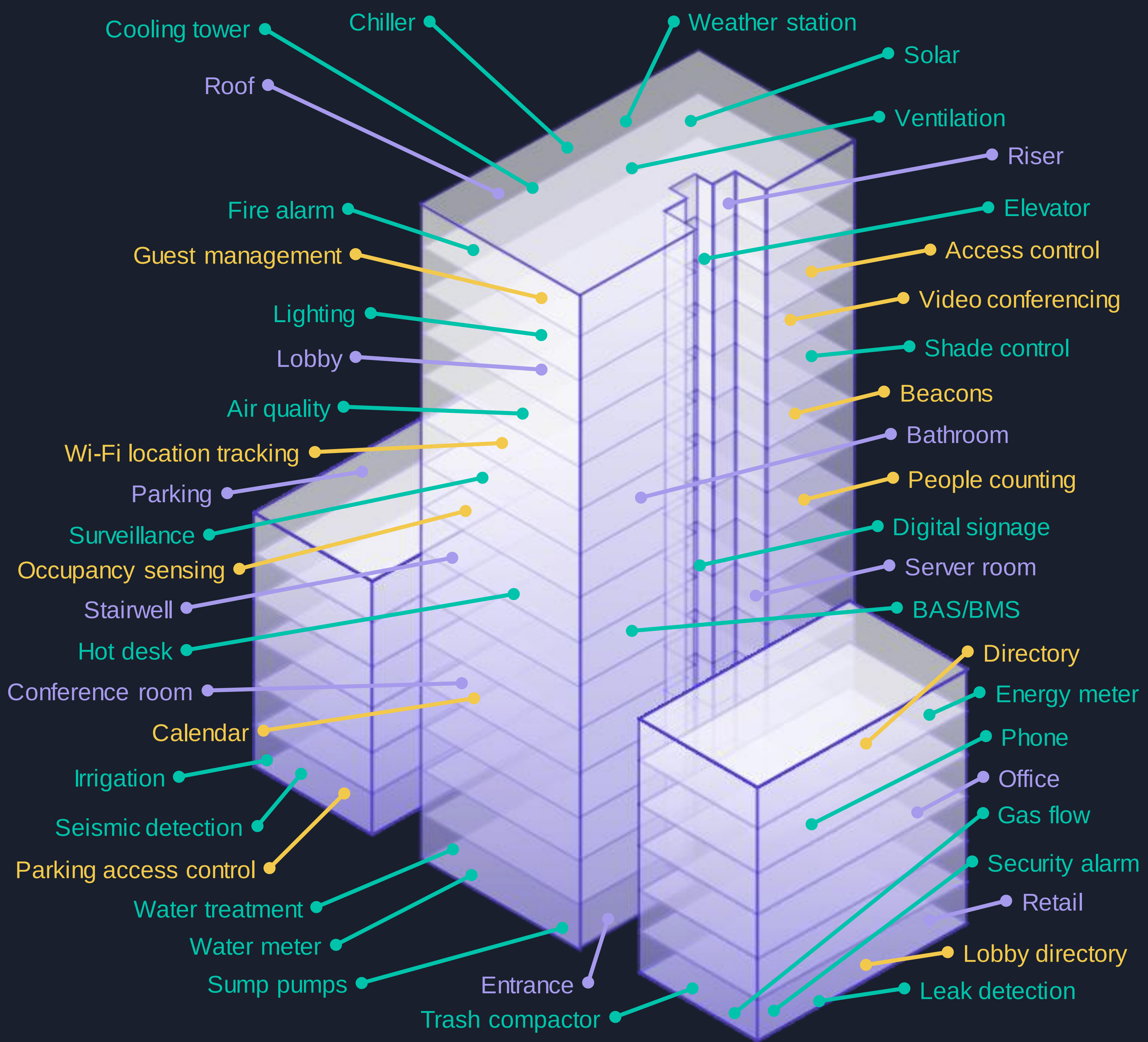
What does it Really Take to be an **Independent Data Layer**?

Dr. Jason Koh, Chief Data Scientist, Mapped
jason@mapped.com

Millions of data points

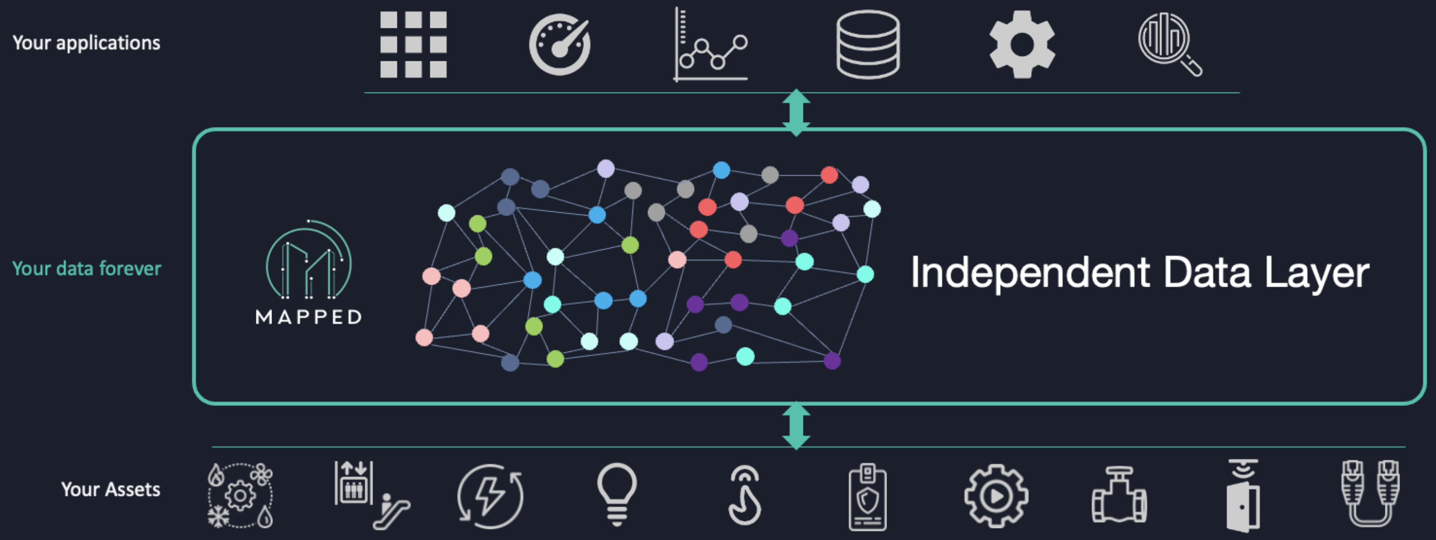
One Independent Data Layer

MAPPED



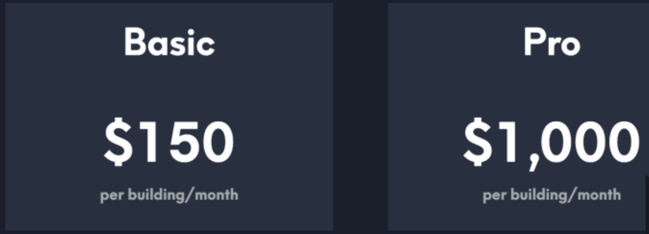
What is Independent Data Layer (IDL)?

- IDLs abstract heterogeneous data sources for the end users. Applications can offload the integration effort. Apps can be independent to the data sources.
- Gives owners control over who can access what subset of their data (e.g., integrators)
- An IDL makes data available to any application via a standard, open interface.



Common Misconception about IDLs

- IDLs are expensive



Basic

$150

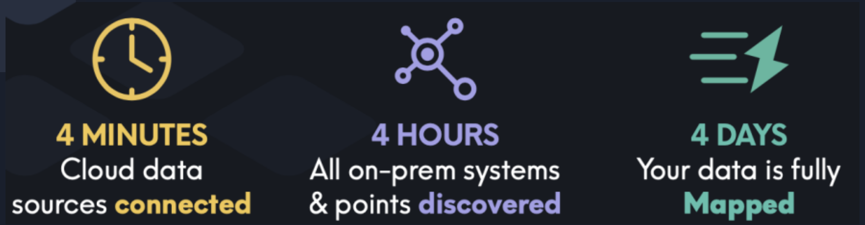
per building/month

Pro

$1,000

per building/month

- IDLs are slow to deploy



4 MINUTES
Cloud data sources connected

4 HOURS
All on-prem systems & points discovered

4 DAYS
Your data is fully Mapped

- IDLs are redundant

- IDLs add complexity
- IDLs are not my concern, but somebody else's.

IDLs Should Be

- Independent from both equipment and solution vendors
  - Anybody can benefit from an IDL

- Agnostic to data sources
  - Make deployment fast, scalable, inexpensive
- Using standard ontologies
  - Remove redundancy, reduce complexity by abstractions

Project Haystack

Brick

RealEstateCore

Who benefits from an IDL?

DATA CONSUMERS

CHALLENGES AND NEEDS

1st Party

BUILDING OWNERS & CORPORATE TENANTS

- Large portfolios
- ESG
- Gov't compliance
- Operations
- Monitoring
- Maintenance
- Facilities management
- Asset disposition
- Cost allocation
- Indoor air quality
- Space planning

Market Examples:

PROLOGIS SIMON FORTUNE 500 LaSalle INVESTMENT MANAGEMENT
DukeREALTY PGIM FORTUNE 1000 Inc. 5000
Blackstone GGP nuveen

2nd Party

SERVICE PROVIDERS, CONTRACTORS, OPERATORS

- Huge customer bases
- Margin optimization
- Insurance: Liability, equipment, crime, fire, ...
- Energy service companies
- Facilities, operations, and maintenance
- Property & investment management

Market Examples:

ABM Building Value aramark CUSHMAN & WAKEFIELD
Munich RE AECOM ECOLAB mitie
CBRE sodexo Liberty Mutual INSURANCE

3rd Party

PROPTECH SOLUTIONS & SOFTWARE VENDORS

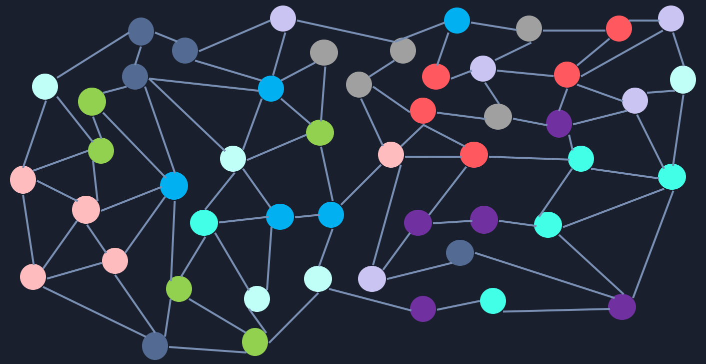
- Replace slow human-based onboarding with one-click deployments
- Normalized access to systems across all deployments
- Vendor agnostic data
- Focus on innovation rather than integration
- Recommend vendors for missing sensors

Market Examples:

Microsoft intel Johnson Controls SWITCH AUTOMATION wipro
amazon BuildingEngines aquicore
Honeywell Capgemini accenture
SIEMENS KONE willow

Dataflow in an IDL

CONNECT
Discover & Extract

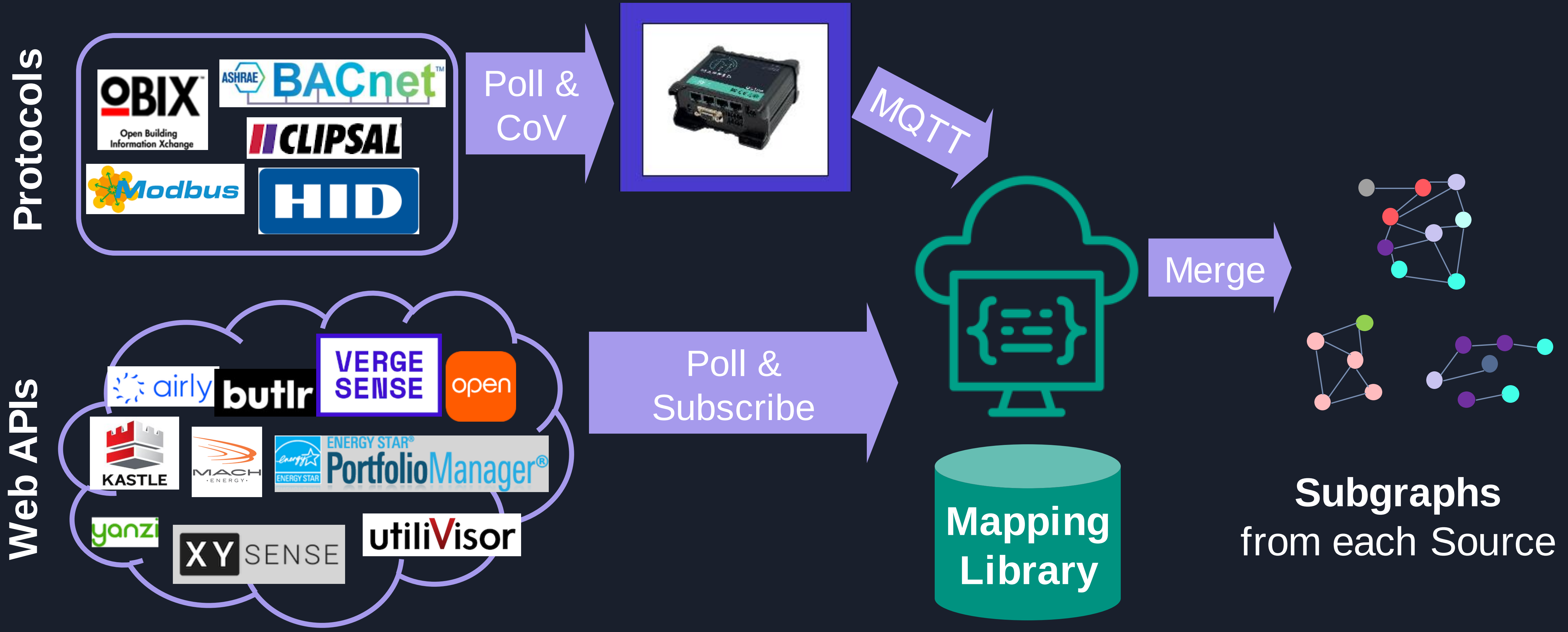


MAP
Data normalized to BRICK

```
{
  building(id:"175A7C19") {
    floors(index: "3") {
      spaces {
        id
        name
        geoshape
      }
    }
  }
}
```

ACCESS
Solutions & Apps

Data Source Connection



Protocols

Web APIs

Poll & CoV

MQTT

Poll & Subscribe

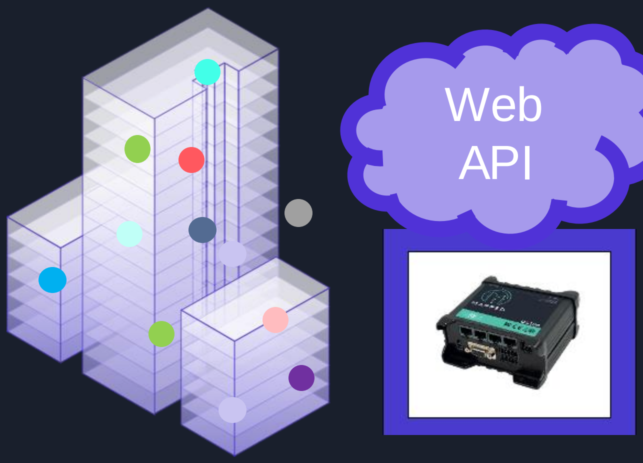
Merge

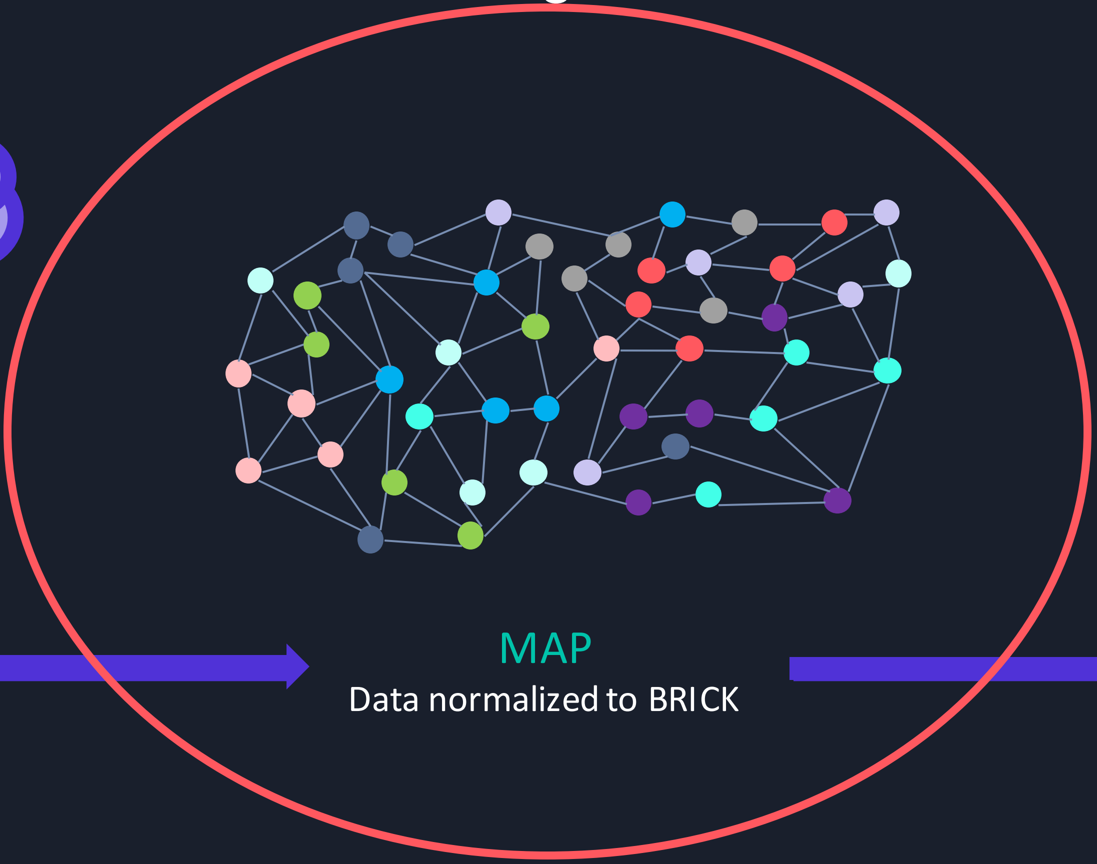
Mapping Library

Subgraphs
from each Source

Mapping and Normalization at Scale

AI Engine





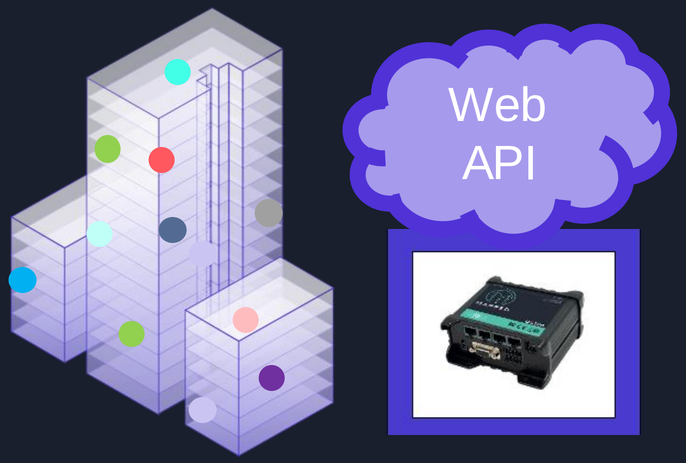
```
{
  building(id:"175A7C19") {
    floors(index: "3") {
      spaces {
        id
        name
        geoshape
      }
    }
  }
}
```

CONNECT
Discover & Extract

MAP
Data normalized to BRICK

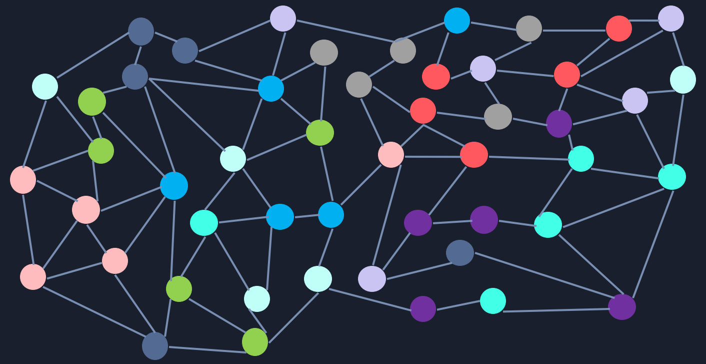
ACCESS
Solutions & Apps

A Unified Interface through GraphQL + BRICK



CONNECT

Discover & Extract



MAP

Data normalized to BRICK

```
{
  building(id:"175A7C19") {
    floors(index: "3") {
      spaces {
        id
        name
        geoshape
      }
    }
  }
}
```

ACCESS

Solutions & Apps

Mapped GraphQL

- A standard API model for **structured data**
- A client can define **exactly what they need**
- **Standard types, relations, and properties** by an ontology

```
{
  buildings (id: <building_id>) {
    things (type: AHU) {
      feeds (type: VAV) {
        id
        points (type: Supply_Air_Flow_Sensor){
           id
           name
           series (startTime: 2023-06-01,
                  endTime: 2023-06-06) {
             timestamp
             value
}}}}}}
```

What are Ontologies?

- A common way to represent concepts of interest
- What concepts? **Types, annotations, relationships, properties**
- Haystack started with tagging. BRICK started with type systems.

Why are there different ontologies?

- Different design principles
- Different eco system
- Different life cycles

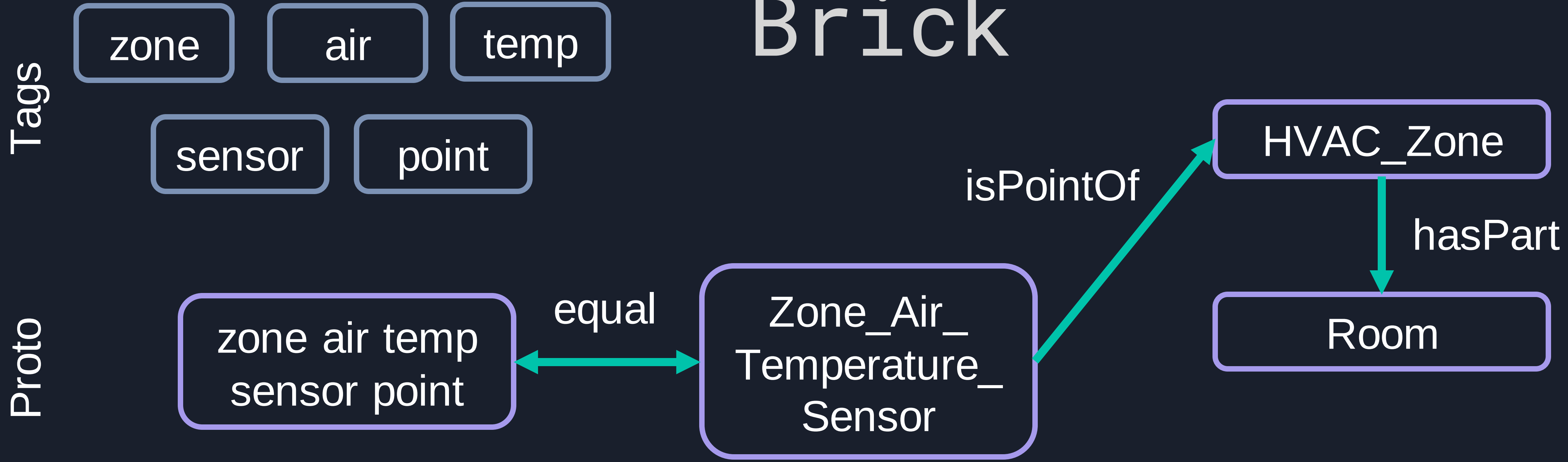
They all want the same:
interoperability

Relationships between Ontologies

Project Haystack

Brick

RealEstateCore



Tags

zone

air

temp

sensor

point

Proto

zone air temp sensor point

equal

Zone_Air_Temperature_Sensor

isPointOf

HVAC_Zone

hasPart

Room

Haystack API over Mapped in BRICK

Haystack Client

1. Haystack Read filter

```
temp and sensor and
   equipRef=@VAV1
```

7. Grid Response

```
id,dis,curVal
...,...,...
```

Mapped-Haystack

2. Parsed Query

```
Tags: temp, sensor
 Ref: equip:@vav1
```

3. Haystack-BRICK Lookup

Haystack tags	Haystack proto	BRICK class
…	…	…

4. GraphQL Construction

```
things (id: vav1) {
  points (type: {in: [“Temp_Sensor”, “Air_Temp_Sensor”,…]
    ...
}
```

5. GraphQL Execution

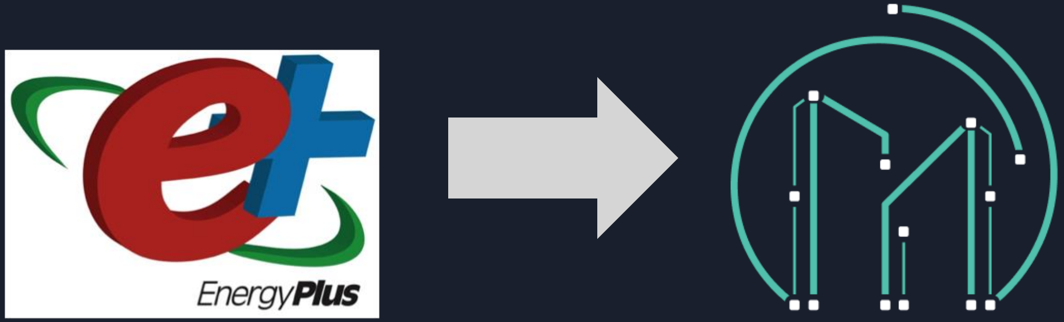
6. GraphQL -> Grid

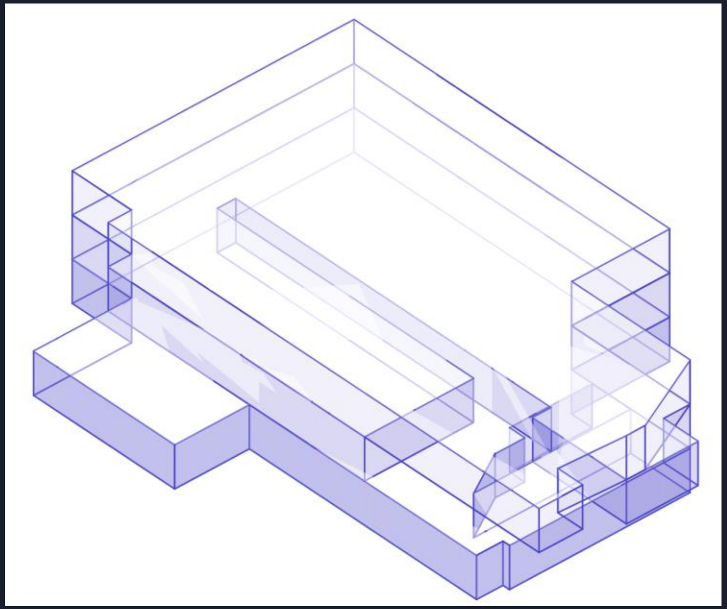
Opensource: https://github.com/mapped/mapped-haystack

Mapped Sandbox

- Need realistic sample data to test our platform.
- Converted a reference building in EnergyPlus into BRICK
  - ASHRAE901_OutPatientHealthCare_STD2019
- Run the actual simulator in real time to feed the data
- https://developer.mapped.com/docs/sandbox

Outpatient Clinic

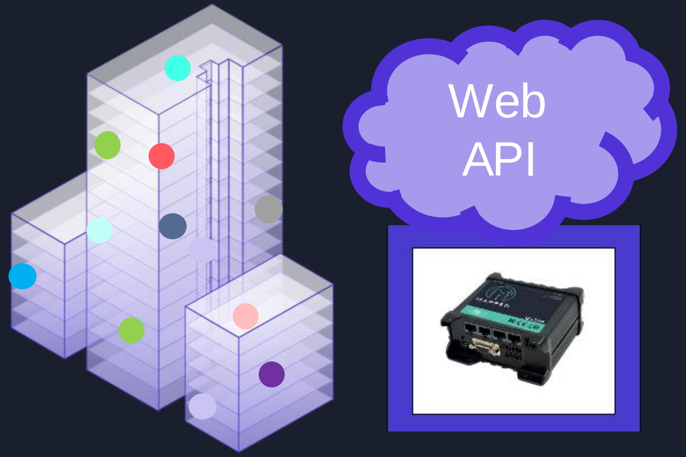


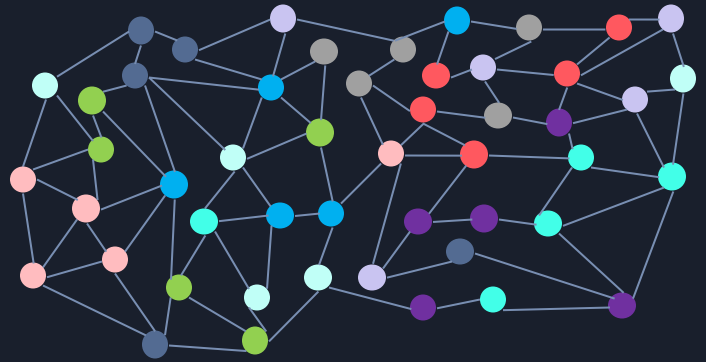


Intellicare Infirmary

An IDL should be

- Independent from both equipment and solution vendors
- Agnostic to any particular data sources
- Supporting standard ontologies





```
{
  building(id:"175A7C19") {
    floors(index: "3") {
      spaces {
        id
        name
        geoshape
      }
    }
  }
}
```

CONNECT
Discover & Extract

MAP
Data normalized to BRICK

ACCESS
Solutions & Apps

References

- https://blog.mapped.com/demystifying-the-search-for-a-perfect-ontology-through-mapping-and-evolution-51571501115e
- Cloud function by Ahmad Roaayala from https://thenounproject.com/browse/icons/term/cloud-function