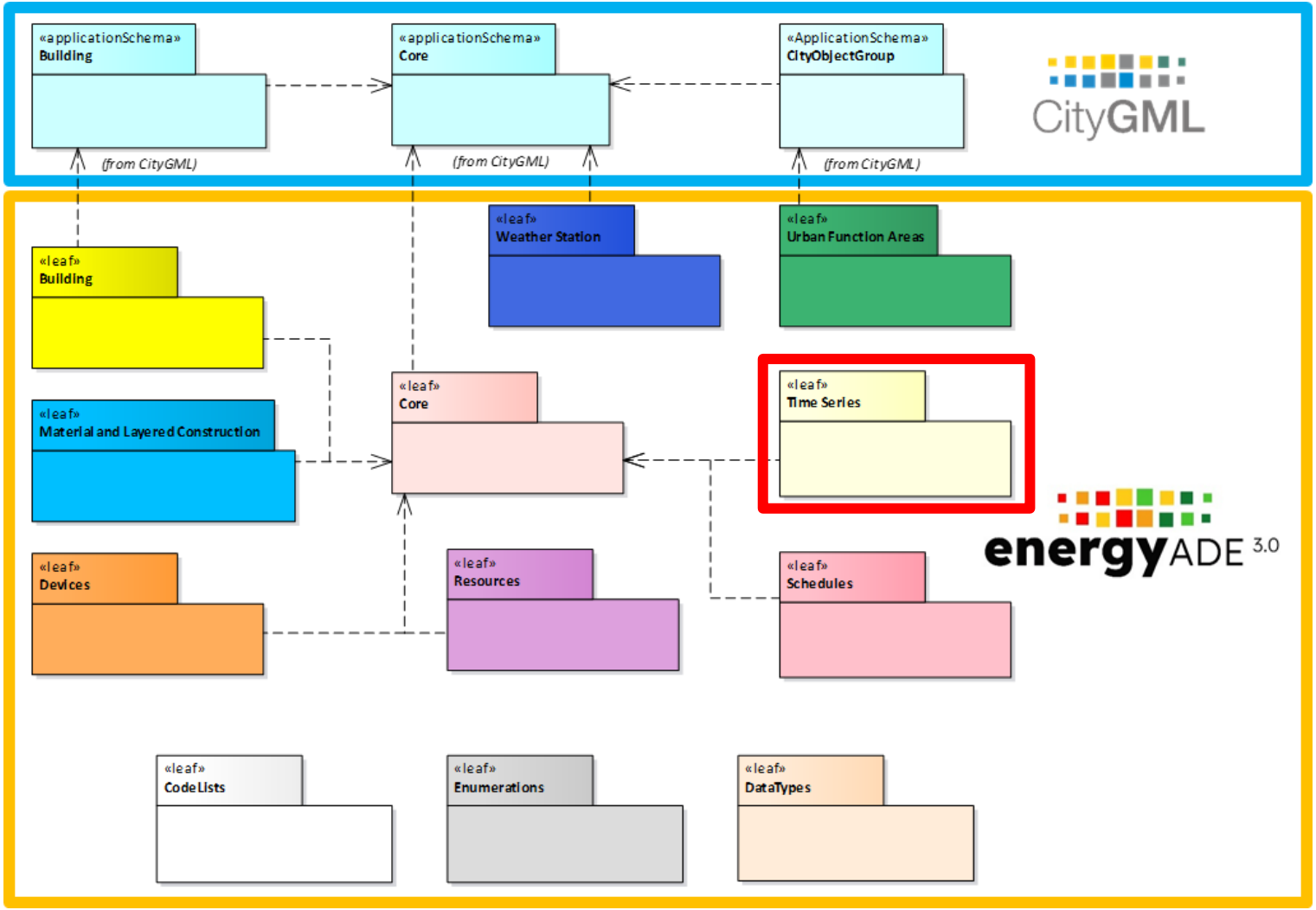
Schedule module

- Allows to model 4 types of schedules
  - **ConstantValueSchedule**
  - **DualValueSchedule**
  - **TimeSeriesSchedule**
  - **CompositeSchedule**

- Reworks the Schedule classes of Energy ADE 1.0
  and brings them closer to the logic of CityGML 3.0
  (see Dynamizer module)

- All schedule objects must be contained in their **ScheduleLibrary** and referenced (e.g. via Xlinks)

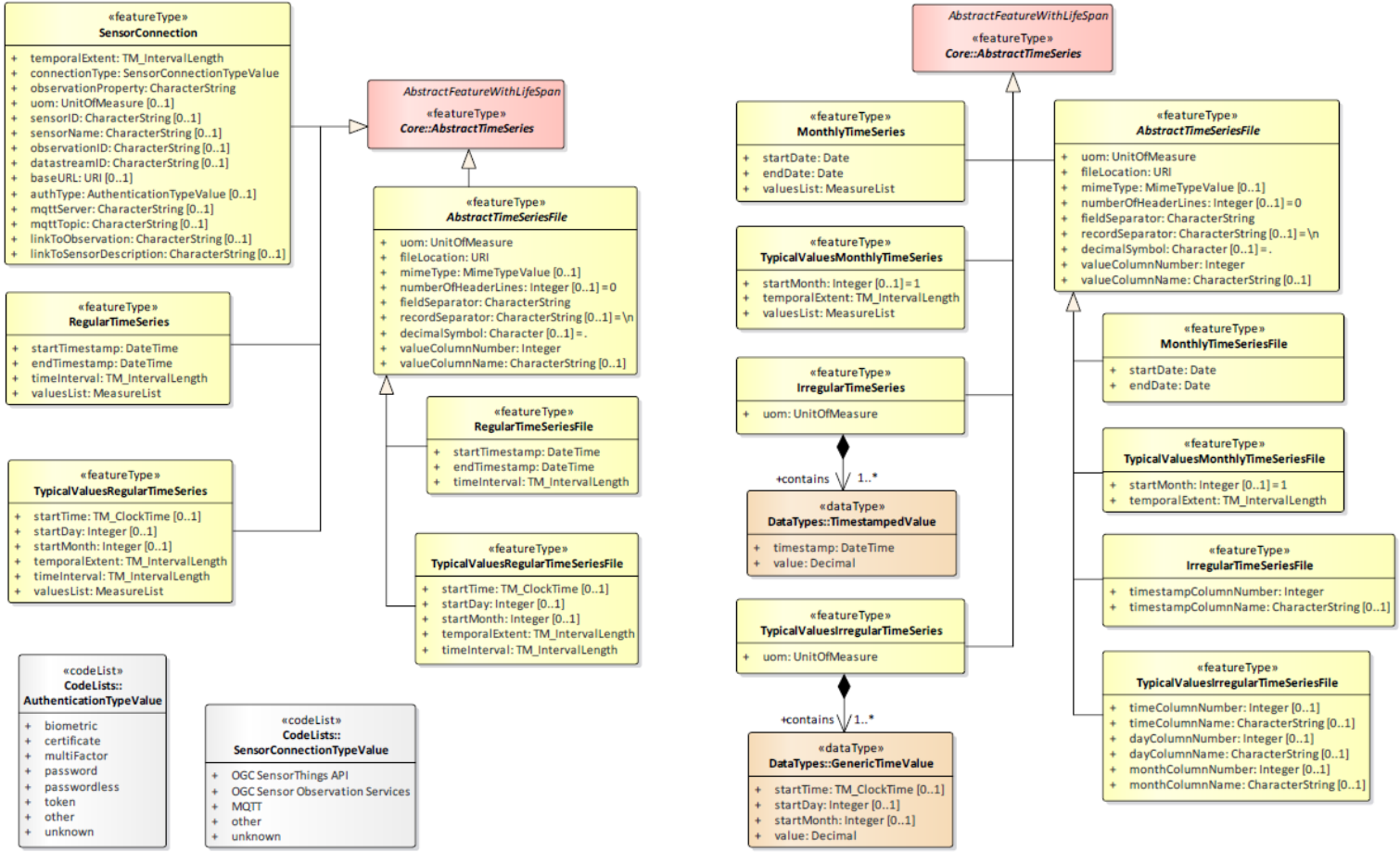


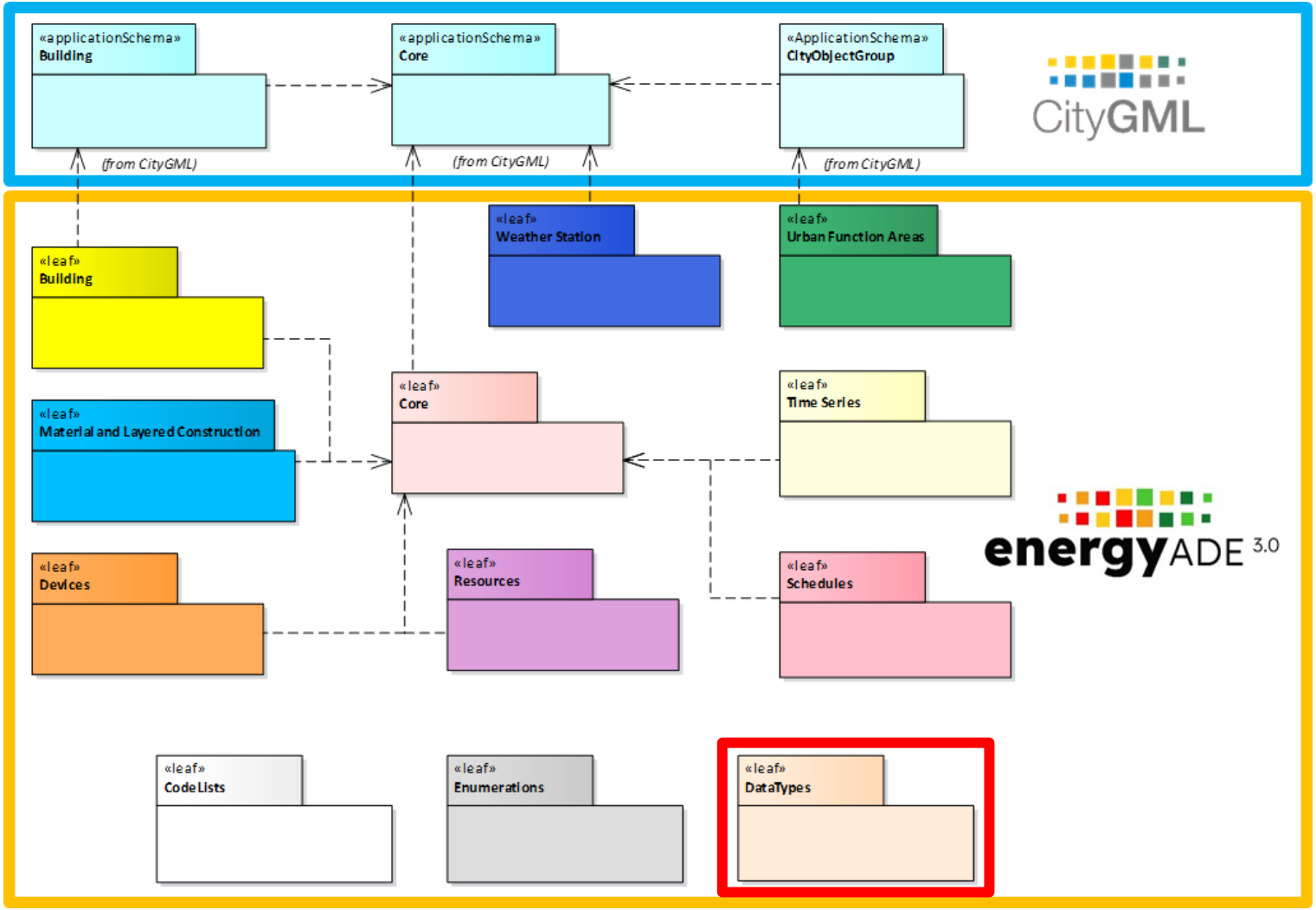


Time Series module 1/2

- Allows to store **regular** and **irregular** time series
- Allows to store time series of **measured** and **typical values**
- Allows to store values **in line** (in the same document) or **in external files**
- Allows to store metadata to connect to sensors via class **SensorConnection** (backported from CityGML 3.0)
- Offers a "special" class for **monthly values**

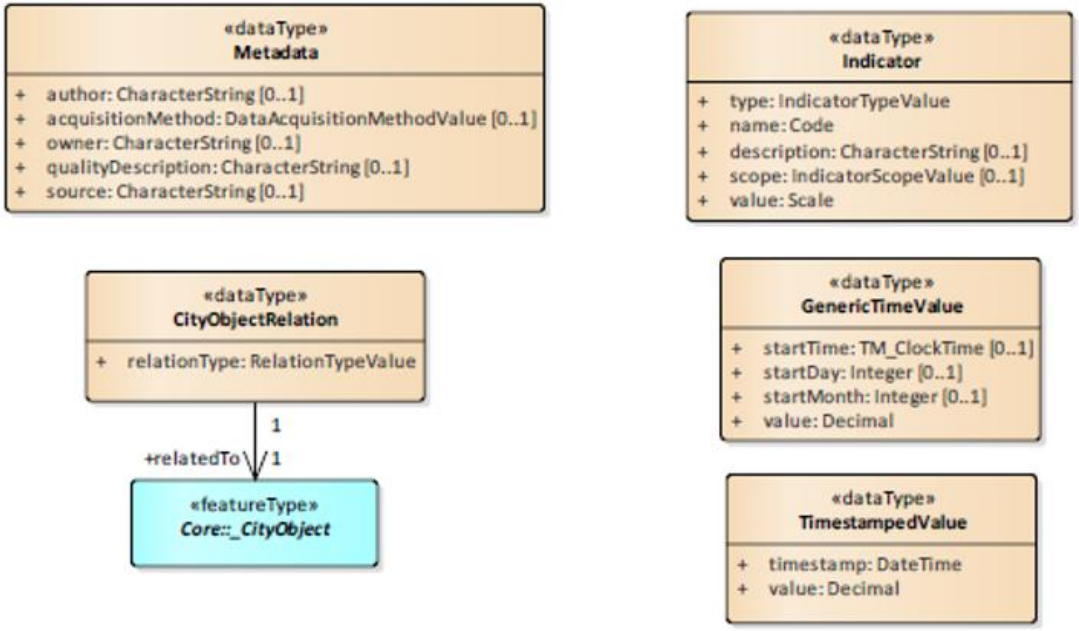
Time Series module 2/2



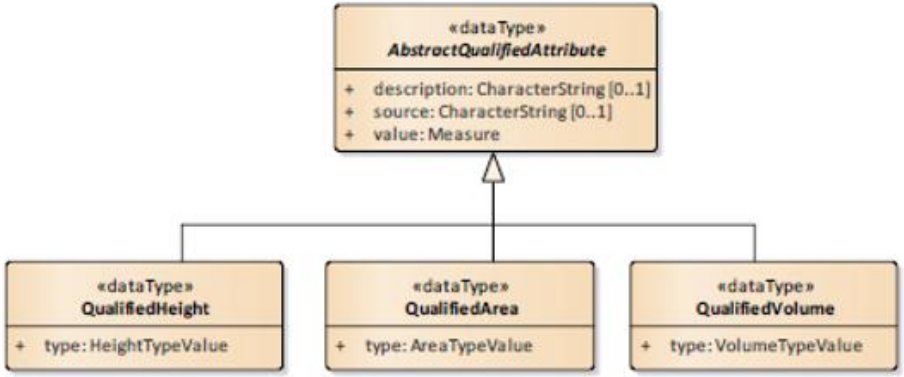


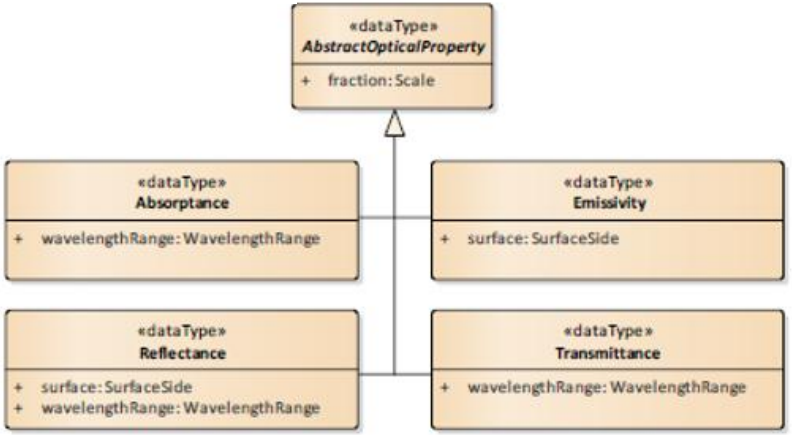
Datatypes module 1/2

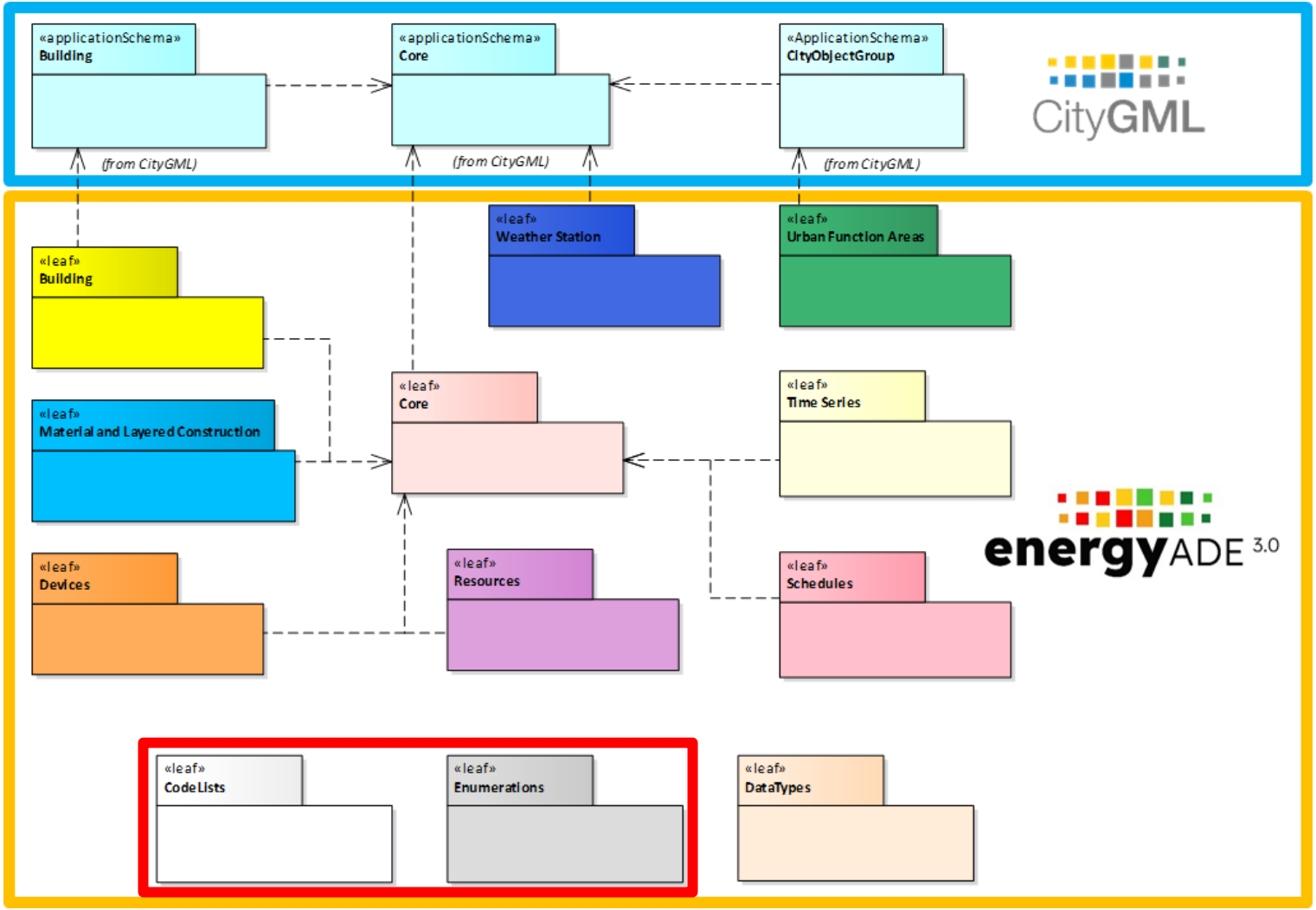
- Contains classes (they are *datatypes*, **neither Features nor _*CityObjects***!) that are used by other modules
- A *datatype* is "a group of properties without identity" (in simple words: no *id* for each instance of a datatype class)



Datatypes module 2/2

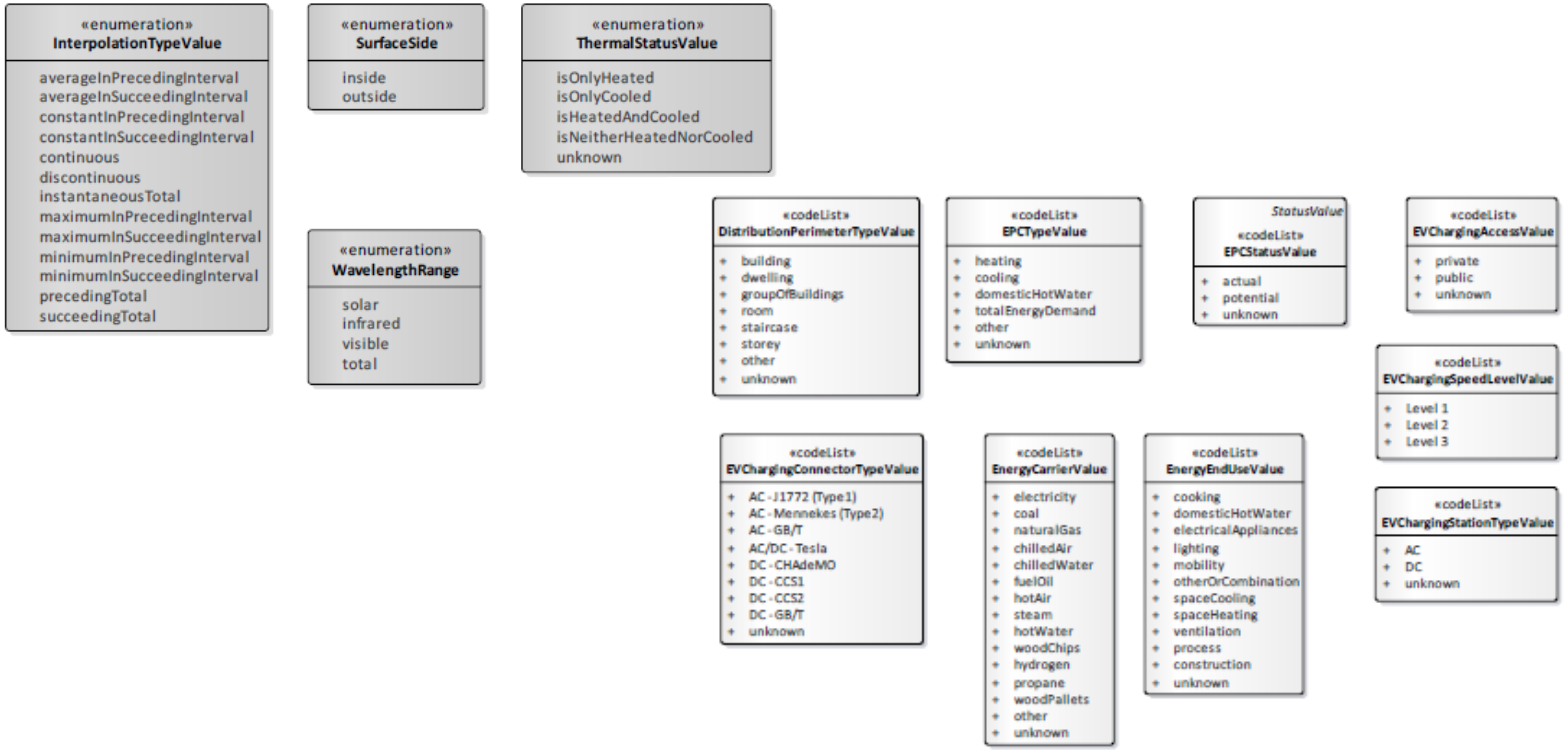






Enumerations and CodeLists

- **Enumerations** are normative closed sets of valid values
- **CodeLists** are non-normative open sets of proposed values



energyADE 3.0

Available resources

CityGML: Conceptual model and encodings

CONCEPTUAL DATA MODEL
("definitions, relations and hierarchies between objects")
Technical specifications, UML diagrams, etc.

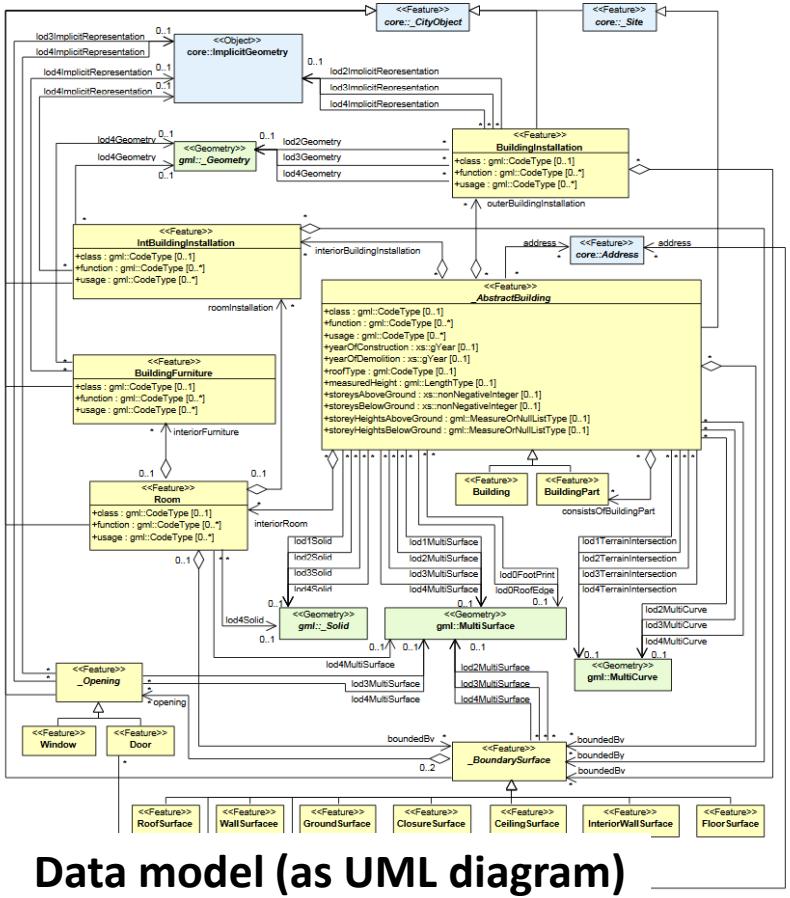
CityGML

Application
+ **D**omain
Extensions

ENCODINGS
("how data are actually written")

Lossless encoding **Full** CityGML	**Lossy** encoding **Subset** of CityGML	**Lossless** encoding **Full** CityGML
XML	JSON	SQL
CityGML	CityJSON	3D City Database

CityGML: a closer look

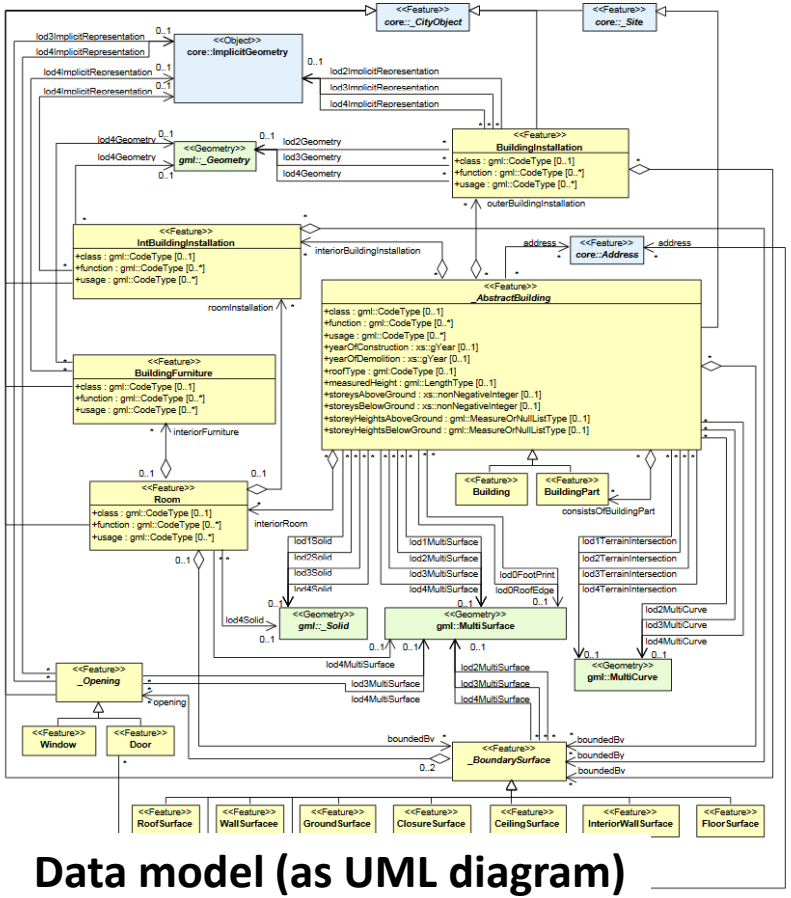


Data model (as UML diagram)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Written by 3D City Database Importer/Exporter, version "5.5.3" -->
<!-- Chair of Geoinformatics, Technical University of Munich -->
<CityModel xmlns:xAL="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0" xmlns:gml="http://www.opengis.n
  <cityObjectMember>
    <bldg:Building gml:id="id_building_04">
      <gml:description>This is Building 4</gml:description>
      <gml:name>Fin's Cabin</gml:name>
      <gml:boundedBy>
        <gml:Envelope srsName="urn:ogc:def:crs,crs:EPSG::28992,crs:EPSG::5109" srsDimension="3"
          <gml:lowerCorner>85880.0 448700.0 2.0</gml:lowerCorner>
          <gml:upperCorner>85890.0 448710.0 17.0</gml:upperCorner>
        </gml:Envelope>
      </gml:boundedBy>
      <creationDate>2026-03-24</creationDate>
      <gen:intAttribute name="n_adjacent_buildings">
        <gen:value>1</gen:value>
      </gen:intAttribute>
      <gen:intAttribute name="lod_max">
        <gen:value>2</gen:value>
      </gen:intAttribute>
      <gen:measureAttribute name="lod2_volume">
        <gen:value uom="m^3">1250.0</gen:value>
      </gen:measureAttribute>
      <bldg:class codeSpace="http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuil
      <bldg:function codeSpace="http://www.sig3d.org/codelists/standard/building/2.0/_AbstractB
      <bldg:yearOfConstruction>1955</bldg:yearOfConstruction>
      <bldg:roofType codeSpace="http://www.sig3d.org/codelists/standard/building/2.0/_AbstractB
      <bldg:measuredHeight uom="m">15.0</bldg:measuredHeight>
      <bldg:storeysAboveGround>3</bldg:storeysAboveGround>
      <bldg:storeysBelowGround>0</bldg:storeysBelowGround>
      <bldg:storeyHeightsAboveGround uom="m">3.0</bldg:storeyHeightsAboveGround>
      <bldg:lod0FootPrint>
        <gml:MultiSurface gml:id="id_building_4_footprint_multisurf_1">
          <gml:surfaceMember>
            <gml:OrientableSurface orientation="-">
              <gml:baseSurface>
                <gml:Polygon gml:id="id_building_4_polygon_3">
                  <gml:exterior>
                    <gml:LinearRing gml:id="id_building_4_polygon_3_0_">
                      <gml:posList srsDimension="3">85890.0 448700.0 2.0 85880.0 448700.0 2.0 85
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Polygon>
              </gml:baseSurface>
            </gml:OrientableSurface>
          </gml:surfaceMember>
        </gml:MultiSurface>
      </bldg:lod0FootPrint>
```

Example of XML encoding (file)

CityGML: a closer look

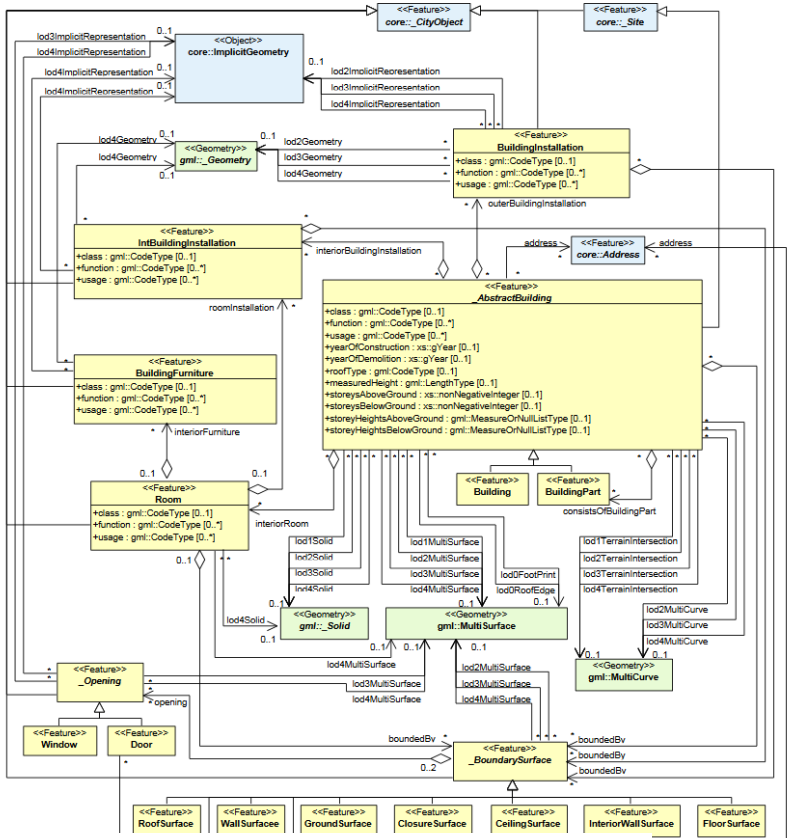


Data model (as UML diagram)


```
{
"type": "CityJSON",
"version": "1.0",
"CityObjects": {
 "id_building_04": {
  "address": {
   "CountryName": "Bespin Territories",
   "LocalityName": "Alderaan",
   "ThoroughfareNumber": "4",
   "ThoroughfareName": "Bespin Square",
   "PostalCode": "1977SW",
   "location": {
    "type": "MultiPoint",
    "boundaries": [
     24
    ],
    "lod": 0
   }
  },
  "type": "Building",
  "attributes": {
   "measuredHeight": 15.0,
   "roofType": "gabled roof",
   "storeysAboveGround": 3,
   "storeysBelowGround": 0,
   "storeyHeightsAboveGround": [
    3.0
   ],
   "yearOfConstruction": 1955,
   "description": "This is Building 4",
   "name": "Fin\u0027s Cabin",
   "creationDate": "2026-03-24",
   "class": "habitation",
   "function": "residential building",
   "n_adjacent_buildings": 1,
   "lod2_volume": 1250.0,
   "lod_max": 2
  },
  "geographicalExtent": [
   85880.0,
   448700.0,
   2.0,
   85890.0,
   448710.0,
   17.0
  ],
  "geometry": [
   {
    "type": "MultiSurface",
```


Example of JSON encoding (file)

CityGML: a closer look



Data model (as UML diagram)

id [PK] bigint	objectclass_id integer	gmlid character varying (256)	gmlid_codespace character varying (1000)	name character varying (1000)	name_codespace character varying (4000)	description character varyin
1	26	id_building_02	[null]	Rey's Hut	[null]	This is Building
2	26	id_building_01	[null]	Snoke's Palace	[null]	This is Building
3	36	id_building_2_closuresurface_1	[null]	WallSurface 4906-3 (shared)	[null]	[null]
4	36	id_building_1_closuresurface_1	[null]	WallSurface 4906-8 (shared)	[null]	[null]
5	35	id_building_1_groundsurface_1	[null]	GroundSurface 1 (Building 1)	[null]	This is GroundS
6	36	id_building_2_closuresurface_2	[null]	WallSurface 4906-12 (share…	[null]	[null]
7	33	id_building_1_roofsurface_1	[null]	RoofSurface 1 (Building 1)	[null]	This is Roofsurf
8	35	id_building_2_groundsurface_1	[null]	GroundSurface 1 (Building 2)	[null]	This is GroundS
9	9					
10	10					
11	11					
12	12					

id [PK] bigint	objectclass_id integer	building_parent_id bigint	building_root_id bigint	class character varying (256)	class_codespace character varying (4000)
1	26	[null]	1	habitation	http://www.sig3d.org/codelists/standard/building
2	26	[null]	2	habitation	http://www.sig3d.org/codelists/standard/building
19	26	[null]	19	traffic	http://www.sig3d.org/codelists/standard/building
20	26	[null]	20	habitation	http://www.sig3d.org/codelists/standard/building
36	26	[null]	36	habitation	http://www.sig3d.org/codelists/standard/building
37	26	[null]	37	healthcare	http://www.sig3d.org/codelists/standard/building
52	26	[null]	52	habitation	http://www.sig3d.org/codelists/standard/building

id [PK] big	gmlid character varying (256)	gmlid_cod character	parent_id bigint	root_id bigint	is_solid numeric	is_composite numeric	is_triangulated numeric	is_xlink numeric	is_reverse numeric	solid_geon geometry	geometri geometri
1	id_building_2_footprint_multisurf_1	[null]	[null]	1	0	0	0	0	0	[null]	[null]
2	id_building_2_polygon_3	[null]	1	1	0	0	0	2	1	[null]	01030000A040
4	id_building_1_footprint_multisurf_1	[null]	[null]	4	0	0	0	0	0	[null]	[null]
5	id_building_1_polygon_3	[null]	4	4	0	0	0	2	1	[null]	01030000A040
7	id_building_01_lod1_Solid_1	[null]	[null]	7	1	0	0	0	0	010F0000A04…	[null]
8	id_building_02_lod1_Solid_2	[null]	[null]	8	1	0	0	0	0	010F0000A04…	[null]
9	id_building_01_lod1_CompSurf_1	[null]	7	7	0	1	0	0	0	[null]	[null]
10	id_building_02_lod1_CompSurf_2	[null]	8	8	0	1	0	0	0	[null]	[null]
11	id_building_01_lod1_Polygon_1	[null]	9	7	0	0	0	0	0	[null]	01030000A040
12	id_building_02_lod1_Polygon_7	[null]	10	8	0	0	0	0	0	[null]	01030000A040
13	id_building_01_lod1_Polygon_2	[null]	9	7	0	0	0	0	0	[null]	01030000A040
14	id_building_										
15	id_building_										
16	id_building_										

id [PK] bigint	parent_g bigint	root_gen bigint	attname character varying (256)	datatyp integer	strval character varyin	intval integer	reaival double pr	urival character var	dateval timestamp wi
1	[null]	1	lod2_volume	6	[null]	[null]	1250	[null]	[null]
2	[null]	2	lod2_volume	6	[null]	[null]	1250	[null]	[null]
3	[null]	3	lod_max	2	[null]	2	[null]	[null]	[null]
4	[null]	4	lod_max	2	[null]	2	[null]	[null]	[null]
5	[null]	5	n_adjacent_buildings	2	[null]	2	[null]	[null]	[null]
6	[null]	6	n_adjacent_buildings	2	[null]	1	[null]	[null]	[null]
7	[null]	7	lod2_volume	6	[null]	[null]	1250	[null]	[null]
8	[null]	8	num_residents	2	[null]	5	[null]	[null]	[null]
9	[null]	9	lod_max	2	[null]	2	[null]	[null]	[null]
10	[null]	10	test_date_att	5	[null]	[null]	[null]	[null]	2020-08-06 0
11	[null]	11	n_adjacent_buildings	2	[null]	2	[null]	[null]	[null]
12	[null]	12	test_real_att	3	[null]	[null]	12.34	[null]	[null]
13	[null]	13	test_string_att	1	This is a tes…	[null]	[null]	[null]	[null]
14	[null]	14	test_uri_att	1	https://www…	[null]	[null]	[null]	[null]

Example of SQL encoding (tables)

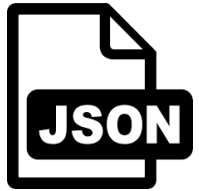
Energy ADE 3.0

CONCEPTUAL DATA MODEL
("definitions, relations and hierarchies between objects")
Technical specifications, UML diagrams, etc.

(as of spring 2026)

ENCODINGS
("how data are actually written")

CityGML



CityJSON

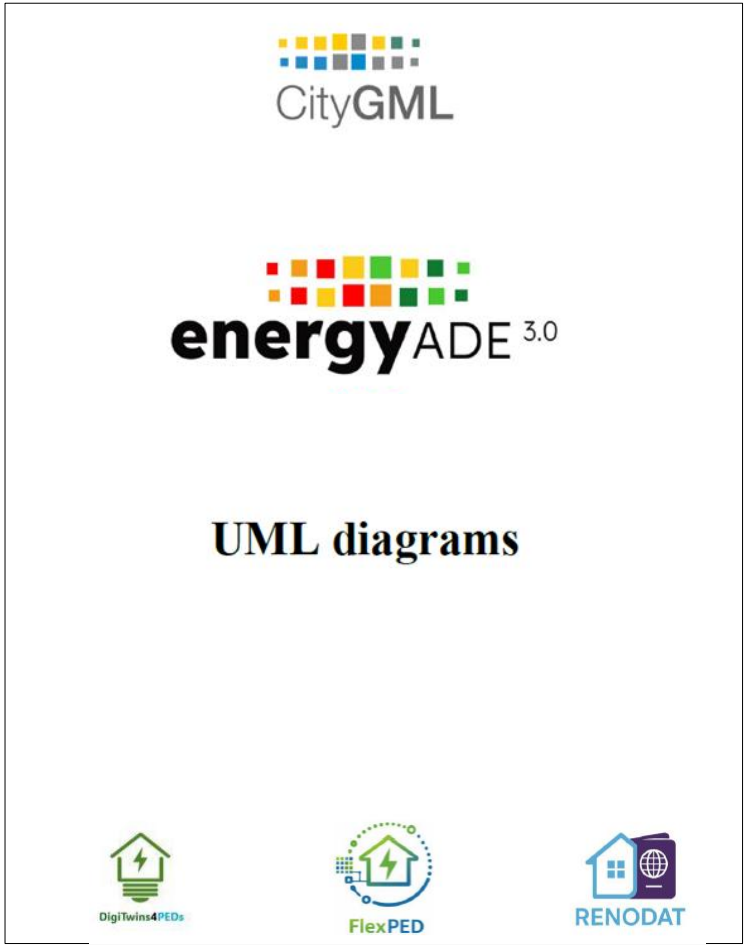
3D City Database

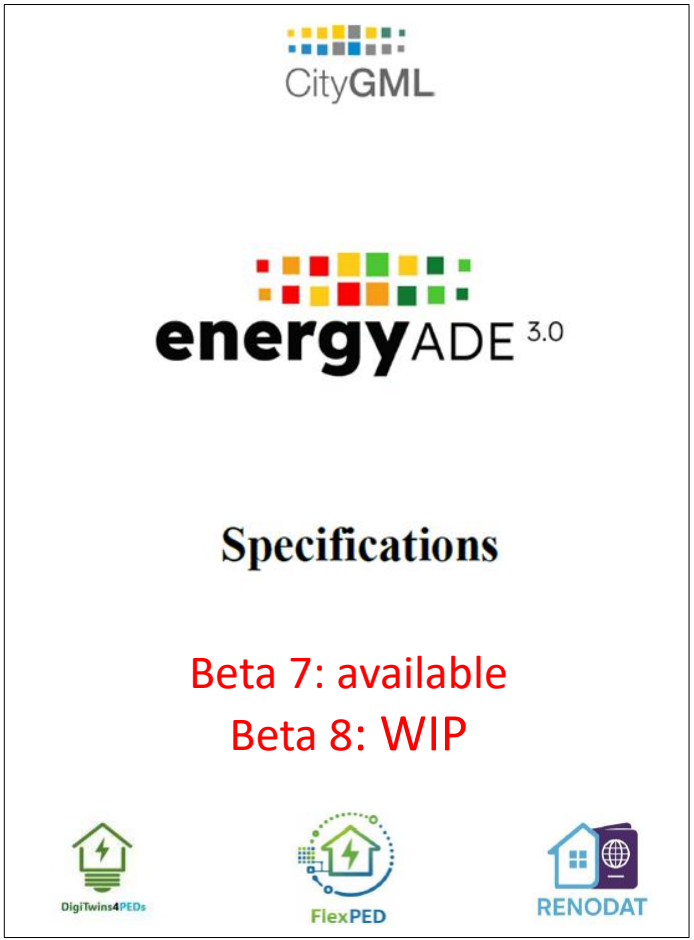
Energy ADE 3.0

- **Beta 7** available since October 2025
  - **UML** diagrams
  - **Specifications** document, modelling rules
  - **XSD** file
  - **DDL** scripts for the open-source database **3D City Database**
  - **Test data**
  - **FME** Workbench
  - Preliminary **Java-based libraries** to support, among the rest, Web Feature Services

- **Beta 8** preview released in May 2026. So far:
  - **UML** diagrams
  - **XSD** file
  - **DDL** scripts for the open-source database **3D City Database**
  - **Test data**
  - **FME** Workbench

Documentation





Modelling rules

Beta 7: available

Beta 8: WIP

Example 6: LoD3 building modelled via thematic surfaces and with openings. Roof edge extruded downwards to create the LoD3 envelope. Openings are "simply" cut into the hull surfaces. No overhangs With ADE attributes for thematic surfaces and openings, information about azimuth, inclination, sky view factor, etc. can be stored.	
Special case 6.1: Single LoD3 thermal zone, with thermal boundaries and thermal openings. All LoD3 thermal boundaries/opening coincide exactly with the building LoD3 thematic surfaces/openings • Basically, same rules as for Case 5.1 • One thermal zone must be created. Set property *coincidesWithLoD3Hull* of the thermal zone to **TRUE** • The thermal zone can optionally be modelled as a LoD3 solid geometry. If applicable, XLinks can be used to reference the homologous polygons of the LoD3 solid • Neither the LoD3 thermal boundaries nor the LoD3 thermal openings must be modelled. The corresponding building LoD3 thematic surfaces and openings must be used instead • *LayeredConstructions* (i.e. U and g values) are associated to the building thematic surfaces/openings	

4.3.2 Using the new ADE classes for thermal boundaries

The Energy ADE 3.0 adds four new classes derived from *_BoundarySurface*: *PartyWallSurface*, *AtticFloorSurface*, *IntermediateFloorSurface*, and *BasementCeilingSurface*. Figure 22 and Figure 23 provide simple, schematic examples of when to use such new classes to model adjacent buildings and thermal zones, respectively. In these examples, class *PartyWallSurface* is represented in red and is intended to be used when modelling:

- Party walls between two adjacent buildings (e.g. between buildings A and B, and B and C)
- Party walls between two adjacent thermal zones (e.g. in building D)

Class *IntermediateFloorSurface* is represented in azure (see buildings D and E), and is intended to be used to model the horizontal surface subdividing two thermal zones, unless they correspond with the following two special cases.
Class *AtticFloorSurface* is used to model the horizontal surface of a thermal zone which is located under an "empty" space which is not modelled as a thermal zone. An example of such empty space could be an attic under the roof (see building E).
Class *BasementCeilingSurface* is used to model the horizontal surface of a thermal zone which is located above an "empty" space which is not modelled as a thermal zone. An example of such empty space could be an underground garage (see building E).

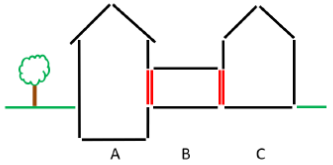


Figure 22. Schematic examples of surfaces modelled using class *PartyWallSurface* (in red).

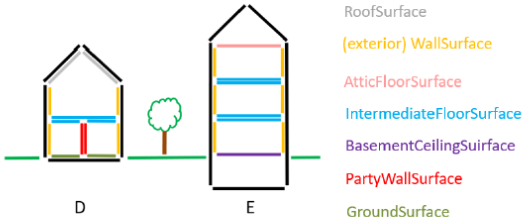


Figure 23. Schematic examples of thermal boundary surfaces modelled using different classes.

4.4 Occupancy module

4.4.1 Rule 5: Modelling and writing *Address* objects

Address objects must be always modelled as belonging to the respective parent *Building* (or *BuildingPart*) objects, as in standard CityGML. Additionally, in accordance with Rule 1 and as suggested by the annotations in the UML diagrams of the Occupancy module, the association between *BuildingUnit* and *Address* is of type "byReference", therefore it must be written using an XLink. An example is provided in Figure 17 (line 18).

4.4.2 Modelling geometries of *UsageZone* and *BuildingUnit* objects

Rule 2 applies also to modelling the geometries of *UsageZone* and *BuildingUnit* objects. All (optional) representations by means of solid geometries (for LoD1, LoD2 and LoD3), inherited from *AbstractBuildingPartition*, can be either explicit, or references to geometries used elsewhere, but within the same building *and* the same LoD. For example, if a LoD2 usage zone corresponds geometrically to the LoD2 thermal zone that is part of, then XLinks can be used between the LoD2 solid of the usage zone and the LoD2 solid of the thermal zone.

Test datasets

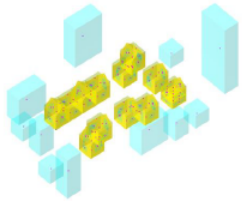
Alderaan_ADE_Building_Physics.gml

CityGML:
- CityObjectGroup
- Building, BuildingPart
- (Building) GroundSurface, WallSurface, RoofSurface
- (Building) Window
- Address
- Appearance, X3DMaterial

Energy ADE 2.0:
- CityObjectRelation
- ADE Building, ADE BuildingPart
- BasementCeilingSurface, IntermediateFloorSurface, PartyWallSurface
- ThermalZone

This file is the same as the previous one, but it adds ThermalZone objects, which are modelled via solid geometries (LoD1, LoD2 and LoD3) and via thermal boundaries (LoD2 and LoD3) and thermal openings (LoD3).

This dataset uses classes from the Energy ADE 2.0 Core and Building physics modules.



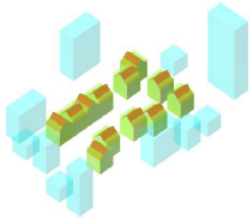
Alderaan_ADE_ThermalZone_CoincidesWithLoD2Hull.gml

CityGML:
- CityObjectGroup
- Building, BuildingPart
- (Building) GroundSurface, WallSurface, RoofSurface
- Address
- Appearance, X3DMaterial

Energy ADE 2.0:
- CityObjectRelation
- ADE Building, ADE BuildingPart
- PartyWallSurface
- ThermalZone

This file is contains ThermalZone objects which are modelled via solid geometries (LoD1, LoD2 and LoD3). The LoD2 representation via thermal boundaries coincides with the building envelope defined by the thematic surfaces. Therefore, thermal boundaries are not defined explicitly, instead each thermal zone has the coincidesWithLoD2Hull attribute set to TRUE.

This dataset uses classes from the Energy ADE 2.0 Core and Building physics modules.



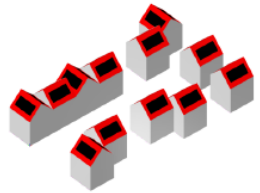
Alderaan_ADE_Devices.gml

CityGML:
- Building, BuildingPart
- (Building) GroundSurface, WallSurface, RoofSurface
- Address
- Appearance, X3DMaterial

Energy ADE 2.0:
- CityObjectRelation
- ADE Building, ADE BuildingPart
- Boiler, HeatPump, LightingDevice, GenericEletricalDevice, GenericDevice
- MovableShadingDevice
- ElectricalStorageDevice, ThermalStorageDevice
- GenericSolarCollector, SolarThermalCollector, PhotovoltaicCollector, PhotovoltaicThermalCollector

This file contains only the 11 "core" buildings, modelled only in LoD2 via thematic surfaces, and without shared walls. Additionally, all city objects are modelled via referencePoints.
The buildings contain objects from all classes of the Devices module. In particular, solar collectors are modelled in LoD2 and LoD3.

This dataset uses classes from the Energy ADE 2.0 Core and Devices modules.



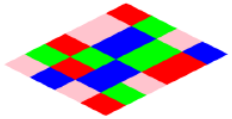
Alderaan_ADE_UrbanFunctionAreas.gml

CityGML:
- Appearance, X3DMaterial

Energy ADE 2.0:
- UrbanFunctionArea
- Energy
- Water

This file contains several UrbanFunctionAreas objects, organised in 3 hierarchical levels. The first-level object contains 6 second-level objects, which again contain each 4 third-level objects. The image on the right represents only the 24 third-level objects.
Each UrbanFunctionArea object is represented via a multisurface geometry and a referencePoint.
Some Energy and Water objects are associated with the UrbanFunctionArea objects, in order to provide an example of resources associated at each level.

This dataset uses classes from the Energy ADE 2.0 Core, Urban function areas and Resources modules.



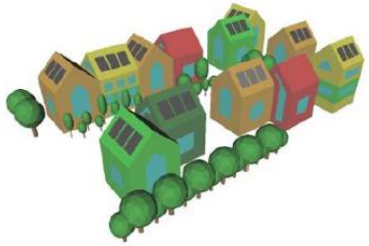
3D City Database 4.x

CityGML

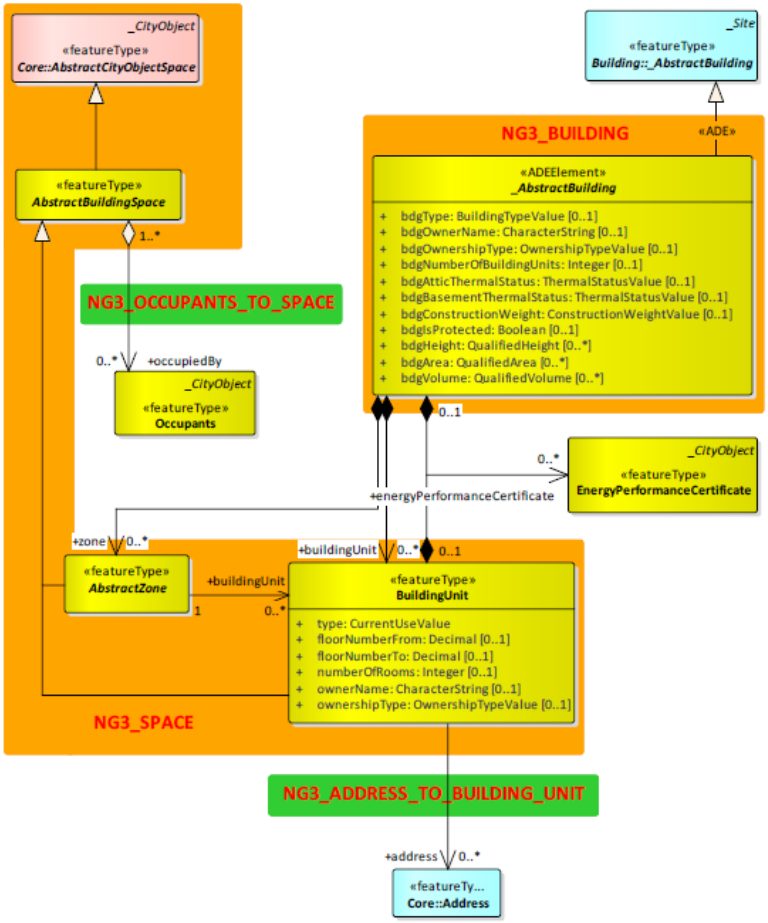
energyADE 3.0

Beta 8

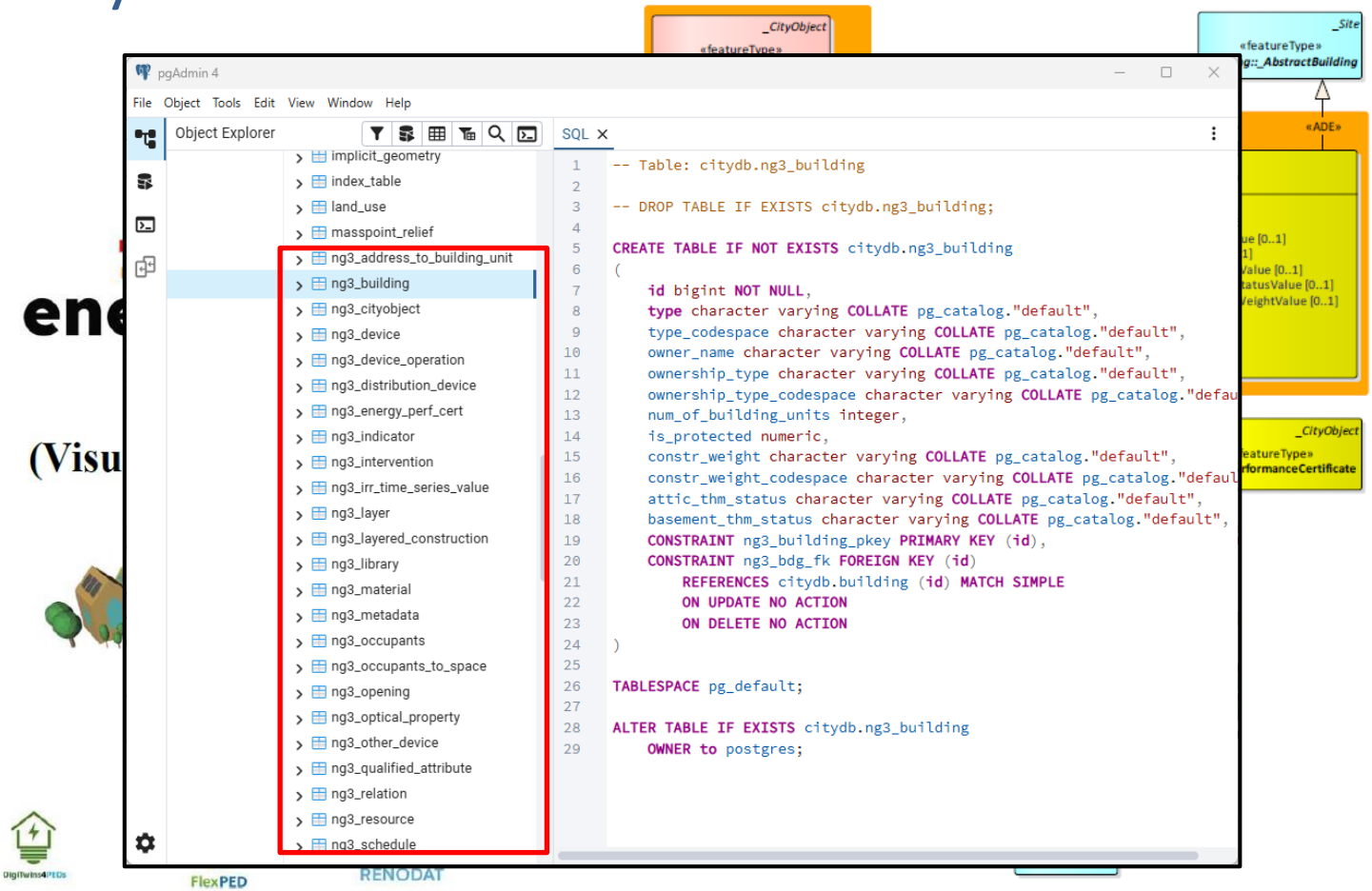
(Visual) ER Mapping



DigiTwins4PEDs FlexPED RENODAT

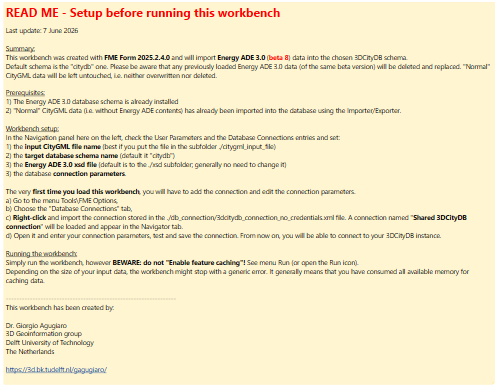


3D City Database 4.x



FME Workbench

- Import Energy ADE 3.0 data into the **3DCityDB 4.x** with Energy ADE extension



READ ME - Setup before running this workbench

Last update: 7 June 2026

Summary:
This workbench was created with **FME Form 2025.2.4.0** and will import **Energy ADE 3.0** (beta 8) data into the chosen 3DCityDB schema.
Default schema is the "citydb" one. Please be aware that any previously loaded Energy ADE 3.0 data (of the same beta version) will be deleted and replaced. "Normal" CityGML data will be left untouched, i.e. neither overwritten nor deleted.

Prerequisites:
1) The Energy ADE 3.0 database schema is already installed
2) "Normal" CityGML data (i.e. without Energy ADE contents) has already been imported into the database using the Importer/Exporter.

Workbench setup:
In the Navigation panel here on the left, check the User Parameters and the Database Connections entries and set:
1) the **Input CityGML file name** (best if you put the file in the subfolder ./citygml_input_file)
2) the **target database schema name** (default it "citydb")
3) the **Energy ADE 3.0 xsd file** (default is to the ./xsd subfolder; generally no need to change it)
3) the database **connection parameters**.

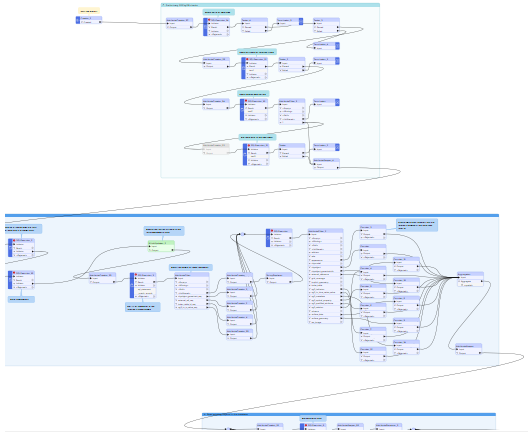
The very **first time you load this workbench**, you will have to add the connection and edit the connection parameters.
a) Go to the menu Tools\FME Options,
b) Choose the "Database Connections" tab,
c) **Right-click** and import the connection stored in the ./db_connection/3dcitydb_connection_no_credentials.xml file. A connection named "**Shared 3DCityDB connection**" will be loaded and appear in the Navigator tab.
d) Open it and enter your connection parameters, test and save the connection. From now on, you will be able to connect to your 3DCityDB instance.

Running the workbench:
Simply run the workbench, however **BEWARE: do not "Enable feature caching"!** See menu Run (or open the Run icon).
Depending on the size of your input data, the workbench might stop with a generic error. It generally means that you have consumed all available memory for caching data.

This workbench has been created by:

Dr. Giorgio Agugiaro
3D Geoinformation group
Delft University of Technology
The Netherlands

https://3d.bk.tudelft.nl/gagugiaro/



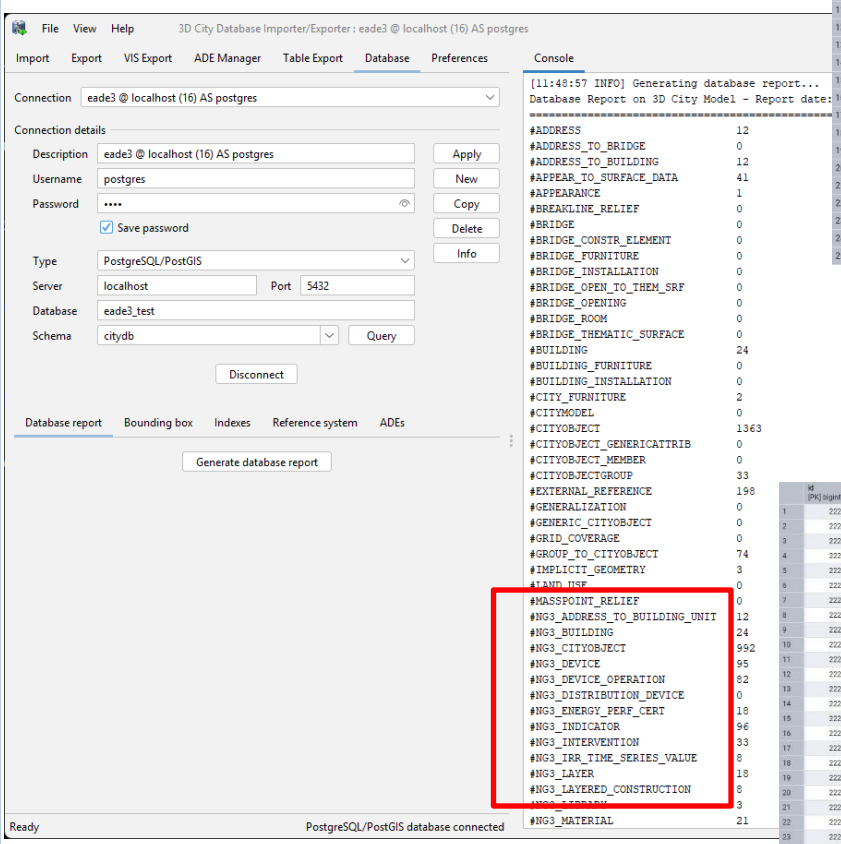
FME + 3DCityDB 4.x

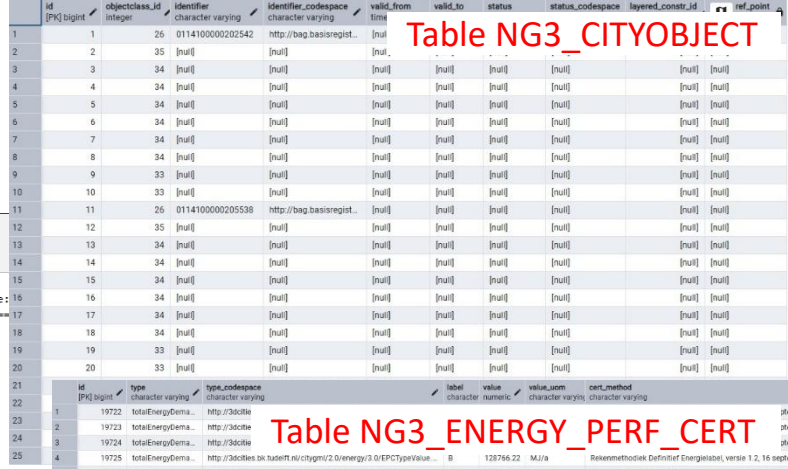
Introduction

Energy ADE 3.0

- Data model
- **Resources**

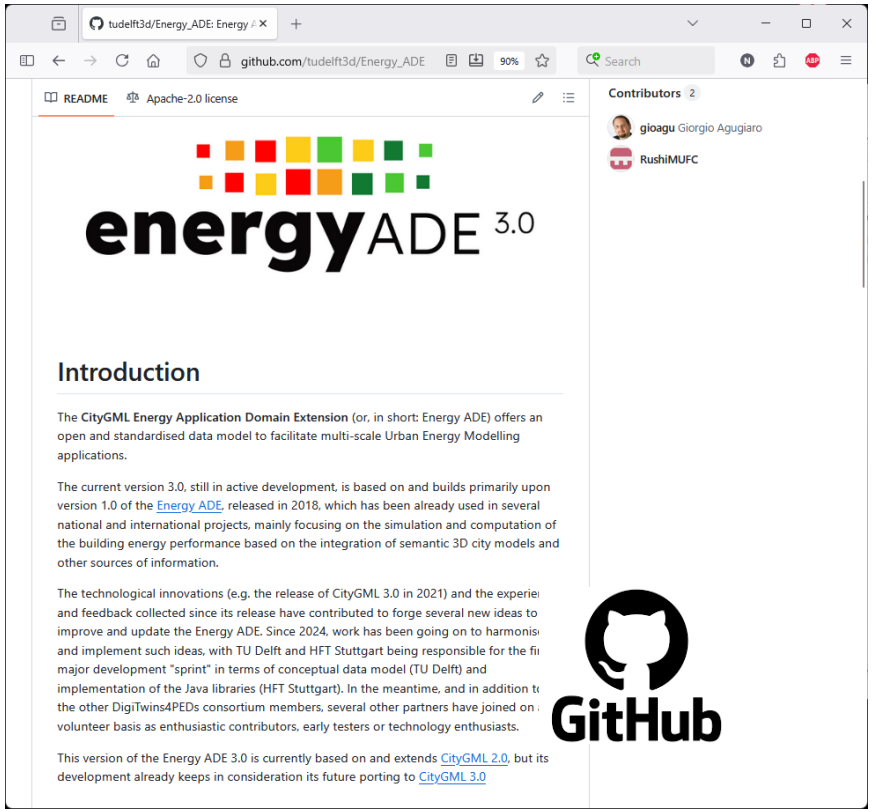
Conclusions





GitHub

- GitHub **open** since September 2025
  - https://github.com/tudelft3d/Energy_ADE



Conclusions

- The **Energy ADE 3.0** is the result or 2 years of development, started in spring 2024 within project DigiTwins4PEDs, and now continues in projects FlexPED and RenoDAT
- Additionally, several interviews with experts and stakeholders have been conducted to gather ideas, suggestions and to check the current maturity of the data model
- Focus is not only on the conceptual data model, but also on testing it in "real life" via some implementations
  - XML and SQL encodings are being kept "up to date" with the running development
- (The impression is that) it is already quite mature and complete, with some possible minor adjustments and improvements due to further testing
- However, some areas are still rather "new" and need some further thinking
  - The role of **metadata** and the need to find a balance to avoid "overbloating" the data model
  - The concept of "**money**" (costs, investments, etc.) is still very very simple. How far does it need to improve?
  - No **scenario management** so far, but see parallel (preliminary) development of **Scenario ADE** at TU Delft (Link)

Conclusions and next steps

- Small revisions / suggestions / bug fixes are still possible for beta 8
- If necessary, major change requests will go into beta 9 (not planned, but we'll see)
- Documentation for beta 8 planned to be written during summer 2026
- Java implementation will (re)start after beta 8 (or 9) is frozen

- **To do:**
  - ALL: go once over the current data model and spot errors, inconsistencies, ...
    - Report on GitHub!
  - ALL: Try and test it with your use cases, and provide feedback!
  - Careful check of description of classes, attributes, and UoM
  - Move documentation to online resources? Who wants to help?

- **Future steps** (some ideas)
  - Test with 3DCityDB 5.x (might requires some major customisation)
  - Start deeper analysis for porting to CityGML 3.0
  - Perform a thorough check of the mapping from Energy ADE to gbXML
  - Develop a Python library to work with CityGML and (any?) ADE...

Why not CityJSON (yet)?

Simple pragmatic reasons

- Development of Energy ADE 3.0 is on CityGML 2.0; CityJSON 2.0 is based on CityGML 3.0
- Better focus on getting the data model done first, before spreading effort on too many encodings at the same time

Some technical reasons

- CityJSON does not support/enforce natively **units of measure**. They can be added via an Extension, but why developing and extension for something native in (City)GML?
- CityJSON does not support natively **CodeLists** (i.e code + codespace). See above
- CityJSON does not support natively **ExternalReference**. See above
- CityJSON does not support **CityObjectRelation**. Only indirectly, only via hierarchies
- CityJSON does not support **Dynamizer** module. See above
- CityJSON does not support **Versioning** module. Alternative is still "work-in-progress"
- Extension mechanism does not support direct definition of children classes for existing classes
  - E.g. you need to "copy/paste" an existing "Building" class into a new "NewBuilding" one to have a new "Balcony" child class (when defining the JSON schema file)
- Which CityJSON version? **2.0.0** (OGC Community standard), or 2.0.1, or 2.0.2, or... **2.x.x**?

Thank you for your attention!

Dr. Giorgio Agugiaro
g.agugiaro@tudelft.nl
3D Geoinformation group
TU Delft - The Netherlands

https://3d.bk.tudelft.nl/gagugiaro
https://3d.bk.tudelft.nl/urbanrg/