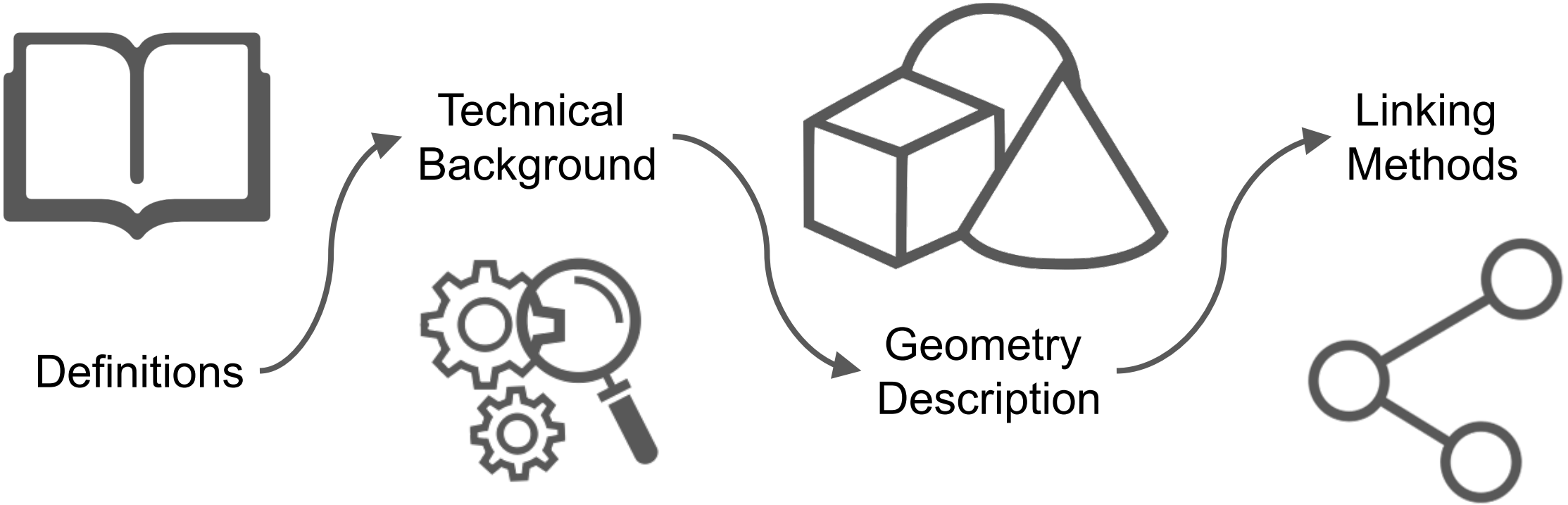
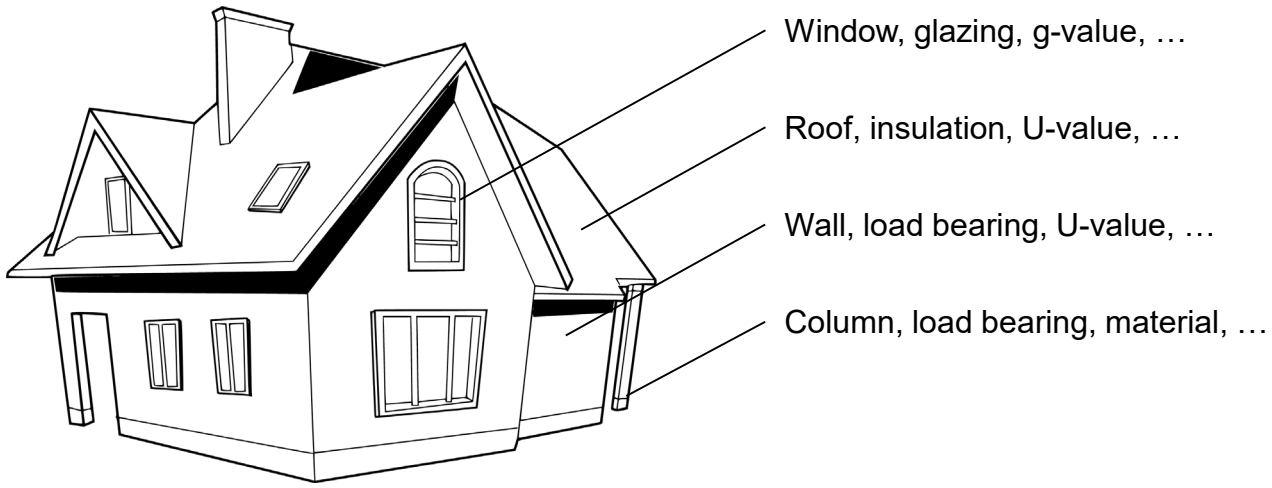
Geometry in a Semantic Web Context

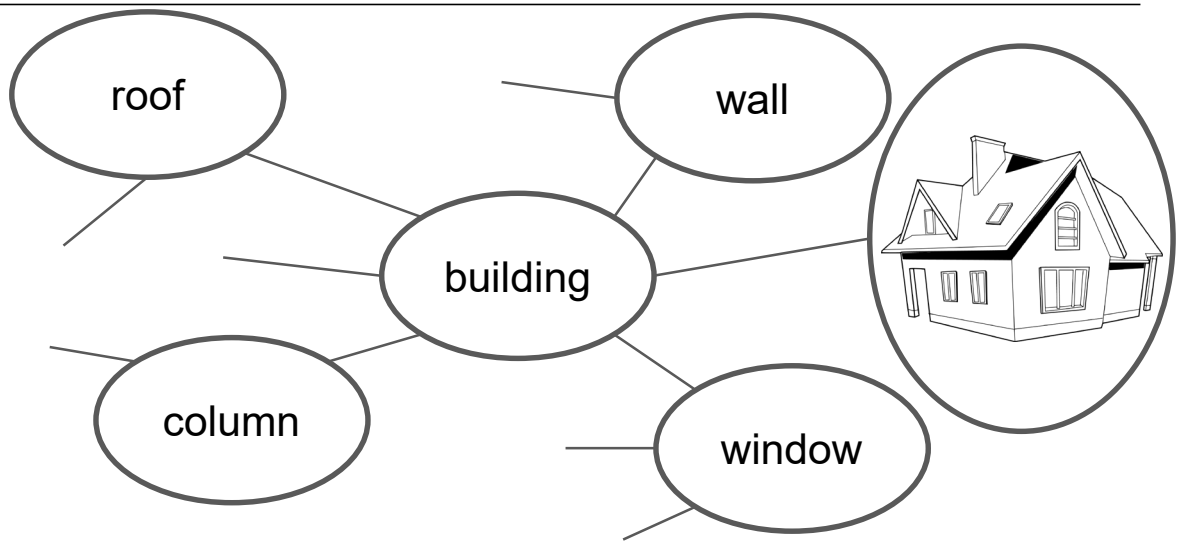
Lecturer: Anna Wagner, wagner@iib.tu-darmstadt.de

Assistants: Mathias Bonduel, Kris McGlinn



Geometry in a Semantic Web Context





Common CAD & BIM applications:

- Geometry as structure for information
- Details as materials & non-geometric properties are added to the geometry

→ Geometry represents its object

Semantic Web context:

- Data structured without geometry
- Each object might have ist own geometry description

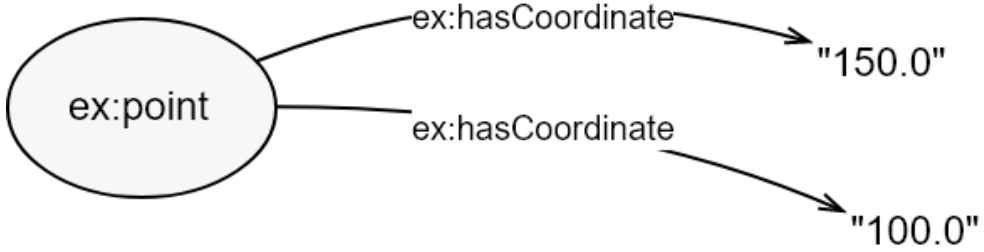
→ Geometry as „special kind“ of property

Ordered lists in RDF (1)

- Most geometry descriptions rely on ordered lists (e.g. arrays)
  - e.g. points, polylines, meshes
- In RDF, ordered lists are complex to describe
  - Lists (multiple values for the same property) are un-ordered and multiple values for the same property are treated as one triple
  - rdf:first, rdf:next, rdf:nil to structure ordered lists
  - If amount of values is known, dedicated properties can be introduced
    - e.g. x/y/z coordinates for points

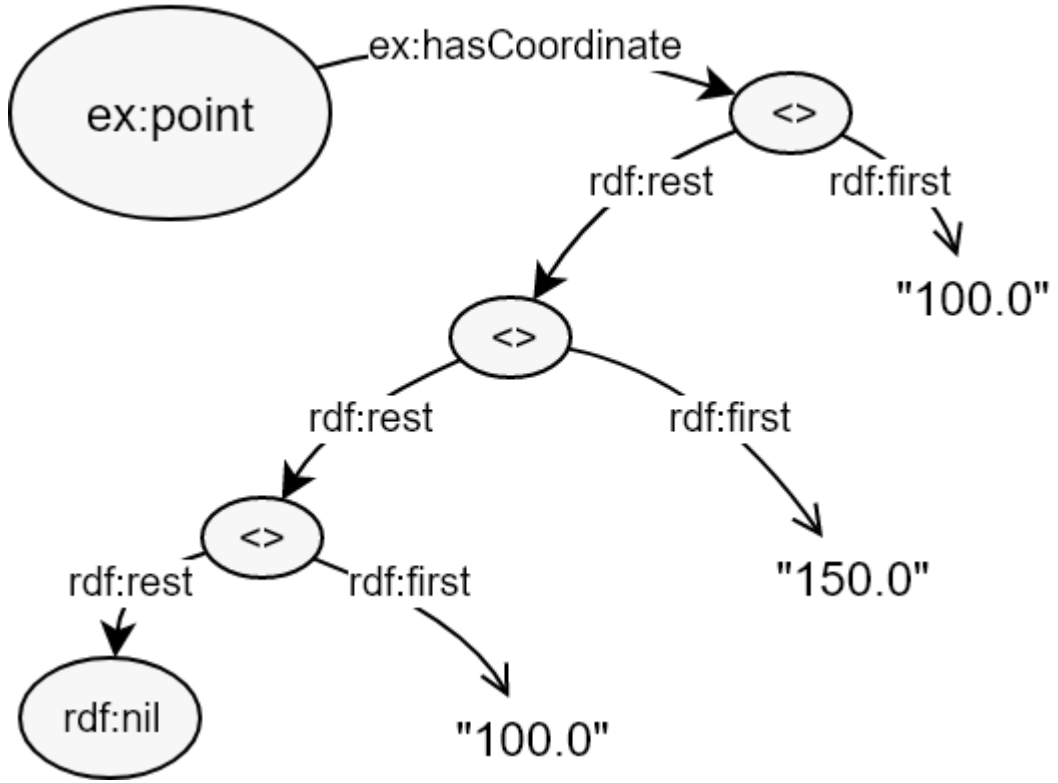
```
POINT(100.0, 150.0, 100.0)
```

```
ex:point a ex:Point ;
  ex:hasCoordinate “100.0”, “150.0”, “100.0” .
```



Ordered lists in RDF (2)

```
ex:point a ex:Point ;
  ex:hasCoordinate [
    rdf:first “100.0” ;
    rdf:rest [
      rdf:first “150.0” ;
      rdf:rest [
        rdf:first “100.0” ;
        rdf:rest rdf:nil . ] ; ] ; ] .
```

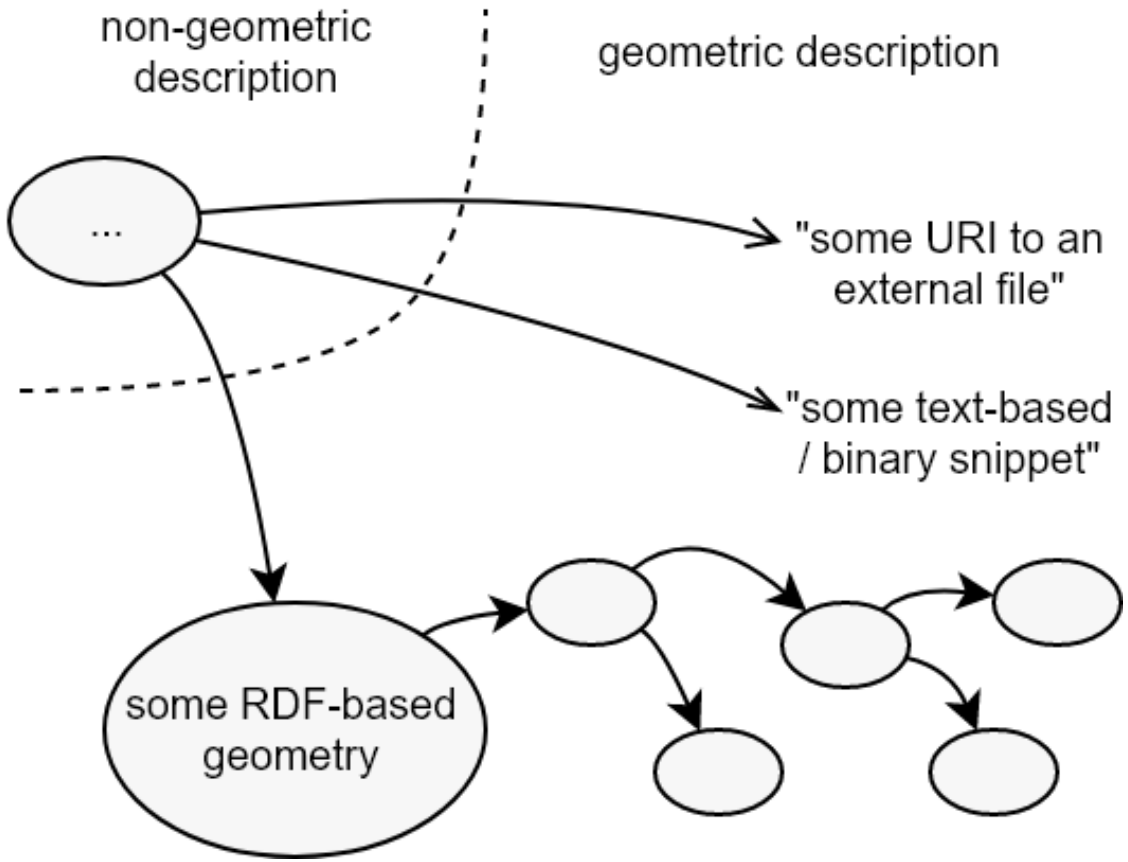


Turtle syntax (syntactic sugar)

```
ex:point a ex:Point ;
  ex:hasCoordinate (“100.0” “150.0” “100.0”) .
```

Describing geometry in a Semantic Web context

- Due to the inefficiency of describing ordered lists in RDF, non-RDF geometry schemes may be preferrable in certain use cases
- Different levels of integrity of geometry descriptions possible
  - Light integration by linking to external files
  - Medium integration of embedding snippets or entire geometry descriptions in RDF literals
  - High integration with using RDF-based geometry descriptions



Embedding geometry descriptions using RDF literals

TECHNISCHE UNIVERSITÄT DARMSTADT

Supporting implementations and demos for

- Well Known Text (WKT)
  - (mostly) 2D geometry as polygons
- Geography Markup Language (GML)
  - 2D geometry, focus on geospatial data

Spatial querying using SPARQL extensions:
GeoSPARQL (2D), stSPARQL (2D) and BimSPARQL (3D, WKT only)

- OBJ
  - Tessellated, 3D geometry

Visualisation of geometry in demo web applications
https://madsholten.github.io/ng-plan/index.html

GeoSPARQL with WKT
Defining geometries

TECHNISCHE UNIVERSITÄT DARMSTADT

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix ex: <http://www.example.org/POI#> .
@prefix sf: <http://www.opengis.net/ont/sf#> .

# Object of interest
ex:NationalMall a ex:Park;
 rdfs:label "National Mall";
 geo:hasGeometry ex:NMPoly . # Connection to geometry

# Geometry as polygon
ex:NMPoly a sf:Polygon;
 geo:asWKT "POLYGON((-77.050125 38.892086, -77.039482 38.892036, -77.039482
 38.895393, -77.033669 38.895508, -77.033585 38.892052, -77.031906 38.892086,
 77.031883 38.887474, - 77.050232 38.887142, -77.050125 38.892086
))"^^geo:wktLiteral. # Polygon description in WKT
```

GeoSPARQL with WKT
Querying

TECHNISCHE UNIVERSITÄT DARMSTADT

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
SELECT ?what
WHERE {
  ?what geo:hasGeometry ?geometry .
  FILTER(geof:sfWithin(?geometry,
   "POLYGON((-77.089005 38.913574,-77.029953 38.913574,-77.029953
   38.886321,-77.089005 38.886321,-77.089005 38.913574))"^^geo:wktLiteral))
```

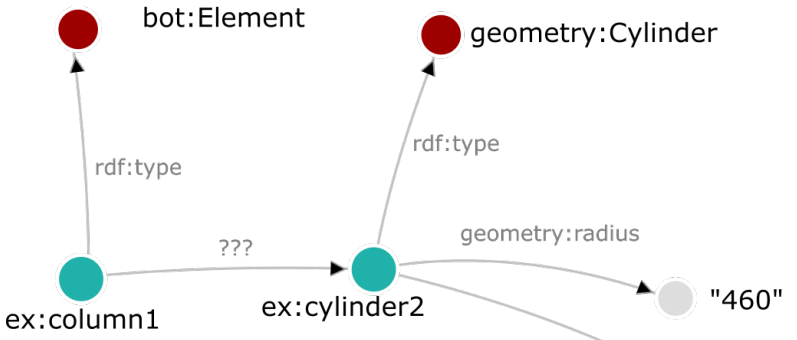
“What is within the bounding box defined by 38.913574°N 77.089005°W and 38.886321°N 77.029953°W ?”

RDF-based geometry descriptions

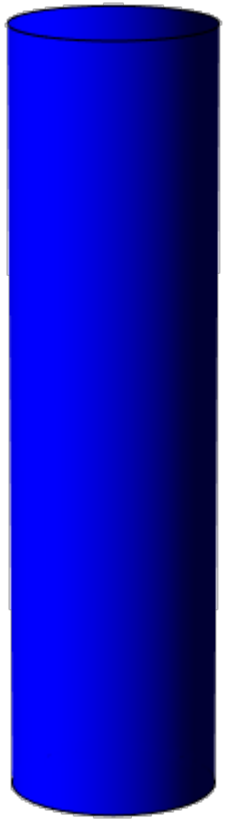
TECHNISCHE UNIVERSITÄT DARMSTADT

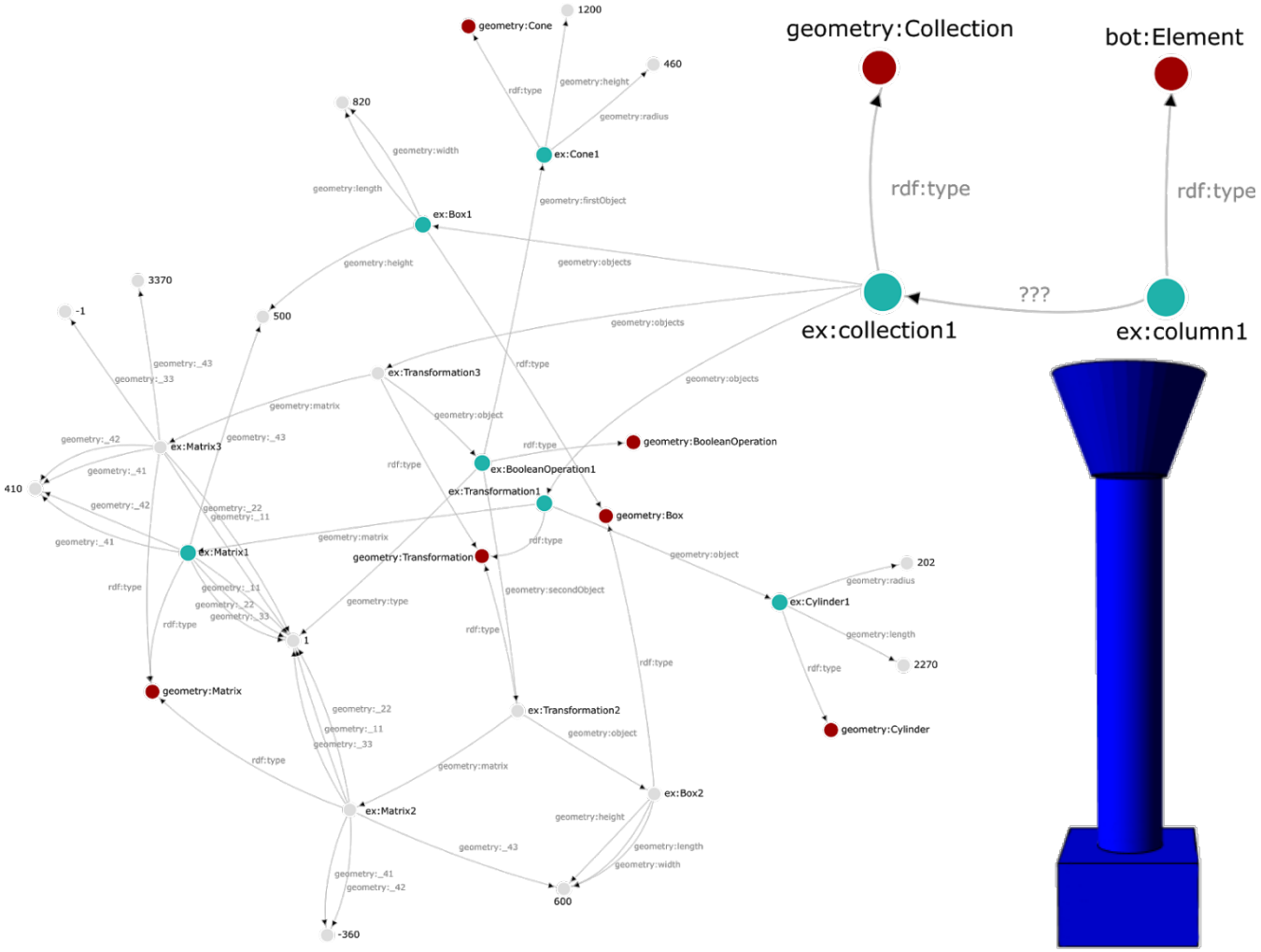
- **ifcOWL:** based on the IFC schema and thereby includes the geometric part of the IFC, but cannot be extracted easily due to its lack of modularity; https://github.com/buildingSMART/ifcOWL
- **GEOM:** dedicated ontology for geometry descriptions, does not support ordered lists in RDF, only applicable for feature-based geometry descriptions; http://rdf.bg/geometry.ttl
- **OntoBREP:** dedicated ontology for geometry descriptions in a BREP geometry representation; https://github.com/OntoBREP/ontobrep
- **OntoSTEP:** based on the STEP schema, can be applied for any AP; https://www.nist.gov/services-resources/software/ontostep-plugin
- **3DMO:** based on the X3D schema, originates as annotation ontology; http://web.archive.org/web/20180831114523/http://3dontology.org/3d.ttl

GEOM example
feature-based geometry description



Since GEOM does not implement ordered lists in RDF, any geometry representation that requires arrays cannot be described by the schema (e.g. tessellated or BREP)





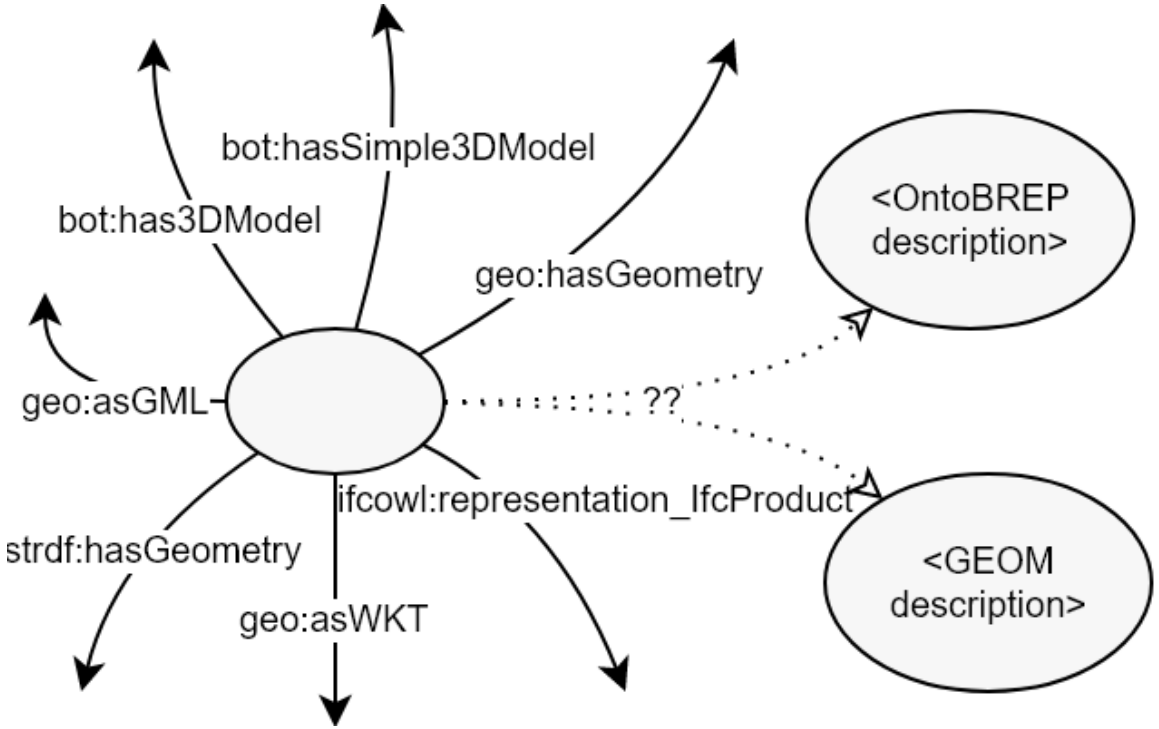
Overview for geometry in a Semantic Web context

TECHNISCHE UNIVERSITÄT DARMSTADT

Linking to external files	Embedding snippets	RDF-based descriptions
+ can be applied to any geometry schema + no additional efforts needed + small overall file size and triple count	+ can be applied to any geometry schema + spatial querying for some geometry schemes supported + geometry is part of the RDF graph	+ querying directly on geometry description + linking between non-geometric parts and geometric objects or properties possible
− geometry description not as part of the RDF graph (must be kept available elsewhere) − geometry cannot be included in any form of querying − linking to geometric objects or properties not possible	− additional efforts (for escaping and encoding) required − larger file sizes than linking to ext. files (in case of text-encoded binary files) − linking to geometric objects or properties not possible	− large file sizes / triple counts − few geometry schemes available − little support of existing schemes − spatial querying currently not available for any schema

Linking between non-geometric objects and their geometry description

- Different methods for describing geometry in a Semantic Web context
- Various geometry schemes that can be used
- Inconsistent definition of terms for linking a non-geometric object to its geometry description
  - GeoSPARQL
  - stSPARQL/stRDF
  - ifcOWL
  - BOT
  - …



Ontologies for uniform linking
OMG & FOG

TECHNISCHE UNIVERSITÄT DARMSTADT

Ontology for Managing Geometry (OMG)

- Defines relations between non-geometric objects and geometries for a uniform way of modelling
- Allows multiple geeometry representations per object
- Provides means to describe dependencies between geometry descriptions and / or properties
- Introduces concepts for version management and grouping of geometries

File Ontology for Geometry formats (FOG)

- Extension of OMG
- Taxonomy of specific relations between non-geometric objects and geometry per geometry schema to enhance data exchange

OMG
Basic concept



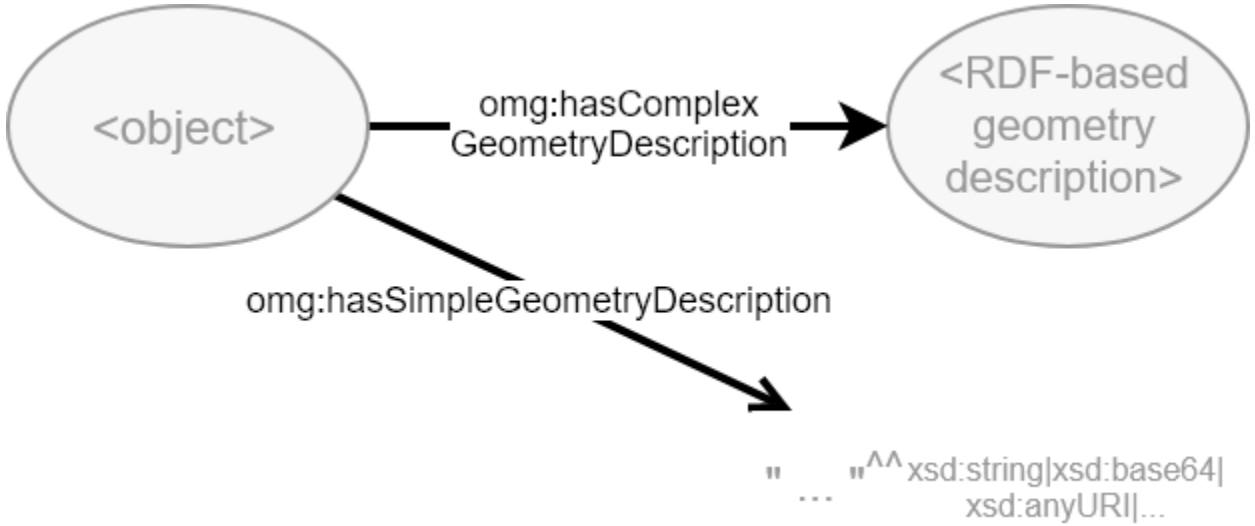


TECHNISCHE UNIVERSITÄT DARMSTADT

- Small ontology (4 classes, 1 datatype property, 13 object properties)
- Re-uses concepts and vocabulary from SEAS, PROV-O and OPM
- Introduces three levels for connecting geometry to objects (similar to OPM)
  - **Level 1:** direct link from object to geometry
  - **Level 2:** adding a ‚Geometry‘ node between geometry and object for meta-data
  - **Level 3:** adding a further ‚GeometryState‘ node between the ‚Geometry‘ node and geometry for version control
- Provides concepts to describe derivatives of geometries
  - Geometry to geometry
  - Property to geometry and geometric properties
- Introduces a grouping mechanism for easier querying (geometry context)

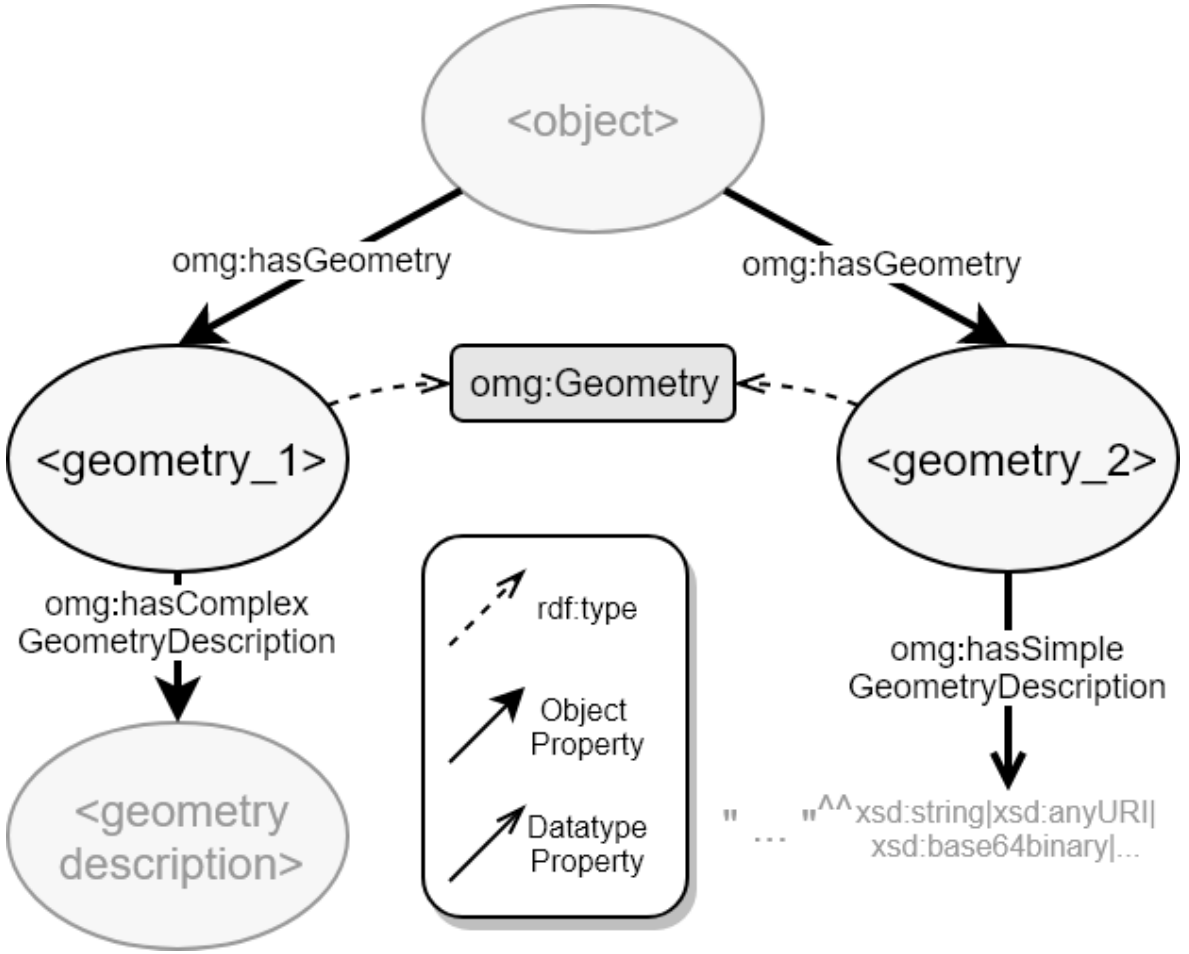
OMG
Level 1

- Directly connects geometry to any object
- Object property for complex, RDF-based geometry descriptions
- Datatype property for simple, non-RDF based geometry descriptions that can be added to the graph as snippets in RDF literals or via links to external files



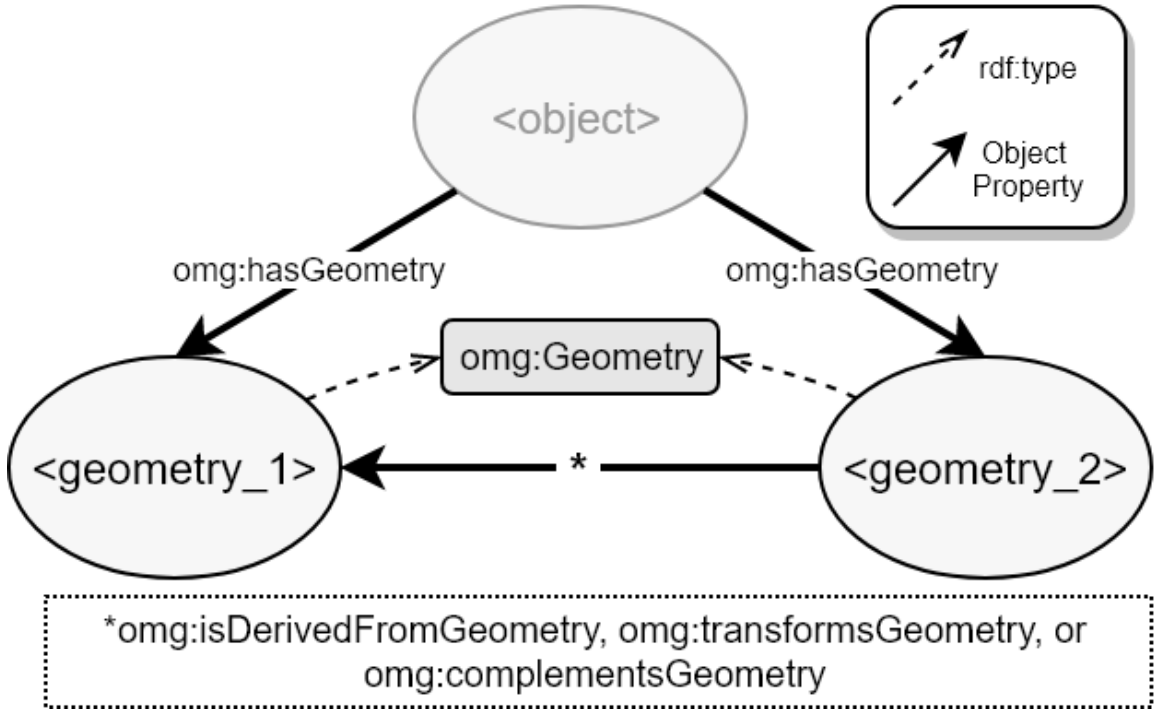
OMG
Level 2

- Intermediate ‚Geometry' node for meta-data
- One object can have multiple geometry descriptions that can be clearly differentiated and addressed separately
- Generic relation between object and geometry nodes
- Relations of OMG level 1 applied from geometry node to geometry description



OMG
Relating geometry descriptions

- **isDerivedFromGeometry:** *geometry_2* can be calculated from *geometry_1*, in case *geometry_1* is manipulated
- **transformsGeometry:** if *geometry_2* was the same geometry as *geometry_1* but in a different location (transformed), *geometry_2* may hold information regarding the transformation only while the geometry description itself is defined in *geometry_1*
- **complementsGeometry:** *geometry_1* describes an object, but *geometry_2* describes additional details of the same object (e.g. detailed carvings)



OMG
Overview of functionalities per level

TECHNISCHE UNIVERSITÄT DARMSTADT

	level 1	level 2	level 3
connecting geometry	yes	yes	yes
multiple geometry representations	no*	yes	yes
defining dependencies between geometries	no	yes	yes
versioning of geometry	no	no	yes
explicitly derived properties	yes	yes	yes
implicitly derived properties	no	yes	yes

* no differentiation between different representations possible, especially for descriptions relying on multiple parts

OMG
Demo and documentation

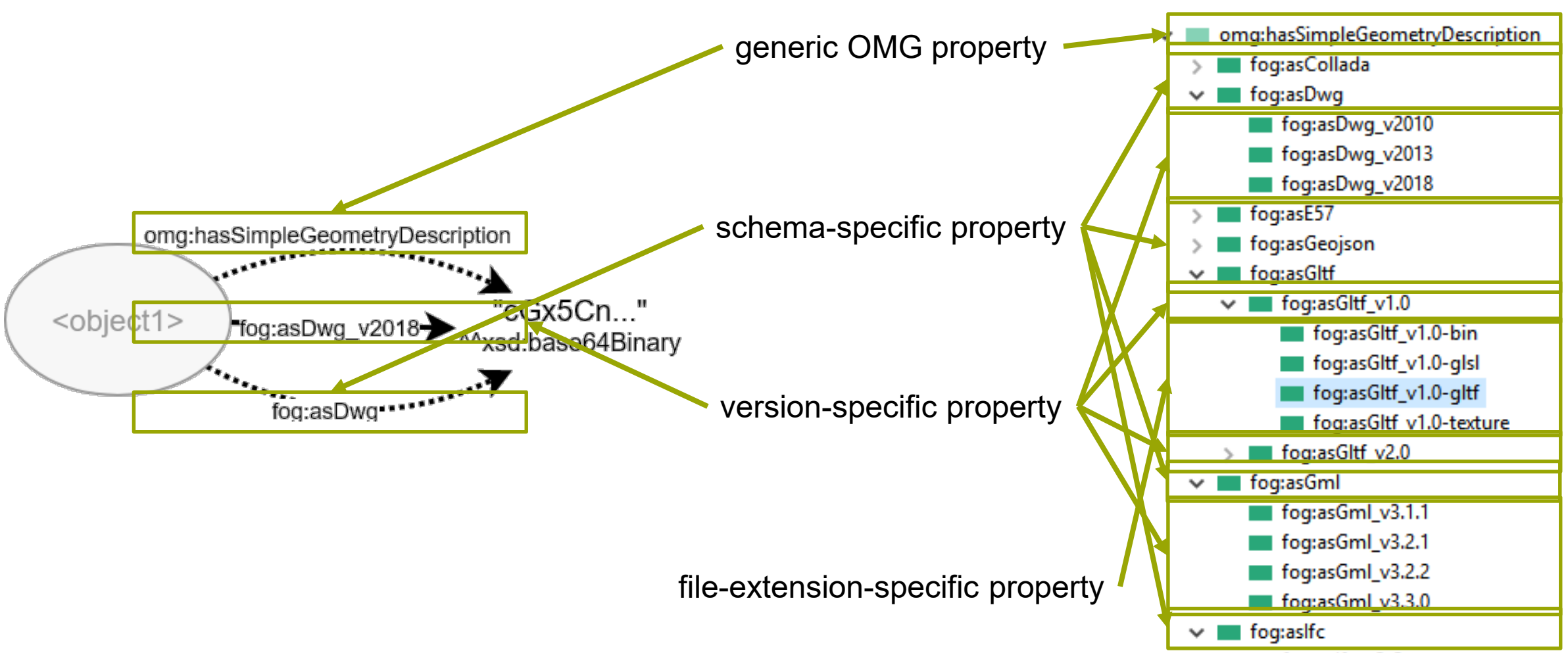
- Namespace https://w3id.org/omg#
  - HTML documentation: https://w3id.org/omg/
  - Raw ontology: https://gitlab.iib.tu-darmstadt.de/Wagner/omg.git
- Demo
  - SPARQL visualiser: https://madsholten.github.io/sparql-visualizer/?file=https:%2F%2Fwww.dropbox.com%2Fs%2Fg1c9oclaxv1l8ud%2Fomg-demo.json
- Publications
  - EC3 paper in print

FOG
Basic concept

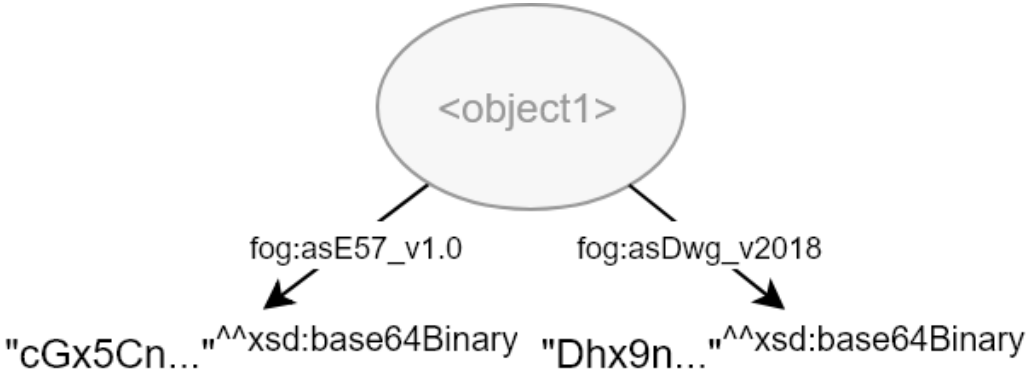
TECHNISCHE UNIVERSITÄT DARMSTADT

- Extends the `omg:hasSimpleGeometryDescription` and `omg:hasComplexGeometryDescription` properties to add specific properties for geometry schemes
- Introduces a taxonomy of properties
  - geometry schema including ist version
  - file extension
    - enables connecting geometry schemes with multiple associated files, e.g. OBJ with .obj and .mtl files
- Adds metadata per geometry schema
  - specification and documentation
  - associated files and their extensions
- Taxonomy maintenance as community effort: https://github.com/mathib/fog-ontology
  - for adding uncovered or new geometry schemes

FOG
Taxonomy



FOG
Application

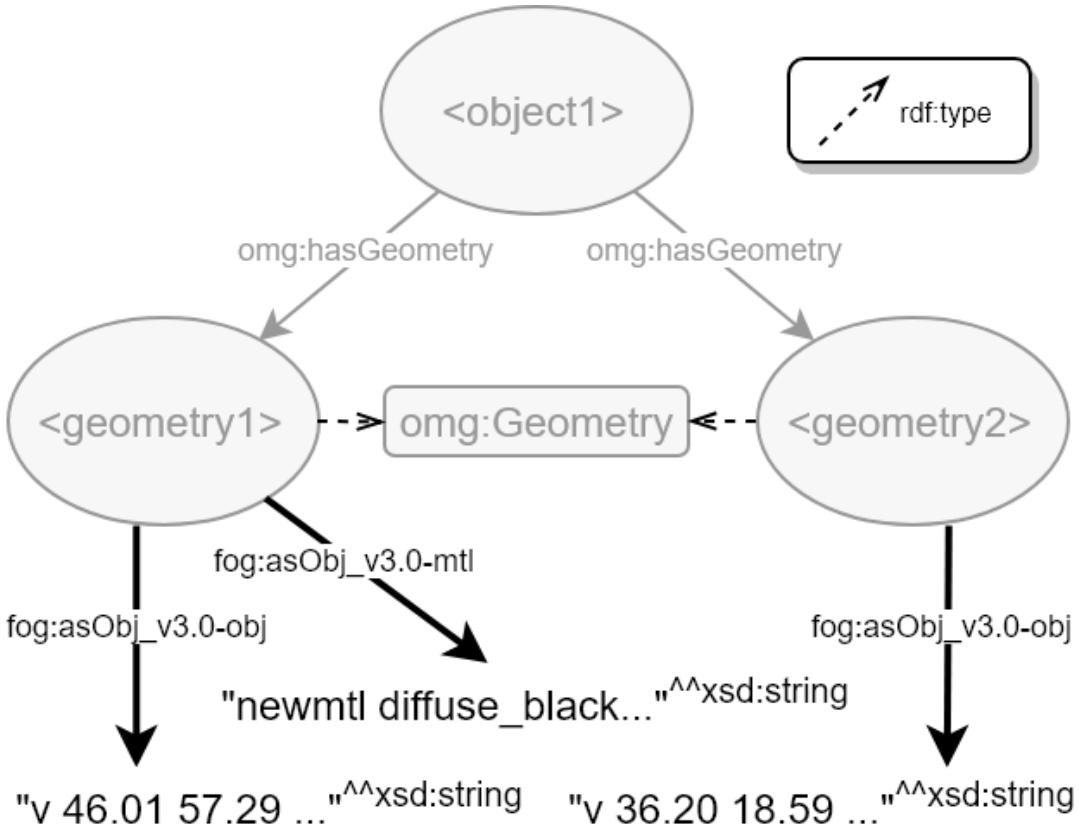


Level 1

- Multiple geometry descriptions of the same schema cannot be differentiated

Level 2

- Differentiation on node-level allows application of multiple geometry descriptions



FOG
Demo and documentation

Namespace https://w3id.org/fog#

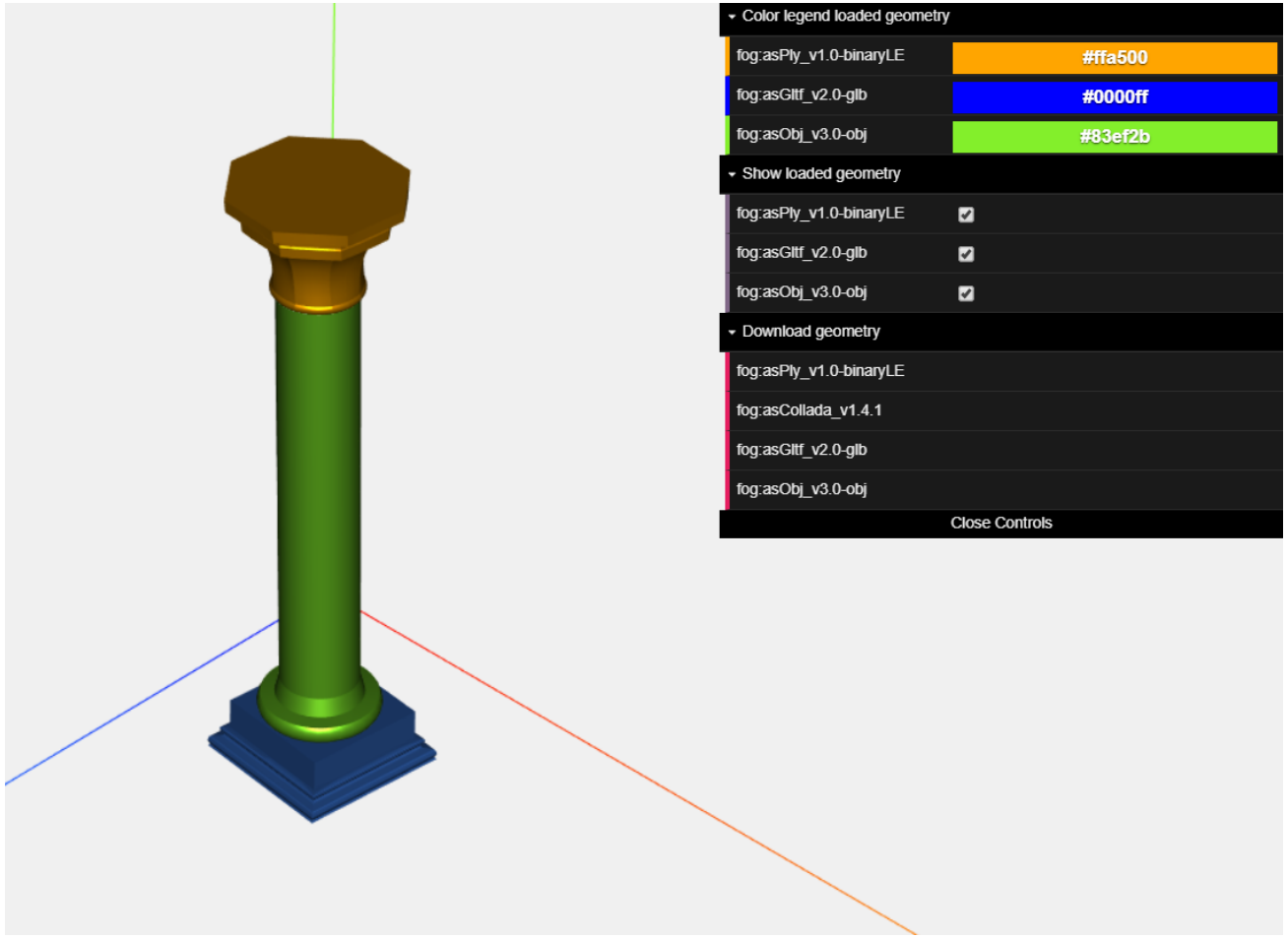
- HTML documentation: https://w3id.org/fog/
- Raw ontology: https://github.com/mathib/fog-ontology

Demo

- SPARQL-visualiser https://madsholten.github.io/sparql-visualizer/?file=https:%2F%2Fwww.dropbox.com%2Fs%2Fbch9flrxoskw29v%2Ffog-demo.json
- Webviewer (right)

Publications

- EC3 paper Bonduel



OMG & FOG
Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

- Different methods for describing geometry in a Semantic Web context
  - Linking to external files, embeddings snippets, using RDF-based geometry schemes
- For unified querying, a uniform linking approach should be applied
  - OMG:
    - introduces concepts to implement any method above
    - enables the handling of multiple geometry descriptions
    - provides means for tracking versions and dependencies
  - FOG:
    - extends OMG to define geometry-schema-specific properties
    - unambiguous definition of used schema, version and file extension

Any Questions?

THANK YOU FOR YOUR ATTENTION