Semantics and ontologies for operational buildings: caveats, pitfalls, and opportunities.

MONDAY 15 NOVEMBER 2021 – WEBINAR – IBPSA PROJECT 1 PRESENTATION

Pieter Pauwels, Associate Professor

Department of the Built Environment, Information Systems in the Built Environment

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Who am I?

Associate Professor TU Eindhoven (2019-…)

Assistant Professor Ghent University (2016-2019)

Postdoc Ghent University (2014-2016)

Postdoc University of Amsterdam (2012-2014)

Master & PhD in Civil Engineering – Architecture @Ghent University (2008, 2012)

B4B: Brains for Building's Energy Systems

BRAINS 4 BUILDINGS

- Buildings are to serve people's needs:
  - Occupants and FM: Health, comfort, ease of use & ease of operation, affordability
  - Humankind: energy efficiency, renewables
- Building operation is key (Energy & Indoor climate systems)
  - Lots of occupants & FM dissatisfaction
  - Lots of energy wastage
- Operation data & data analytics, ML, AI are key to:
  - Understand
  - Steer & Control optimally
  - Adapt to renewable energy
  - Make better designs

April 1, 2021

Consortium Meeting

TUDelft

• Image adapted by TU Delft from: [illegible] from Pixabay; [illegible] from [illegible] Pixabay

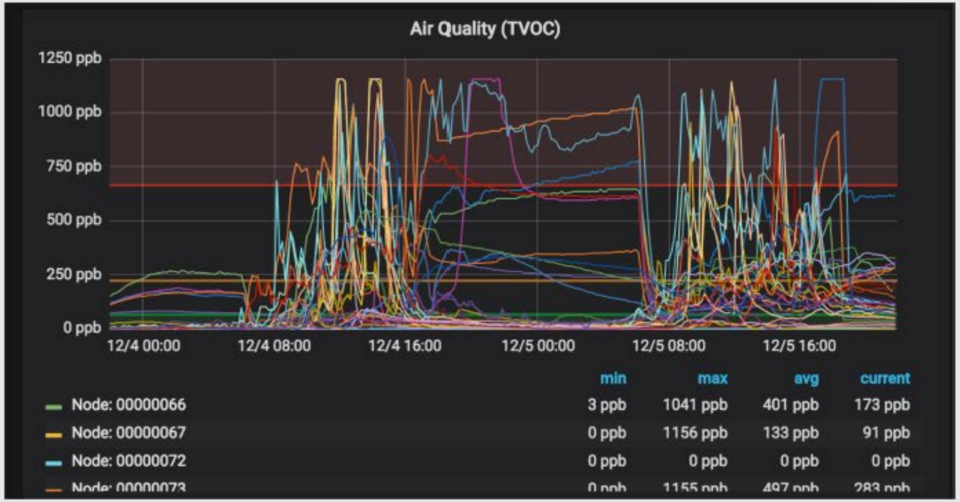
Presentation Outline

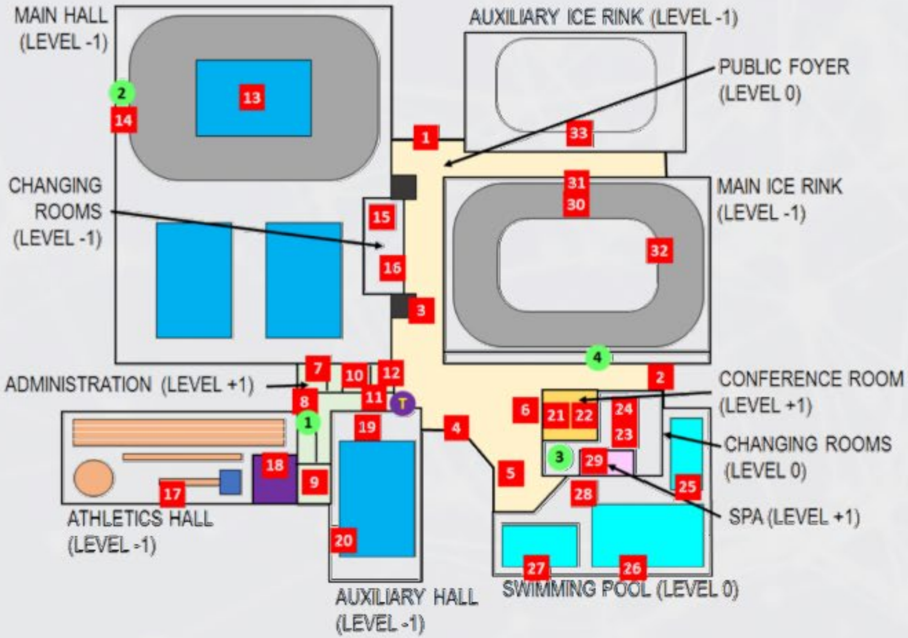
1. Stepping through Building Data Semantics: BIM, IFC, LBD, BRICK, Haystack, etc.
2. The ‘Scale of AI Methods’: ML and semantics
3. Towards scalable system integration that combines multiple AI fields

SENSOR DATA EXAMPLE: GIGANTIUM IOT LIVING LAB AALBORG

- 35 sensor nodes monitoring Temperature (°C), Relative Humidity (%), Air Pressure (hPa), Indoor Air Quality (Total Volatile Organic Compounds ((TVOC), ppb) and CO_2 (ppm)), illuminance (lux) and motion

- Data storage in SQL database
- Data monitoring and visualization in Grafana

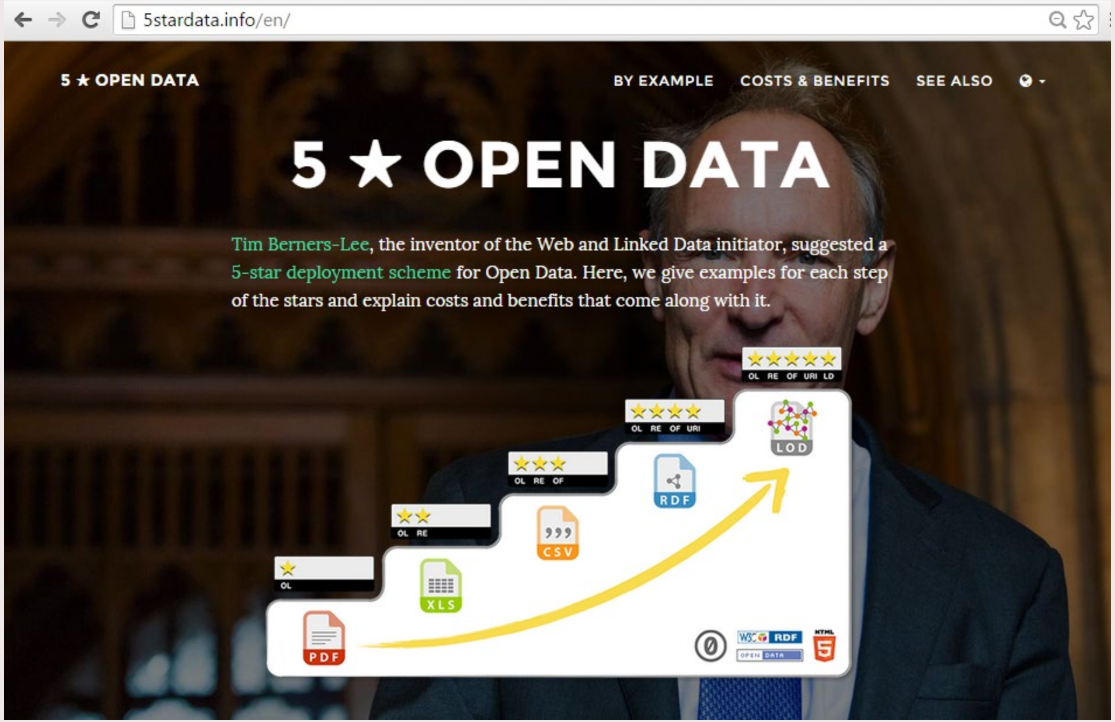




Source: Rodriguez et al. (2018)

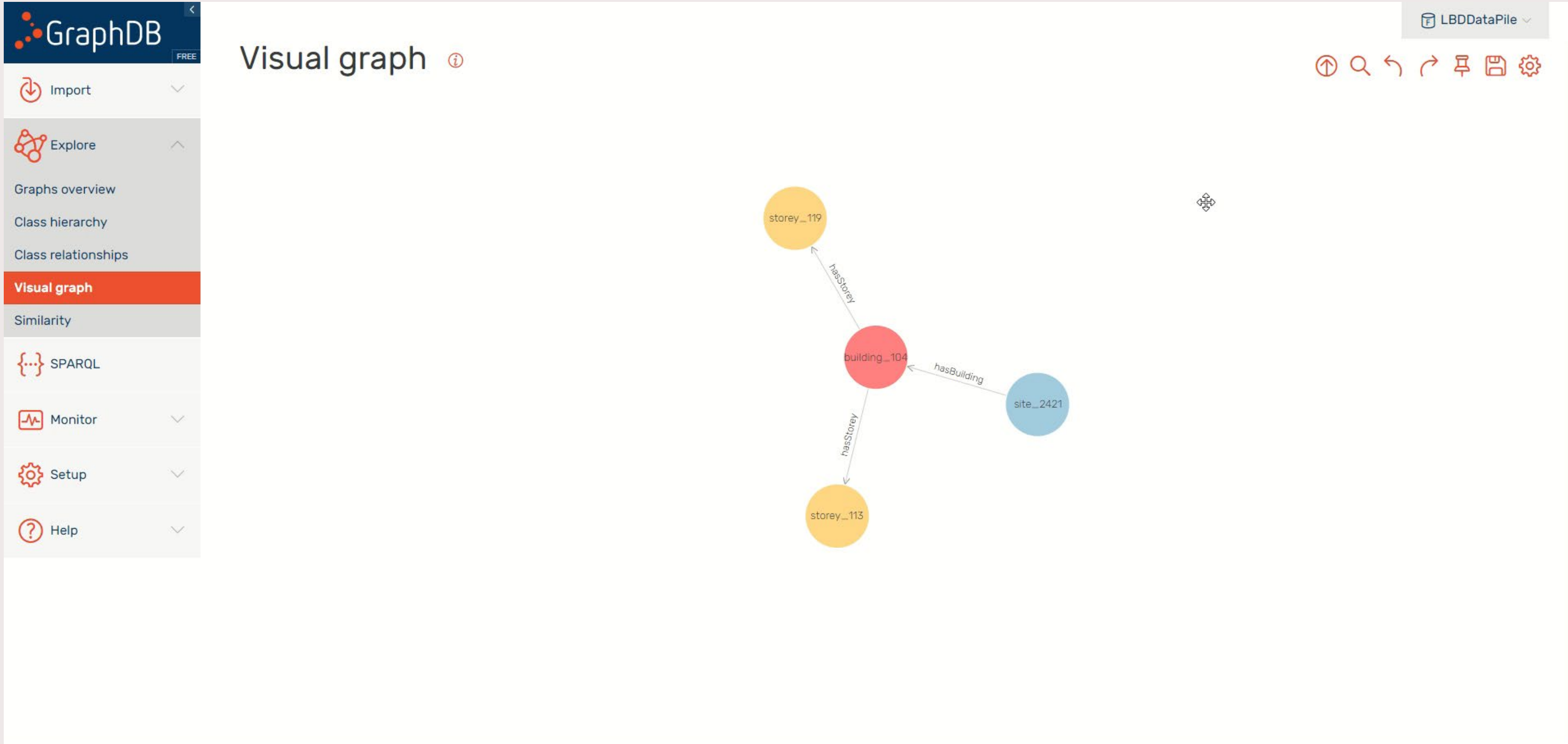
Petrova, E., Pauwels, P., Svidt, K., Jensen, R.L. (2018) From patterns to evidence: Enhancing sustainable building design with pattern recognition and information retrieval approaches. Proceedings of the 12th ECPPM conference, pp. 391-399.

Cloud of linked data

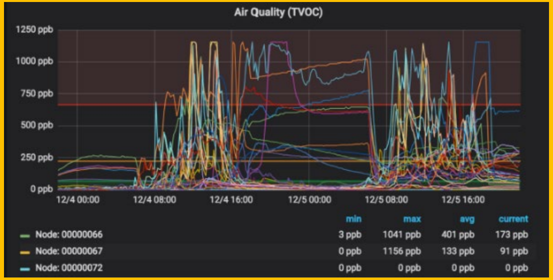


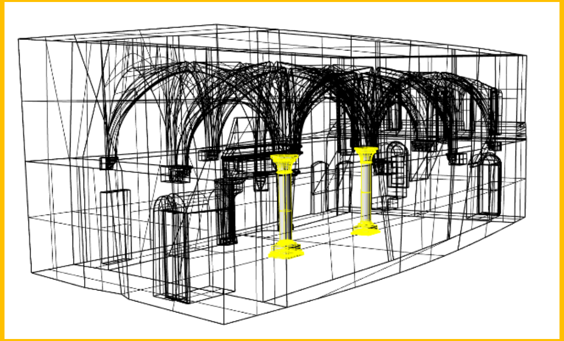
https://5stardata.info/en/

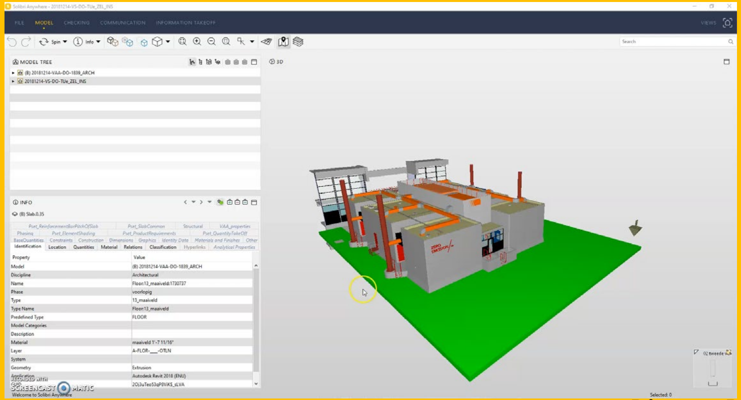
Cloud of linked building data

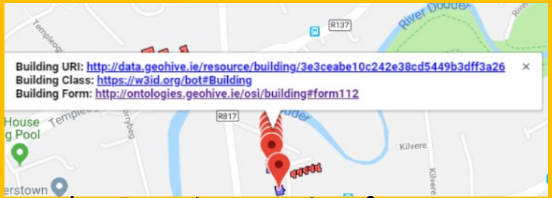


A building has different types of data associated







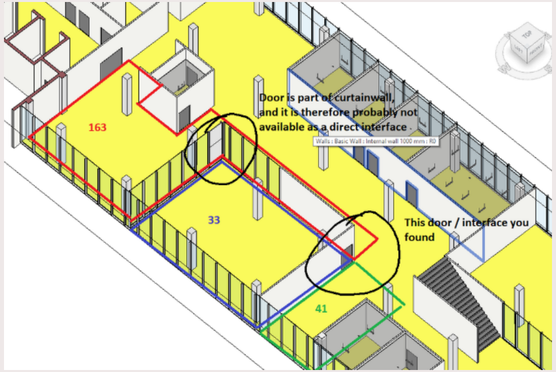


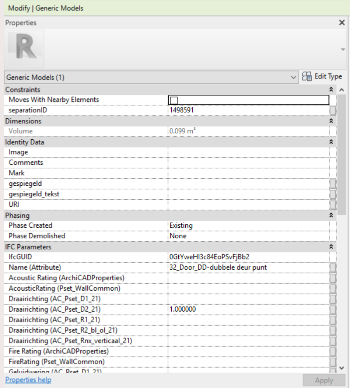
Categories of data?

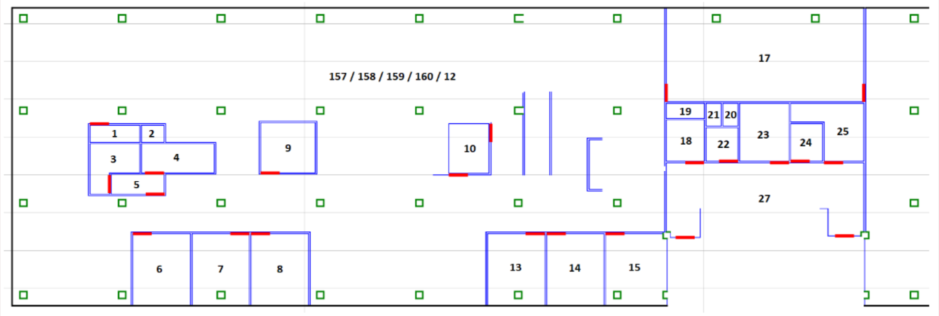
- **Data streams:**
  - Ordered lists of values, typically floating point values
  - Large amount of data -> data lakes
  - Almost no semantics, at best a few labels for categorization
  - E.g. temperature measurements, system logs (e.g. triggering of actuators), etc.
- **Semantically rich and interconnected data:**
  - Seldom including large data streams
  - Long debates about the semantics of things -> standardisation
  - Complex and brittle (breaks easily)
  - Small amount of very important data
  - Easy to combine with rule-based and/or logic-based technologies (inference and query)
  - E.g. BIM models, semantic web ontologies, taxonomies, OTLs, etc.
- **Control models:**
  - Algorithms for control, parametric functions
  - Communication system, signal processing, direct control, low latency
  - Typically located on the edge (devices with embedded functions)
  - E.g. Control Description Language (CDL), modelica models
- **User data:**
  - Outside of the system
  - Different privacy and security requirements
- **Files:**
  - No semantically rich encodings, no data streams
  - Highly valuable, and seldom machine-processable
  - E.g. PDFs, Images, Geometry ‘blobs’

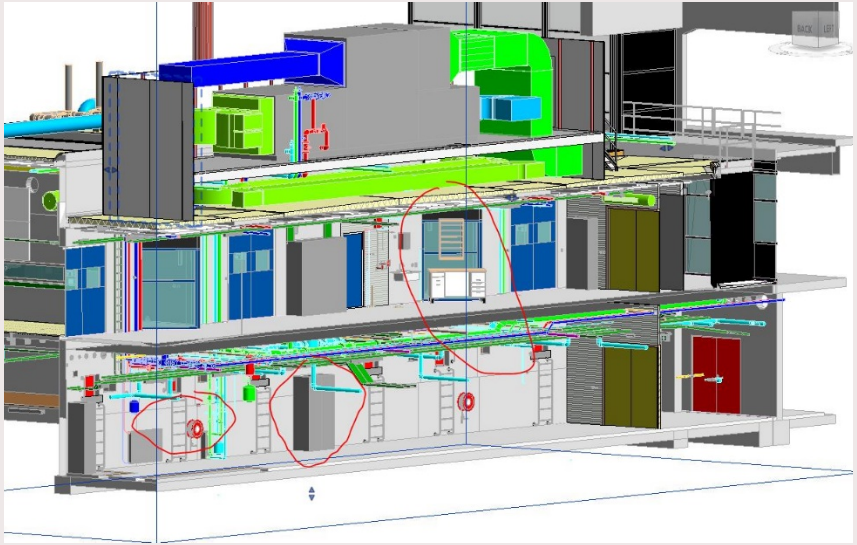
BIM Data

BIM data: Revit, modelling guidelines, agreements, 3D modelling, and IFC









Luckily we have IFC…

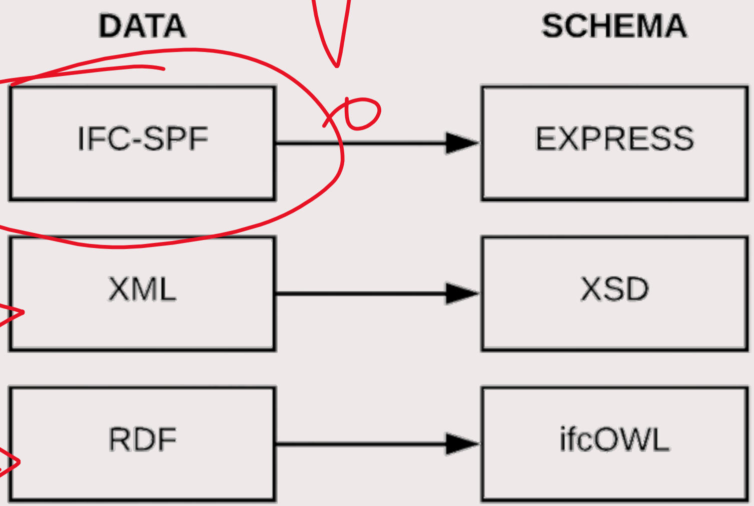
STEP (ISO 10303) can represent 3D objects in Computer-Aided design (CAD) and related information.

STEP has an **ASCII** structure- a character encoding standard for electronic communication.

IFC- SPF (STEP Physical File Format) Text format defined by ISO 10303-21 ("STEP-File"), where each line consists of a single object record.

Extensible Markup Language (XML) is a markup language designed to store and transport data.

Resource Description Framework (RDF) is a standard model for data interchange on the Web.



EXPRESS is a data definition standard developed to enable a formal definition of industrial data.

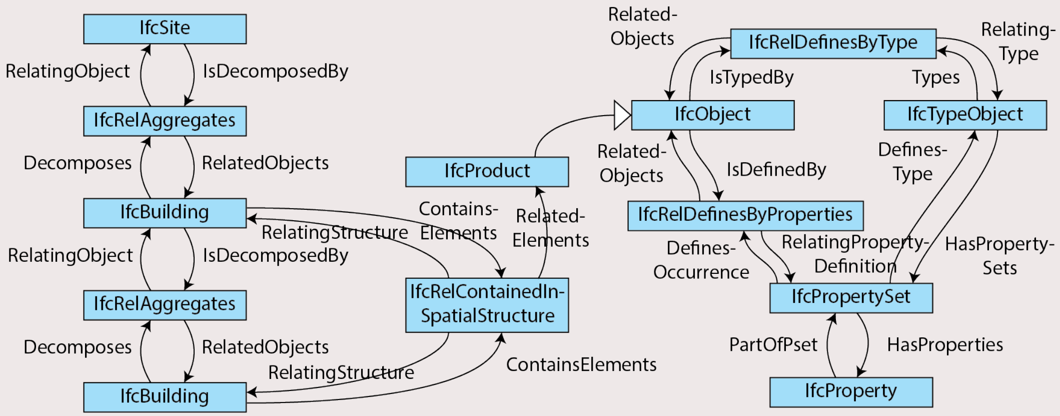
XSD (XML Schema Definition) specifies how to formally describe the elements in a XML document.

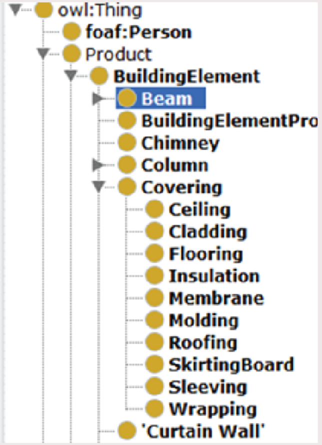
ifcOWL provides a Web Ontology Language (OWL) representation of the IFC schema.

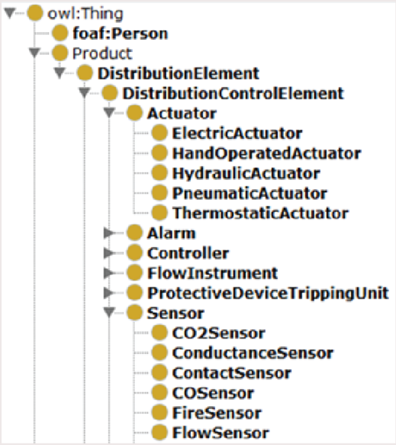
Conversions aiming at roundtripping and backwards compatibility

Data in the Industry Foundation Classes (IFC)

- Overall building shape and topology easy
- Classification of elements possible, but not many classes => extension with bSDD classes and properties possible
- Difficult (not impossible) to include sensor data (timeseries data)
- Availability in STEP, XML, RDF, and JSON







W3C Linked Building Data

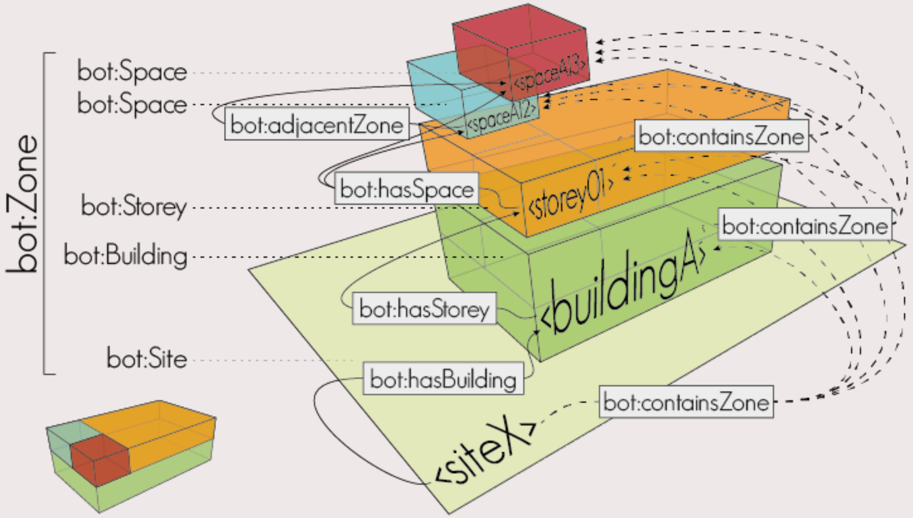
Emergence of W3C LBD Community Group: Mission Statement

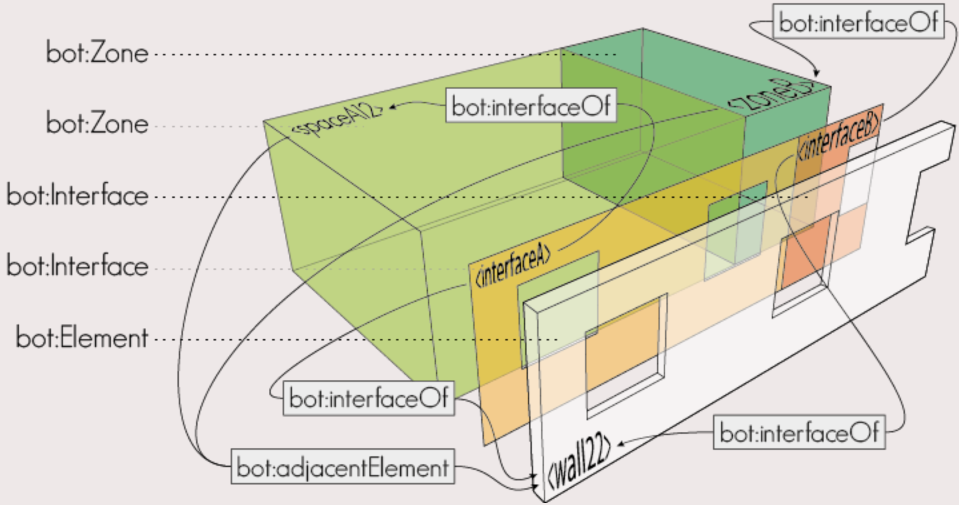
Bring together experts in the area of Building Information Modeling (BIM) and Web of Data technologies to:

1. define existing and future use cases and requirements for Linked Data based applications across the life cycle of buildings.
2. discuss best practices for publishing building data on the Web propose ontology models to describe:
   1. Buildings and building elements (topology, associate values to properties)
   2. Products and product properties
3. discuss how they can be used together with other specifications:
   1. existing standards (IFC, GeoSPARQL, Semantic Sensor Network, ...)
   2. separate initiatives (schema.org, Haystack, BRICK, ...)

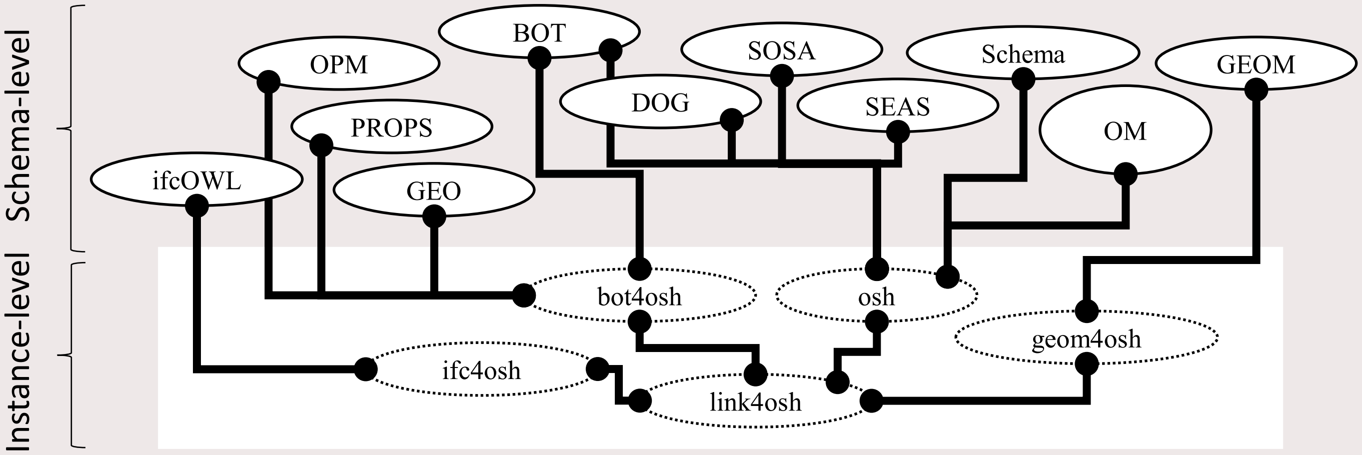
Scope of BOT: just the start, allowing to extend

- Limited set of classes
- Extensible and easy to combine with other ontologies and data sets
- Comprehensible





Modular ontology modelling advocated by LBD group

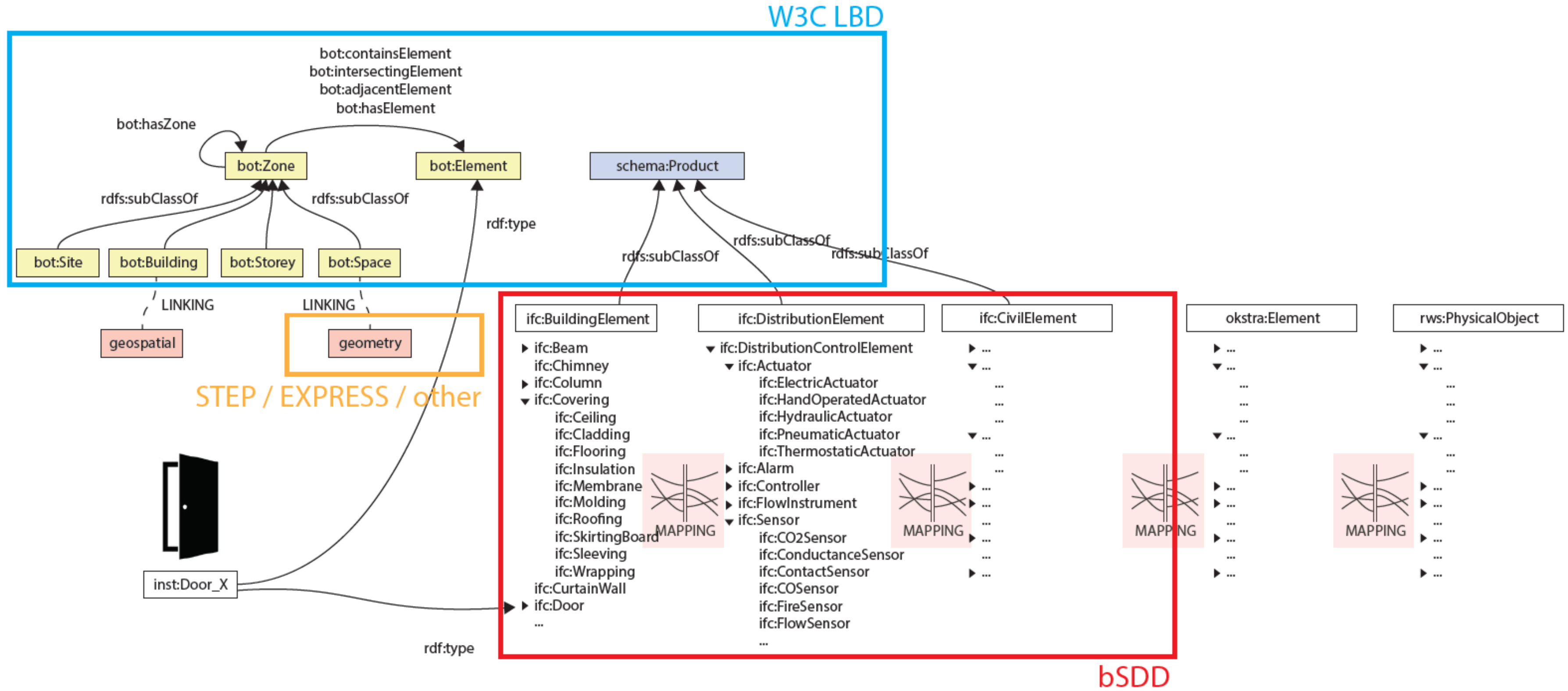


- Implemented using Semantic Web Technologies -> Web-scale, queryable
- Reuse of existing ontologies -> Modular
- Linking at instance level -> Multi-model method

Sample dataset available at:
https://github.com/TechnicalBuildingSystems/OpenSmartHomeData

TU/e

Modular approach to building data



Reference ontologies

BOT	https://w3id.org/bot#
BEO	https://pi.pauwel.be/voc/buildingelement/
MEP	https://pi.pauwel.be/voc/distributionelement/
OMG	https://w3id.org/omg#
FOG	https://w3id.org/fog#
BPO	https://www.w3id.org/bpo#
OPM	https://www.w3id.org/opm#

Revit to LBD exporter: on demand

IFC to LBD converter: on demand

BRICK, HTO, SAREF, and REC

BRICK, HTO, SAREF, REC, etc.

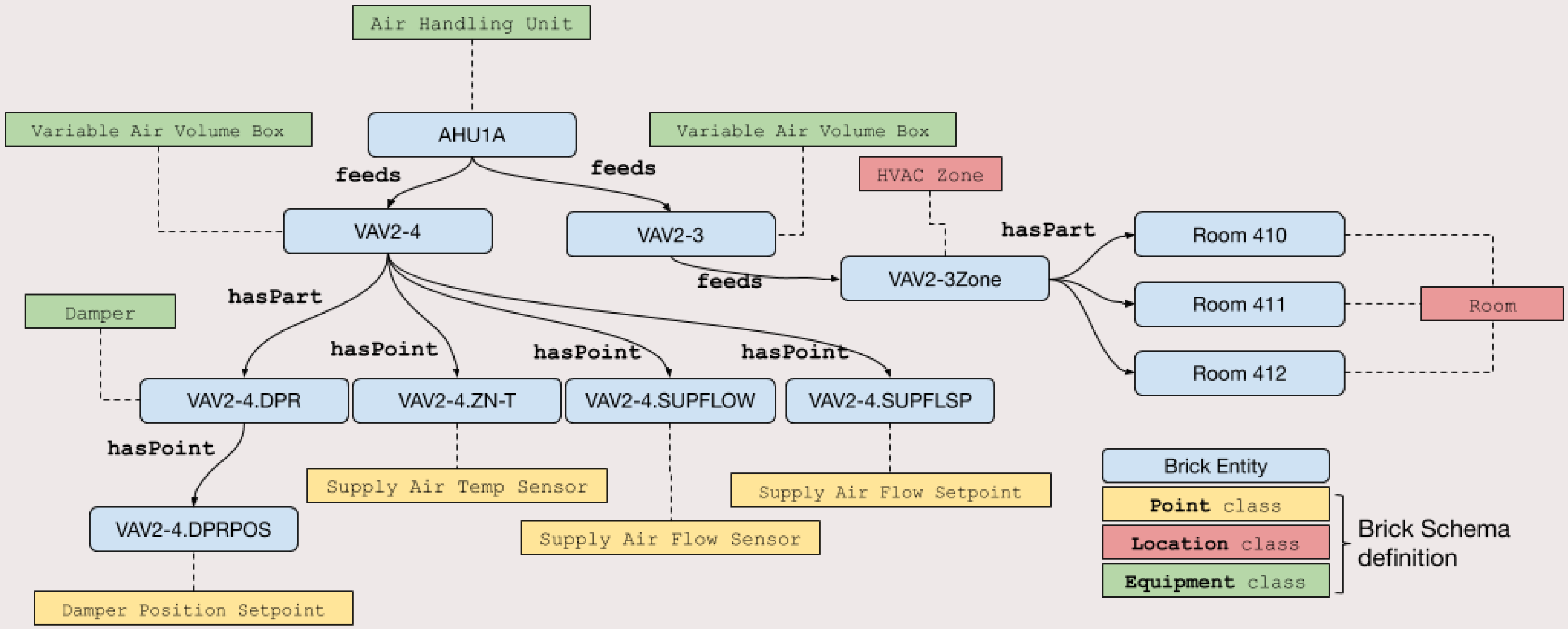
- Developments are rather disconnected from any BIM- or building-related area
- Focus on systems, incl. operation and control
- Focus on the sensor Point and Equipment types

- Beware of biased overview tables

Modeling Support	Brick	Project Haystack	IFC	BOT	SAREF
HVAC Systems	yes	yes	yes	no	no
Lighting Systems	yes	partial	yes	no	no
Electrical Systems	yes	yes	yes	no	no
Spatial Information	yes	no	yes	yes	no
Sensor Systems	yes	yes	generic	no	yes
Control Relationships	yes	no	generic	no	no
Operational Relationships	yes	no	generic	no	no
Formal Definitions	yes	no	yes	yes	yes

https://brickschema.org/

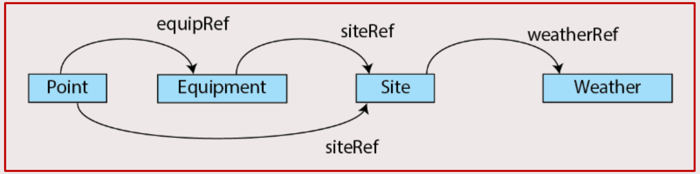
BRICK - A uniform metadata schema for buildings



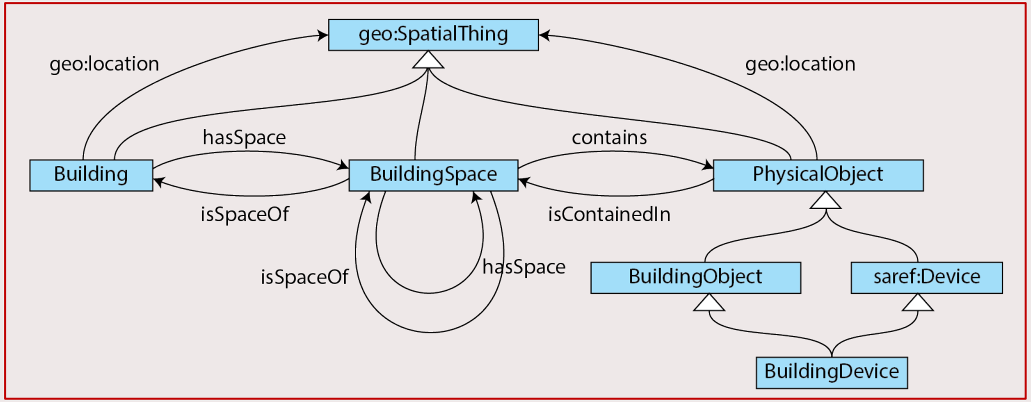
https://brickschema.org/

BRICK, HTO, SAREF, REC, etc.

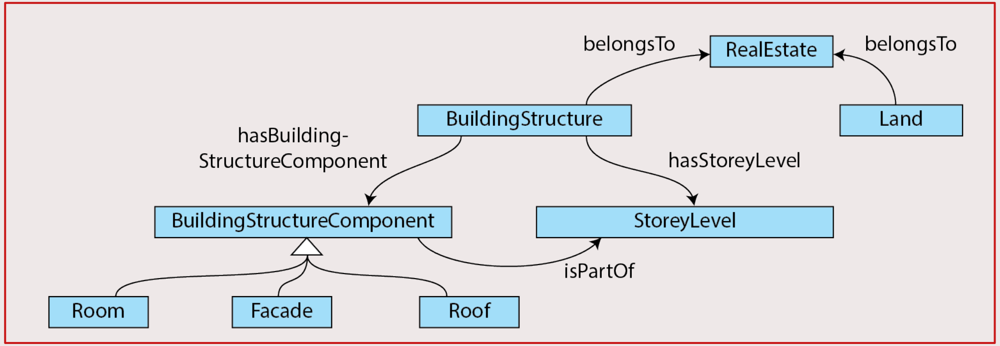
Haystack



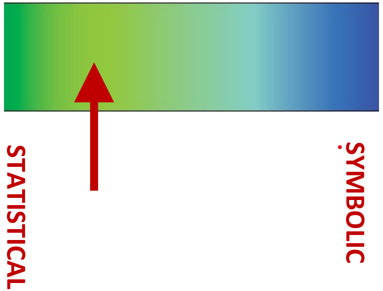
SAREF4BLDG



Real Estate Core



Haystack Tags



“We standardize semantic data models and web services with the goal of making it easier to unlock value from the vast quantity of data being generated by the smart devices that permeate our homes, buildings, factories, and cities.” (https://project-haystack.org/)

Haystack = stack of technologies:

- ***Data Types**: a fixed set of data types for modeling information*
- ***File Types**: a set of text formats to encode and exchange those data types*
- ***HTTP API**: a protocol to exchange data over HTTP using those file types*
- ***Ontology**: a standard way to model common concepts such as buildings, equipment, and sensors*
- ***Defs**: a standard way to define and extend the ontology*

Individual aspects of the technology stack can be used on their own. For example you can use the Haystack data types as an "enhanced JSON". Or you could use just the terms in the ontology without the Haystack data types.

RDF encoding / mapping example

URI

```
def: ^lib:phIoT
doc: "Project Haystack definitions for Internet of Things"
version: "4.0"
baseUri: `https://project-haystack.org/def/phIoT/`
depends: [^lib:ph, ^lib:phScience]
```

```
https://project-haystack.org/def/phIoT/4.0#site
```

Ontology (vocabulary)

```
def: ^site
is: [^entity, ^geoPlace]
mandatory
doc: "Site is a geographic location of the built environment"
```

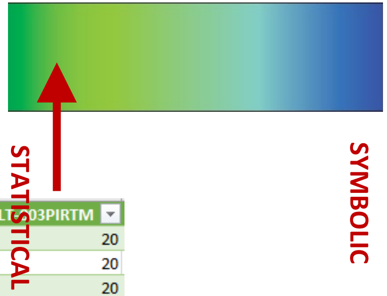
```
phIoT:site a owl:Class ;
    rdfs:subClassOf ph:entity,
        ph:geoPlace ;
    rdfs:comment "Site is a geographic location of the built environment" ;
    ph:is ph:entity,
        ph:geoPlace ;
    ph:lib phIoT:lib:phIoT ;
    ph:mandatory ph:marker .
```

Instances (data)

```
id:@24192ca1-0c85f75d "Headquarters"
site
area:140797ft²
tz:New_York
dis:Headquarters
geoAddr:"600 W Main St, Richmond, VA"
geoCoord:C(37.545826,-77.449188)
hq
metro:Richmond
primaryFunction:Office
yearBuilt:1999
```

```
_:24192ca1-0c85f75d
    a phIoT:site ;
    ph:hasTag site,
    phIoT:area 140797 ;
    ph:tz "New_York" ;
    ph:dis "Headquarters" ;
    ph:geoAddr "600 W Main St, Richmond, VA" ;
    ph:geoCoord "C(37.545826,-77.449188)" ;
    phIoT:primaryFunction "Office" ;
    phIoT:yearBuilt 1999 .
```

Labels and Tags



Extract from BMS (labels):

[illegible]	11NR008FT-013FLW	11NR008FT-038FLW	11NR008FT-039FLW	11NR008FT-040FLW	11NR008FT-301FLW	11NR008FT-302FLW	11NR008LT-001PIRTM	11NR008LT[illegible]03PIRTM
01/01/2021 00:02	0	0	0	280	0	0	20	20
01/01/2021 01:02	0	0	0	300	0	0	20	20
01/01/2021 02:02	0	0	0	300	0	0	20	20
01/01/2021 03:02	0	0	0	280	0	0	20	20
01/01/2021 04:02	0	0	0	300	0	0	20	20
01/01/2021 05:02	0	0	0	300	0	0	20	20
01/01/2021 06:02	0	0	0	300	0	0	30	30
01/01/2021 07:02	0	0	0	300	0	0	30	30
01/01/2021 08:02	0	0	0	300	0	0	30	30
01/01/2021 09:02	0	0	0	300	0	0	30	30
01/01/2021 10:02	0	0	0	300	0	0	30	30
01/01/2021 11:02	0	0	0	300	0	0	30	30
01/01/2021 12:02	0	0	0	300	0	0	30	30
01/01/2021 13:02	0	0	0	300	0	0	30	30
01/01/2021 14:02	0	0	0	280	0	0	30	30
01/01/2021 15:02	0	0	0	300	0	0	30	30

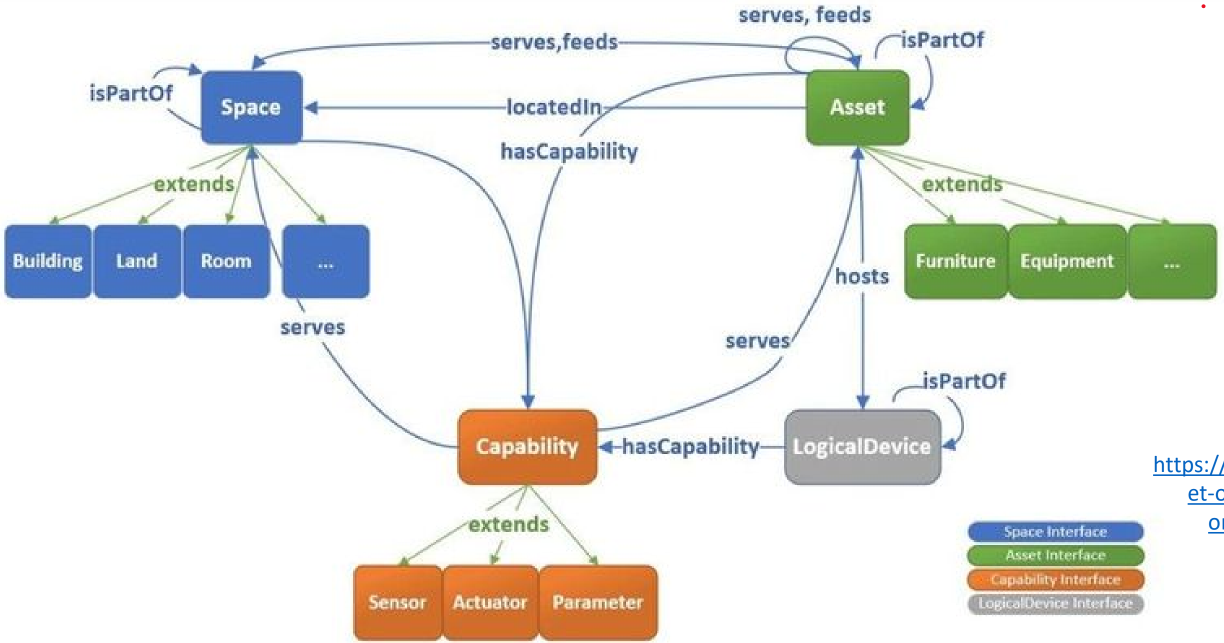
Mapping of sensors to rooms in BMS and then building model:

Column1	ItemName	ItemDescriptionDutch	ItemDescriptionEnglish	Spaces
6358	11NR009TE-033TRL	RUIMTETEMPERATUUR 9_Z01	ROOM TEMPERATURE 9_Z01	9_Z01
6359	11NR009TE-030TRL	RUIMTETEMPERATUUR 9_Z01	ROOM TEMPERATURE 9_Z01	9_Z01
2089	11NR009TE-030TRLBP	ACTUEEL SETPOINT REGELING 9_Z01	CURRENT SETPOINT SCHEME 9_Z01	9_Z01
6824	11NR009SCR030BNDRS	STAND ZONWERING 9_Z01	STAND AWNINGs 9_Z01	9_Z01
5900	11NR009CV-030RWWRS	REGELSIGNAAL VERWARMING 9_Z01	CONTROL SIGNAL HEATING 9_Z01	9_Z01
5899	11NR009CV-033RWWRS	REGELSIGNAAL VERWARMING 9_Z01	CONTROL SIGNAL HEATING 9_Z01	9_Z01
4349	11NR009TE-033MAXBP	KOELINGSETPOINT 9_Z01	COOLING SETPOINT 9_Z01	9_Z01
4350	11NR009TE-030MAXBP	KOELINGSETPOINT 9_Z01	COOLING SETPOINT 9_Z01	9_Z01
3568	11NR009TE-030CPA	CPA VIA WANDMODULE RUIMTE 9_Z01	CPA VIA WALL MODULE ROOM 9_Z01	9_Z01
3567	11NR009TE-033CPA	CPA VIA WANDMODULE RUIMTE 9_Z01	CPA VIA WALL MODULE ROOM 9_Z01	9_Z01
6823	11NR009SCR033BNDRS	STAND ZONWERING 9_Z01	STAND AWNINGs 9_Z01	9_Z01

Real Estate Core

Microsoft RealEstateCore JÖNKÖPING UNIVERSITY willow know your world Azure Digital Twins

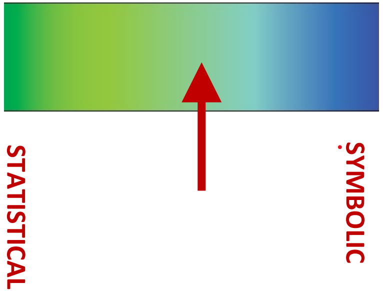
STATISTICAL SYMBOLIC



https://techcommunity.microsoft.com/t5/internet-of-things/realestatecore-a-smart-building-ontology-for-digital-twins-is/ba-p/1914794

Hartig et al., 2019

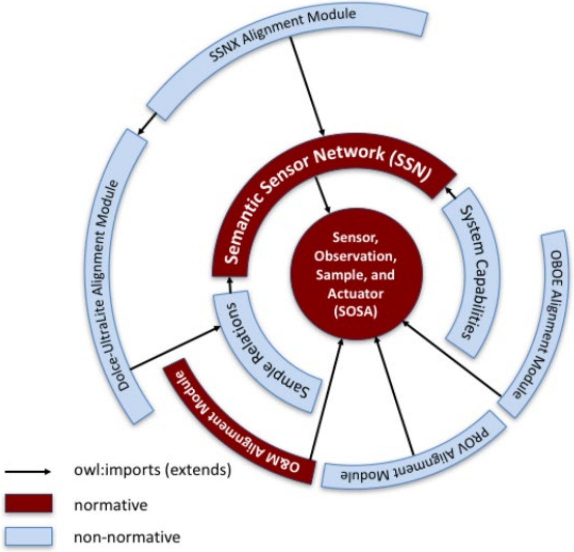
SAREF, SOSA and SSN



Example:

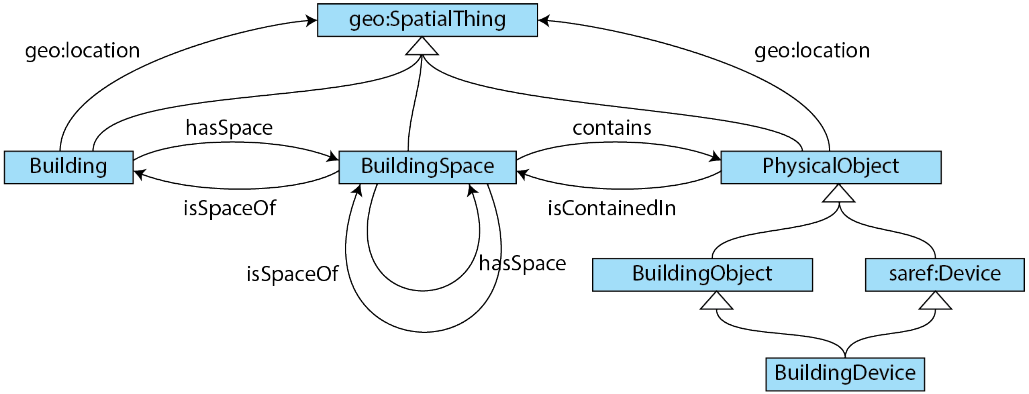
```
@prefix cdt:
  <http://w3id.org/lindt/custom_datatypes#> .
BASE <http://example.org/>
<observation/235727> a sosa:Observation ;
  sosa:hasFeatureOfInterest <house/134/kitchen> ;
  sosa:observedProperty <electricConsumption> ;
  sosa:madeBySensor <sensor/927> ;
  sosa:hasSimpleResult "22.4 kWh"^^cdt:ucum .
```

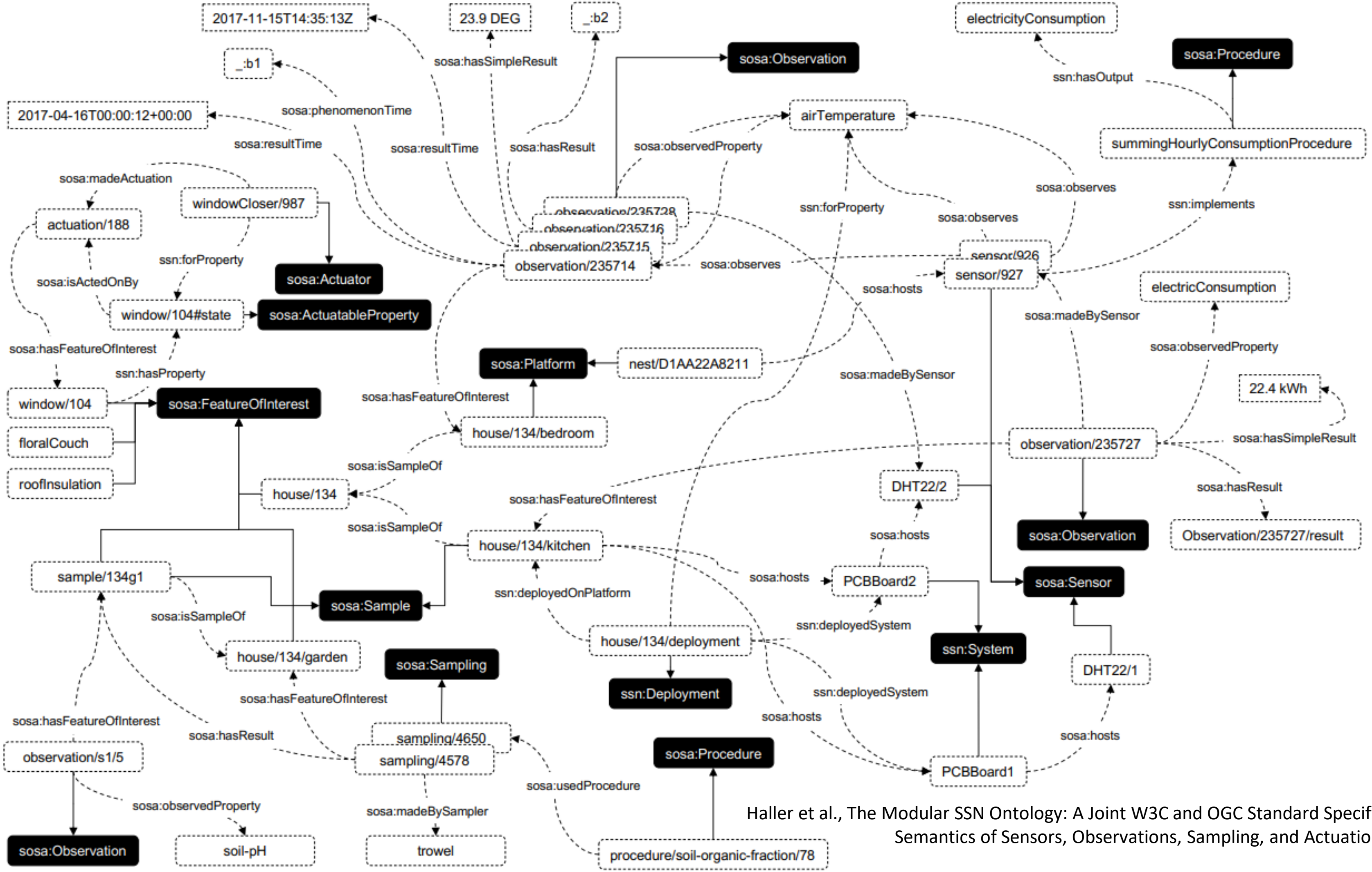
```
<observation/235715> a sosa:Observation ;
  sosa:resultTime
      "2017-11-15T14:35:13Z"^^xsd:dateTime ;
  sosa:hasSimpleResult "23.9 DEG"^^cdt:temperature .
```



Haller et al., The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation, 2018.

Ontology Core:





Haller et al., The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation, 2018.

Presentation Outline

1. Stepping through Building Data Semantics: BIM, IFC, LBD, BRICK, Haystack, etc.
2. The ‘Scale of AI Methods’: ML and semantics
3. Towards scalable system integration that combines multiple AI fields

Categories of data?

- Data streams:
  - Ordered lists of values, typically floating point values
  - Large amount of data -> data lakes
  - Almost no semantics, at best a few labels for categorization
  - E.g. temperature measurements, system logs (e.g. triggering of actuators), etc.
- Semantically rich and interconnected data:
  - Seldom including large data streams
  - Long debates about the semantics of things -> standardisation
  - Complex and brittle (breaks easily)
  - Small amount of very important data
  - Easy to combine with rule-based and/or logic-based technologies (inference and query)
  - E.g. BIM models, semantic web ontologies, taxonomies, OTLs, etc.
- Control models:
  - Algorithms for control, parametric functions
  - Communication system, signal processing, direct control, low latency
  - Typically located on the edge (devices with embedded functions)
  - E.g. Control Description Language (CDL), modelica models
- User data:
  - Outside of the system
  - Different privacy and security requirements
- Files:
  - No semantically rich encodings, no data streams
  - Highly valuable, and seldom machine-processable
  - E.g. PDFs, Images, Geometry ‘blobs’

Categories of data?

- Data streams:
  - Ordered lists of values, typically floating point values
  - Large amount of data -> data lakes
  - Almost no semantics, at best a few labels for categorization
  - E.g. temperature measurements, system logs (e.g. triggering of actuators), etc.
- Semantically rich and interconnected data:
  - Seldom including large data streams
  - Long debates about the semantics of things -> standardisation
  - Complex and brittle (breaks easily)
  - Small amount of very important data
  - Easy to combine with rule-based and/or logic-based technologies (inference and query)
  - E.g. BIM models, semantic web ontologies, taxonomies, OTLs, etc.

MACHINE LEARNING
STATISTICAL AI METHODS

- Control models:
  - Algorithms for control, parametric fu… ons
  - Communication system, signal processing, direct control, low latency
  - Typically located on the edge (devices with embedded functions)
  - E.g. Control Description Language (CDL), modelica models
- User data:
  - Outside of the system
  - Different privacy and s…
- Files:
  - No semantically rich e… ams
  - Highly valuable, and s… sable
  - E.g. PDFs, Images, Geo…

SEMANTICS
SYMBOLIC AI METHODS

Strengths and methods in Artificial Intelligence

Data streams

MACHINE LEARNING
STATISTICAL AI METHODS

- **Reinforcement learning**: data comes from an available experimentation environment
- **Neural networks**: ANNs, GNNs, CNNs, …
- **Traditional ML**:
  - **Regression**: predict next value
  - **Classification**: predict category / classification label
  - **Clustering**: group based on similarity
  - **Association**: identify sequences and combinations

SEMANTICS
SYMBOLIC AI METHODS

- **Semantic Web** technologies:
  - **Ontologies** and **formal** vocabularies
  - **Logics**: Description Logic, Defeasible Logic, etc.
- **Expert Systems**
- **Rule-based inference:** if-then rules

Semantically rich and interconnected data

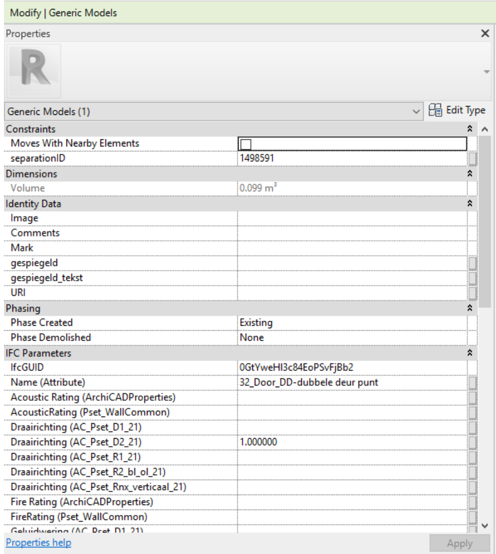
The Scale of AI Methods

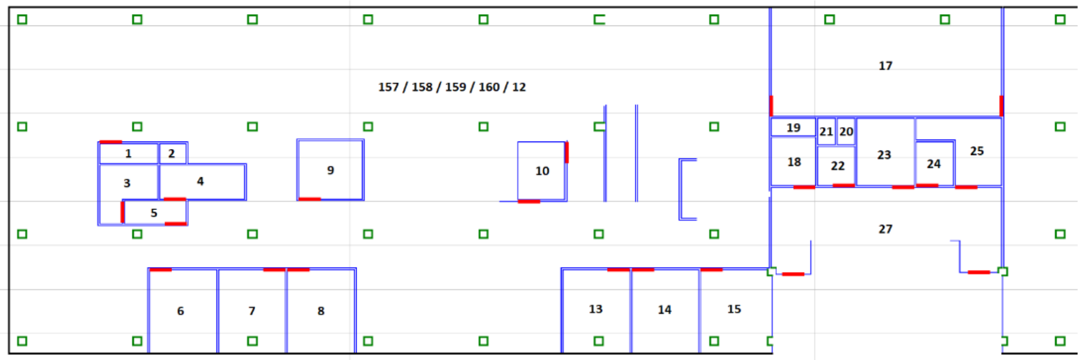
The Scale of AI Methods

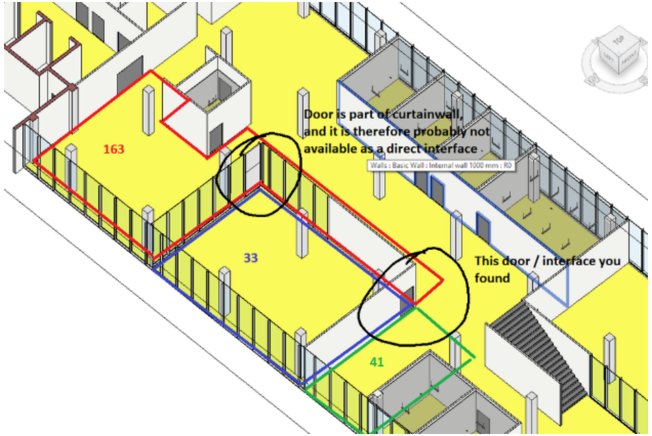
STATISTICAL AI **SYMBOLIC AI**

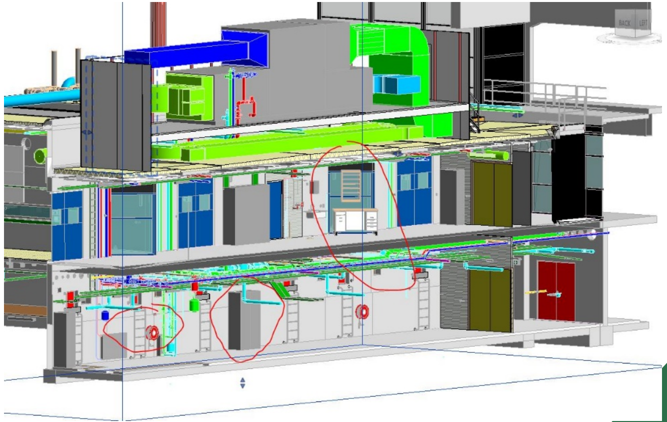
Can we integrate these methods and use all of them?

Semantic encoding of buildings

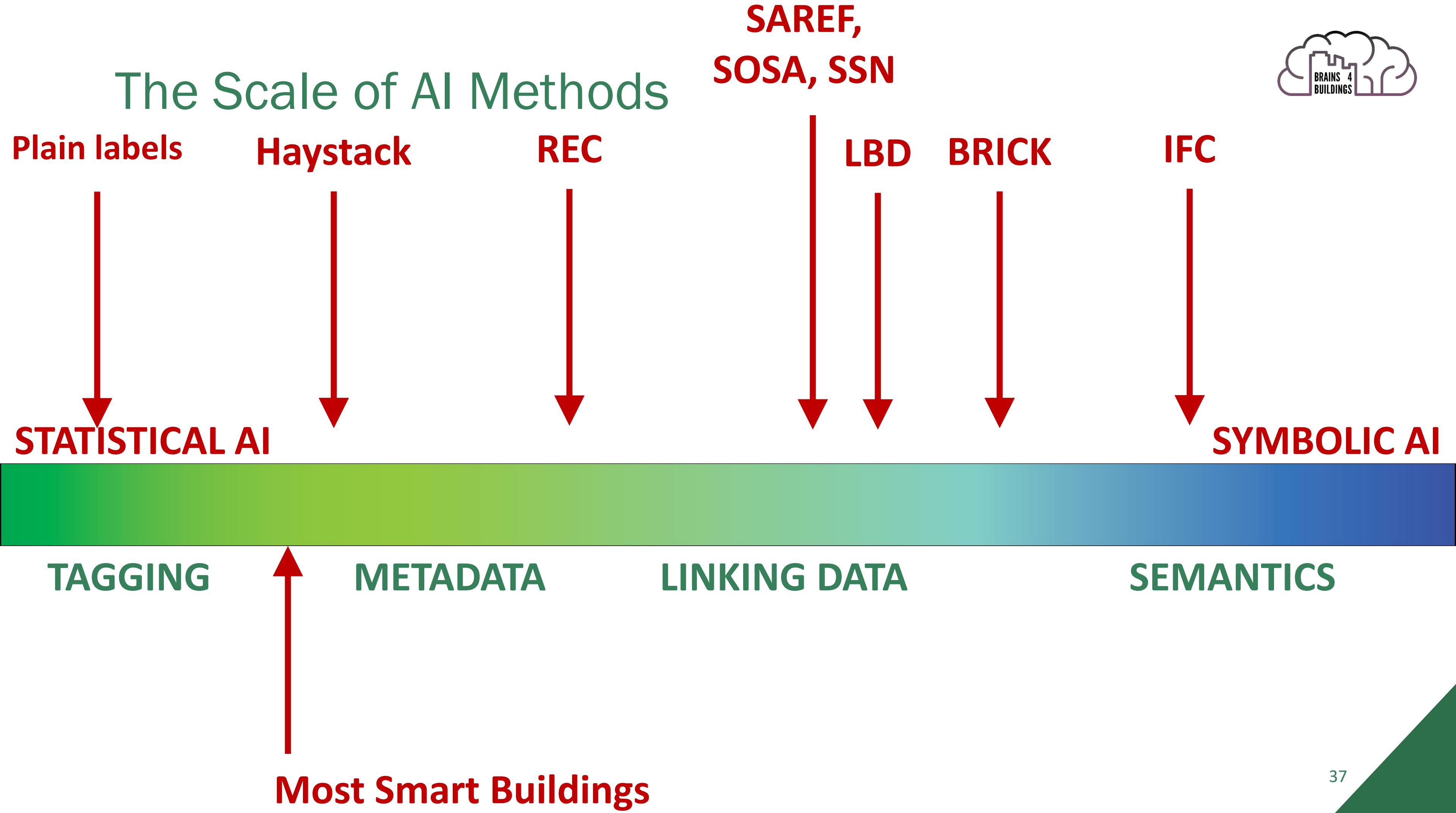




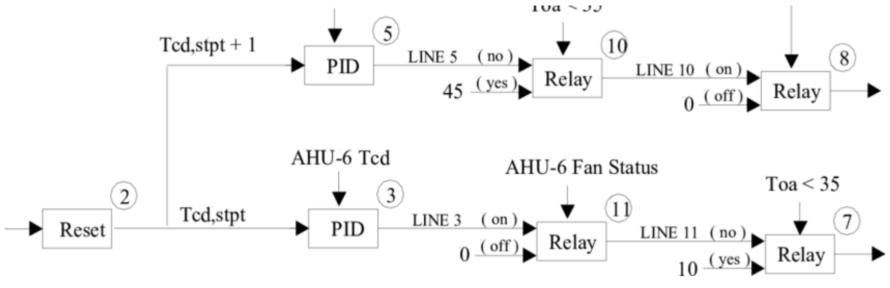


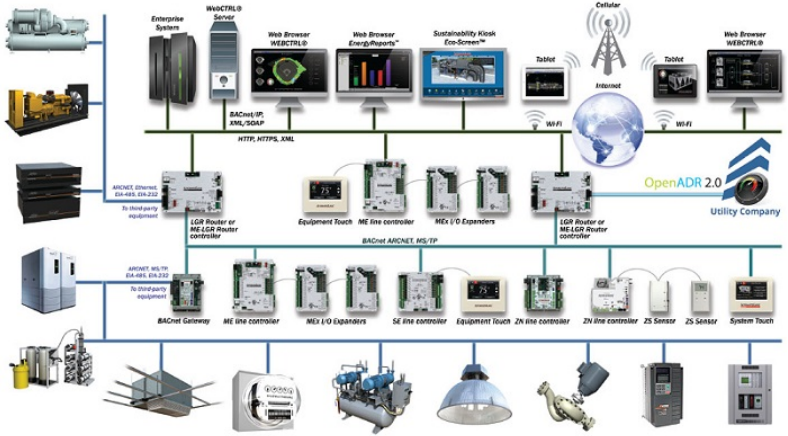


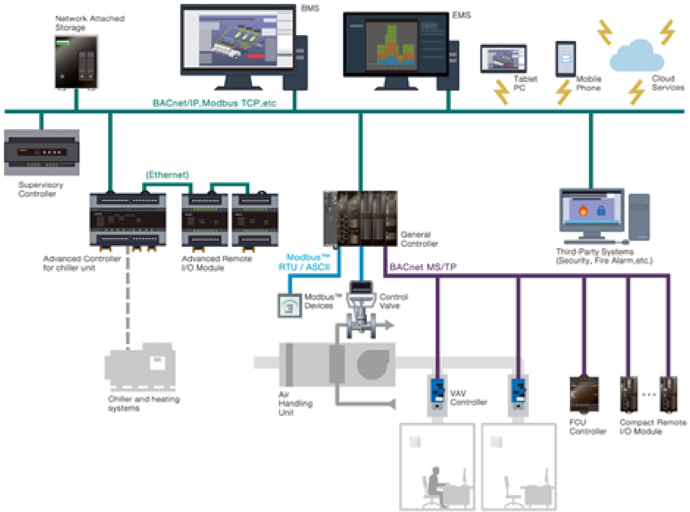
The Scale of AI Methods

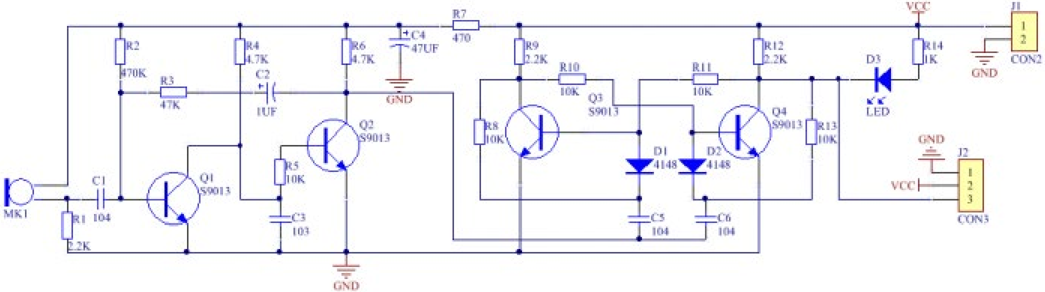


And all of this excludes control systems and control logic!!









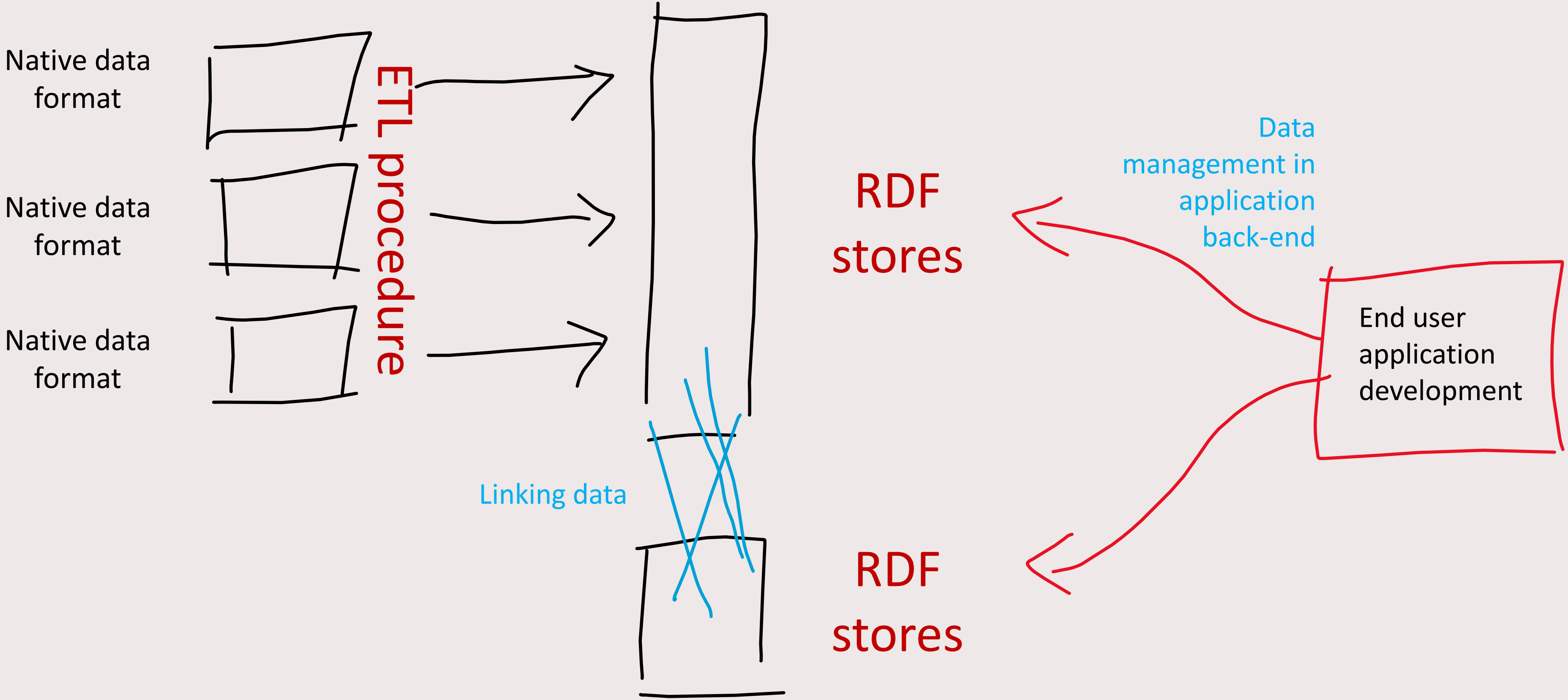
Presentation Outline

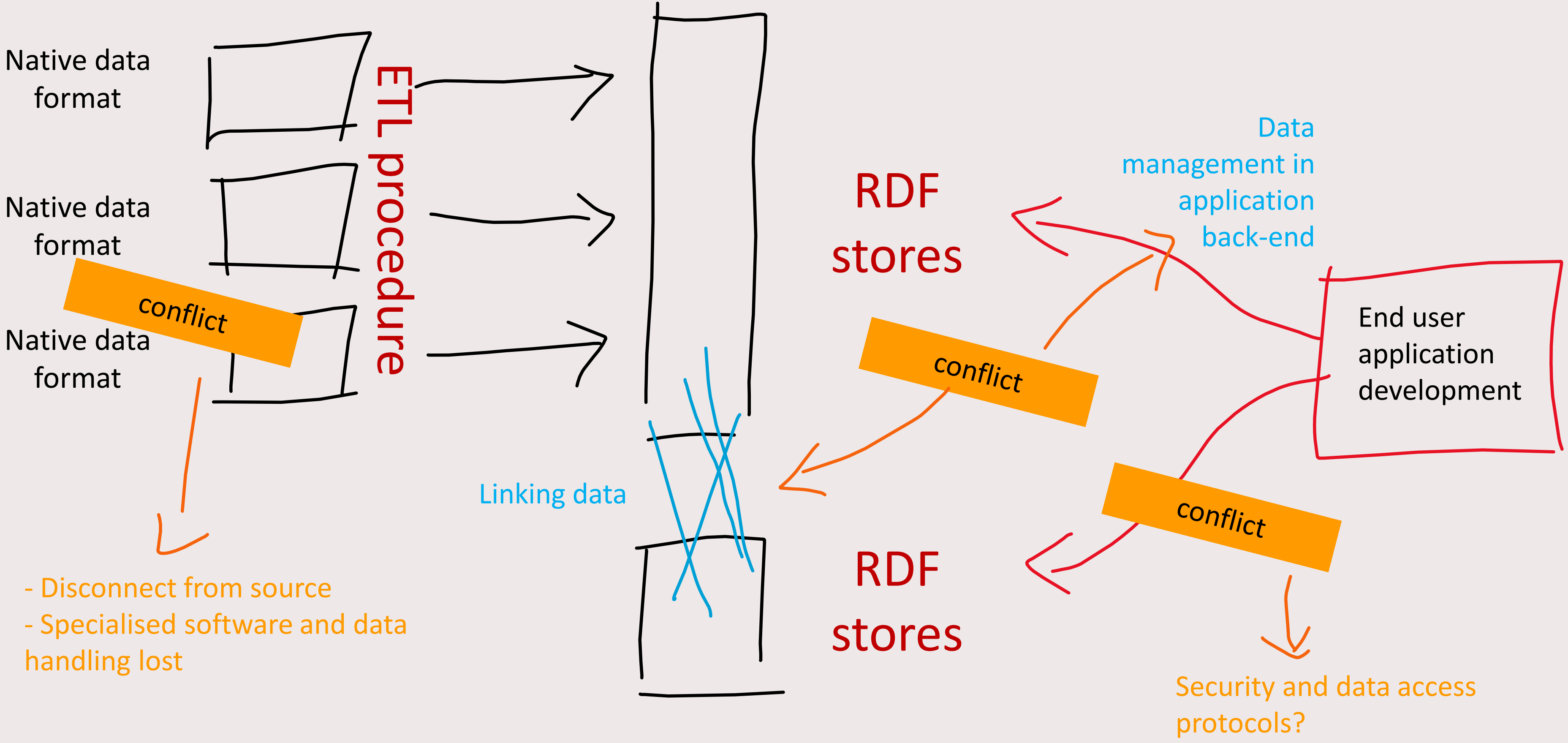
1. Stepping through Building Data Semantics: BIM, IFC, LBD, BRICK, Haystack, etc.
2. The ‘Scale of AI Methods’: ML and semantics
3. Towards scalable system integration that combines multiple AI fields

Available options for integration in trustworthy manner

OPTION 1: Transform all into semantic graphs (e.g. R2RML or custom data transformers) and do data integration

- Plus: all in same format
- Plus: inference possible
- Minus: unfit storage
- Minus: disconnect from origin
- Minus: no ML algorithms nor procedural code possible
- Minus: how to handle privacy and security (trust?)


Native data format
Native data format
Native data format
ETL procedure
RDF stores
Linking data
RDF stores
Data management in application back-end
End user application development

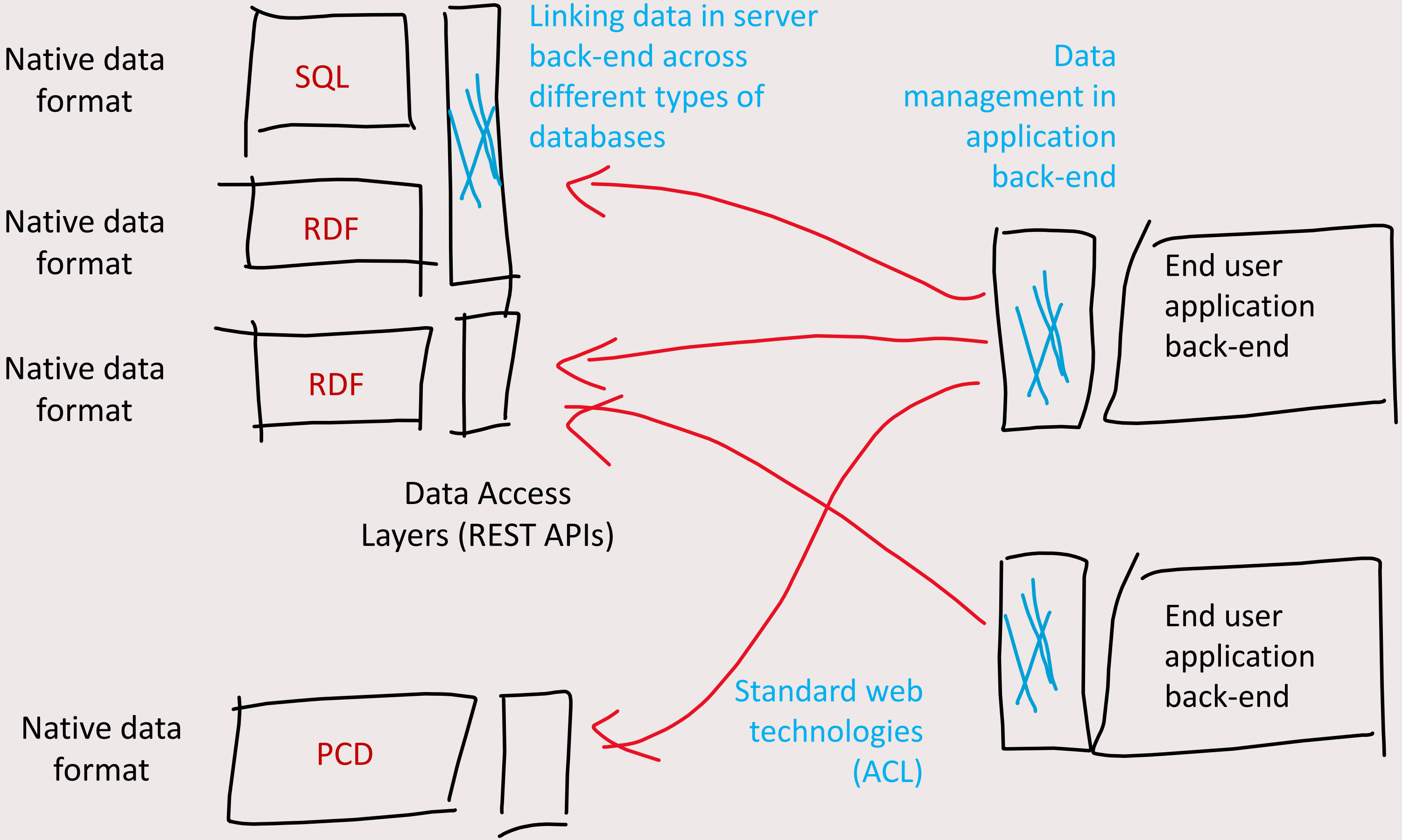


Native data format
Native data format
Native data format
conflict
ETL procedure
- Disconnect from source
- Specialised software and data handling lost
Linking data
RDF stores
RDF stores
conflict
Data management in application back-end
End user application development
conflict
Security and data access protocols?

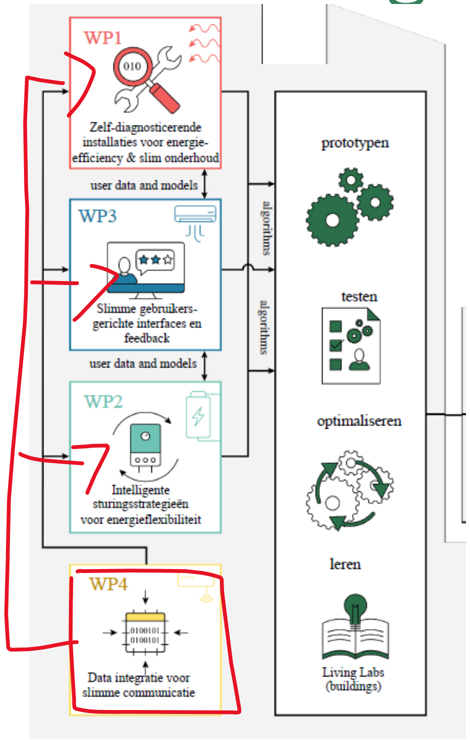
Available options for integration in trustworthy manner

OPTION 2: Store all in well-fit data stores (KV stores, graphDBs, relational DBs, timeseries stores, etc.) and perform data integration (also) on a system and API level (system integration)

- Plus: apt data storage
- Plus: data stays at source -> web-based connections needed
- Plus: ML algorithms and procedural algorithms not blocked
- Plus: Privacy and security can be easily handled at the gates of APIs and DBs.
- Minus: multitude of systems requires lots of diverse software and expertise


Native data format
SQL
Native data format
RDF
Native data format
RDF
Native data format
PCD
Linking data in server back-end across different types of databases
Data management in application back-end
Data Access Layers (REST APIs)
End user application back-end
End user application back-end
Standard web technologies (ACL)

Data Integration for Smart Buildings

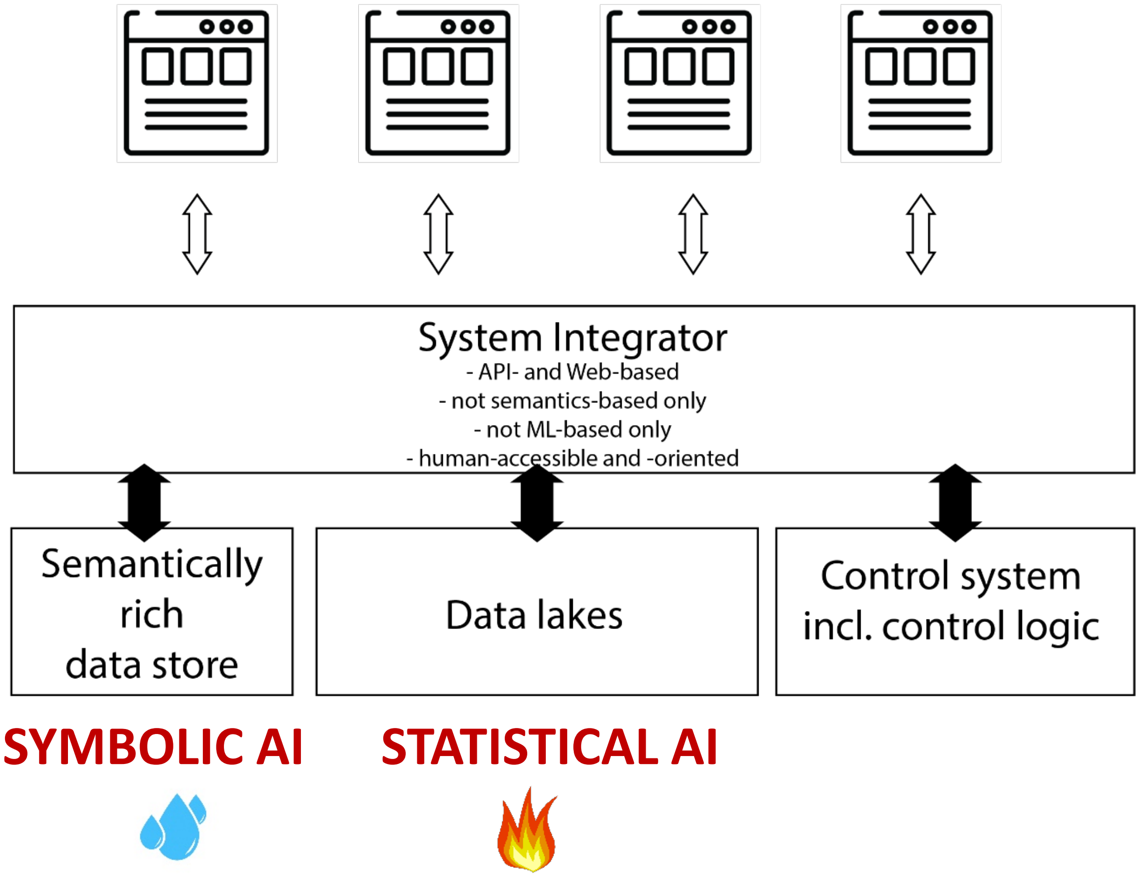


To enable making our buildings smarter, advanced data integration is needed (among several other matters):

- Ensure **data connectivity** between applications
- Ensure security, **ethical use and privacy of data**
- ~~Standardise data~~ sets and approaches
- Aim for **system integration at API level**, between individual systems of diverse manufacturers

Solution: combine inherently incompatible techniques using a system integration approach

Targeted framework for AI-based smart buildings



Important:

- Include access control (ACL)
- 'linking' of data on system integration level
- Agreement and standardization of labels and metadata tags
- Feed back into control systems!

Presentation Outline

1. Stepping through Building Data Semantics: BIM, IFC, LBD, BRICK, Haystack, etc.
2. The ‘Scale of AI Methods’: ML and semantics
3. Towards scalable system integration that combines multiple AI fields

TU/e

Take-aways to conclude

- Structure your data
- Make agreements about how you structure your data
- Be critical towards acronyms and look beyond them
- Store your data in dedicated technologies
- Aim for (web-ready) system integration