Metadata Schema Generation for Data-driven Smart Buildings

Lasitha Chamari[1], Joep van der Weijden[2], Lolke Boonstra[2], Stefan Hoekstra[2], Ekaterina Petrova[1] and Pieter Pauwels[1]

[1]*Eindhoven University of Technology, The Netherlands*

[2]*Delft University of Technology, The Netherlands*

The project received funding from Dutch Ministry of Economic Affairs and Climate Policy and Ministry of the Interior and Kingdom Relations under the MOOI program

Presentation Outline

1. Background
2. What is a metadata schema of a smart building?
3. #GenerateMyMetadataSchema
4. Conclusion and outlook

Background

- One of the living labs in B4B project
- Tools with live data developed in B4B
  - Therefore, we need live data from the BMS

Fig.1 Living Lab Building

Background

- Usually, a “tag“ or “code” encapsulate data about Point.
- These complex tags are created to be understood by field engineers.
- Need lot of discussions with field engineer and customized translators, making it harder to use available data for any party interested in developing data-driven building controllers[2].
- How these BAS data points can be recognizable and unified across various vendors and buildings and how to make machine-readable?

What is a metadata schema of a smart building?

- **Metadata** (physical, logical and virtual assets in buildings and the relationships between them) represented in standard semantic descriptions.
- Created using ontologies (such as Brick, BOT).
- Use semantic web technologies.

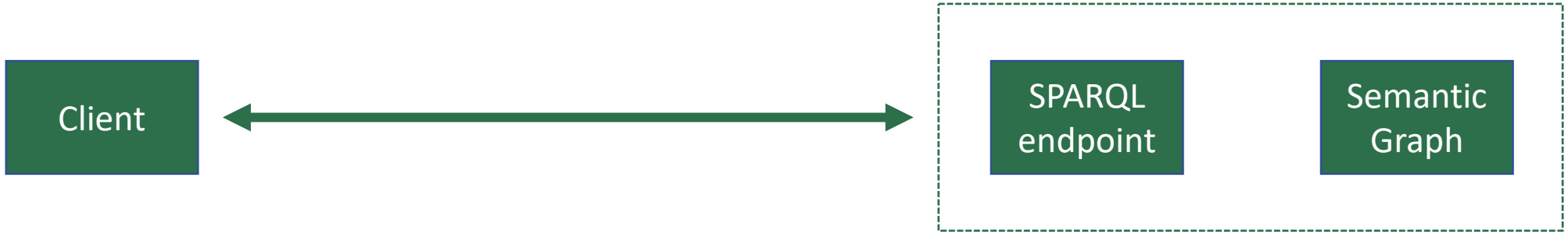


Fig.3 Using semantic web technologies (RDF data and SPARQL)

#GenerateMyMetadataSchema (5 Steps)

1. Identifying the objective
2. Sort the relevant metadata
3. Chose the ontology & identify the mappings to the ontology.
4. Generate the schema.
5. Integrate the schema with the applications.

Step1 : What we have and where we want to go

Table 1
Metadata extraction from BAS containing time-series reference (Item Reference) and descriptions

	Item Reference	Object ID	Object Type	Point Name	Description (NL)	Description (EN)
1	XXX.FEC005.CLG-O	CLG-O	AO Mapper	(33) 201.CV-02V--	Regelafsluiter koeler	Cooler control valve
2	XXX.FEC006.CLG-O	CLG-O	AO Mapper	(33) 202.CV-02V--	Regelafsluiter koeler	Cooler control valve
3	XXX.SHWP1-FAULT	SHWP1-FAULT	BI Mapper	(33) 001.TP-01A--	Transportpomp 1 storing	Transport pump 1 malfunction

Semantic Graph of BAS **to understand and query the timeseries data**

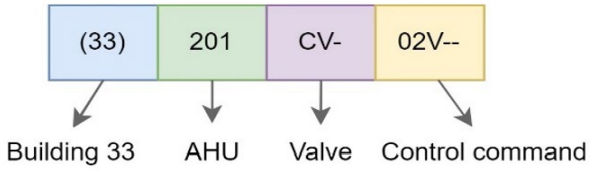


Fig. 4 BMS points naming convention

Systeem nr.	Omschrijving	Voorbeeld
001 t/m **009**	Ketels	001 = Ketel 1 002 = Ketel 2
011 t/m **019**	Transportsysteem Warmwa-ter	011 = Transportsysteem ketel 1 012 = Transportsysteem ketel 2
021 t/m **099**	Warmwater groepen	021 = Warm water groep 1

AKM	Absorptiekoelmachine	MC-	Vochtregeling
BA-	Brandmelding	MT-	Vocht transmitter
BG-	Breekglaasje	NB-	No brak
BKA	Brandklep afvoer	ND-	Nooddrukker
BKT	Brandklep toevoer	O--	Optimalisering
BL-	Blusgasinstallatie	OK-	Overvalknop
BMC	Brandmeldcentrale	PA-	Systeemdruk alarm

Fig. 5 Identifiers in JC BMS Manual (pdf)

Step 2: Selecting the metadata from the Building

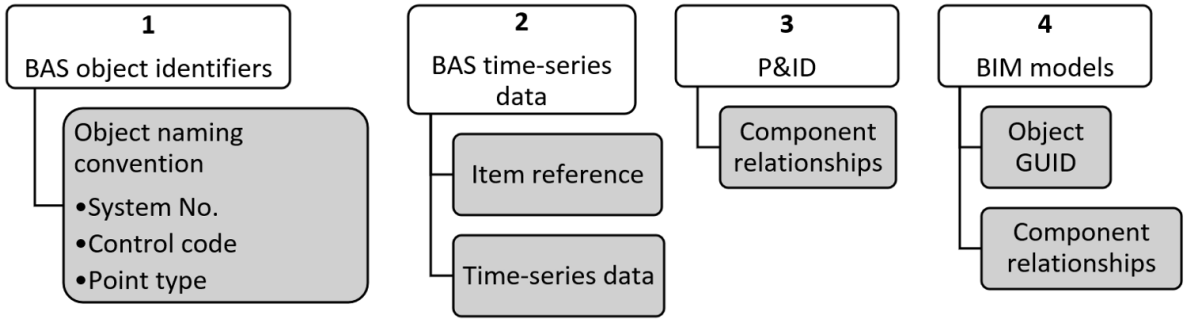


Fig.6 Metadata sources available in the building

Item Reference	Object ID	Object Type	Name	Description
TUDelft/FCB.Local Application.SYS-RESET	SYS-RESET	BI Mapper	(33) 801.RK-01SR2	Reset storing regelkast
TUDelft/FCB.Local Application.MISC3-A	MISC3-A	BI Mapper	(33) 801.RK-01AI-	Installatie automaten
TUDelft/FCB.Local Application.MISC2-A	MISC2-A	BI Mapper	(33) 801.RK-01OB-	Overspanningsbeveiliging
TUDelft/FCB.Local Application.MISC1-A	MISC1-A	BI Mapper	(33) 801.RK-01AN-	Netwachter
TUDelft/FCB.Local Application.Sprinkler-A1	Sprinkler-A1	BI Mapper	(33) 501.XA-01A1-	Sprinkeler storing
TUDelft/FCB.Local Application.Sprinkler-A3	Sprinkler-A3	BI Mapper	(33) 501.XA-01A3-	Sprinkeler reserve
TUDelft/FCB.Local Application.SHWP1-FAULT	SHWP1-FAULT	BI Mapper	(33) 001.TP-01A--	Transportpomp 1 storing
TUDelft/FCB.Local Application.Sprinkler-A2	Sprinkler-A2	BI Mapper	(33) 501.XA-01A2-	Sprinkeler reserve
TUDelft/FCB.Local Application.SHWP2-FAULT	SHWP2-FAULT	BI Mapper	(33) 001.TP-02A--	Transportpomp 2 storing
TUDelft/FCB.Local Application.FIRE-A	FIRE-A	BI Mapper	(33) 801.RK-01BA-	Brandmelding
TUDelft/FCB.Local Application.LA1-A	LA1-A	BI Mapper	(33) 901.LA-01A--	Wateroverlast technische ruimte
TUDelft/FCB.Local Application.LA2-A	LA2-A	BI Mapper	(33) 901.LA-02A--	Wateroverlast MER
TUDelft/FCB.Local Application.LA3-A	LA3-A	BI Mapper	(33) 901.LA-03A--	Wateroverlast DC ruimte
TUDelft/FCB.Local Application.MISC-A1	MISC-A1	BI Mapper	(33) 902.LV-01A--	Tracing vetvangleiding groep 1 storing

Fig.7 BMS Metadata table

Step 3: Mapping to the Brick ontology

Brick's main focus is BMS, matured ontology

Can be linked with other ontologies (BOT, SAREF)

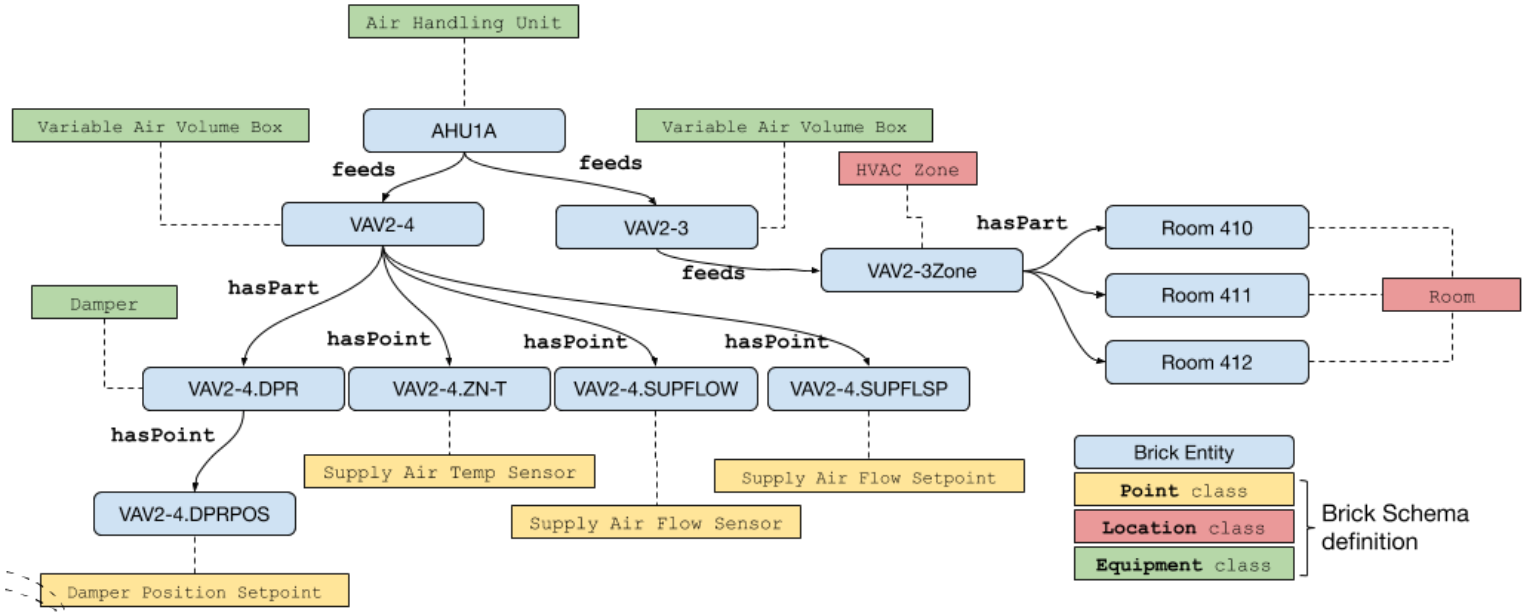


Fig.8 Representing AHU using the Brick ontology
https://brickschema.org/

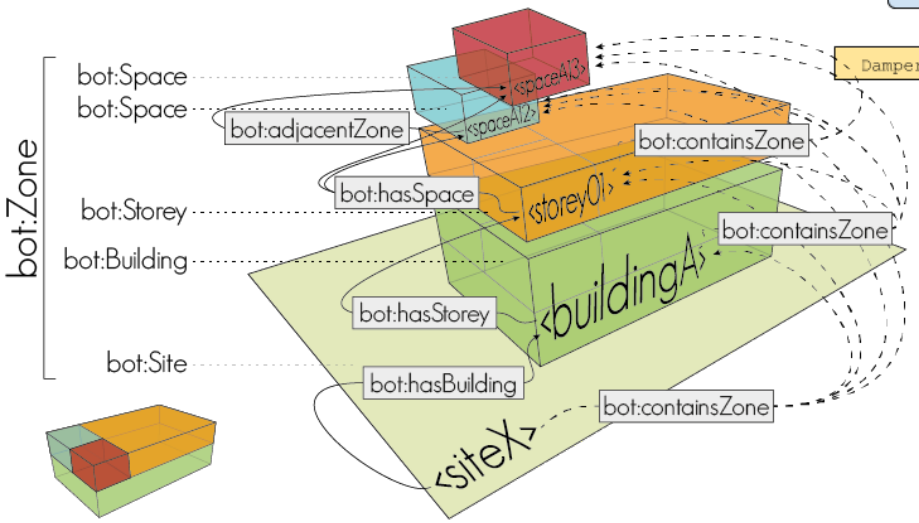


Fig.9 Representing the building topology using the BOT ontology
https://w3c-lbd-cg.github.io/bot/

Text search engine + Human Input

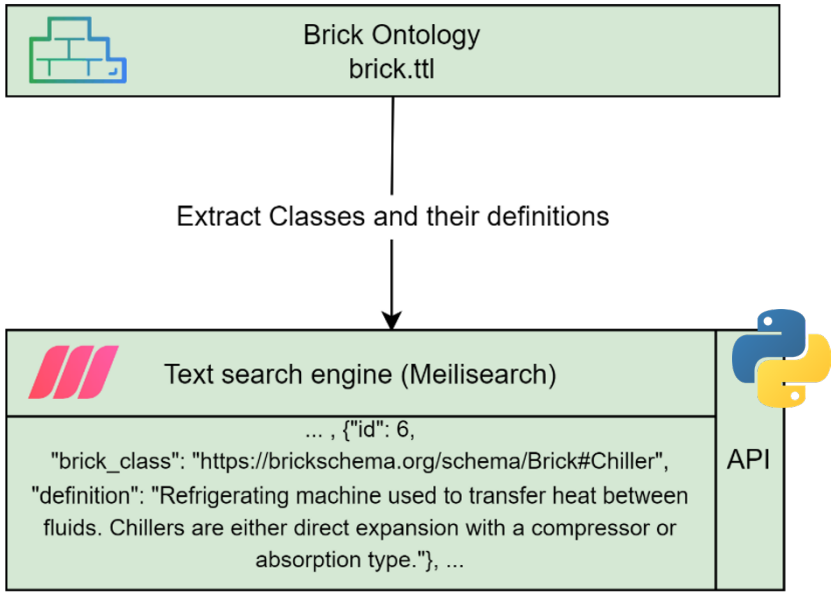


Fig.10 Extracting class names and definitions from the Brick ontology

System No.

Identifier	Description (NL)	Description (EN)	Brick Class
001	Ketel 1	Boiler 1	Boiler
001	Ketel 1	Boiler 1	Boiler
101	Koelmachine 1	Chiller 1	Chiller
201	LBK 1	AHU 1	AHU

brick:Equipment

Control Code

Identifier	Description (NL)	Description (EN)	Brick Class
CV	Regelafsluiter	Regulating valve	Valve
VA-	Afzuig ventilator	Exhaust fan	Exhaust_Fan
CR	Cardreader	Card reader	Access_Reader
HT	Enthalpie Transmitter	Enthalpy Transmitter	Enthalpy_Sensor

brick:Equipment

Point type

Identifier	Description (NL)	Description (EN)	Brick Class
A--	Alarm	Alarm	Alarm
B--	Bedrijfsmelding	Operation status	Status
V--	Sturing	Control	Valve_Command

brick:Point

Fig.12 Three parts of the naming mapped to the Brick ontology

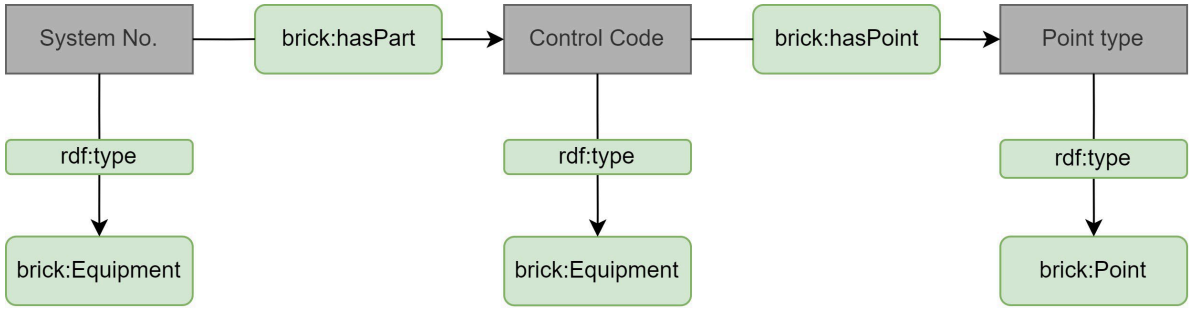


Fig.13 Relevant Brick Classes and Relationships

Step 4: Generating Metadata Schema

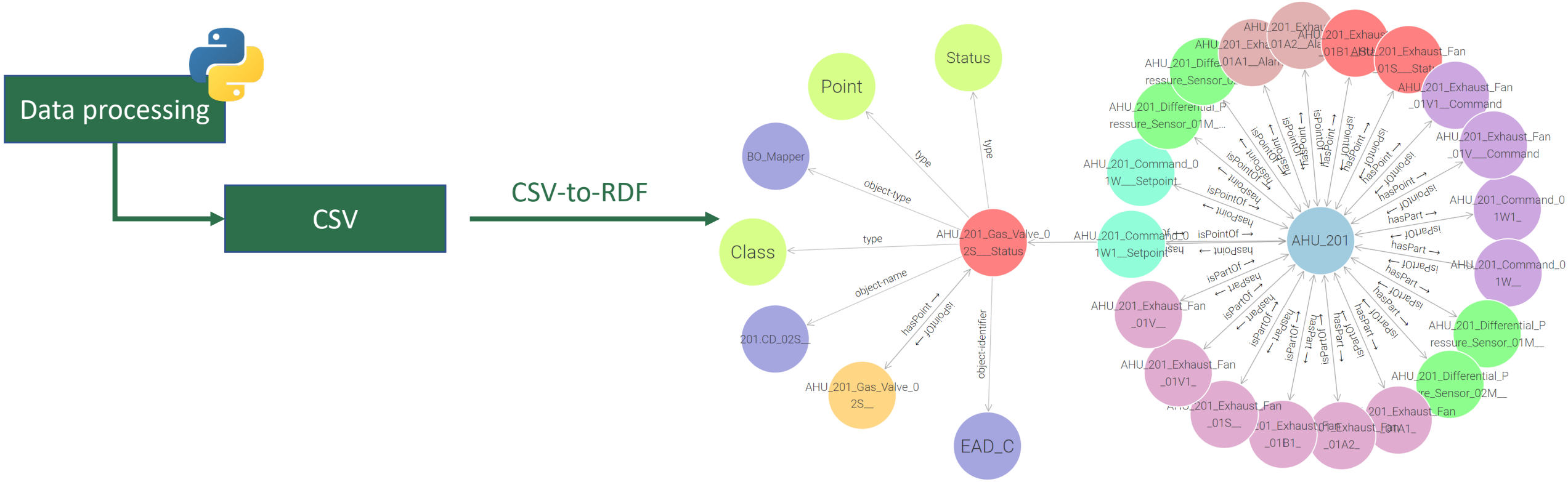


Fig.14 Part of the metadata (AHU 201)

CSV-to-RDF
https://docs.brickschema.org/lifecycle/creation.html#from-structured-tabular-sources

Step 5 : Integrating the schema with a smart building API

- Stored in a **graph database** and integrated with the building API

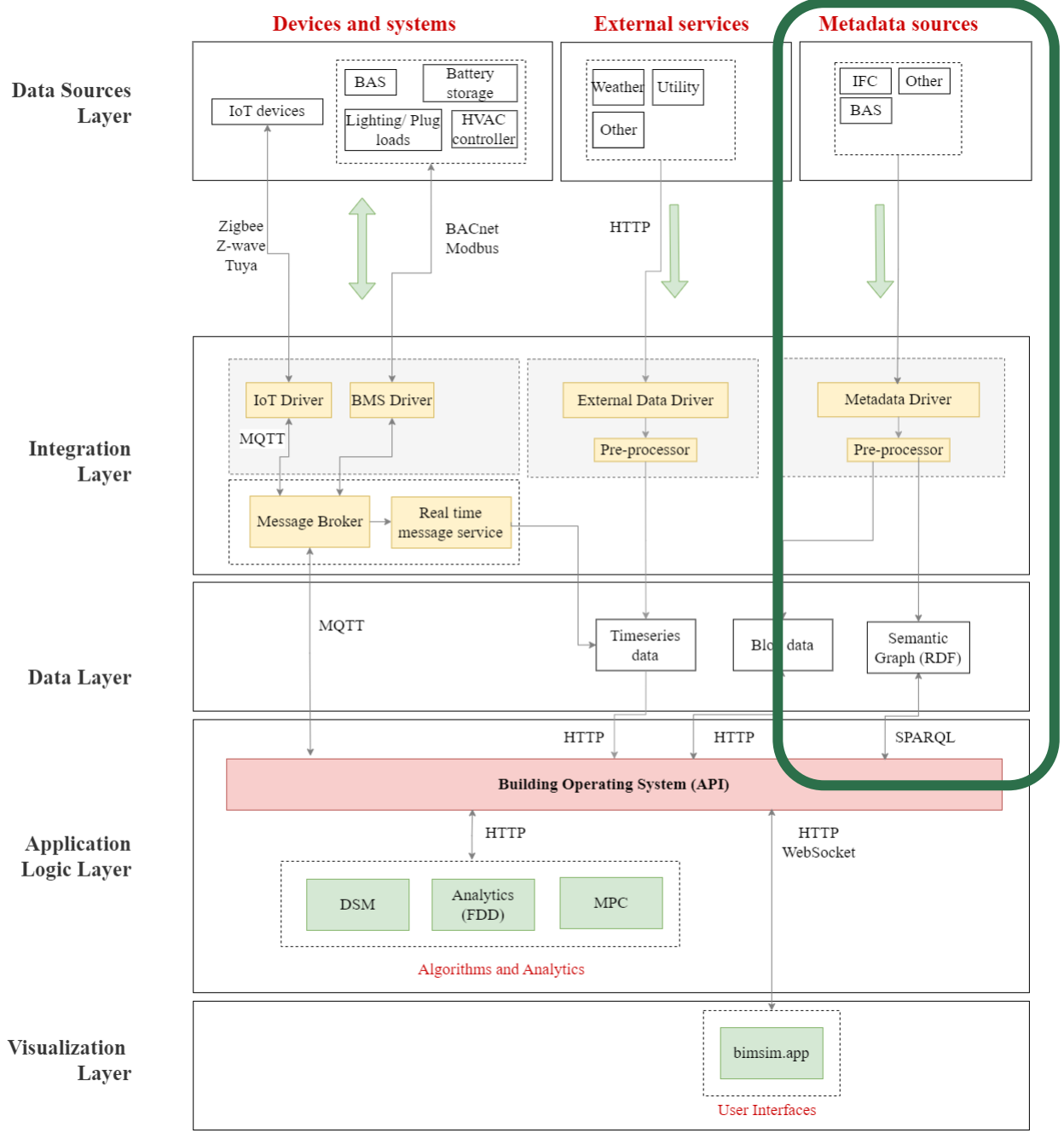
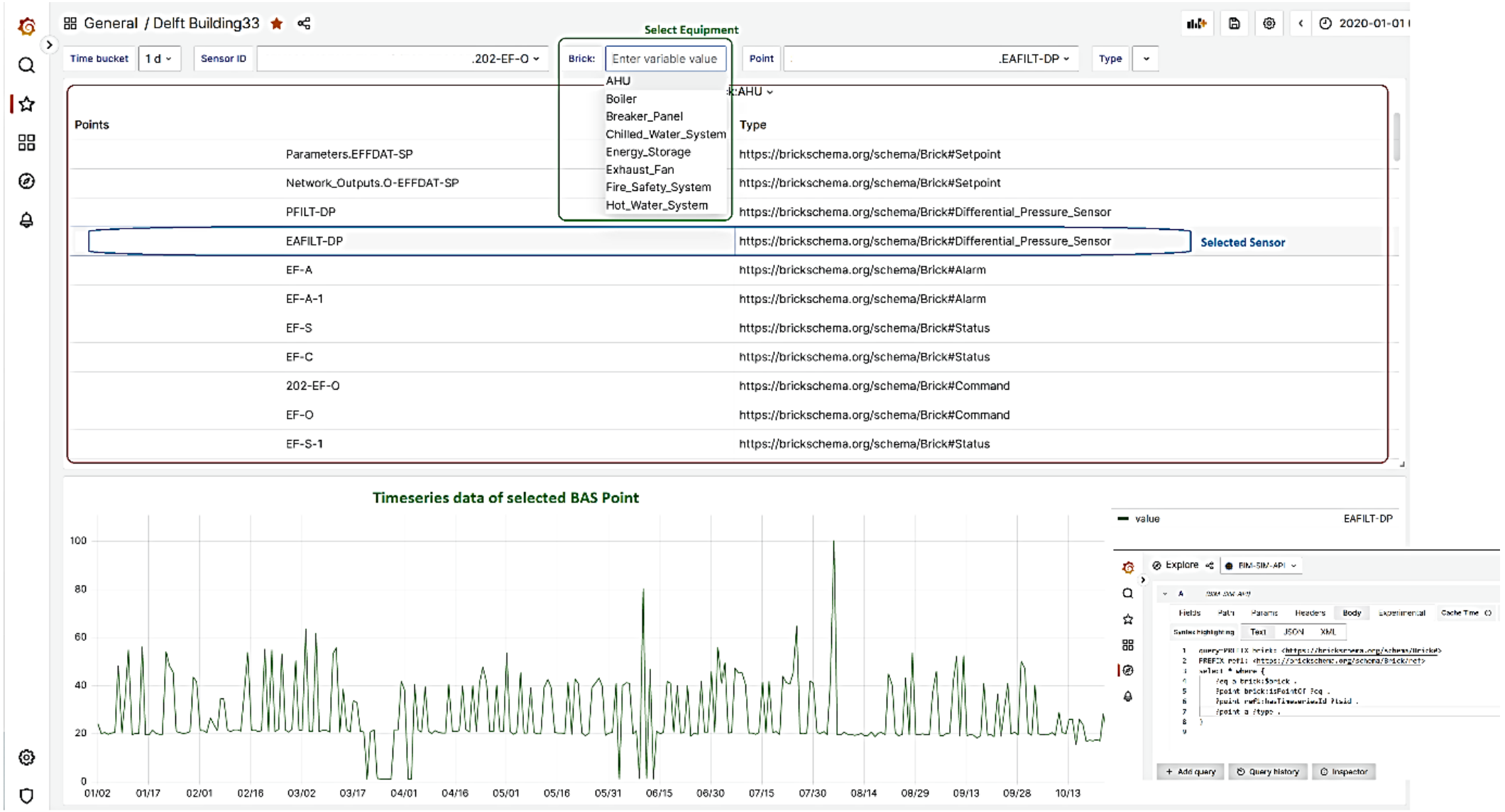


Fig.15 Smart building system architecture[12]

Demo: Grafana dashboard



Demo: Grafana dashboard

Summary and outlook

- Given a proper naming convention, mapping to Brick is achievable, but many buildings do not have a documented naming convention.
- Other improvements include,
    - use sensor readings in the classification (sensor type)

Thank you

Lasitha Chamari
PhD Candidate
Information Systems in the Built Environment
Department of Built Environment
l.c.rathnayaka.mudiyanselage@tue.nl