

# Portable Queries Using the *Brick Schema* for Building Applications

<http://brickschema.org>

Open sourced with BSD License

B. Balaji<sup>1</sup>, A. Bhattacharya<sup>2</sup>, G. Fierro<sup>2</sup>, J. Gao<sup>3</sup>, J. Gluck<sup>3</sup>, D. Hong<sup>4</sup>, A. Johansen<sup>5</sup>, J. Koh<sup>6</sup>, J. Ploennigs<sup>7</sup>, Y. Agarwal<sup>3</sup>, M. Berges<sup>3</sup>, D. Culler<sup>2</sup>, R. Gupta<sup>6</sup>, M. Kjærsgaard<sup>5</sup>, M. Srivastava<sup>1</sup>, K. Whitehouse<sup>4</sup> (<sup>1</sup>UCLA, <sup>2</sup>UCB, <sup>3</sup>CMU, <sup>4</sup>UVA, <sup>5</sup>USD, <sup>6</sup>UCSD, <sup>7</sup>IBM-Ireland)

## Brick Overview

- A common metadata description
- It can express application requirements
- Query metadata without domain knowledge

### Goals

- **Completeness:** capture all sensors/subsystems
- **Expressiveness:** capture all relationships needed to run applications
- **Usability:** easy to understand, easy to port buildings

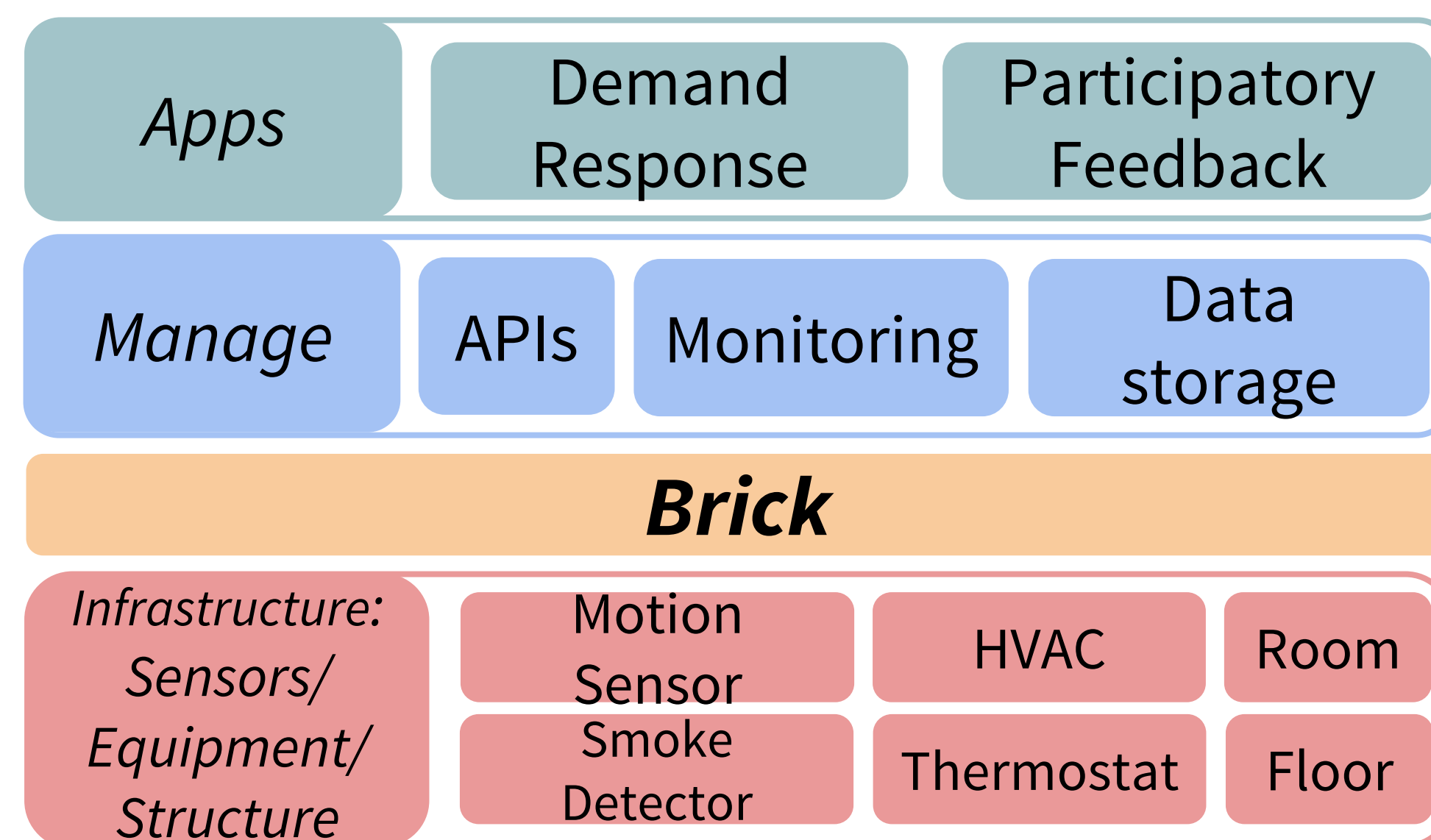


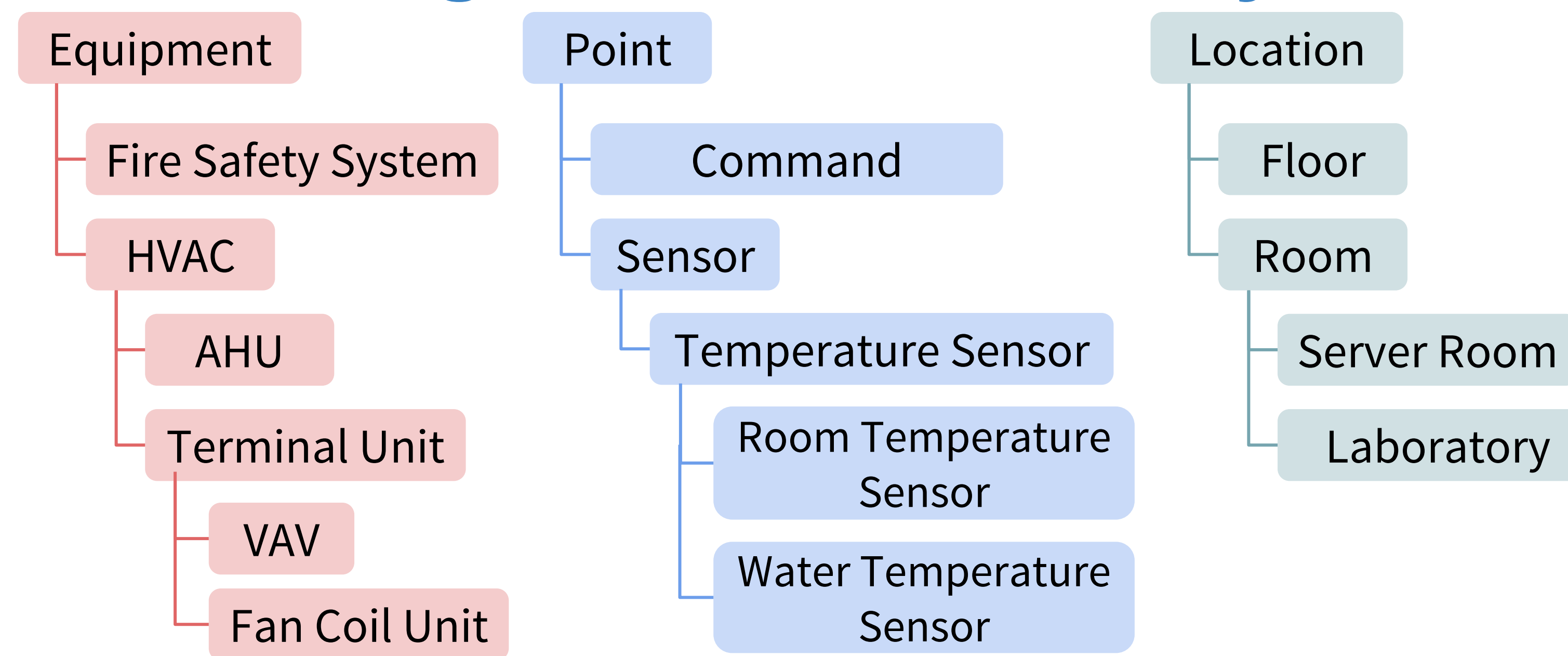
Fig. 1. Architecture for Smart Building Systems.

## Brick Structure

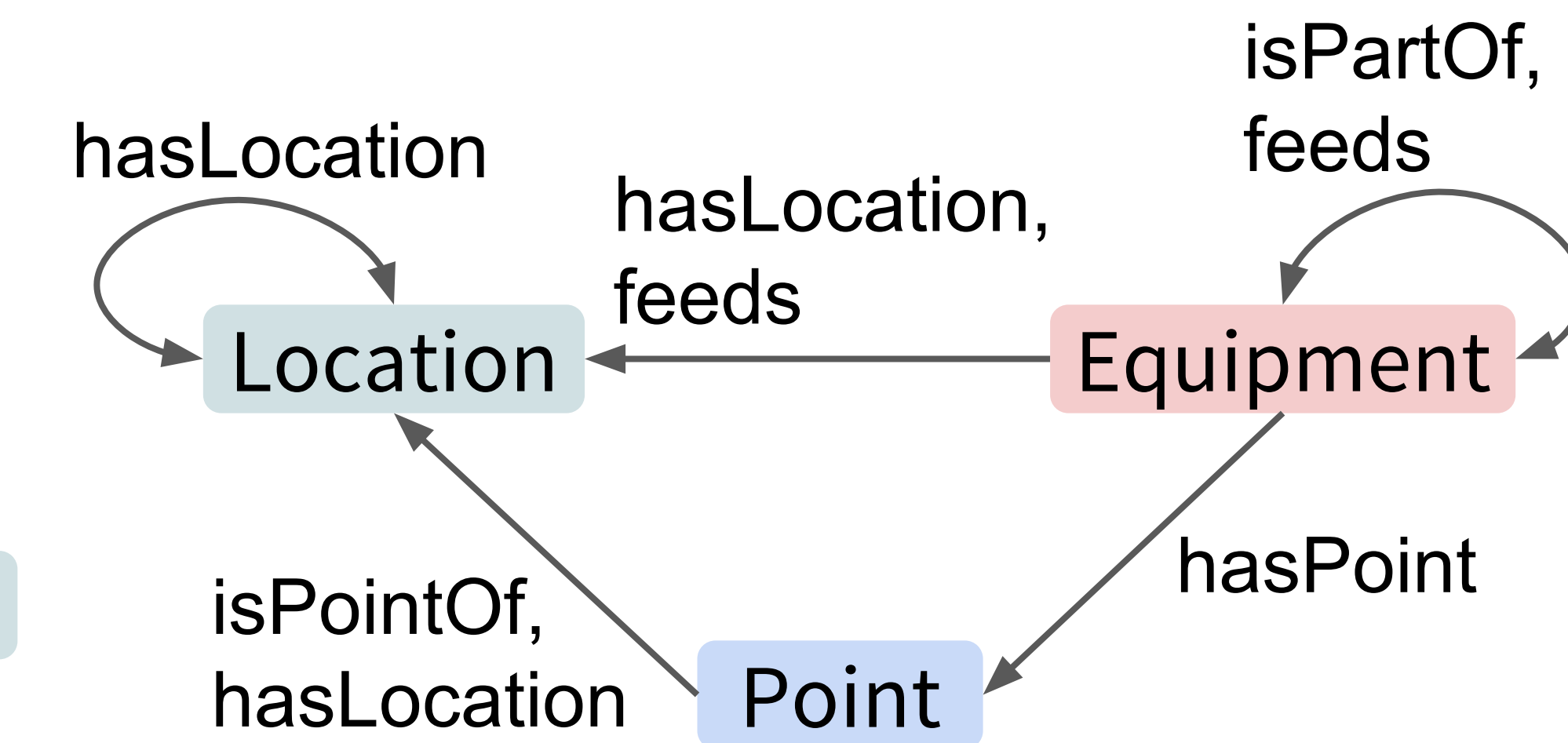
- A Tagset consists of Tags
  - *Room Temperature Sensor* consists of *Room*, *Temperature* and *Sensor*
- Tagsets constitute a taxonomy
  - Four major categories: Equipment, Point, Location, Measurement
- An entity is an instantiation of a Tagset
  - RT2 is an instantiation of *Room Temperature Sensor*.
  - Entities have relationships between them.
    - RT2 *hasLocation* Room-101.

Relationship	Inverse	Transitive?
hasLocation	isLocationOf	Yes
controls	isControlledBy	No
hasPart	isPartOf	Yes
hasPoint	isPointOf	No
feeds	isFedBy	Yes
hasInput	isInputOf	No
hasOutput	isOutputOf	No

## Tagset Class Hierarchy



## Relationships



## Example Building

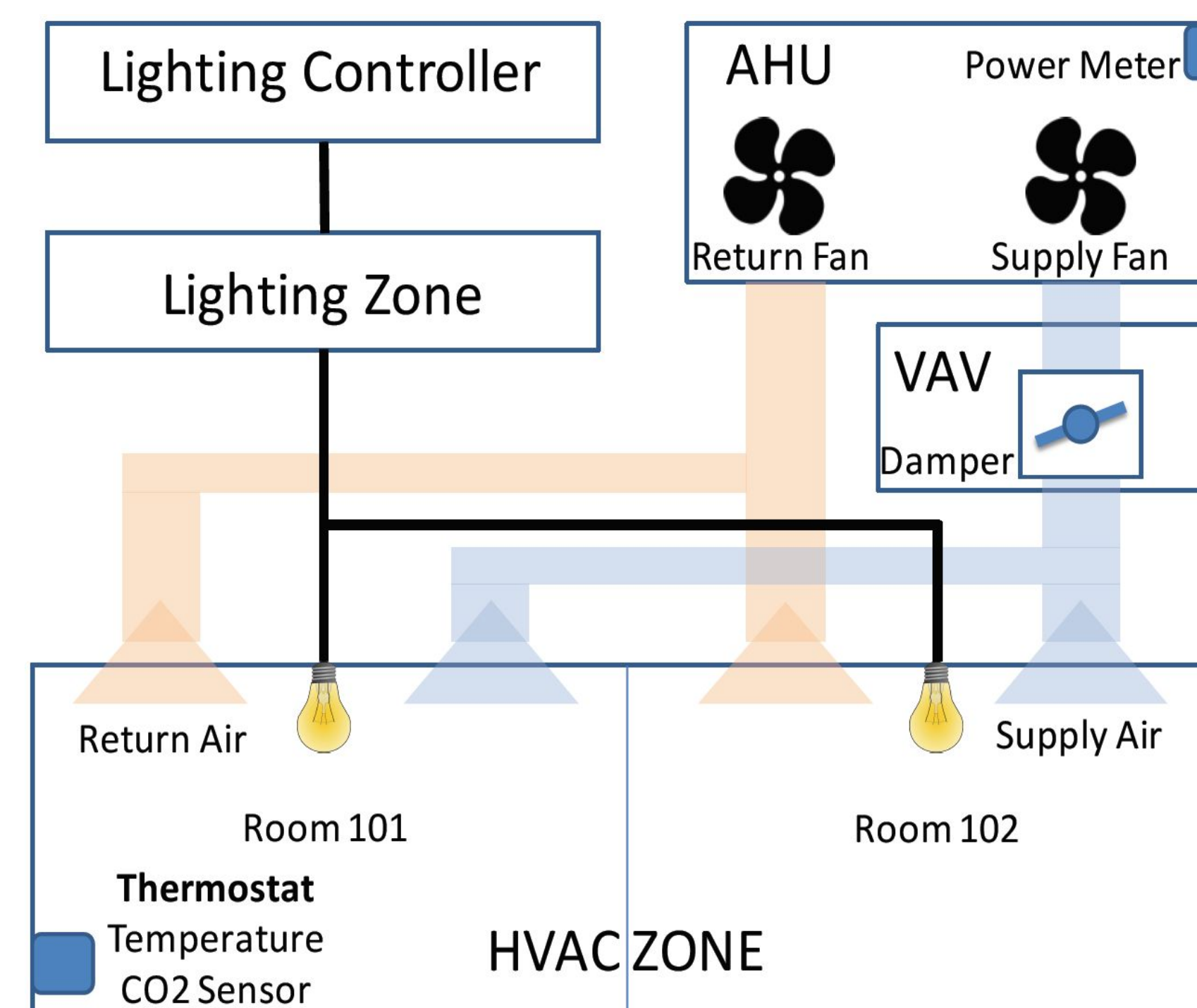


Fig. 2. Example Building Diagram

### Brickify

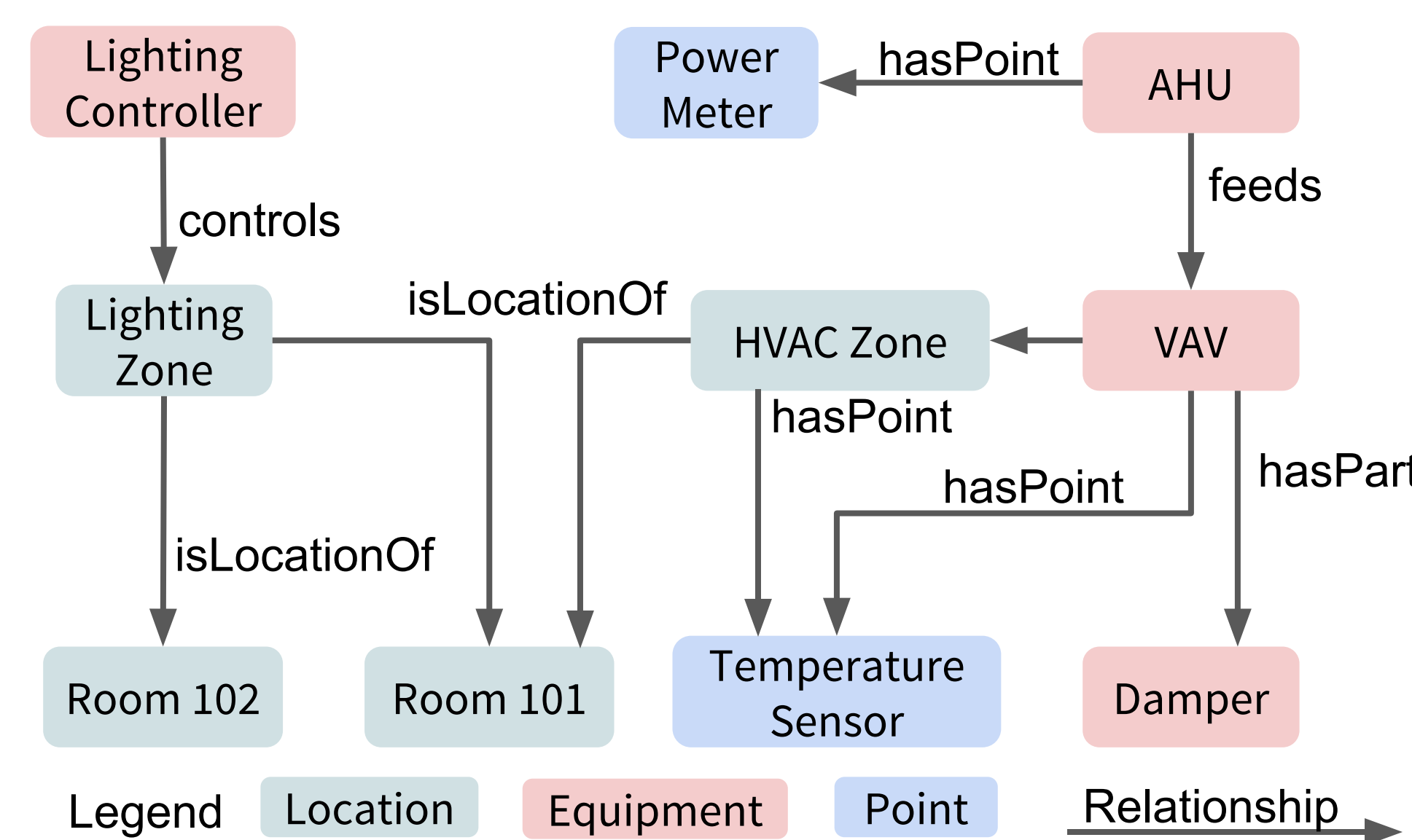


Fig. 3. Brick Representation of the Example Building

## Turtle Example

```
@prefix ex: <http://example-building.com#>
@prefix brick: <http://brickschema.org#>
```

```
ex:temp1 a brick:RoomTemperatureSensor
ex:temp1 brick:hasLocation ex:room101
ex:room101 a brick:Room
ex:vav1 a brick:VAV
ex:vav1 feeds ex:room101
ex:vav1 brick:hasPoint ex:temp1
```

Subject      Predicate      Object

## SPARQL Example

```
SELECT ?ahu ?room
WHERE {
  ?zone rdf:type brick:HVAC_Zone .
  ?room rdf:type brick:Room .
  ?ahu rdf:type/rdfs:subClassOf* brick:AHU .
  ?ahu brick:feeds+ ?zone .
  ?zone brick:hasPart ?room .
}
```

## Brick vs Haystack

Brick	Haystack
6 ported buildings, 8 ported applications	No reference implementation
Supports SPARQL queries that traverse Brick graph	Restrictive query mechanism cannot traverse relationships
Captures relationships within and across building subsystems	Can link entities, but does not classify these relationships
Functional blocks encapsulate complex subsystems; Tagset hierarchy permits partial descriptions	Flat, classless tag structure cannot capture uncertainty or complexity