Portable Building Applications

WoT Use Case

Gabe Fierro
Colorado School of Mines / NREL
gtfierro@mines.edu
https://gtf.fyi

Increasing Amounts of Data Available for Buildings

- Large sensor/actuators networks:
  - thousands, 10s of thousands of data streams and control points
  - Increasing #s of IoT devices available

- Opportunity for wide-scale data-driven analytics and control:
  - AFDD
  - Virtual metering
  - Performance measurement
  - Automated regulatory reporting
  - MPC and advanced controls/sequence-of-operations
  - Demand response

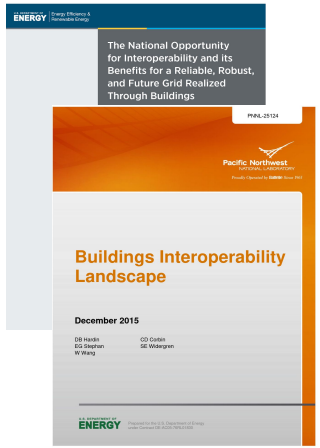
- **Issue**: complex, heterogenous configuration:
  - Every building is a unique, one-off collection of “data silos”
  - Implementing software, extracting data, defining control surfaces is **site-specific**, **error-prone** and **time consuming**

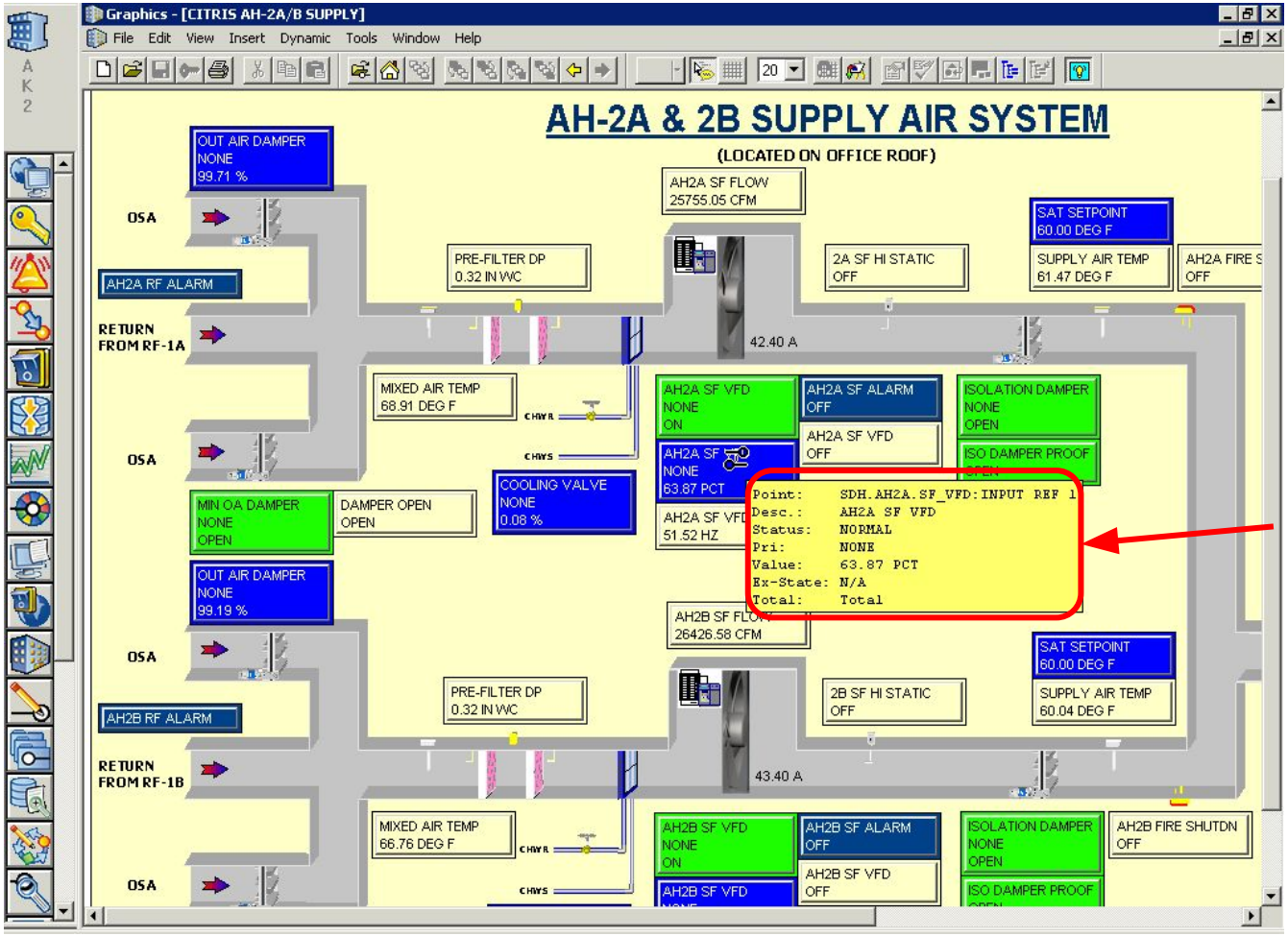
...but that data is increasingly hard to use

- *“Data-rich but information-poor”*
  - Plenty of phenomena, but none of the context

- Understanding this data is a significant bottleneck
  - ~40% of consultant/data-scientist time spent understanding metadata
  - Manual, time-consuming, error-prone, expensive task

- Costs impede adoption of sustainable practices
  - 2004 NIST report estimates lack of interop @ **$15.8 billion/year**
  - 2016 PNNL report: existing work focuses on connectivity, with data semantics still lacking
  - 2018 study: 70% of commercial buildings have digital controls, but only 4% have automated fault detection



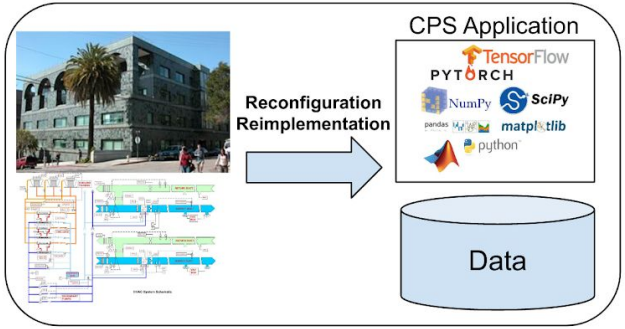


Point: a source of data from a cyber-physical system

10s or 100s of thousands of points in a typical system

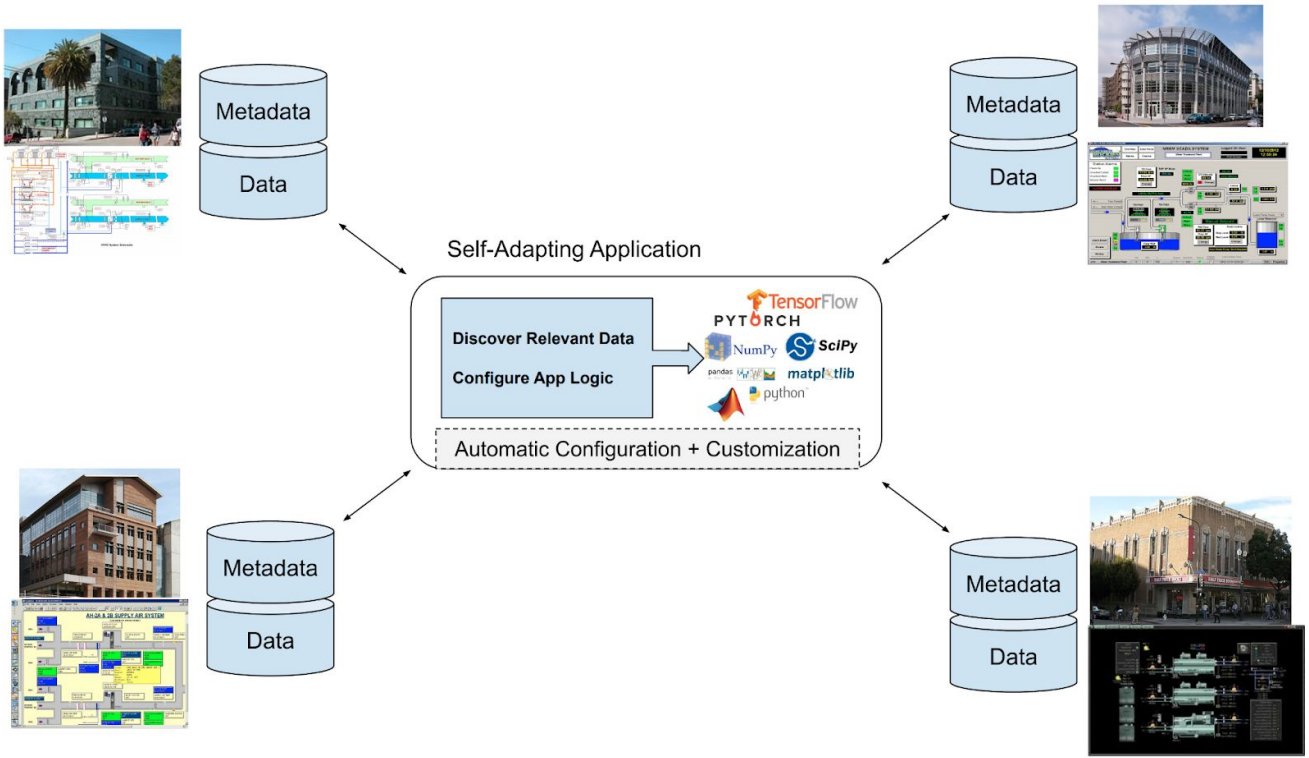
Building Management System (BMS): the “operating system” for your building

Deploying Cyber-Physical Applications at Scale



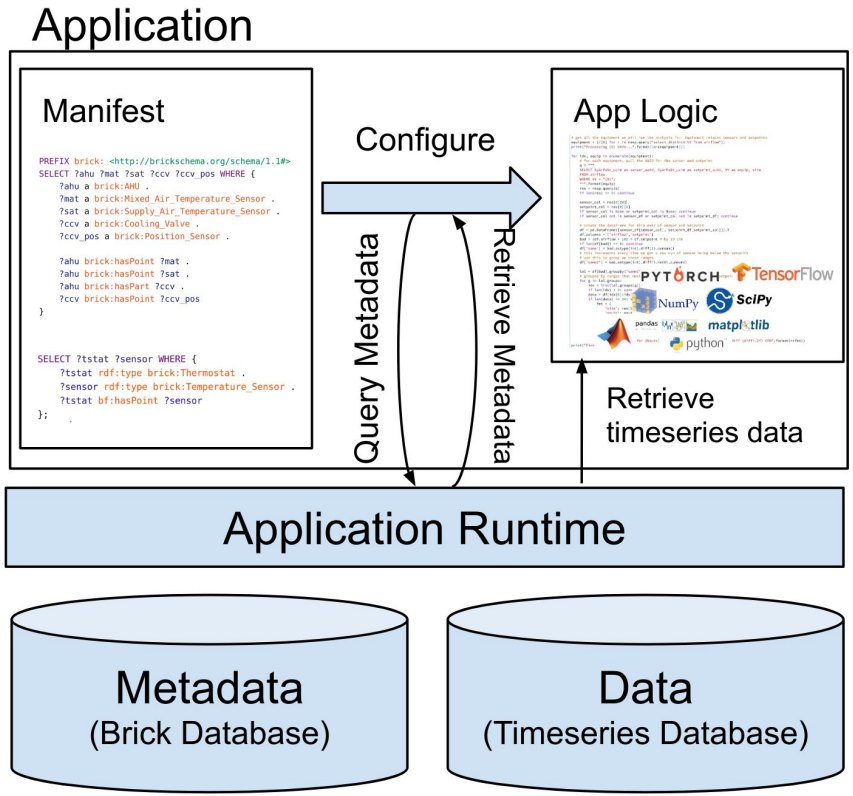
Current Practice: manual, time-consuming, error-prone reimplementation for each site

Deploying Cyber-Physical Applications at Scale



Goal: Write <u>one</u> implementation, run it everywhere without any* manual configuration

Architecture of a Self-Adapting Application



How WoT Fits In

- **Brick** ontology defines the data sources and their context w.r.t. the building subsystem

- Assumes the data is already in a regular format
  - Outsources the “lifting” to the data ingester

- Does not handle the discovery of device capability
  - Assumes this is already modeled

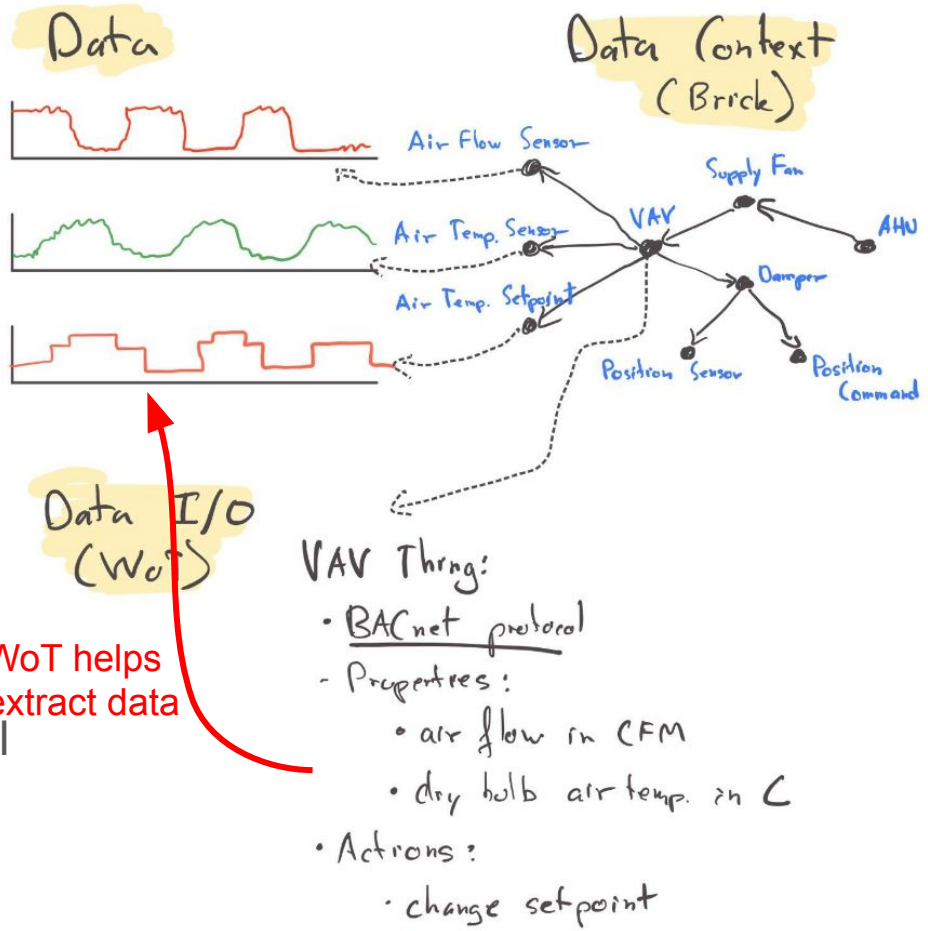
```
PREFIX brick: <http://brickschema.org/schema/Brick#>
SELECT ?term ?zone ?sat ?sp WHERE {
    ?term   a   brick:Terminal_Unit .
    ?zone   a   brick:HVAC_Zone .
    ?sat    a   brick:Supply_Air_Temperature_Sensor .
    ?sp     a   brick:Supply_Air_Temperature_Setpoint .

    ?term   brick:feeds     ?zone .
    ?term   brick:hasPoint  ?sat, ?sp .
}
```

How WoT Fits In

- It is not enough to model the capability of a device
  - Where* is the device?
  - What is it connected to?
  - What do its actions affect?

- WoT standardizes repr. of/access to the I/O capabilities of the IoT devices providing/receiving data

- Abstract over (legacy) industrial control protocols:
  - BACnet, OPC, LonTalk, Modbus, ...



Future Brick Work? *(The work we don't want to do…)*

- Starting to think about the "network/instrumentation view":
  - How are devices connected; how do they communicate?
  - Joel Bender (Cornell, BacPypes, ASHRAE, 223P) working on SHACL representation of BACnet objects

- Standard enumerations of device properties:
  - Thermostat modes, staged equipment, etc
  - Aid in data interpretability

- Schedules of operation:
  - Investigating schema.org-style schedules; RRULE; etc
  - Want to express device behavior in terms of schedules (nighttime setbacks, etc)