Towards a PROPS ontology

W3C Linked Building Data Community Group

12th of March 2018

Mathias Bonduel – PhD researcher @ KU Leuven and Ugent

KU LEUVEN

TECHNOLOGIECAMPUS GENT

mathias.bonduel@kuleuven.be

UNIVERSITEIT GENT

Content

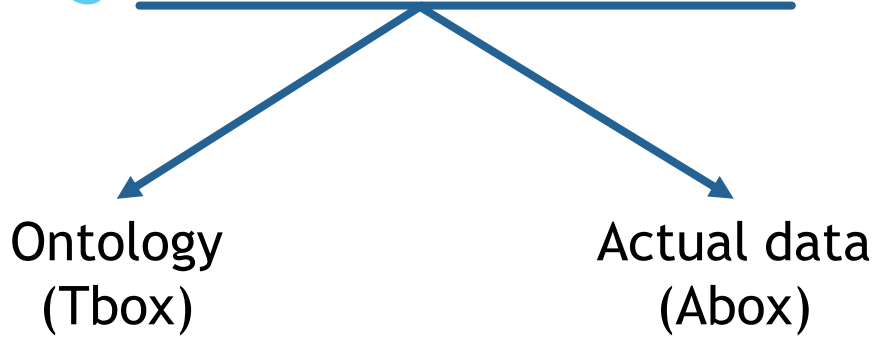
- **0. Introduction:** scope of PROPS ontology
- **1. The need for standardization: existing approaches**
  - Ontology structure: basic data structure / patterns
  - Ontology content
- **2. Considerations for ontology structure**
  - Competency questions / functionality
  - New proposals: pros and cons
  - A range of solutions with different levels of complexity?
- **3. Next steps:** standardization of:
  - Ontology structure
  - Ontology content
- **4. sparql-visualizer tool:** functionality
- **Sources – further reading - links**

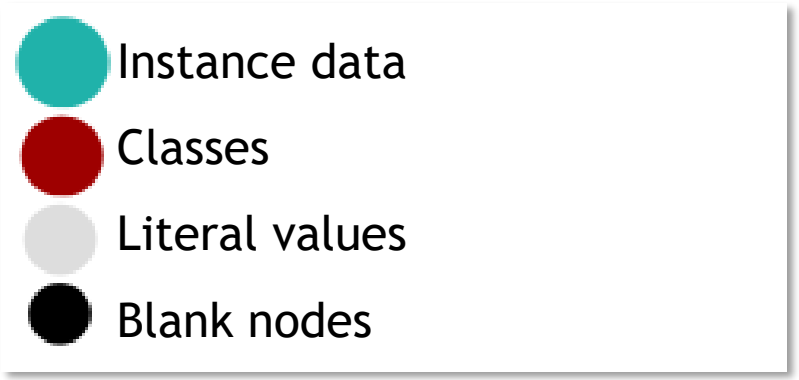
0. Introduction
Scope of PROPS ontology

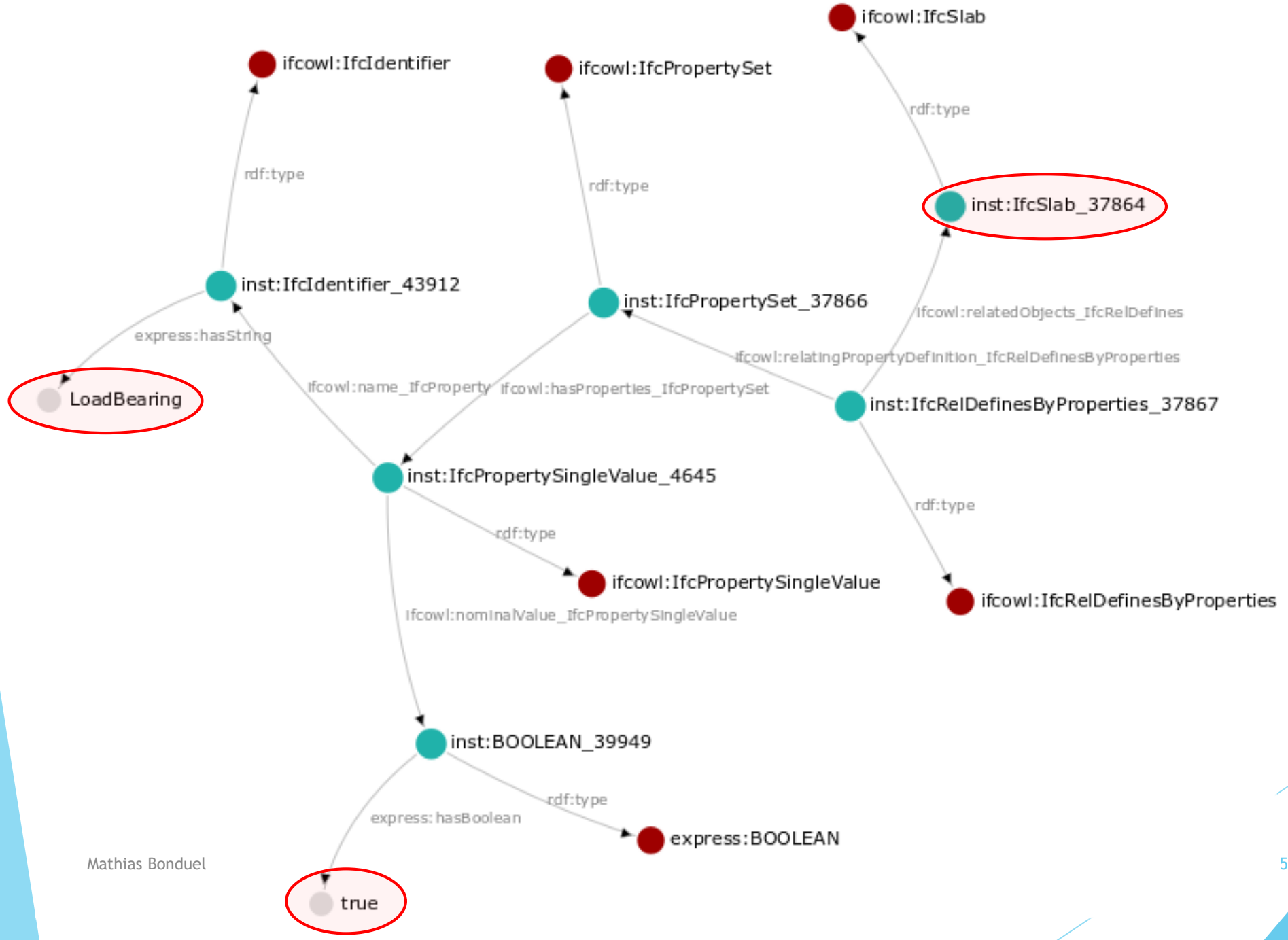
- Building-related properties of:
  - Building elements:
    - bot:Element: elements and parts of elements (bot:aggregates property)
    - PROD ontology classes: products and parts of products
  - Others:
    - bot:Zone (bot:Site, bot:Building, bot:BuildingStorey, bot:Space)
    - bot:Interface
    - Materials? (material-MTL ontology?)
- Alignment to other domains: e-commerce, mechanical engineering, etc.

1. The need for standardization:
existing approaches regarding basic data structure

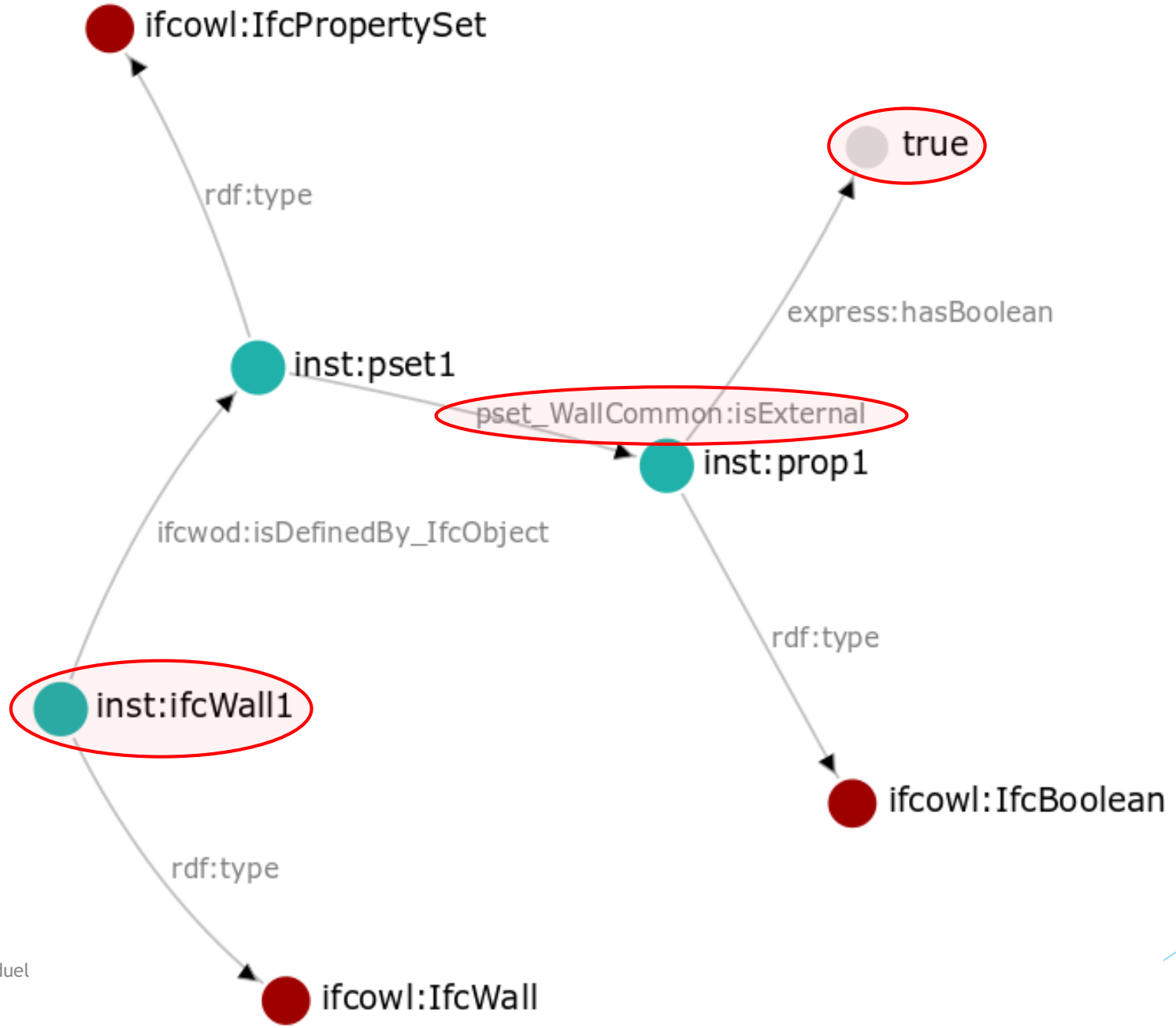
- IFC based
  - ifcOWL
  - IfcWoD and simpleBIM
- BIM based (general)
  - BIMSO – BIMDO
- E-commerce
  - Good Relations – schema.org
- Sensor data?
- Not based on traditional BIM?

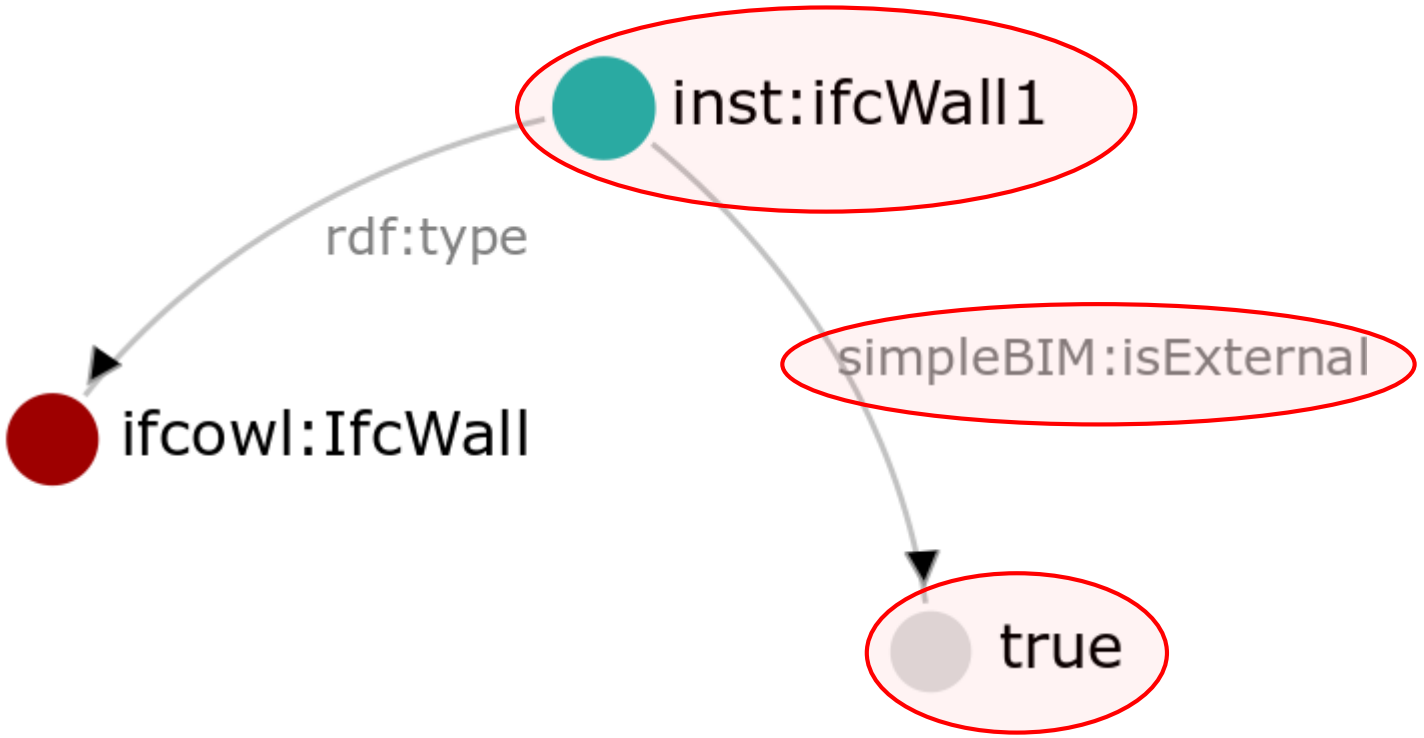




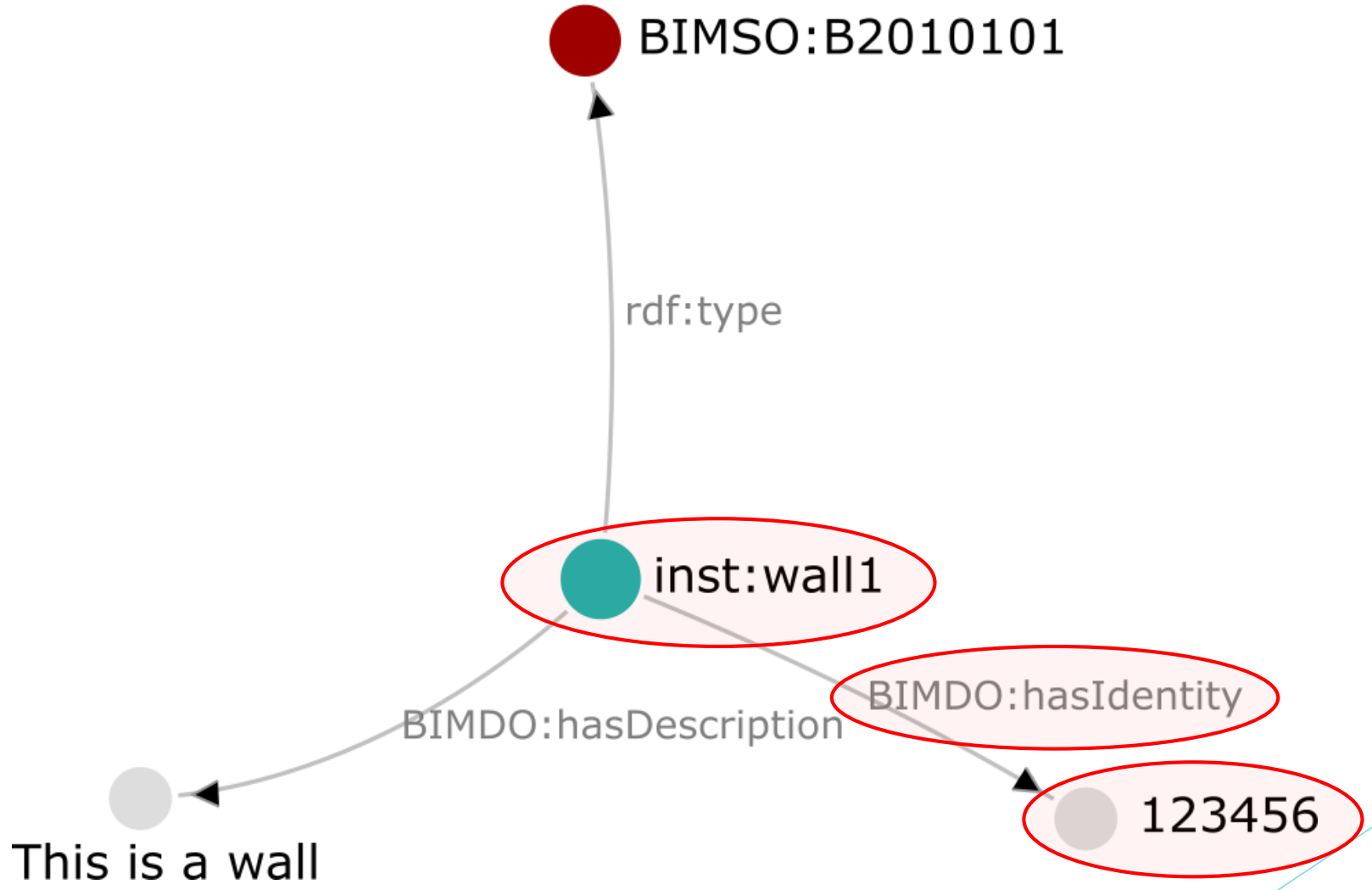

ifcowl:IfcSlab
ifcowl:IfcIdentifier
ifcowl:IfcPropertySet
rdf:type
rdf:type
rdf:type
inst:IfcSlab_37864
inst:IfcIdentifier_43912
inst:IfcPropertySet_37866
ifcowl:relatedObjects_IfcRelDefines
ifcowl:relatingPropertyDefinition_IfcRelDefinesByProperties
express:hasString
ifcowl:name_IfcProperty
ifcowl:hasProperties_IfcPropertySet
LoadBearing
inst:IfcRelDefinesByProperties_37867
inst:IfcPropertySingleValue_4645
rdf:type
rdf:type
ifcowl:IfcPropertySingleValue
ifcowl:IfcRelDefinesByProperties
ifcowl:nominalValue_IfcPropertySingleValue
inst:BOOLEAN_39949
rdf:type
express:hasBoolean
express:BOOLEAN
true

ifcOWL

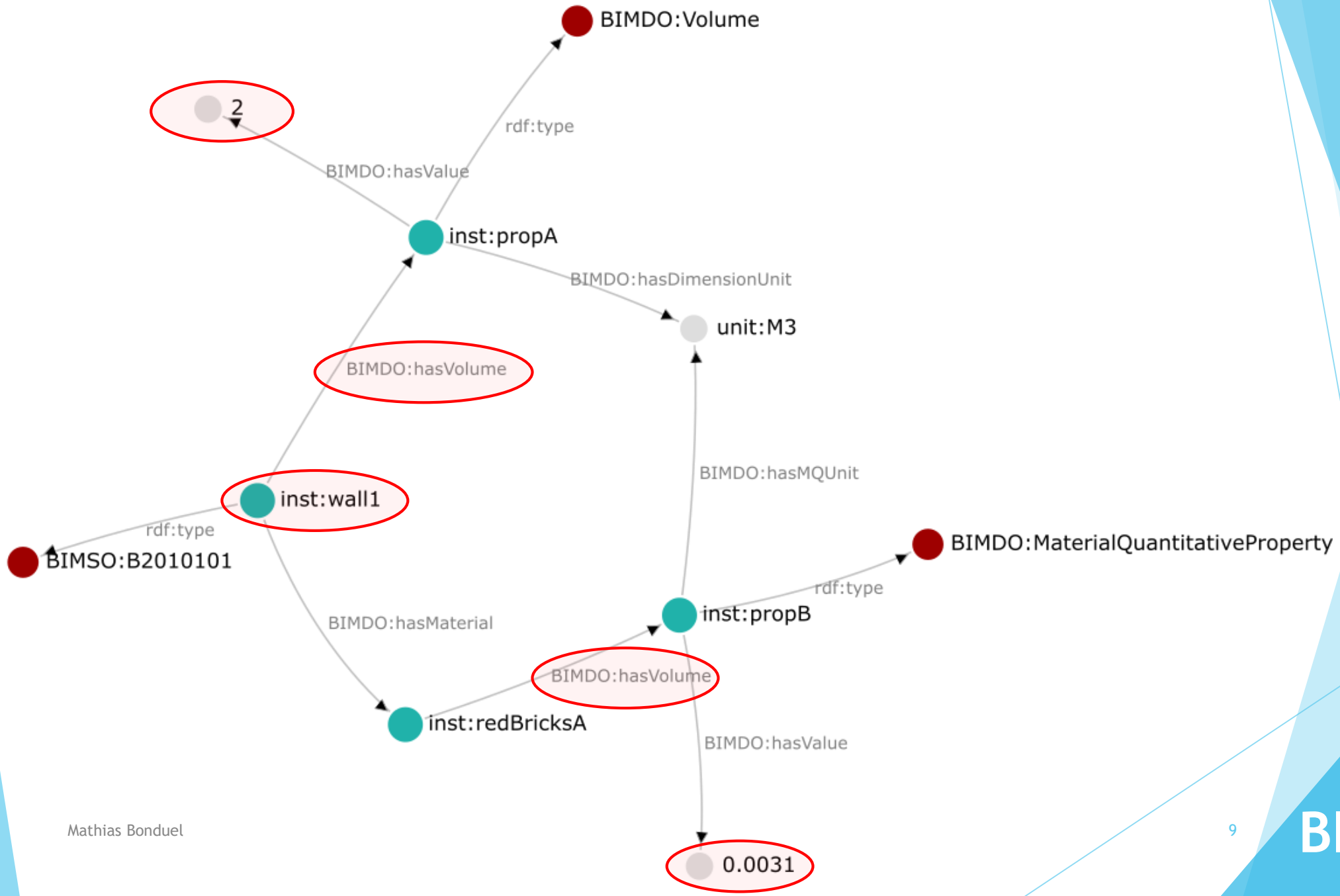

ifcowl:IfcPropertySet
rdf:type
true
express:hasBoolean
inst:pset1
pset_WallCommon:isExternal
inst:prop1
ifcwod:isDefinedBy_IfcObject
rdf:type
inst:ifcWall1
ifcowl:IfcBoolean
rdf:type
ifcowl:IfcWall



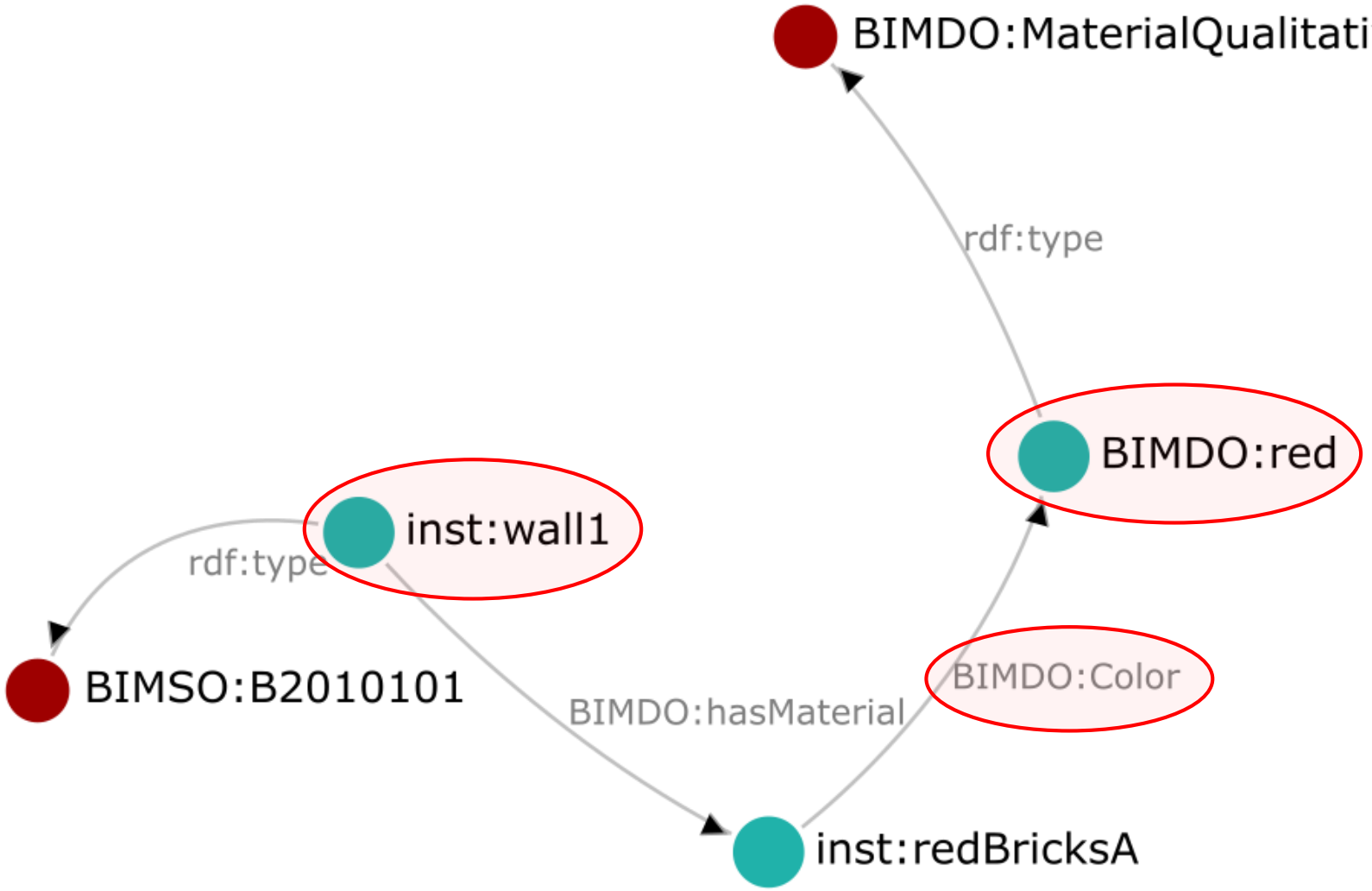
simpleBIM
(~PSET generator)


BIMSO:B2010101
rdf:type
inst:wall1
BIMDO:hasDescription
BIMDO:hasIdentity
This is a wall
123456

BIMSO-BIMDO


BIMDO:Volume
2
rdf:type
BIMDO:hasValue
inst:propA
BIMDO:hasDimensionUnit
unit:M3
BIMDO:hasVolume
BIMDO:hasMQUnit
inst:wall1
rdf:type
BIMSO:B2010101
BIMDO:MaterialQuantitativeProperty
rdf:type
inst:propB
BIMDO:hasMaterial
BIMDO:hasVolume
inst:redBricksA
BIMDO:hasValue
0.0031

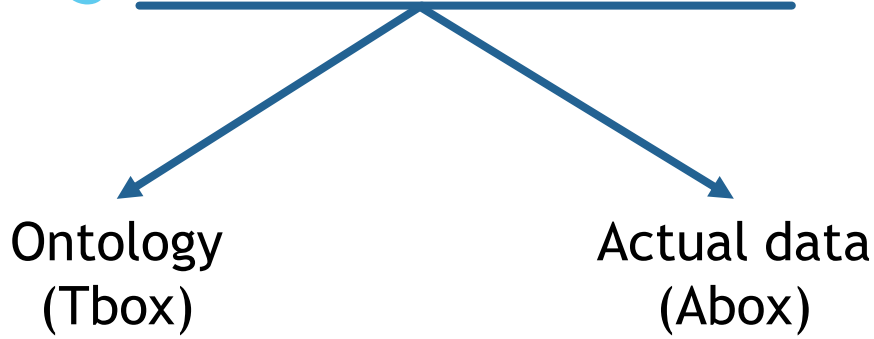
BIMSO-BIMDO

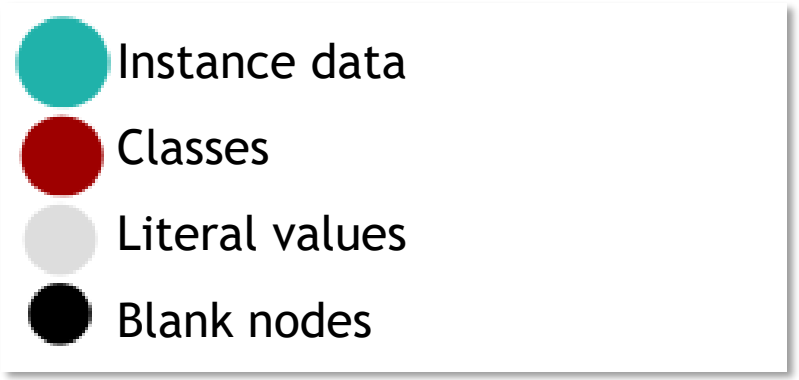

BIMDO:MaterialQualitativeProperty
rdf:type
BIMDO:red
inst:wall1
rdf:type
BIMSO:B2010101
BIMDO:hasMaterial
BIMDO:Color
inst:redBricksA

BIMSO-BIMDO

1. The need for standardization:
existing approaches regarding basic data structure

- IFC based
  - ifcOWL
  - IfcWoD and simpleBIM
- BIM in general
  - BIMSO – BIMDO
- E-commerce
  - Good Relations – schema.org
- Sensor data?
- Not based on traditional BIM?





1. The need for standardization:
existing approaches regarding ontology content

- Based on IFC: PSET generator (Maxime)
- Based on Wikipedia (PTO approach – implemented by Mads)
- Based on Wikidata (Mads)
- Custom properties (user defined): what should be provided + how to extend existing PROPS ontology?

- QUDT ontology
  - Domains: construction / architecture / civil engineering is absent
  - Properties available, but contain no extra information (e.g. qudt:Volume)
- OM ontology
  - Properties available with extra information (om:Volume)
- Schema.org ontology (e.g. schema:width)
  - Limited to generic properties
  - Not OWL but rdf:Property (?)

2. Considerations for ontology structure

- Possibilities:
  - Datatype properties?
  - Object properties?
  - Combination of (one or multiple) object and datatype properties?
  - Combination of (one or multiple) object and datatype properties, together with owl:Classes?

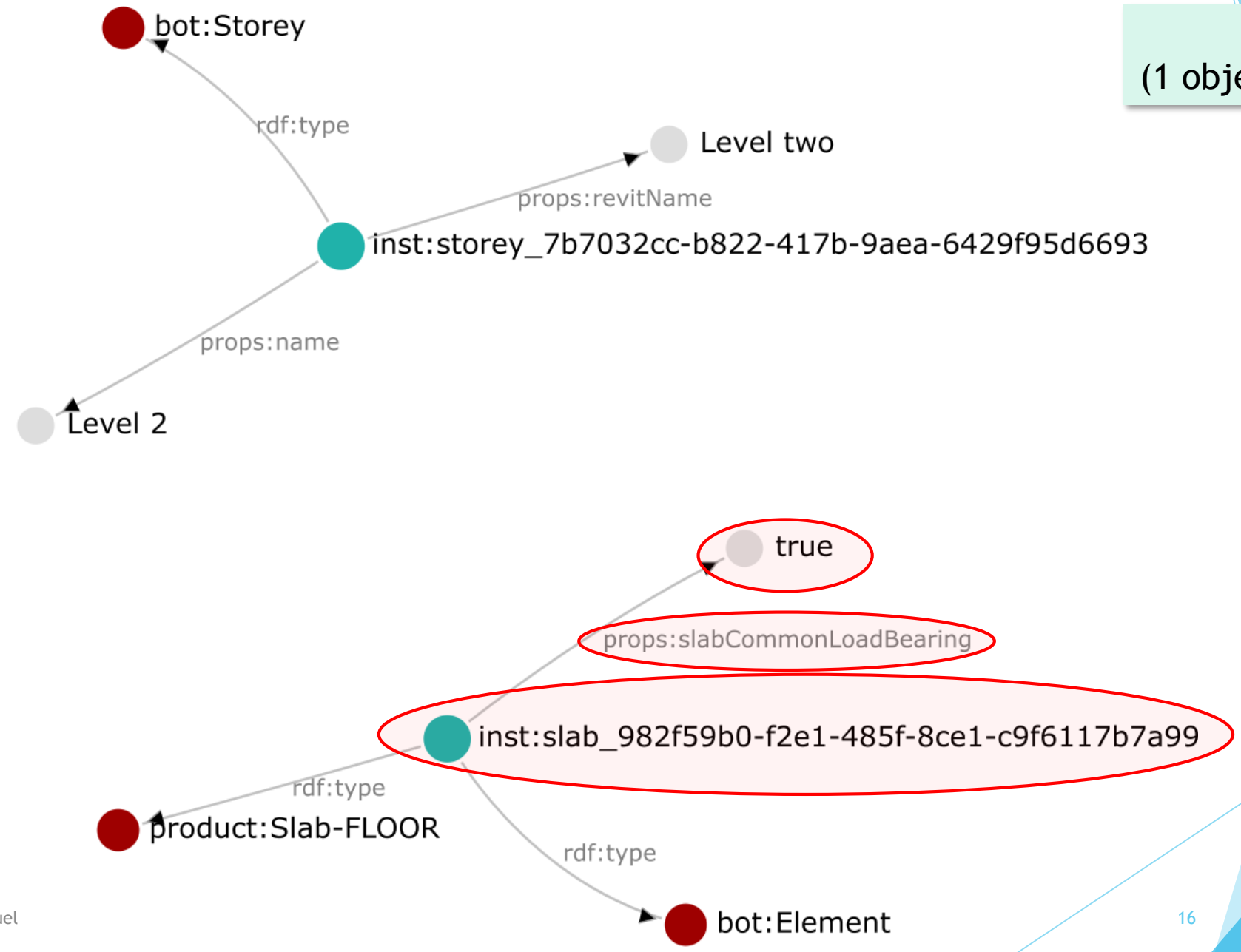
2. Considerations for ontology structure
Competency questions - functionality

- Query execution time
- Easy/intuitive to discover properties
- Reasoning
- Alignment with other existing ontologies (e.g. schema.org)
- PROPS ontology should be easy to maintain and extend
- Extra information about a property (description, label, validity, etc.)
- Grouping props (e.g. IFC psets)
- Versioning of props
- Units for props
- literals: data typing (integer, float, boolean, strings, etc.) + language tags (strings)
- Complex props: depending on other props via a math function + table of values
- ...?

2. Considerations for ontology structure
New proposals (WIP)

- IFCtoLBD converter: converts IFC files to LBD ontologies (BOT, PRODUCT and PROPS) (Jyrki)
  - Different conversion options investigated
    - 1 step/relation => 1 object/datatype property
    - 2 steps/relations => 1 object property + 1 object/datatype property
    - 3 steps/relations => 2 object properties + 1 object/datatype property
  - Optionally: grouping of properties (e.g. IFC psets) + adding units
  - Optionally: versioning of properties of IFC elements
- Building-related requirements (Mads)
  - Versioning of properties necessary + metadata (when, by who changed)
  - Differentiation between properties as required / designed / built / assumption / ...
  - Requirements: exact value ⇔ min value ⇔ max value
  - Units
- Alignment to schema.org (swallowed GoodRelations) (Mads – Georg)

1 step/relation
(1 object/datatype property)



IFCtoLBD converter

Remarks

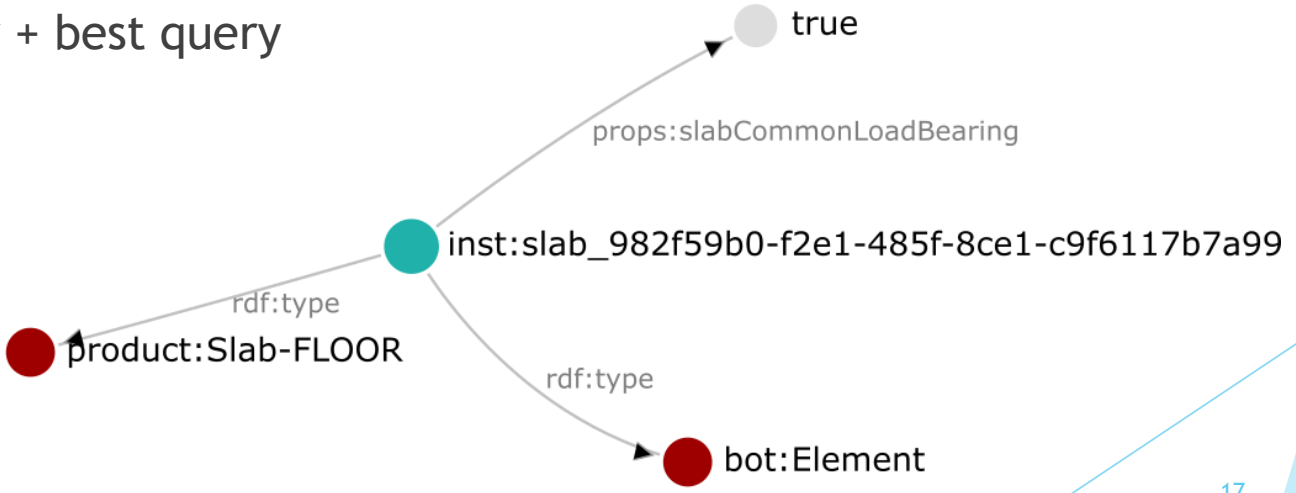
1 step/relation
(1 object/datatype property)

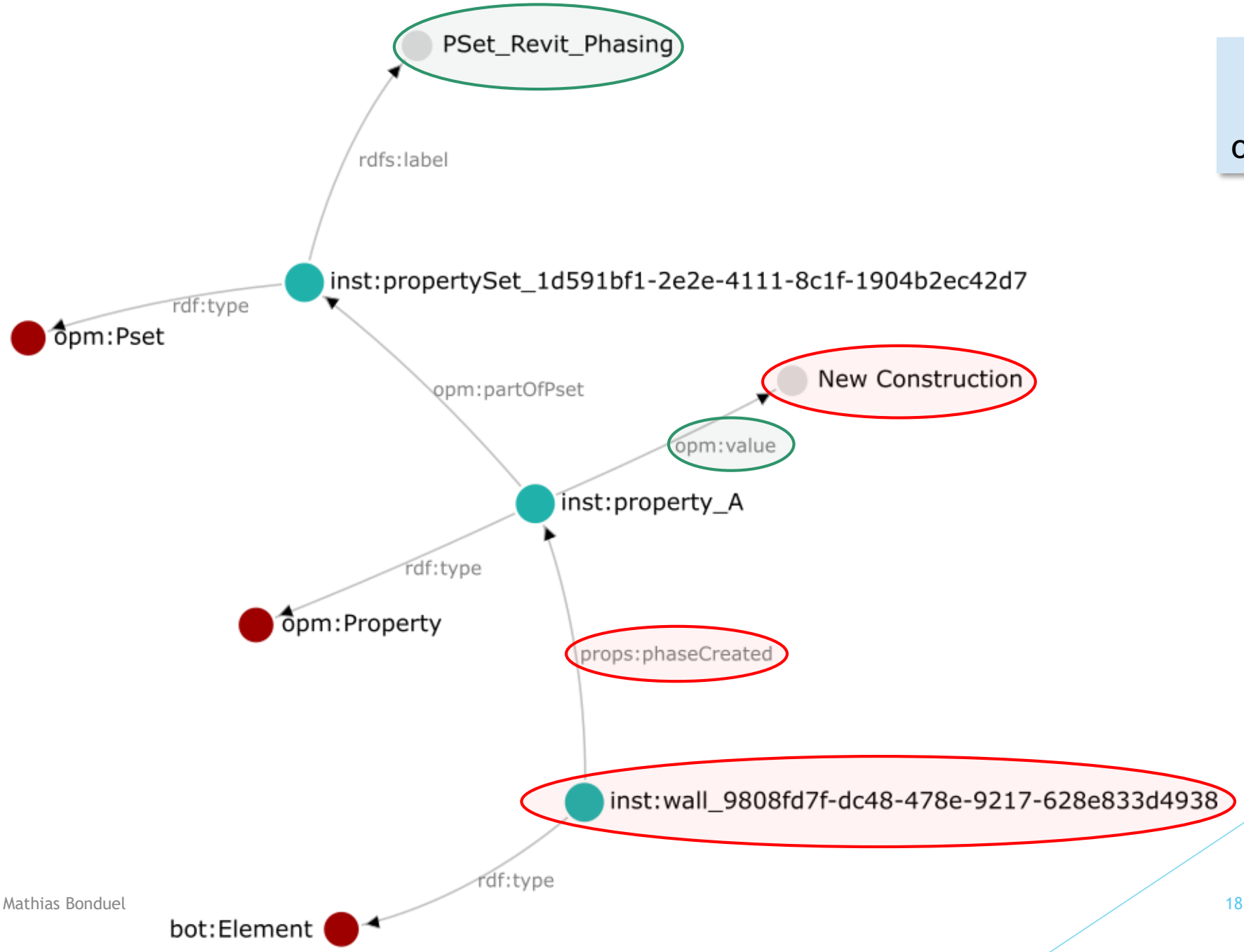
- Not easy to query for psets + hinders readability
- Not easy to maintain ontology if IFC psets are in property name
- No formal units
  - Workaround: mention units in ontology in

  a rdfs:description string
- No extra information about props

- But: easy to discover + best query

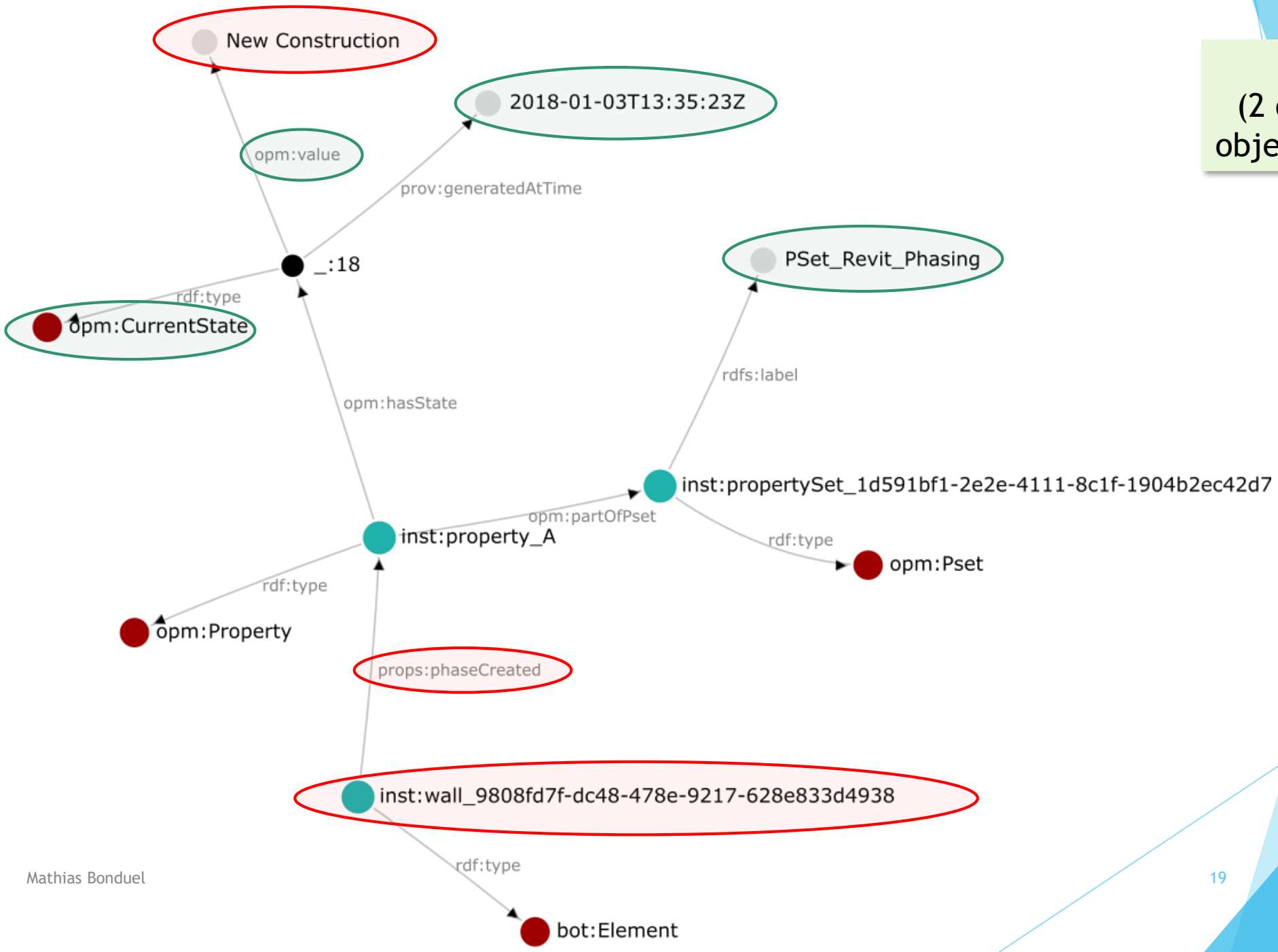
execution times





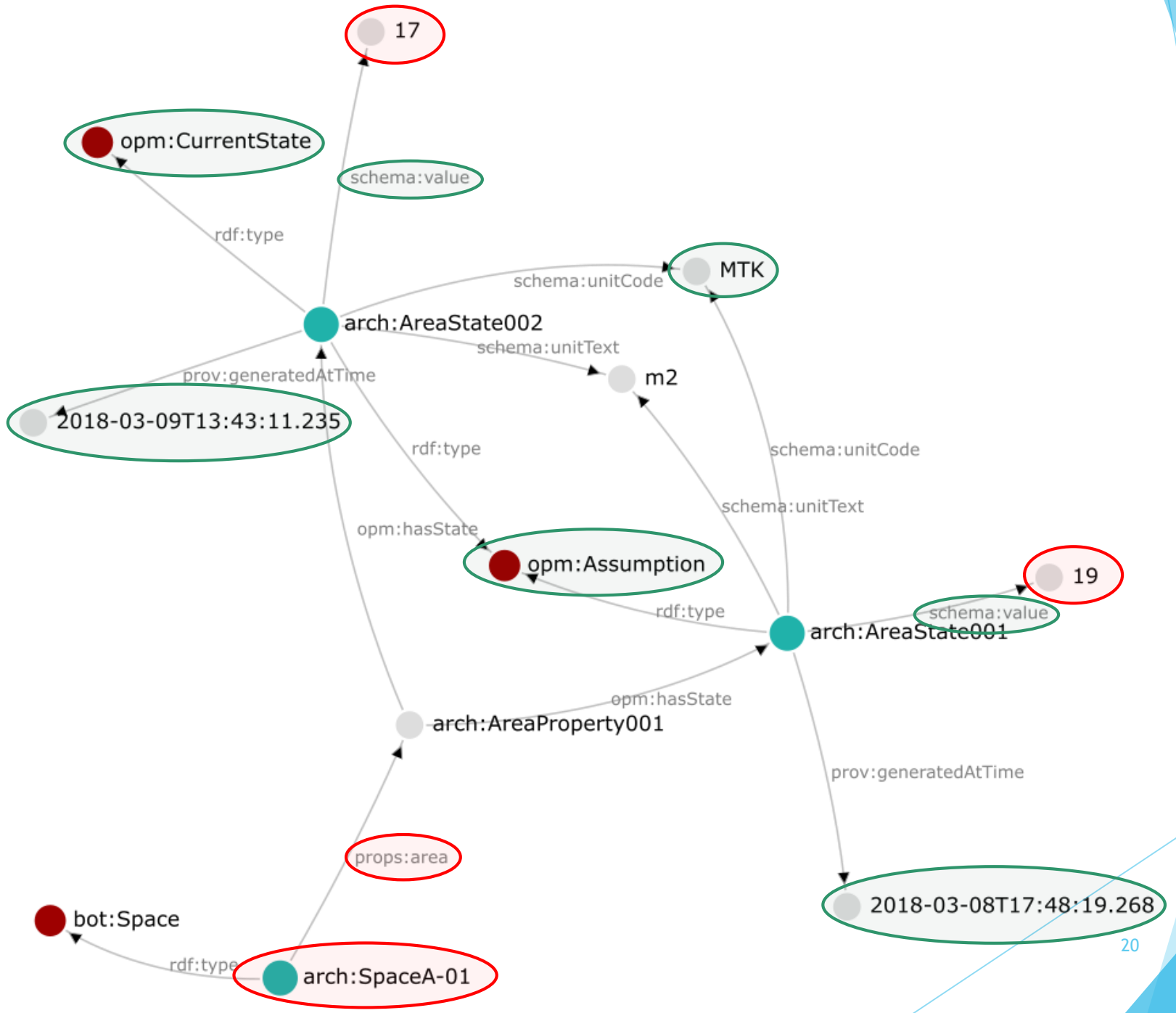
2 steps/relations
(1 object property + 1 object/datatype property)

IFCtoLBD converter

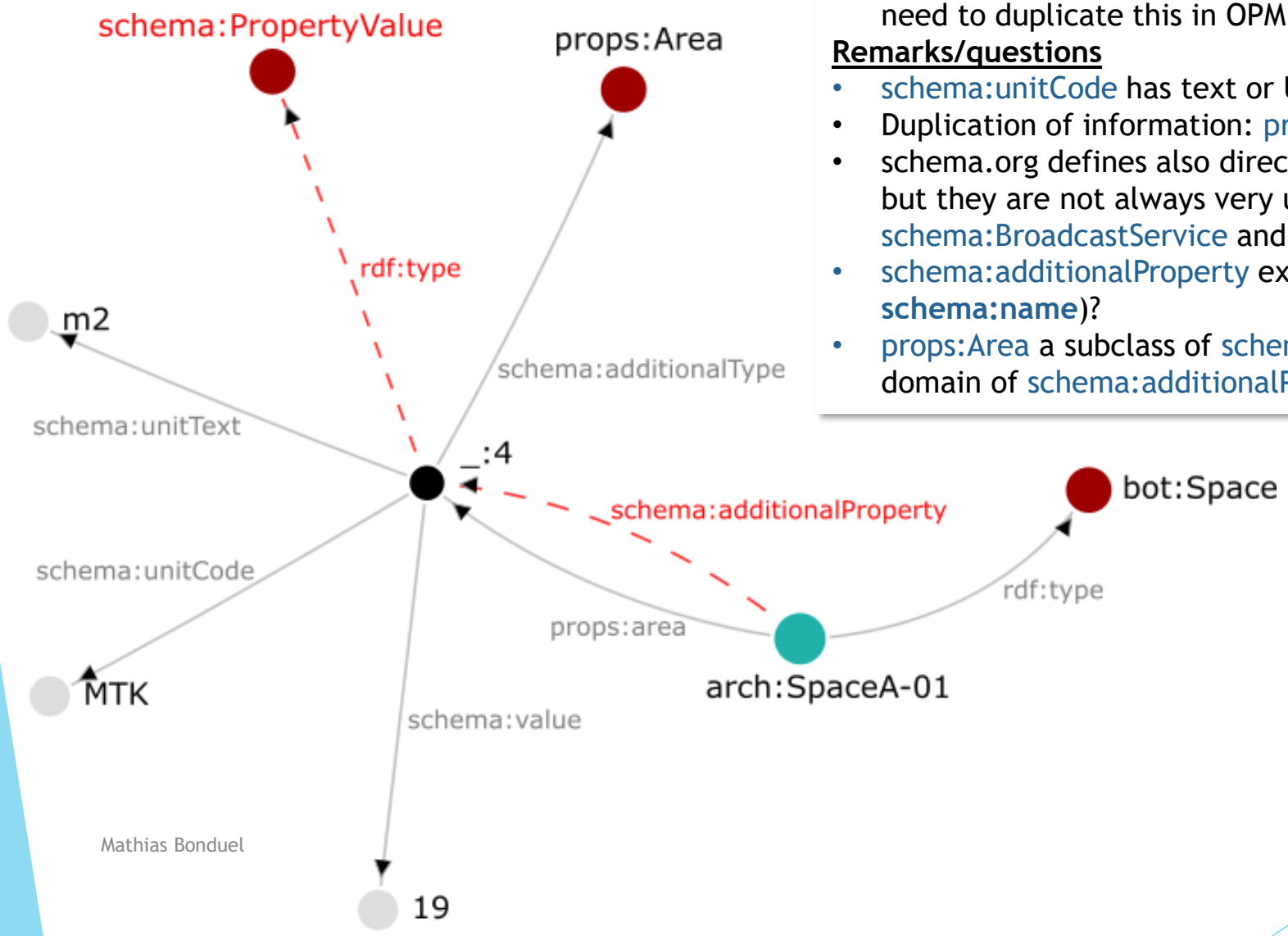


3 steps/relations
(2 object properties + 1 object/datatype property)

IFCtoLBD converter



Building-related requirements



Measures to make the PROPS ontology compatible with schema.org

- props:area is subproperty of schema:additionalProperty (which has as range schema:PropertyValue)
- schema:additionalType (subproperty of rdf:type)
- schema.org defines schema:value, schema:minValue and schema:maxValue => no need to duplicate this in OPM

Remarks/questions

- schema:unitCode has text or URL as domain => not compatible with QUDT / OM?
- Duplication of information: props:area (object property) and props:Area (class)
- schema.org defines also direct object or datatype properties (e.g. schema:area) but they are not always very useful for buildings (e.g. schema:area => domain is schema:BroadcastService and range is schema:Place)
- schema:additionalProperty expects a key-value pair (schema:value and **schema:name**)?
- props:Area a subclass of schema:PropertyValue? (can also be inferred as the domain of schema:additionalProperty)

Alignment to schema.org

2. Considerations for ontology structure

A range of solutions with different levels of complexity?

- Not one ideal solution => depends on the case
  - Different levels of complexity?
  - All in one ontology ⇔ different PROPS ontology for each level?
  - Different levels of complexity in one db ⇔ one level of complexity per db?
    - E.g. ID property only needs one datatype property (no versioning, units, etc.)

One property cannot be an owl:DatatypeProperty / owl:ObjectProperty / owl:Class at the same time (?)

2. Considerations for ontology structure

A range of solutions with different levels of complexity?

- Level 1: (1 step => 1 object/datatype property)
  - searching in large database of e.g. products (public part of e-commerce)
  - All props that don't need versioning, don't have units, are not part of psets (e.g. ID, ...?)
- Level 2: (2 steps => 1 object property + 1 object/datatype property)
  - manufacturer / reseller (private part of e-commerce)
  - During construction projects (shared part of project member)
- Level 3: (3 steps => 2 object properties + 1 object/datatype property)
  - during construction projects (private part of project member)
  - Facility Management, LCA?

2. Considerations for ontology structure

A range of solutions with different levels of complexity?

	Level 1	Level 2	Level 3
Query execution time	+++	++	+
Easy/intuitive to discover props	+++	++	+
Reasoning	?	?	?
Alignment to schema.org	?	?	?
Ontology is easy to maintain and extend	?	?	?
Extra information about property	/	✓	✓
Grouping of props	/	✓	✓
Versioning of props	/	/	✓
Units for props	In rdfs:description (string)	QUDT, OM, other	QUDT, OM, other
Literals: data typing + language tags	✓	✓	✓
Complex props	/	✓ (?)	✓ (?)

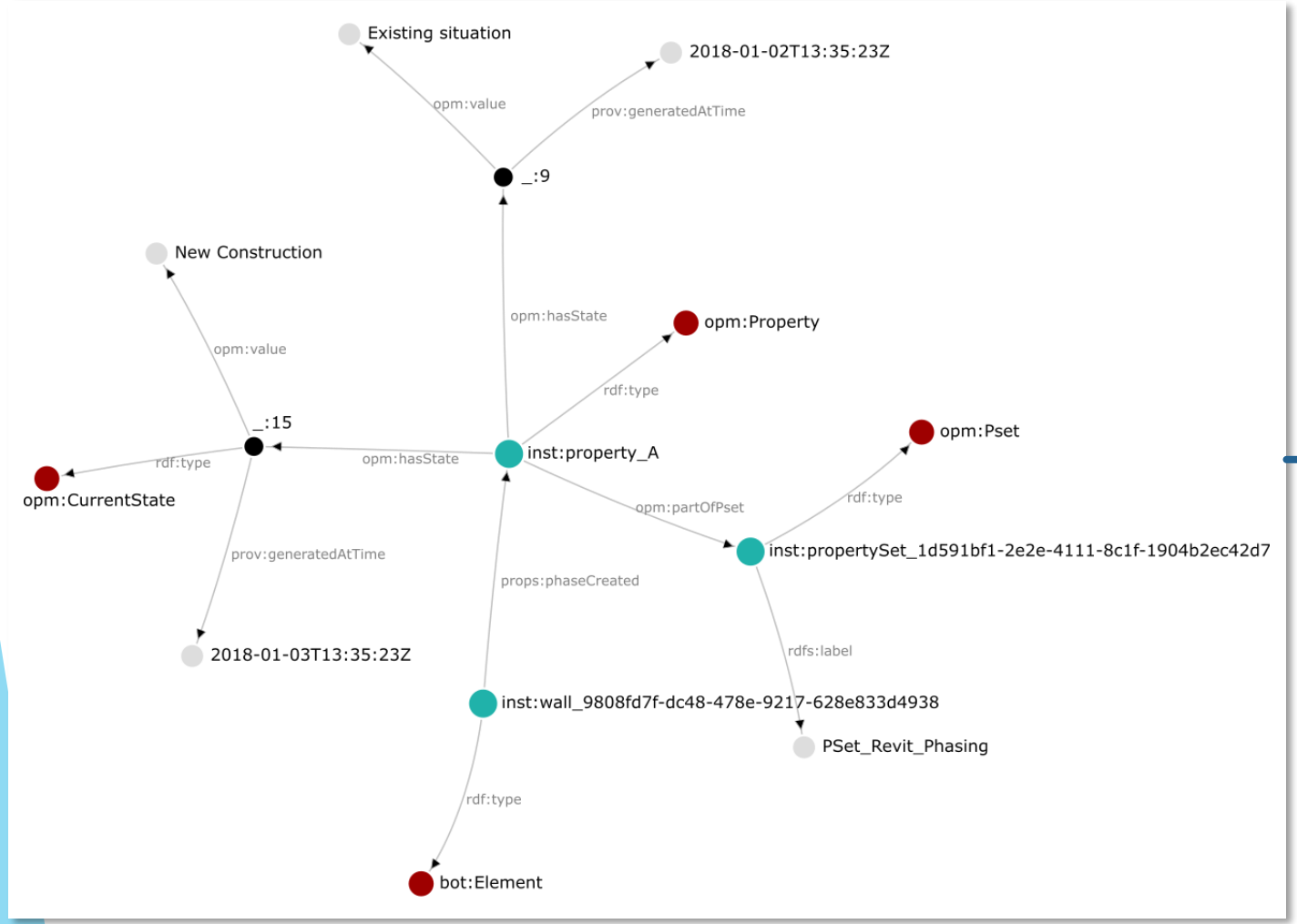
2. Considerations for ontology structure

A range of solutions with different levels of complexity?

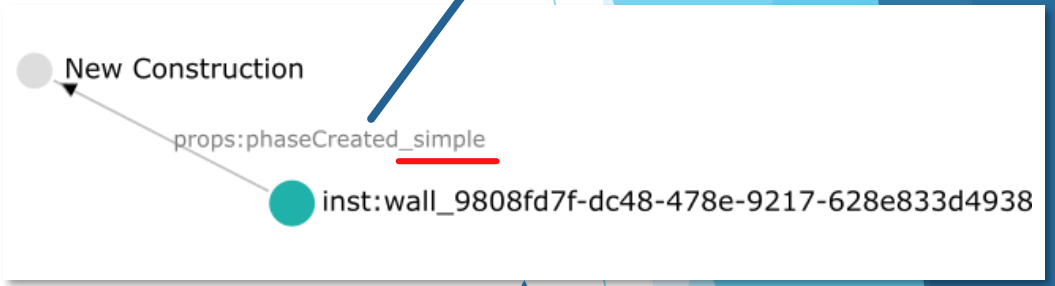
- Not one ideal solution => depends on the case
- Standardize **conversion** between different complexity levels
  - To lower complexity: selection (in the case of versioning)
  - To higher complexity: extra information needed
  - How? => conversions or reasoning
    - SPARQL queries (INSERT / CONSTRUCT) => possibility to place converted part in other DB
    - SWRL (rules) or inferencing/reasoning

2. Considerations for ontology structure

A range of solutions with different levels of complexity?



Datatype property
(⇔ props:phaseCreated
as object property)

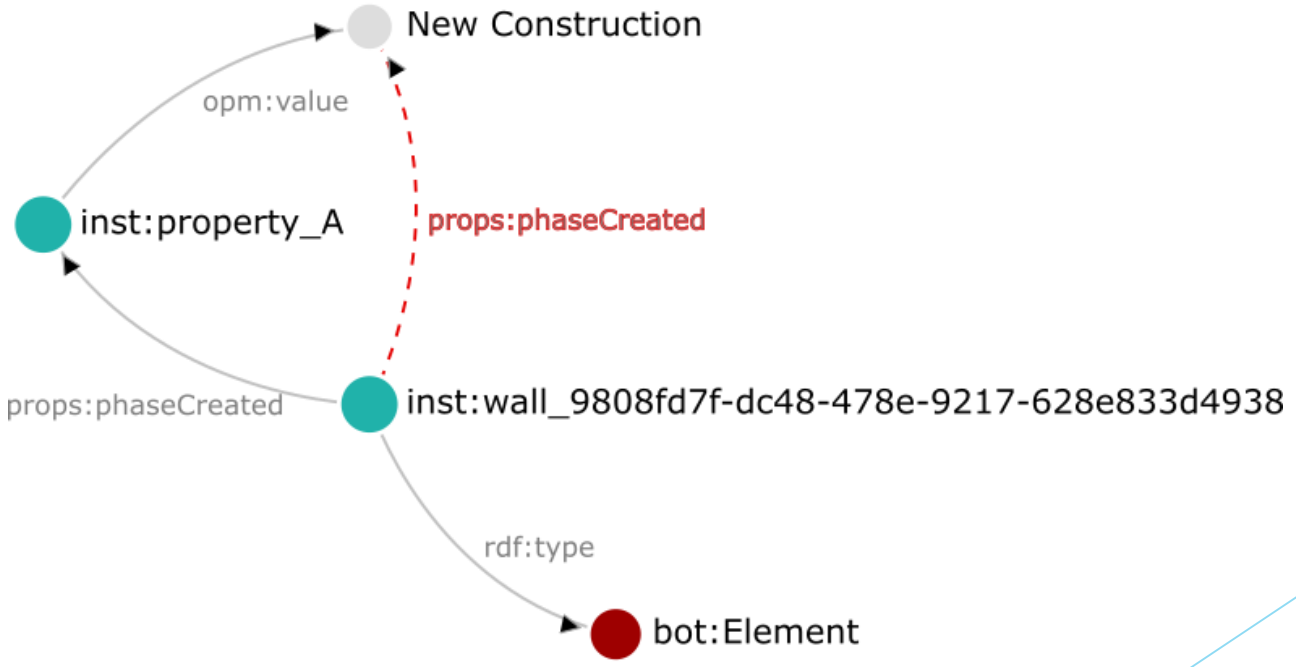


SPARQL INSERT or CONSTRUCT

2. Considerations for ontology structure

A range of solutions with different levels of complexity?

- Only possible for viewing as Level 1, when in fact Level 2 (SWRL needed for Level 3)
- Difficult to update the property (not clear for user if property is inferred or not)
- Filter needed for instances of properties



3. Next steps
Open issues in ontology structure

- Different levels of complexity (interchangeable) ⇔ most complex situation?
  - In the case of different levels of complexity:
    - How many levels?
    - Conversion methods?
    - one ontology for the three levels ⇔ one per level?
    - One level of complexity per DB ⇔ multiple levels?
- What documented where?
  - PROPS ontology/ontologies ⇔ OPM ontology (not yet documented)?
  - Materials in separate ontology?
  - What?
    - Classes: Pset (and other grouping of properties)
    - Relations: partOfPset (and other grouping of properties)

3. Next steps
Open issues in ontology structure

- Level 1:
  - Information about units in rdfs:description string of the property in the PROPS ontology
- Level 2:
  - Units
    - Choose for one units ontology? (QUDT, OM, ...)
    - Compatibility with schema.org? (schema:unitCode => range: text + URL)
  - Blank nodes for:
    - Instances of properties?
    - states?
  - Property name info in object property ⇔ class?
- Level 3:
  - Same as level 2
  - Units and opm:Assumption/opm:asDesigned/opm:asBuilt/... => on state or property instance?

Not possible to refer to it from outside the DB
=> are properties/states supposed to be shared?

3. Next steps
Open issues in ontology content

- Ontology should restrict the kind of units that can be used?
- Methods to extend the ontology with custom properties (will never be complete)
- IFC properties (pset generator) as subproperties of more general properties?
  - Wikipedia?
  - Standards containing definitions?
  - Wikidata?
  - QUDT?
  - OM?
  - schema.org?
  - Custom made properties?

Results compatible with PROD derived from IFC (domain)

Probably combination of these sources (+ alignments between them)?

CEN / TC 442 or others?

4. sparql-visualizer tool

- What? => means to communicate during ontology engineering process between:
  - Domain expert
  - Application engineer
  - Ontology engineer
- No installment necessary:
  - Online
  - Offline: download ready-to-use ZIP version
- Easy to use + documentation online:
  - Github readme (quickstart + detailed functionality)
  - Tutorial videos: pt1. basics + pt2. running locally
- Flexible:
  - Prepare samples in JSON and share one-click links via Dropbox or Github
  - connection with separate triple store possible (atm only Stardog)
  - Loading of turtle files
- Open source: everyone can contribute

✓ Intuitive graph visualization
✓ Application of ontologies: Tbox + Abox

https://madsholten.github.io/sparql-visualizer/?file=https://dl.dropbox.com/s/x7z1aw4hzgtv0c9/ifcOWL-properties.json

Sources – links - further reading

- PROPS discussion: https://github.com/w3c-lbd-cg/props/issues/2#issuecomment-371807503
- PSET Generator (Maxime): https://github.com/w3c-lbd-cg/props
- IFCtoLBD converter (Jyrki): https://github.com/jyrkioraskari/IFCtoLBD
- Demo to get PROPS ontology from Wikipedia: https://objprops-gen.herokuapp.com/id/area
- Ontologies:
  - ifcOWL: http://ifcowl.openbimstandards.org/
  - BOT: https://github.com/w3c-lbd-cg/bot
  - schema.org: http://schema.org/version/3.3/schema.ttl
  - QUDT (2.0)
    - General: http://www.qudt.org/release2/qudt-catalog.html
    - Disciplines: http://qudt.org/doc/2017/DOC_VOCAB-QUDT-DISCIPLINES-v2.0.html
  - OM (ontology of units of measure): https://github.com/HajoRijgersberg/OM

Sources – links - further reading

Literature

- Farias, T. M. De, Roxin, A.-M., & Nicolle, C. (2015). IfcWoD,Semantically Adapting IFC Model Relations into OWL Properties. In *Proc. of the 32nd CIB W78 Conference 2015, 27th-29th October 2015, Eindhoven, The Netherlands* (pp. 175–185).
- Niknam, M., & Karshenas, S. (2017). A shared ontology approach to semantic representation of BIM data. *Automation in Construction*, *80*, 22–36. https://doi.org/10.1016/j.autcon.2017.03.013
- Pauwels, P., & Roxin, A. (2016). SimpleBIM : From full ifcOWL graphs to simplified building graphs. In S. Christodoulou & R. Scherer (Eds.), *EWORK AND EBUSINESS IN ARCHITECTURE, ENGINEERING AND CONSTRUCTION* (pp. 11–18). Limassol, Cyprus.
- Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, *63*, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003
- Rasmussen, M. H., Pauwels, P., Hvidd, C. A., & Karlshøj, J. (2017). Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. In *LC3 2017: Proceedings of the Joint Conference on Computing in Construction* (pp. 237–244). Heraklion, Greece.
- Rasmussen, M. H., Pauwels, P., Lefrançois, M., Schneider, G. F., Hviid, C. A., & Karshøj, J. (2017). Recent changes in the Building Topology Ontology. In *5th LDAC workshop*.