Towards Programmable Smart Buildings

Dr. Gabe Fierro

Assistant Professor, Computer Science Department, Colorado School of Mines

Introduction

- Working in smart buildings/IoT/CPS space for ~10 years
- Computer Science PhD, UC Berkeley, 2021
  - Thesis: *Self-Adapting Software for Cyber-Physical Systems*
- Active ASHRAE member:
  - Regular tutorials at ASHRAE conferences
  - Member of Semantic Interoperability Working Group
  - Core author of new ASHRAE 223P proposed standard
- Co-founder and current tech lead of Brick Schema ontology effort
- Asst Prof in CS Dept at CO School of Mines
- Researcher in Commercial Buildings Research Group at NREL

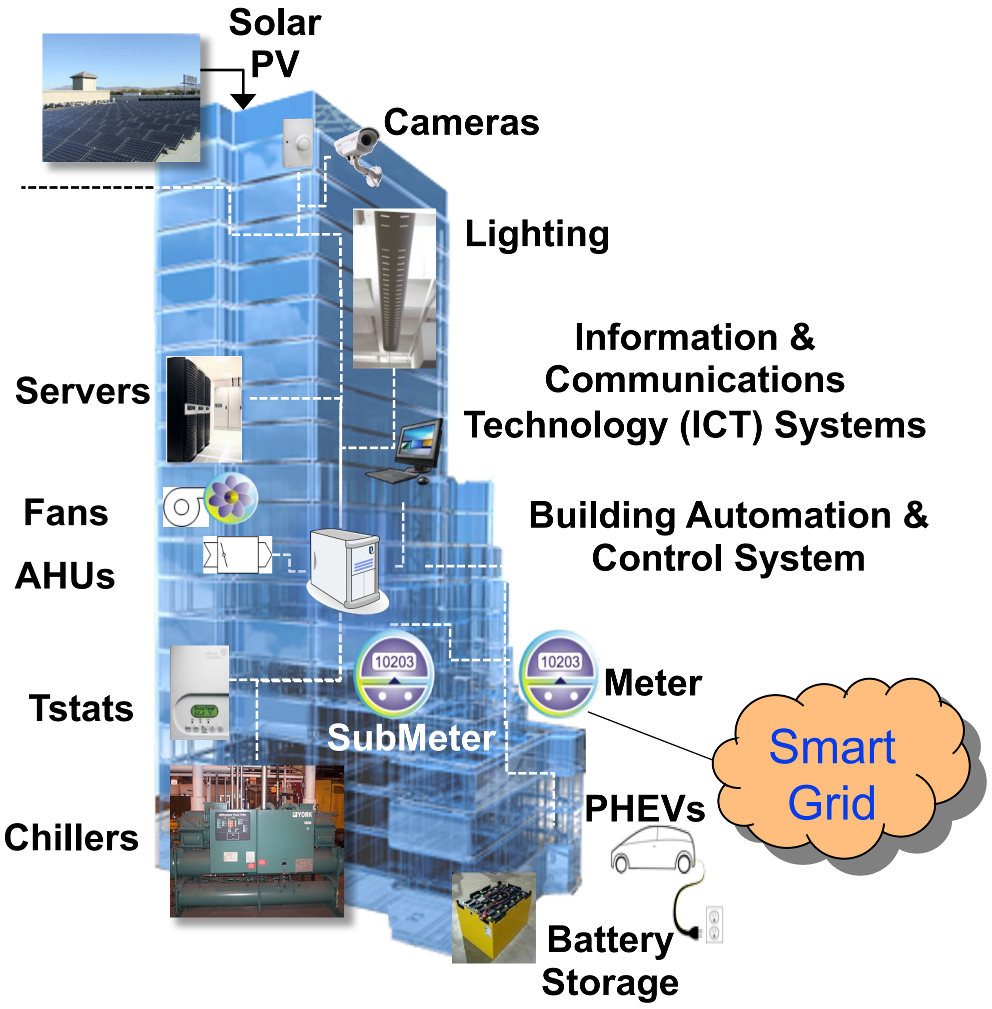
Brief Outline

- Setting the stage: current issues in buildings
- New standards and opportunities
- Unified vision of Programmable Smart Buildings
- Current applications and use cases
- Visions of future Programmable Smart Buildings

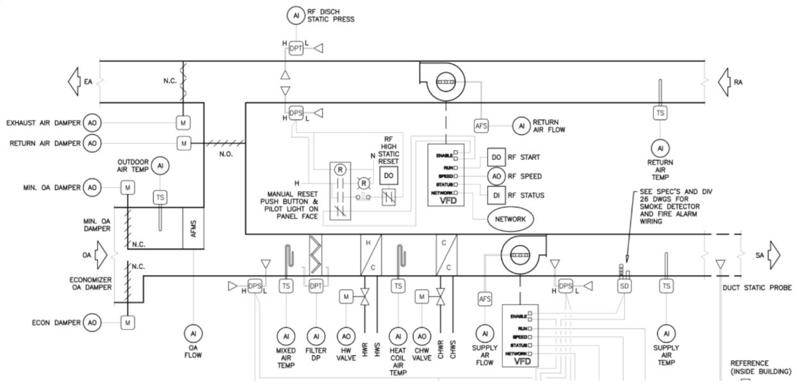
- *Most of my experience is in the commercial building space, but many of these problems generalize*

Setting the Stage

- Modern buildings have numerous subsystems, increasingly digitized
  - Many potential data sources, control points
- No end of emerging digital solutions across the building lifecycle
  - Design, construction, Cx, ops, auditing, maintenance, etc…
- CBECS survey (2018): superlinear adoption rate of digital technologies, but sublinear adoption rate of digital *applications*

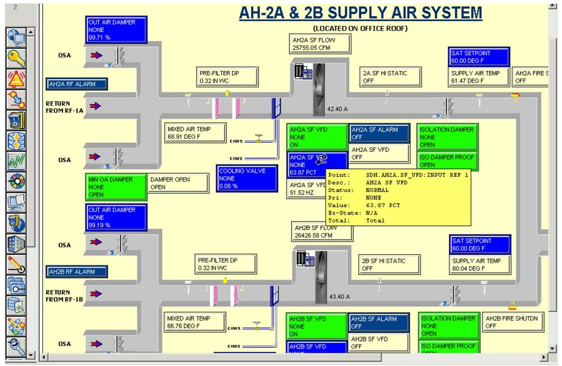


Interoperability Woes



Mechanical Diagrams: human-readable but non-standard

Facility managers, maintenance staff, and others hold implicit knowledge

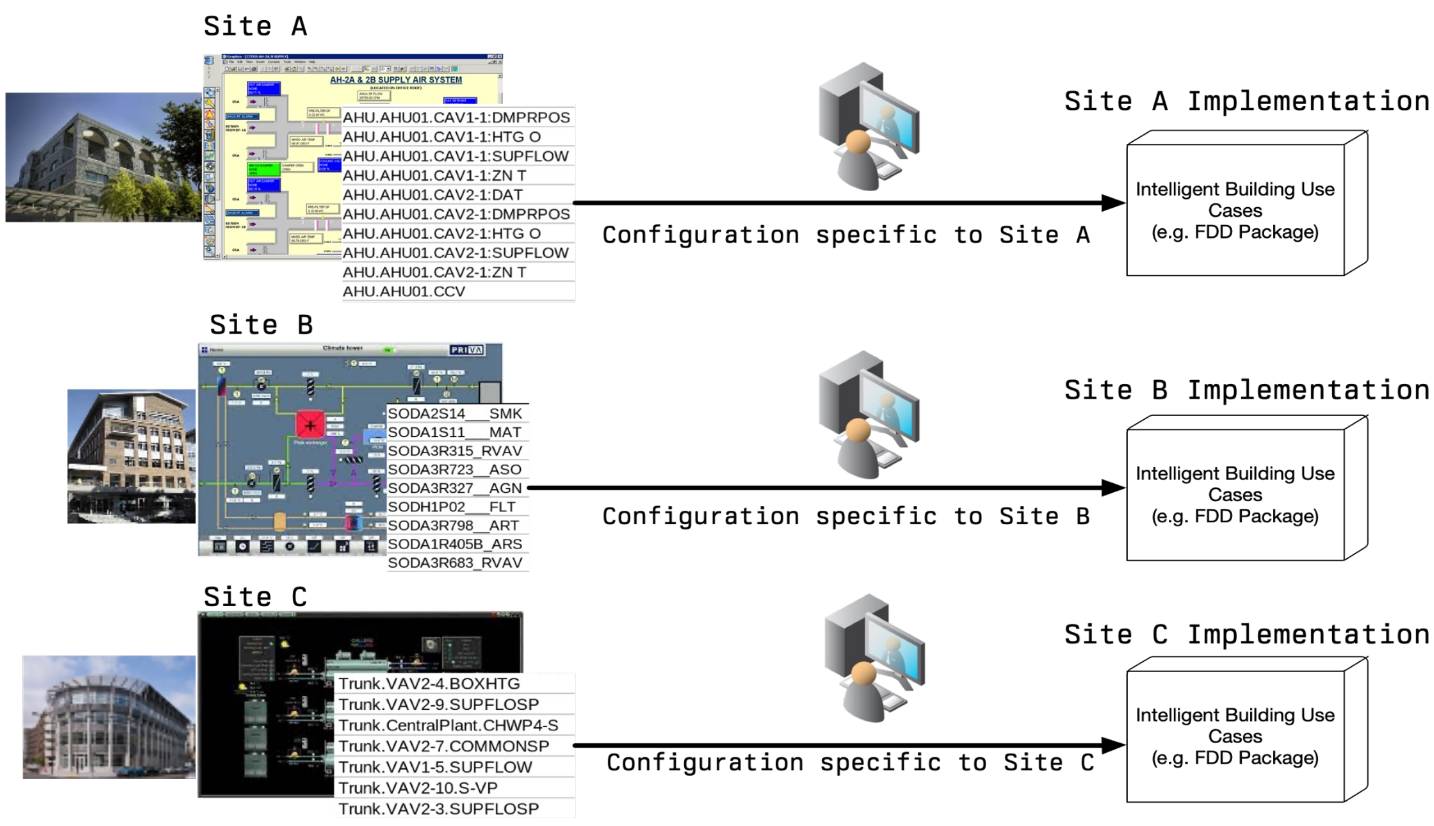


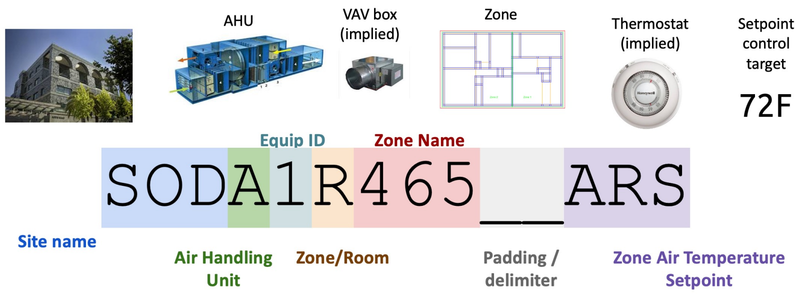
BMS labels and graphics: undocumented naming conventions

Lack of needed digital (and interoperable) digital retrofits

- Emerging data-driven applications like FDD, CX, AI/ML optimization need access to building design, data, control information
- If captured at all, this information is usually locked away behind silos (digital and administrative), may not be digitized, often incomplete/incorrect

Ending Bespoke Development?





Naming conventions and protocol "soup"

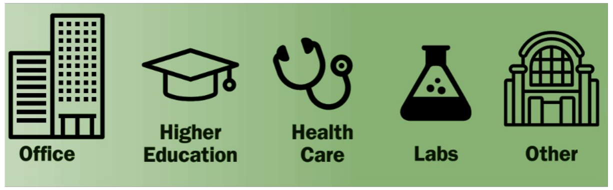
KNX LonWorks ASHRAE BACnet X10 Modbus enocean matter zigbee

- Lack of standardization, interoperability increases soft costs associated with developing and deploying data-driven solutions

Opportunities for Digitization of Buildings

- Software providers:

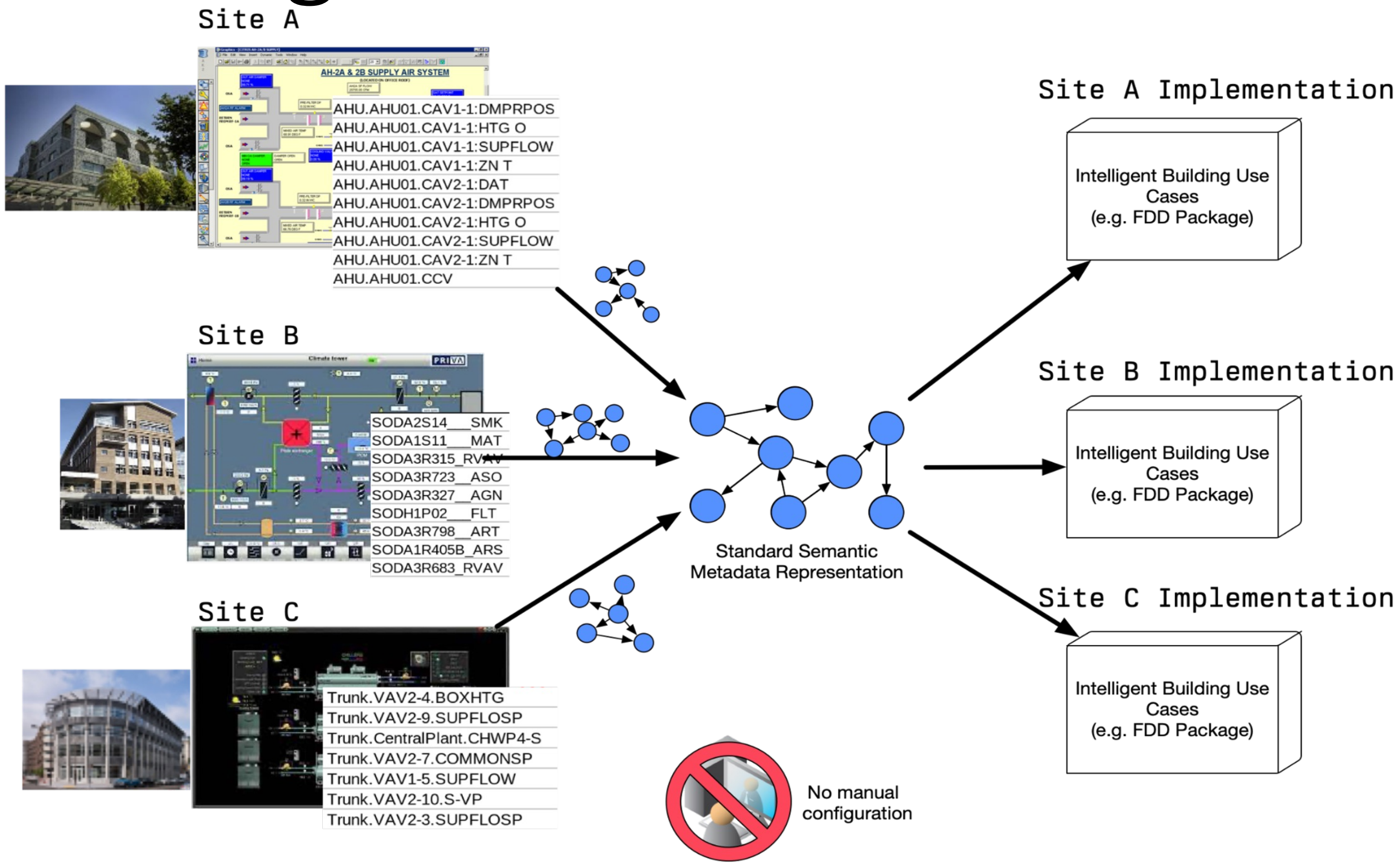
  - Reduce deployment (installation, configuration, etc) time and cost



- Property owners:
  - Enable easier procurement of intelligent controls, FDD, and more
  - Reduce vendor lock-in
  - Allows verification of software conformance

- Society:
  - Facilitate scalable deployment of analytics and controls to support the energy efficiency and decarbonization goals

Standardizing Semantic Metadata



- Standardizing digital representations of building can <u>lower</u> soft costs by removing site-specific configuration and development
- Active conversation in industry: what are these digital representations and how should they be used?

Semantic Metadata for Buildings

- **Digital** representation which can be **accessed** by applications
- **Semantic** metadata retains information that allows applications to understand building composition/data sources *consistently*

Project Haystack

Brick Schema

RealEstateCore

ETSI

SAREF / SAREF4BLDG

LBD

ASHRAE

ASHRAE 223P

- Growing ecosystem of solutions
  - Early solutions (Haystack) are a good start
  - Validation and consistency remain challenges
- Not all solutions are interoperable (yet)
- Variance between models built with same solution!

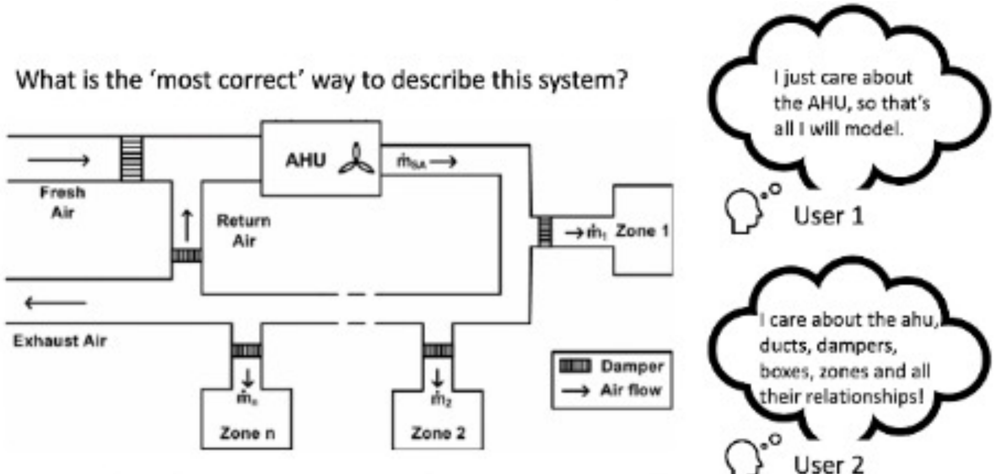
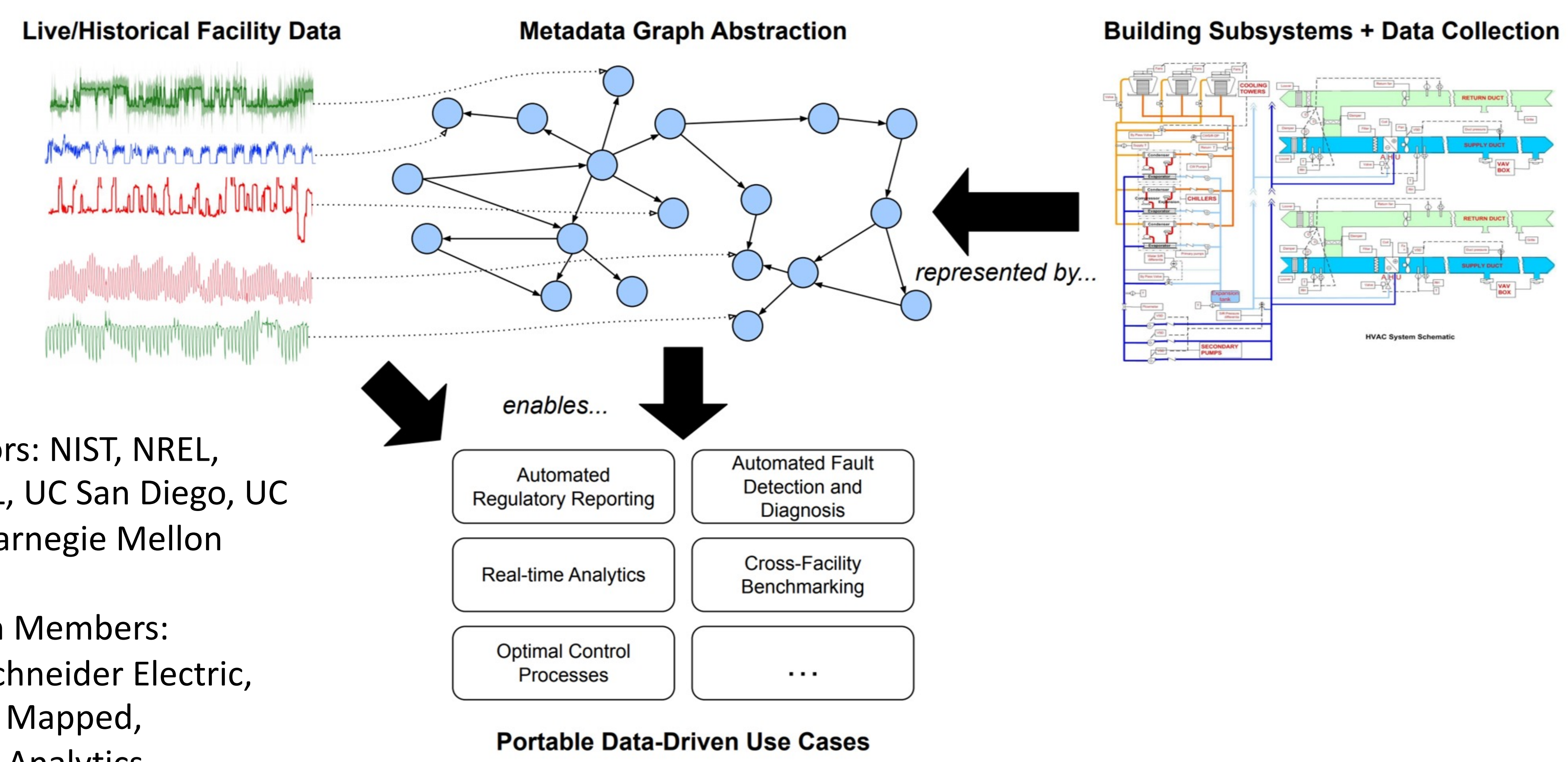


Photo credit: Nikitha Radhakrishnan, Token Based Scheduling of HVAC Services in Commercial Buildings

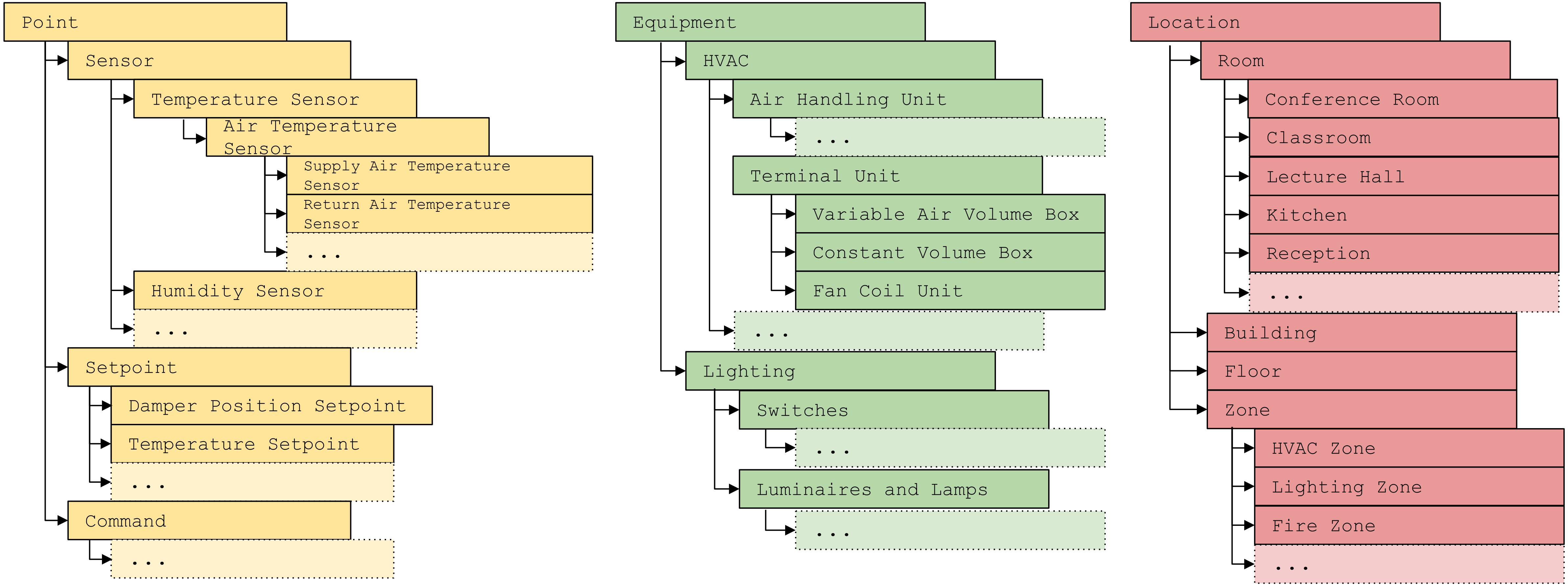
Brick: “Data Twin” for Buildings

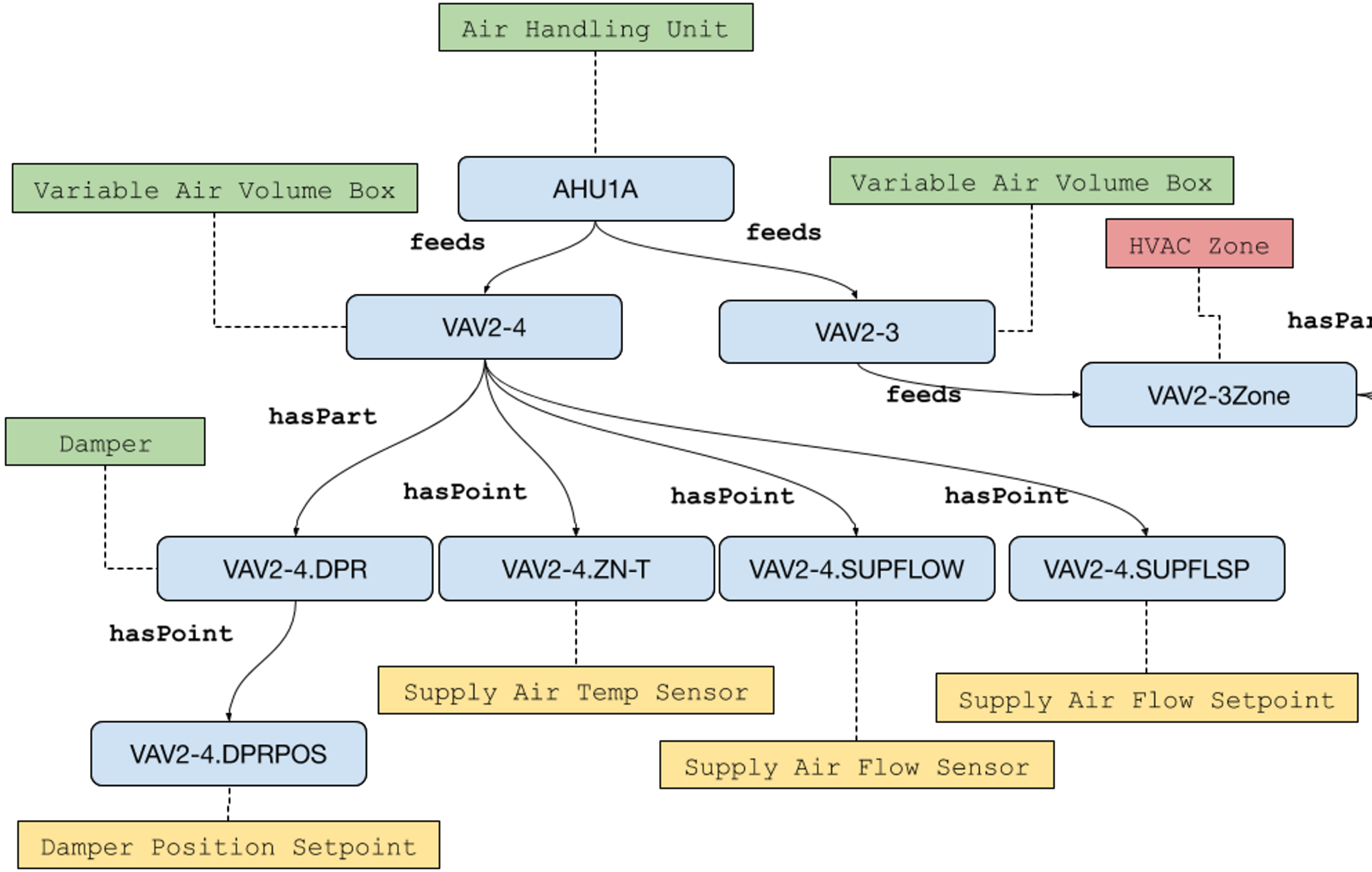


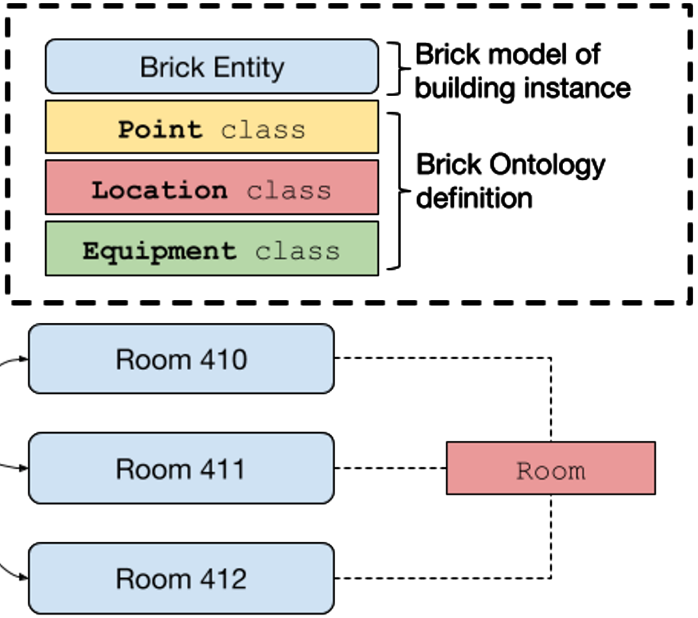
Collaborators: NIST, NREL, PNNL, LBNL, UC San Diego, UC Berkeley, Carnegie Mellon

Consortium Members: Siemens, Schneider Electric, JCI, Carrier, Mapped, Clockworks Analytics

Comprehensive, Extensible Taxonomy

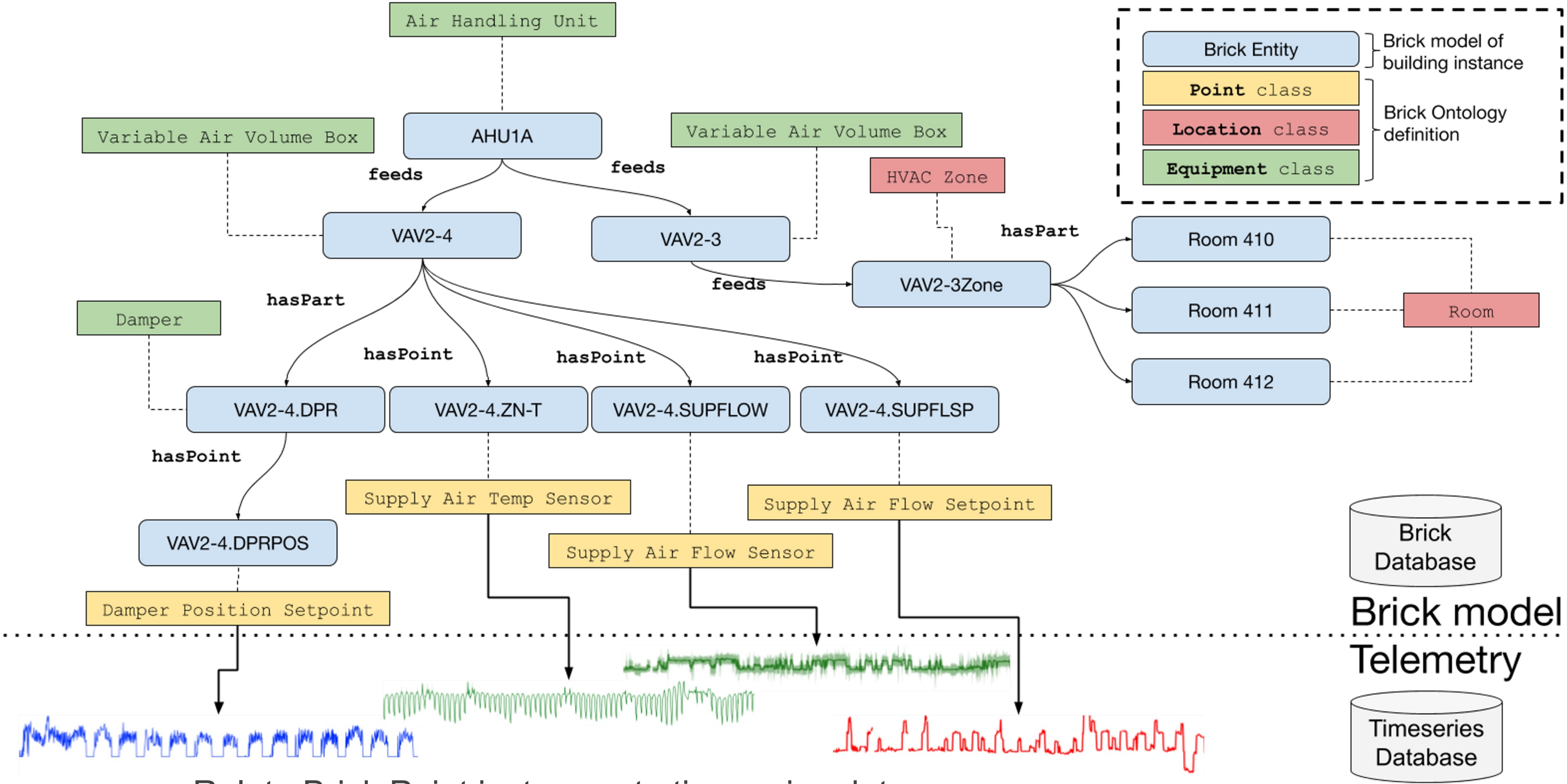






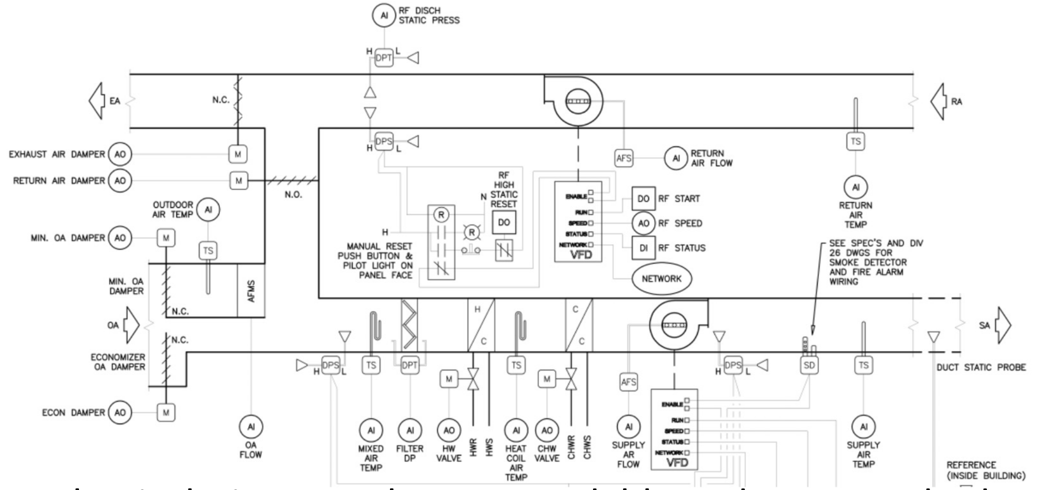
Brick ontology: formal definitions of concepts, relationships

Brick model: the graph representing a particular building

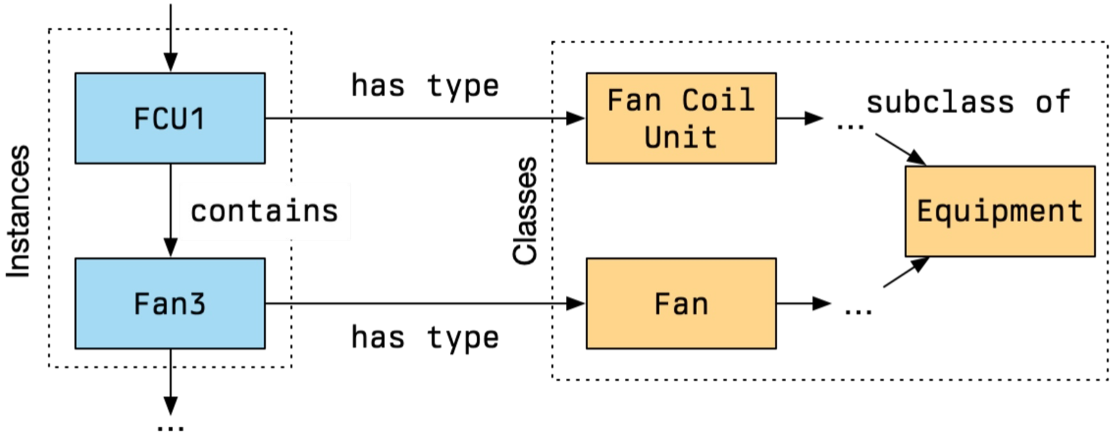


- Relate Brick Point instances to timeseries data
- Contextualize data in existing datastores

Building on Open Standards for Semantic Metadata Graphs



Mechanical Diagrams: human-readable and non-standard

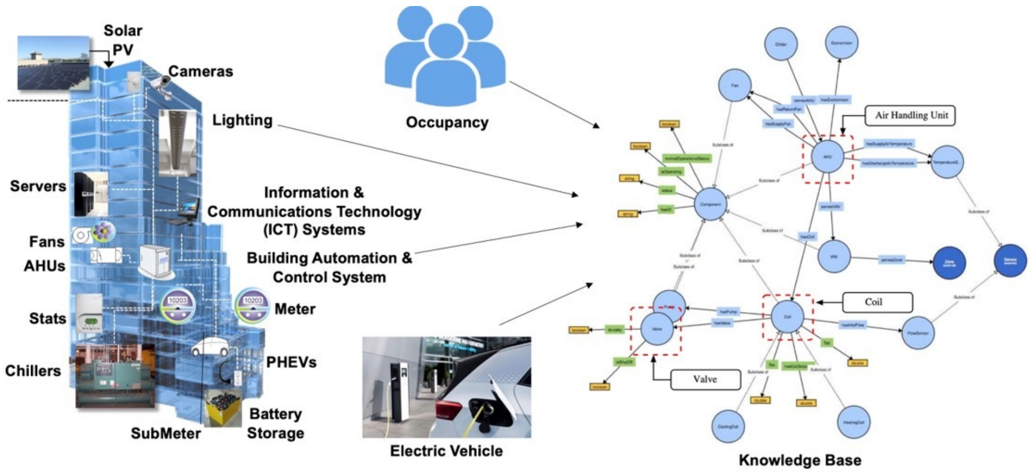


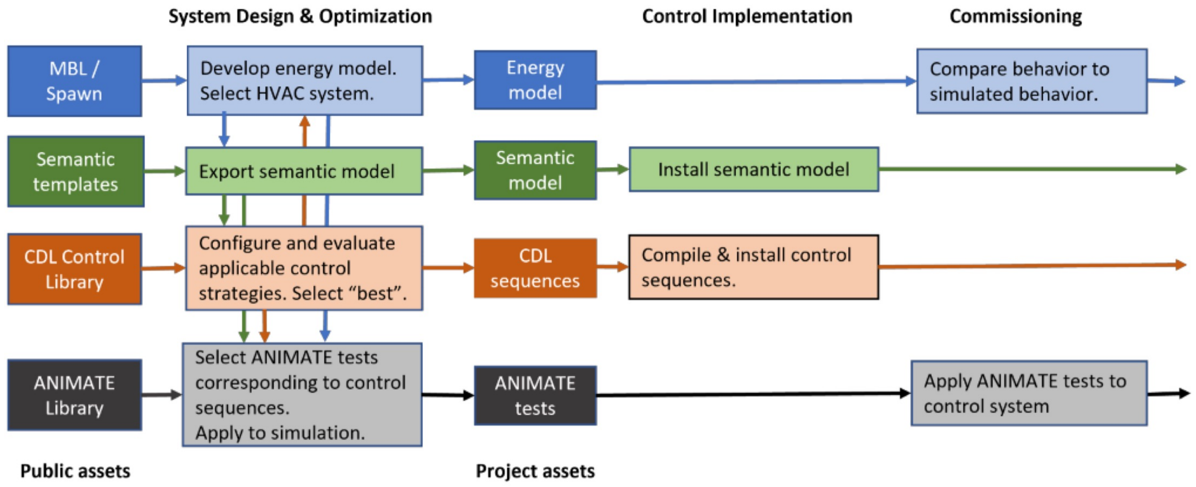
RDF Graphs: standard machine/human-readable models

- Build on **R**esource **D**escription **F**ramework (**RDF**)
  W3C standard for directed, labeled graphs
  - Tap into existing open-source and commercial ecosystems of tools
  - Supports sophisticated search and discovery
- **SPARQL**: Standard graph query language
  - Retrieve information from graphs
- **SHACL**: Constraint language for graphs
  - Enable automated validation of models
  - Acts as a “schema” for graphs

New Standards: ASHRAE 223P and 231P

- **ASHRAE 223P**: Designation and Classification of Semantic Tags for Building Data
  - Graph-based representation of buildings
  - More detail than Brick, Haystack, etc.
  - Models connections, devices, systems, sensors, …
- **ASHRAE 231P:** A Control Description Language for Building Environmental Control Sequences
  - Vendor-agnostic control sequences
  - Validate in simulation and easily deploy on the real thing

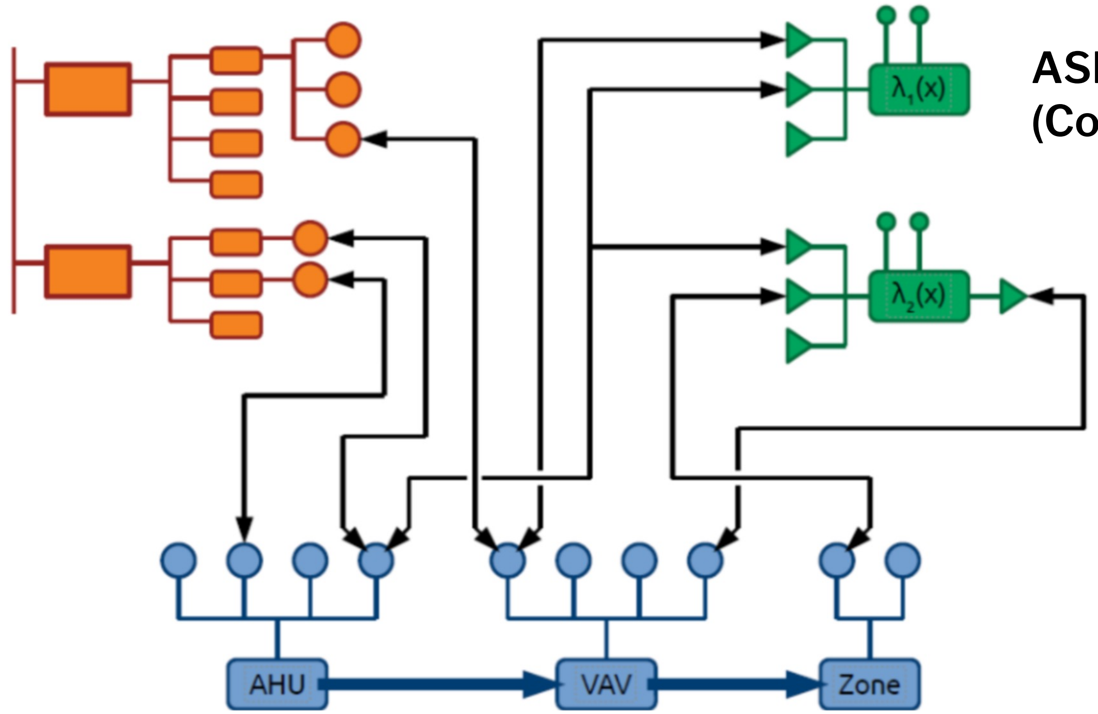




New Standards: ASHRAE 223P and 231P

ASHRAE 135: BACnet (Communication Protocol)

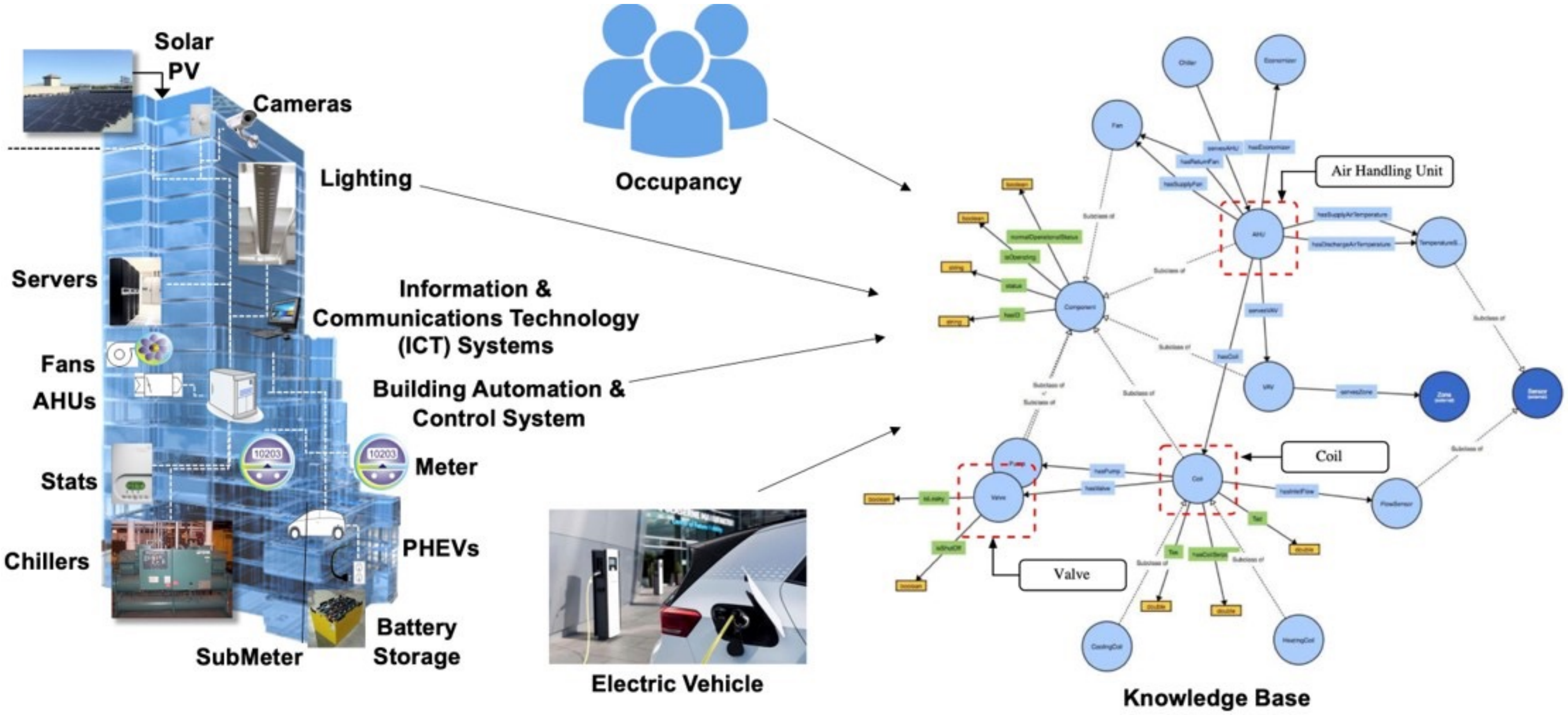
ASHRAE 231p (Control logic)



ASHRAE 223p (Meaning of the data)

Single data structure relating (1) structure/topology of all building subsystems, (2) the networking infrastructure to communicate with data sources, (3) the actual digital logic running the building

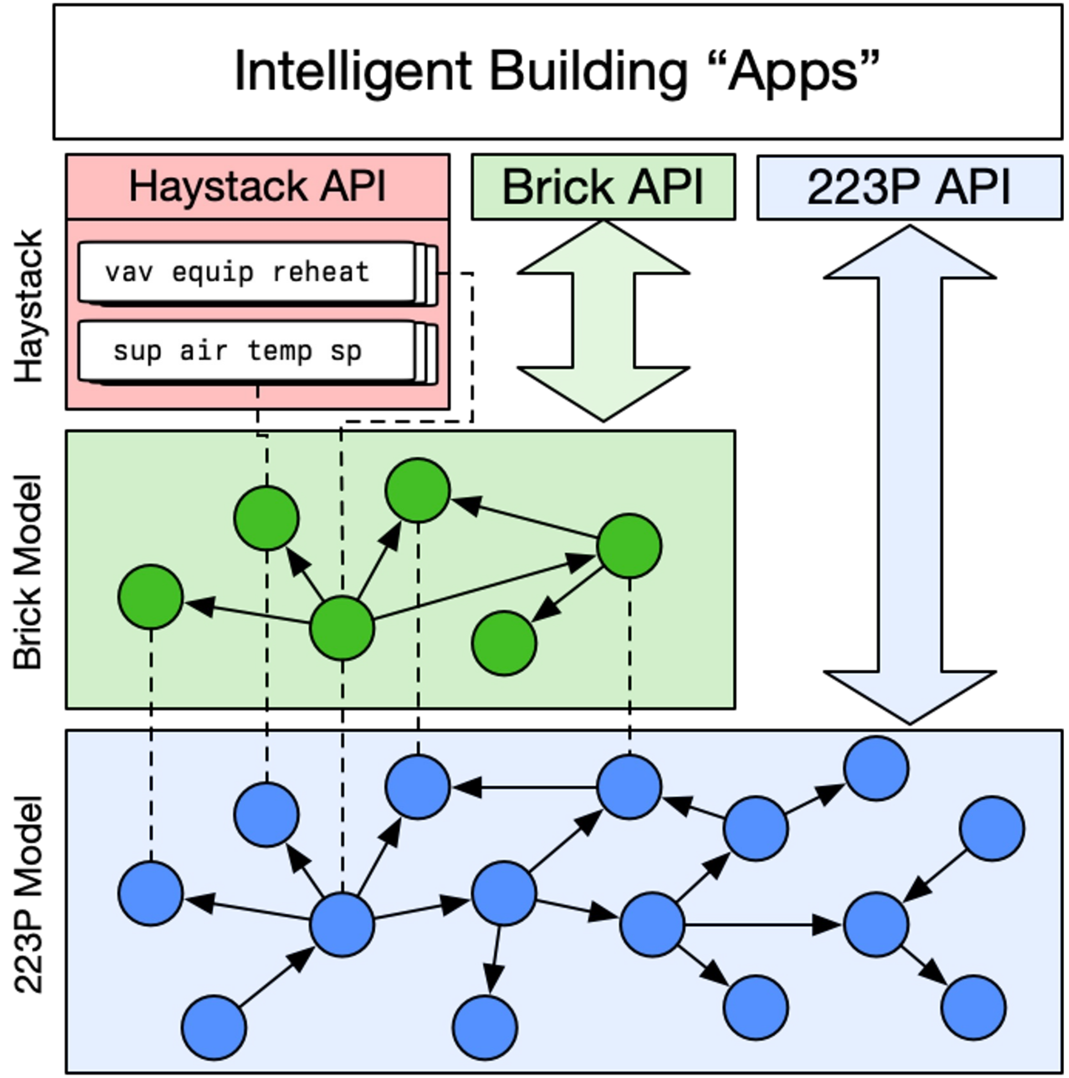
Semantic Metadata Enables Programmability



- Control
- Energy Auditing
- Fault Detection and Diagnostics
- Commissioning
- Smart Grid Interactions
- Dashboards
- **Arbitrary Applications**

Physical, logical systems ← represents ← **Semantic Model** ← access ← **Applications**

Layering Metadata Solutions in Semantic "Stack"

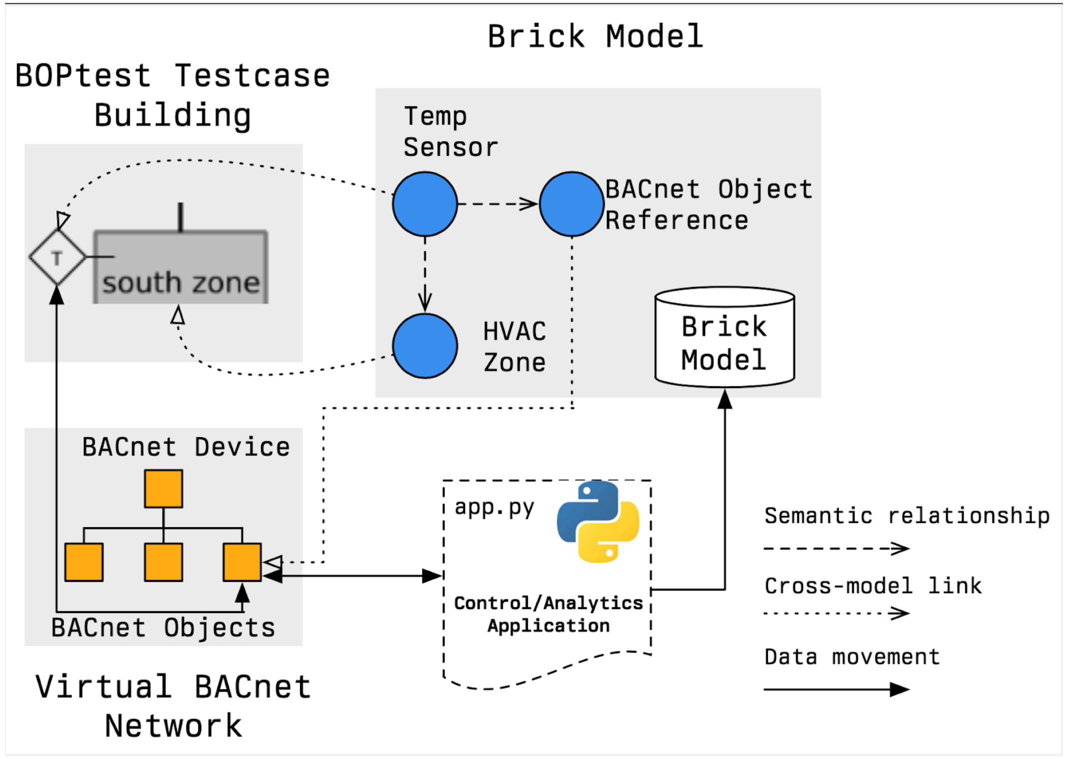


- "Best" model is relative to the applications' needs
- Go "up" the stack
  - More abstracted
  - Easier queries, but less precision
- Go "down" the stack
  - More detail, more formal
  - Queries can be more precise, but may be harder to write
- <u>Active research</u> to automatically derive Haystack, Brick from 223P

Future Opportunities

Connecting Semantic Metadata with Simulations

- **Challenge**: control testbed does not facilitate deployment of these algorithms in actual buildings
- **Solution**: layer virtual building network over the I/O points of simulation
- **Use Brick** (semantic metadata) to provide context over the simulation
- End result is an implementation-agnostic representation of building with realistic behavior



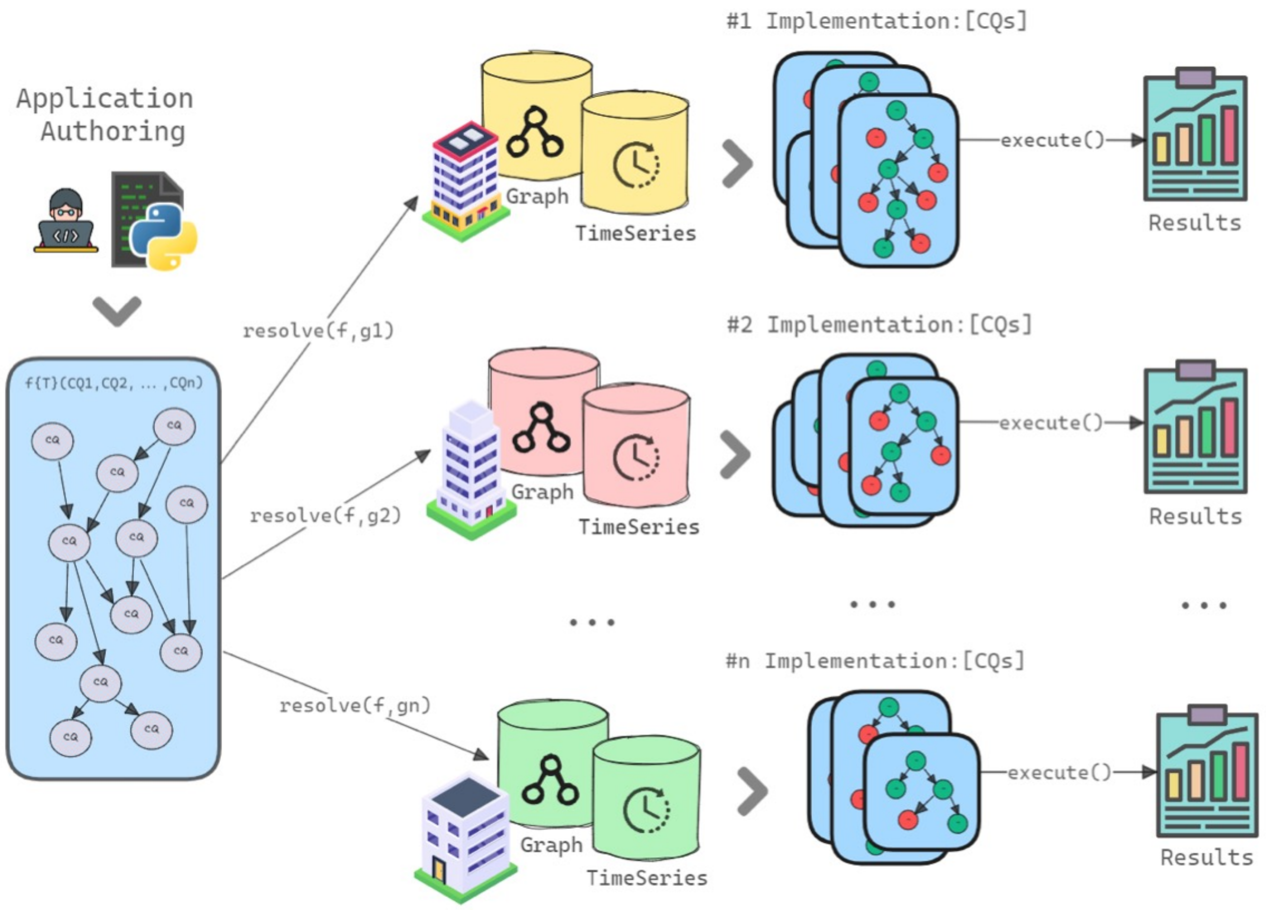
SeeQ: New Programming Model for Building Analytics

- Write Python applications against concepts defined by metadata ontology
- SeeQ “compiles” the Python code against the metadata model for each building
  - Generates building-specific impl.
- Demonstrated on FDD rules
- Step towards fully **portable** applications

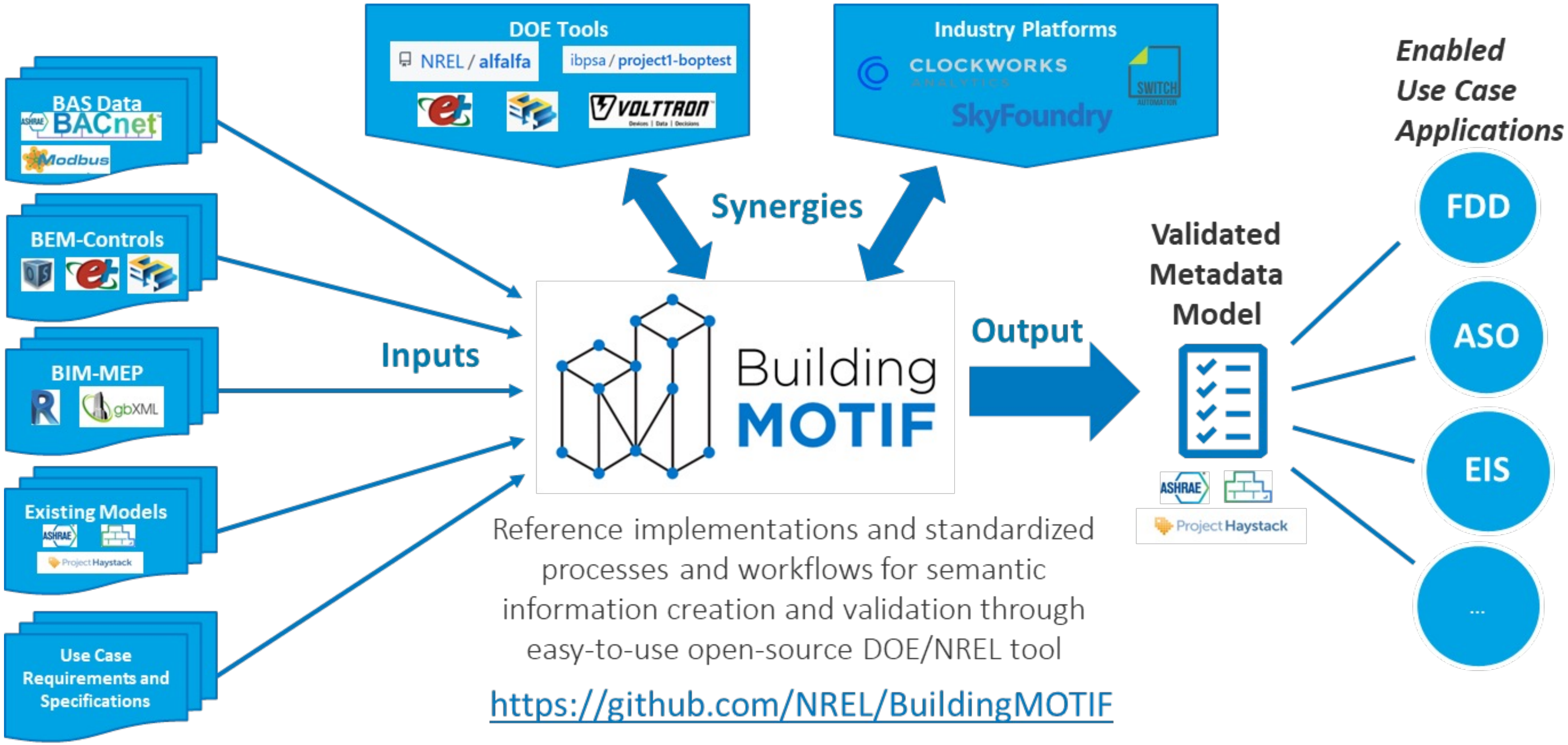
```
from SeeQ import *
from pandas import DataFrame
from G36.CQs import Dmp_Pos, Fsa, Fsp_clg, Fan_s
from APAR.CQs import Tsa, Tma, DelTsf, Hc_pos, Epsilon_t

def APAR_R1(sup: Tsa, mix: Tma, drop: DelTsf, heat_coil: Hc_pos, e: Epsilon_t):
    is_heating: DataFrame = heating_coil.df > 0
    supply_air_low: DataFrame = sup.df < (mix.df + drop.df - error.df)
    violating_records = is_heating & supply_air_low
    # returns fault if more than 10 violiating samples
    if len(violating_records) > 10:
        return "fault detected"

def G36_Dmp_Leaking(pos: Dmp_Pos, sup_flow: Fsa, cool_sp: Fsp_clg, fan: Fan_s):
    if ((pos.df == 0) and (sup_flow.df > max([0.1*cool_sp.df, 50]) \
    and (fan.df == "ON")).for_time(600):
        return "Level 4 alarm"
```

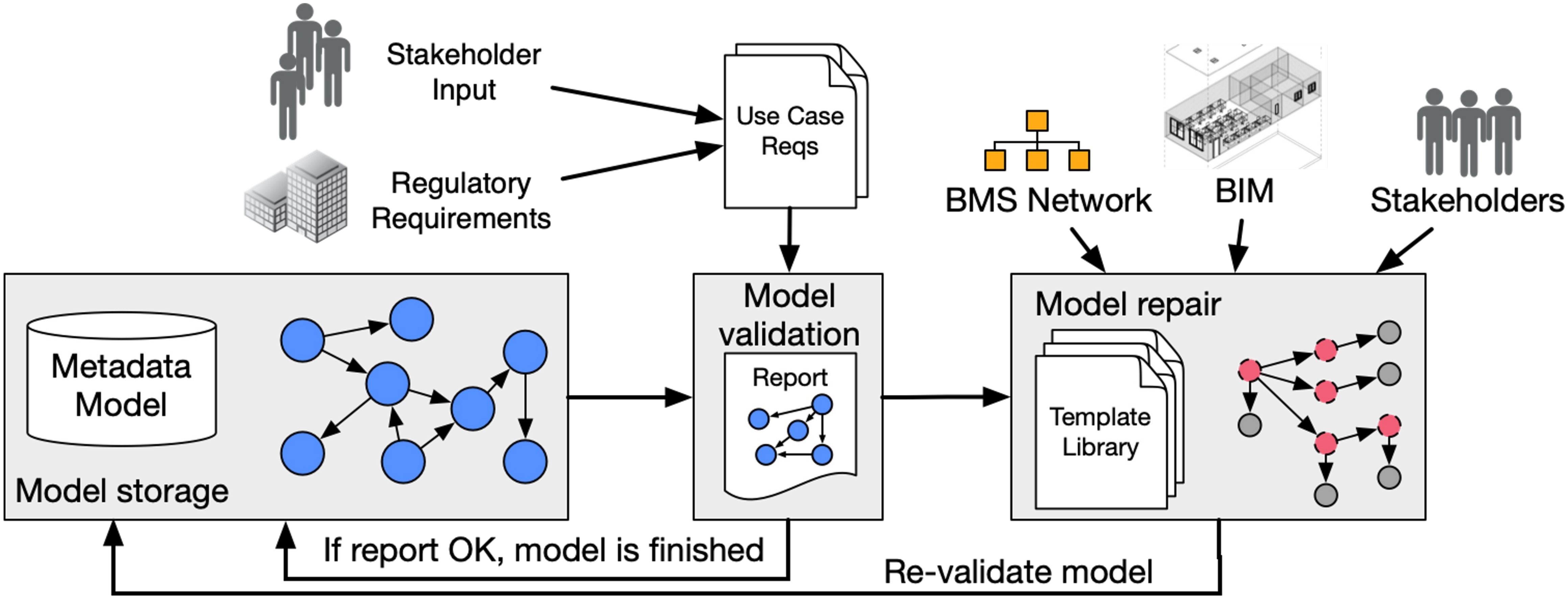


BuildingMOTIF: SDK to Support Semantic Metadata



- US Dept of Energy Building Technologies Office project, NREL led
- Use semantic metadata as “lingua franca” connecting existing tools for simulation, modeling, controls, AFDD, BIM, M&V, data science

BuildingMOTIF: Create and Validate Metadata



- Incorporate formal use case requirements into iterative workflow
- Ensure that delivered metadata model fulfills all use cases
- Automate / simplify authoring of models through templates, imports from other sources
- **Current work**: provide economic transparency on ROI for smart analytics

Future of Programmable Buildings

- Semantic metadata models are a powerful abstraction *underneath* more familiar developer-facing abstractions
  - Automated checking and validation of programs, models
  - Site-specific code generation for “portable applications”
  - Support data warehousing for downstream analytics; use queries to create necessary datasets as needed (“materialized views”)
  - Can replace existing ad-hoc ”device trees” for buildings

- Still lots of work to do!
  - Syntactic interop (RPC?): is Matter/Zigbee sufficient?
  - Better and higher-level programming models
  - Operating system / application platform / software development kits
  - Opportunities to leverage LLMs and emerging AI

Thank you!

- My current research/projects: https://gtf.fyi
  - Contains links to all the works I've mentioned in this talk
  - Most have an open-source GitHub repository associated with them

- Brick ontology project: https://brickschema.org

- ASHRAE 223P development: https://open223.info

- NREL-developed semantic metadata platform: https://github.com/NREL/BuildingMOTIF