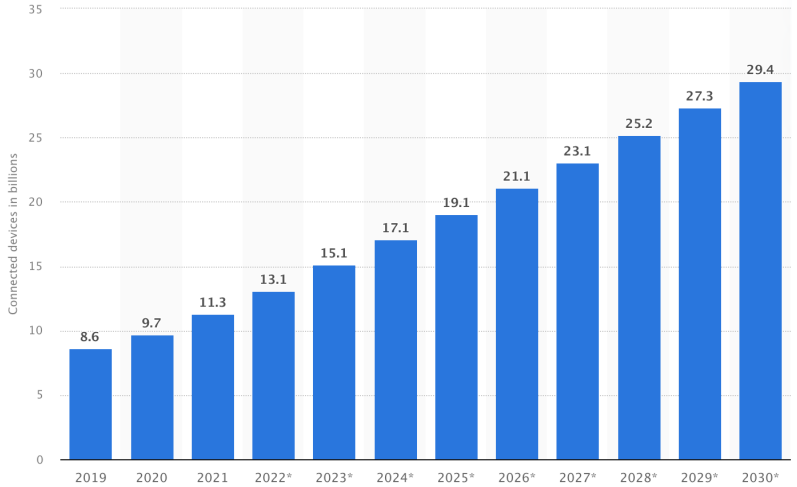
Automatic Composition in IoT Systems

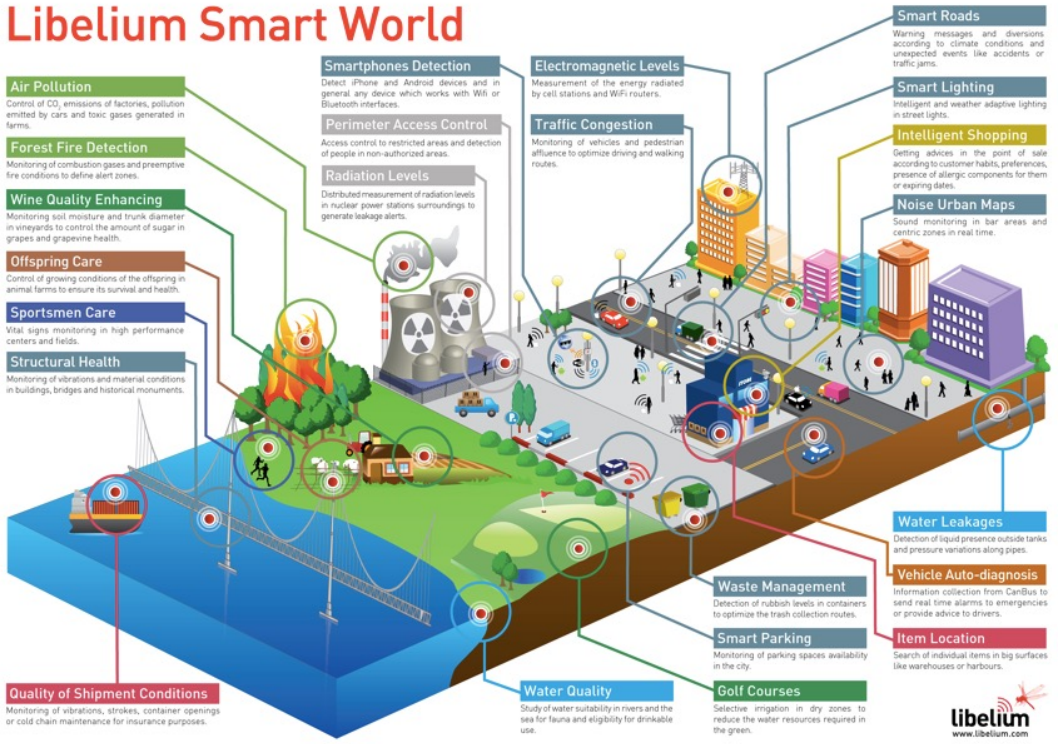
Yehia Elkhatib

The Rise in IoT Uptake

- Technological advances
  - system-on-chip design and manufacturing
  - devices are highly available, affordable, small, power-efficient
  - network connectivity: LTE, 5G, WiFi, Zigbee, LoRaWAN
- Great progress in deploying applications over various fabrics
  - in situ sensing and actuating devices
- Lots of interesting verticals



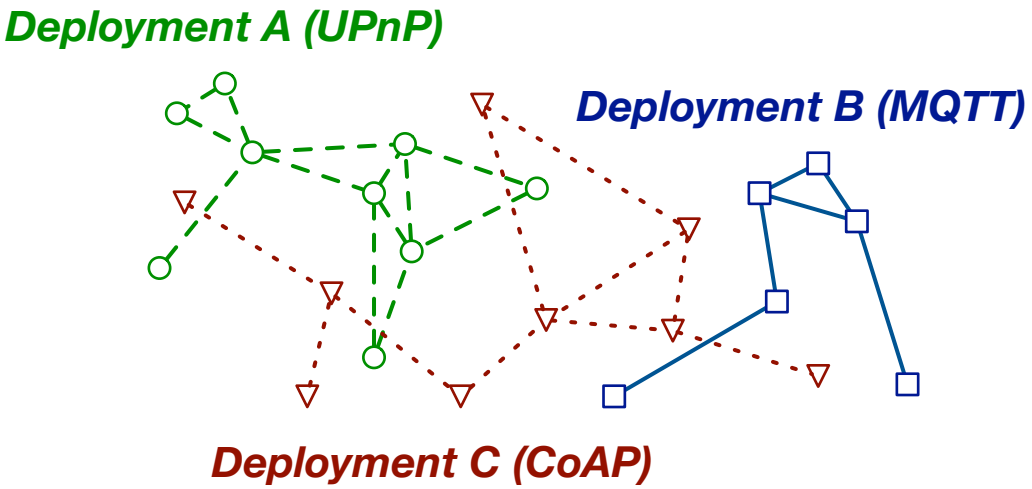
https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/



https://www.libelium.com/libeliumworld/top_50_iot_sensor_applications_ranking/

Problem Statement

- A variety of constituent systems
  - independently deployed
  - use proprietary APIs + data formats
  - co-exist in shared physical spaces
- Challenge:
  - heterogeneity makes it difficult to design, maintain and adapt *integrated systems*
  - system developers are overwhelmed with the amount of knowledge they need to acquire
  - difficult to future-proof applications for opportunistic interaction with other systems

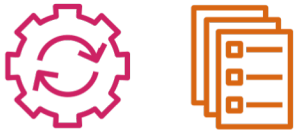


Approach Overview

- *Can we get IoT systems to talk to each other post-deployment?*
  - Yes, with a bit (lot!) of effort
- *How?*
  - Create a means to describe systems
  - Describe systems
  - Discover systems
  - Reason about composition
  - oh! and get users to write abstract workflows
  - … and sometimes we need to deploy mediators
- *Still more work to do?*
  - Always

High-level Vision

(4) System-of-system composition [SEAMS'20, FGCS'24]



(3) Workflow editing [IoT'22]

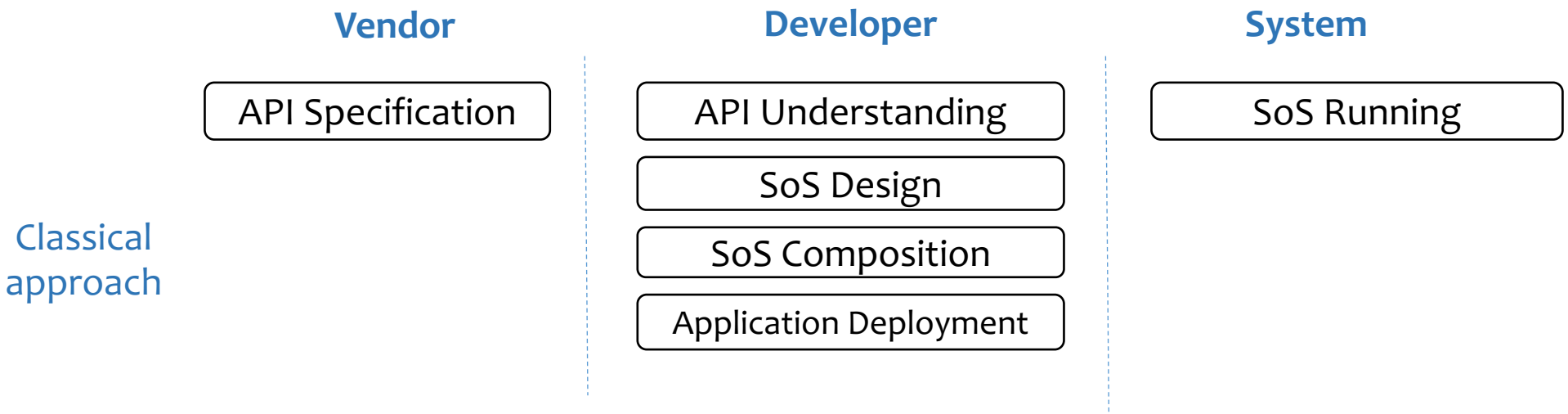
End User

(2) Holon description generation [FiCloud'22, ICWS'23]

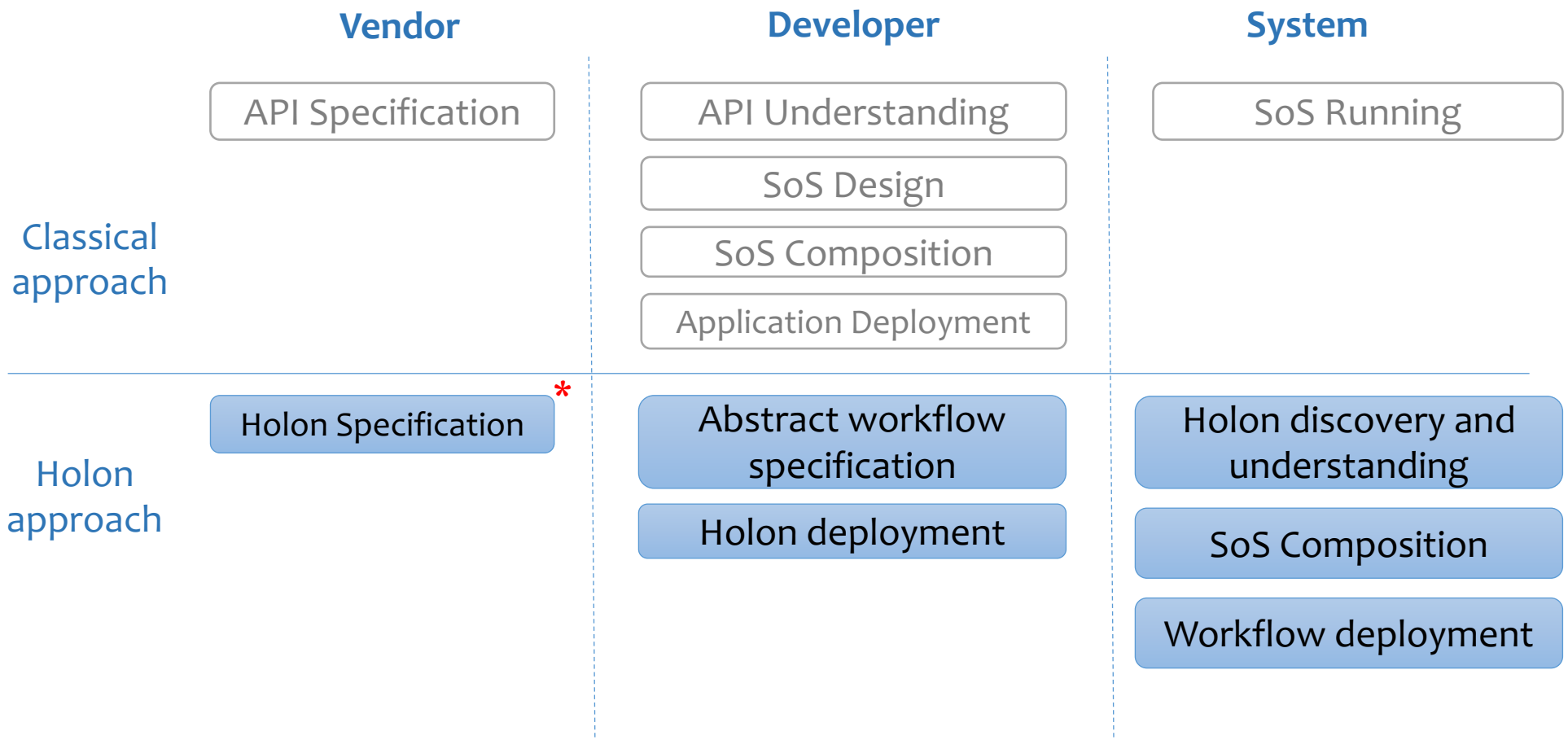
IoT Developer

(1) Device identification [ARM'15, ICIoT'20]

Problem and Solution



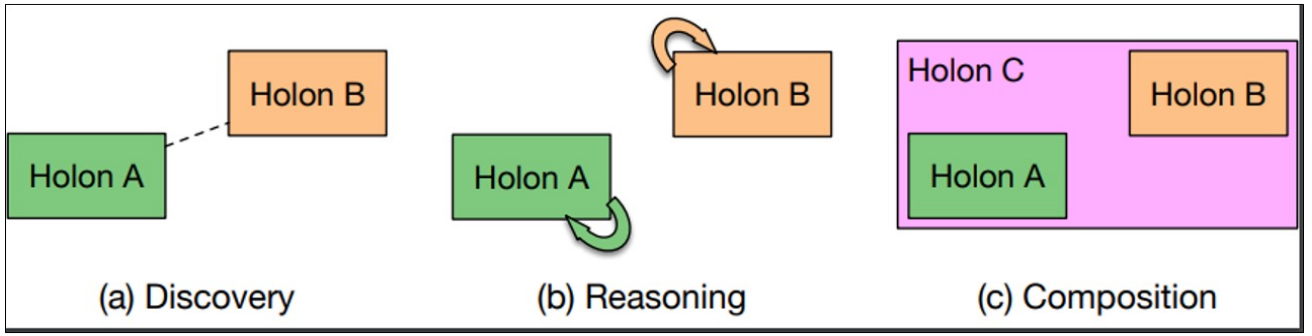
Problem and Solution



(1) Device identification

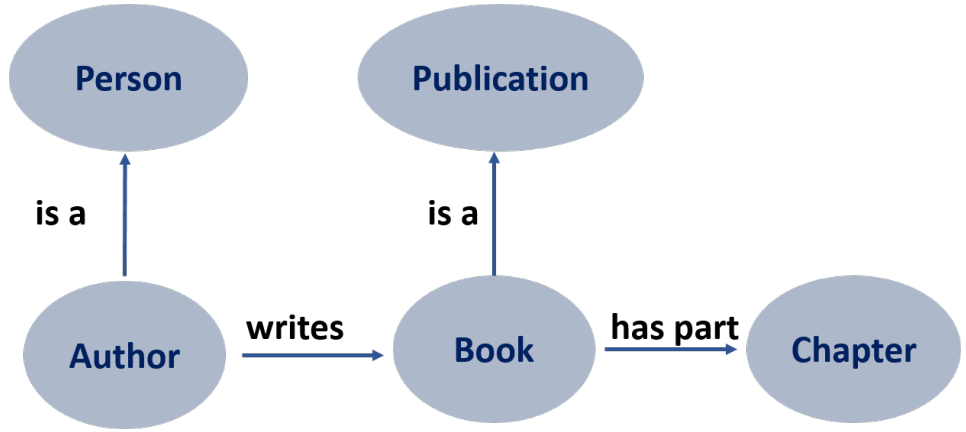
Holon?

- A *holon* is a self-describing system that appears as a whole when viewed from above whilst potentially comprising multiple subsystems when viewed from below
- A holon is described using an *ontology* that can be
  - advertised
  - discovered
  - used to reason about opportunistic composition

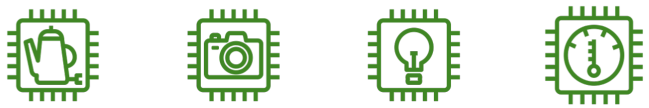


What is an ontology?

- An ontology is an abstract model of the world, or some domain
- An ontology introduces a vocabulary that is relevant to the domain
  - Explicitly defines concepts, properties, relations, and constraints

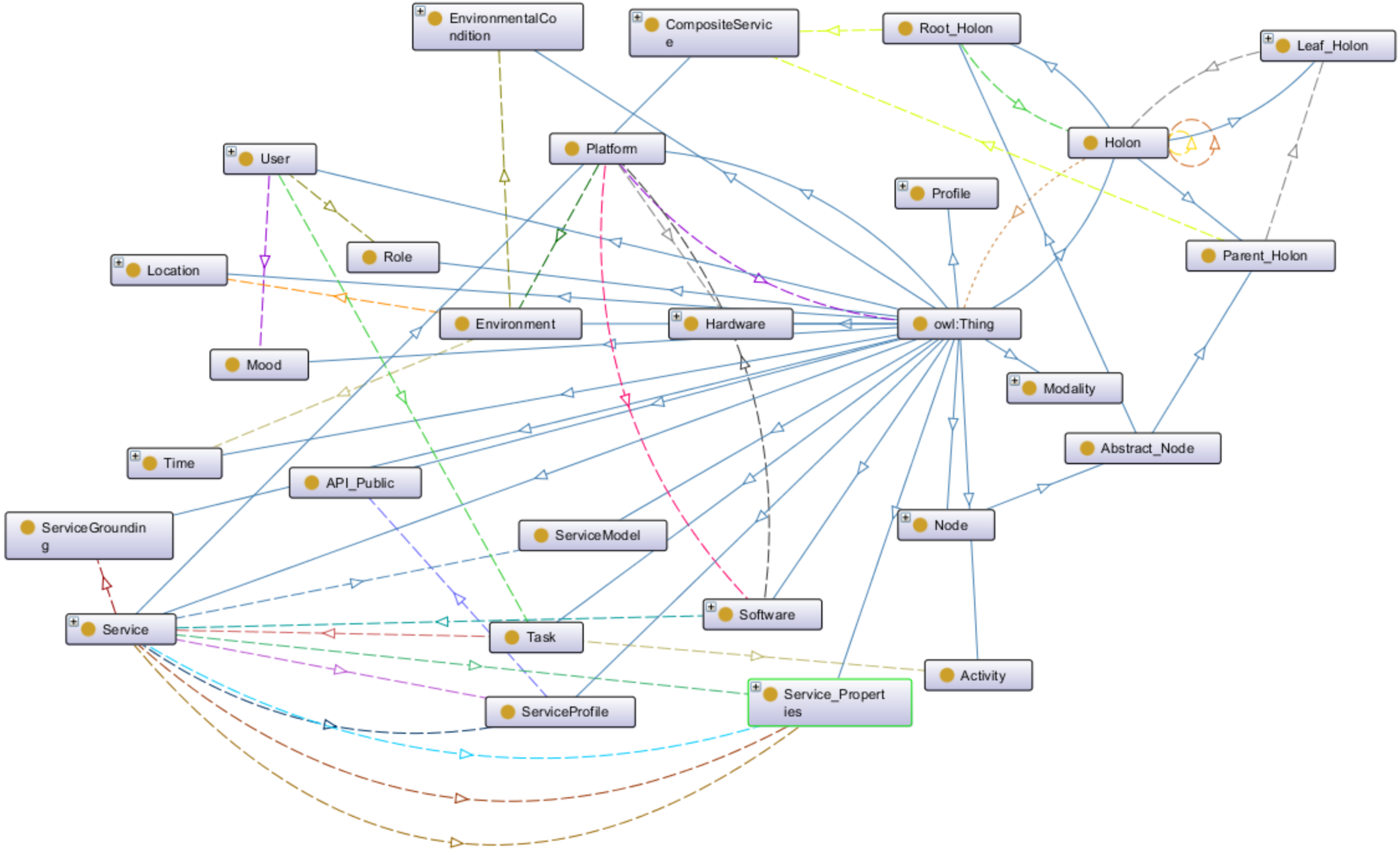


(1) Device identification

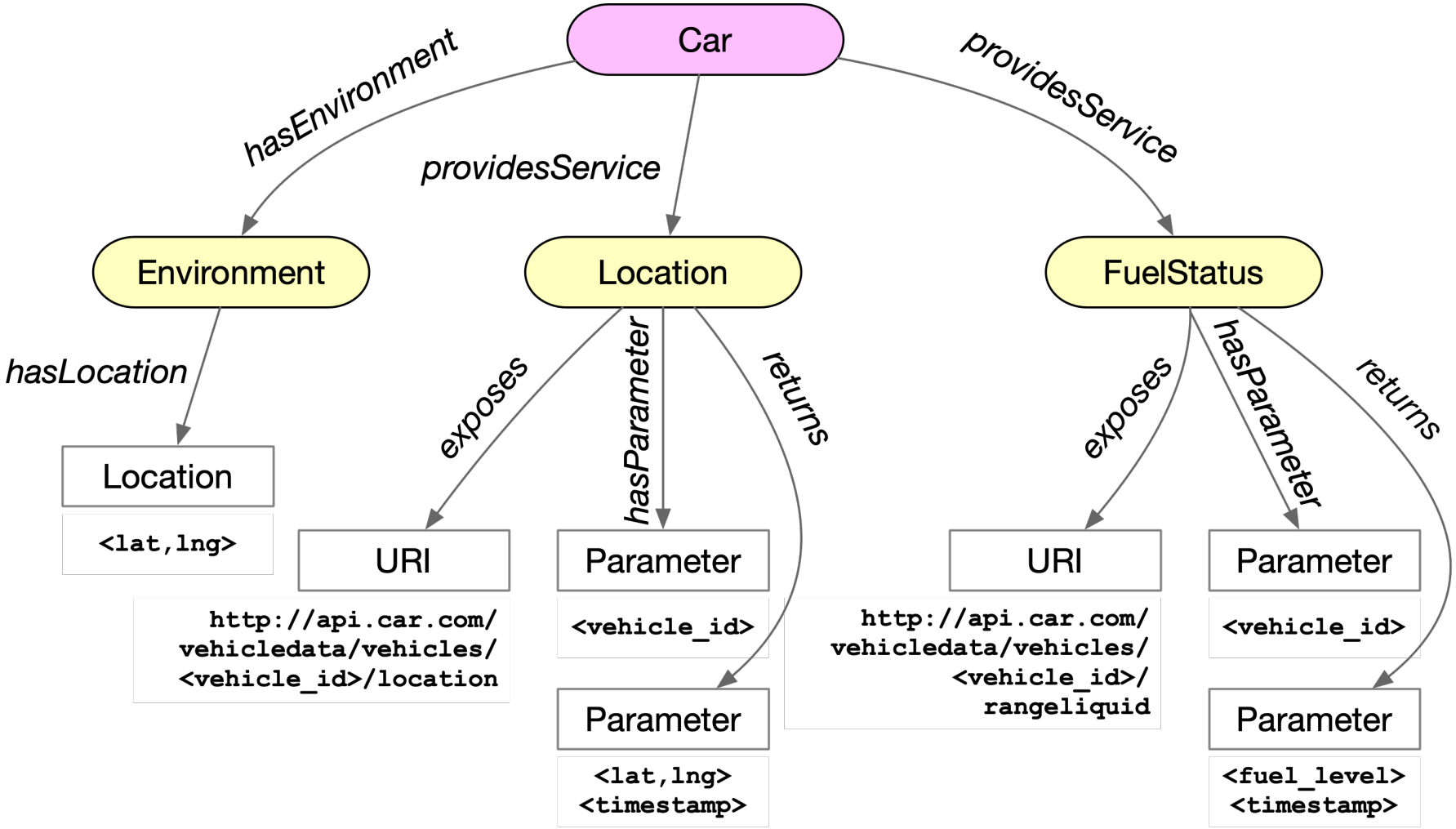


Specifying a Holon

- HOLON:
  - has parent (**holon**); has children (**holons**); supports **services**
  - Membership **properties**: has Children, is Member
  - Physical **properties**: power level; location; mobile; firmware; …
- SERVICE:
  - requires properties; desires properties; guarantees properties; may provide properties; exposes API
- PROPERTY: *(domain-specific)*
  - e.g. smart home: sensing temperature, humidity, motion; data storage; controlling climate, doors, windows, curtains; …

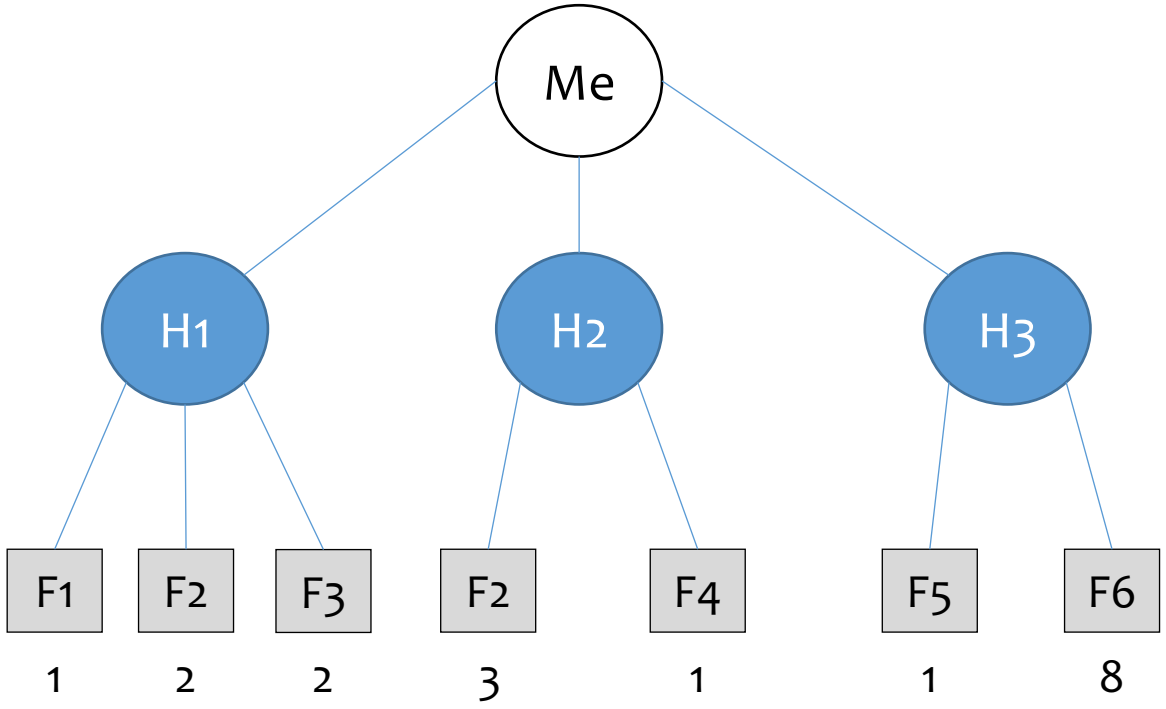

EnvironmentalCondition
CompositeService
Root_Holon
Leaf_Holon
Holon
Platform
User
Profile
Parent_Holon
Location
Role
Environment
Hardware
owl:Thing
Mood
Modality
Time
Abstract_Node
API_Public
ServiceGrounding
ServiceModel
Node
Software
Service
Task
Activity
Service_Properties
ServiceProfile

(1) Device identification



Composition

- The composition model is represented as a weighted tree of depth three



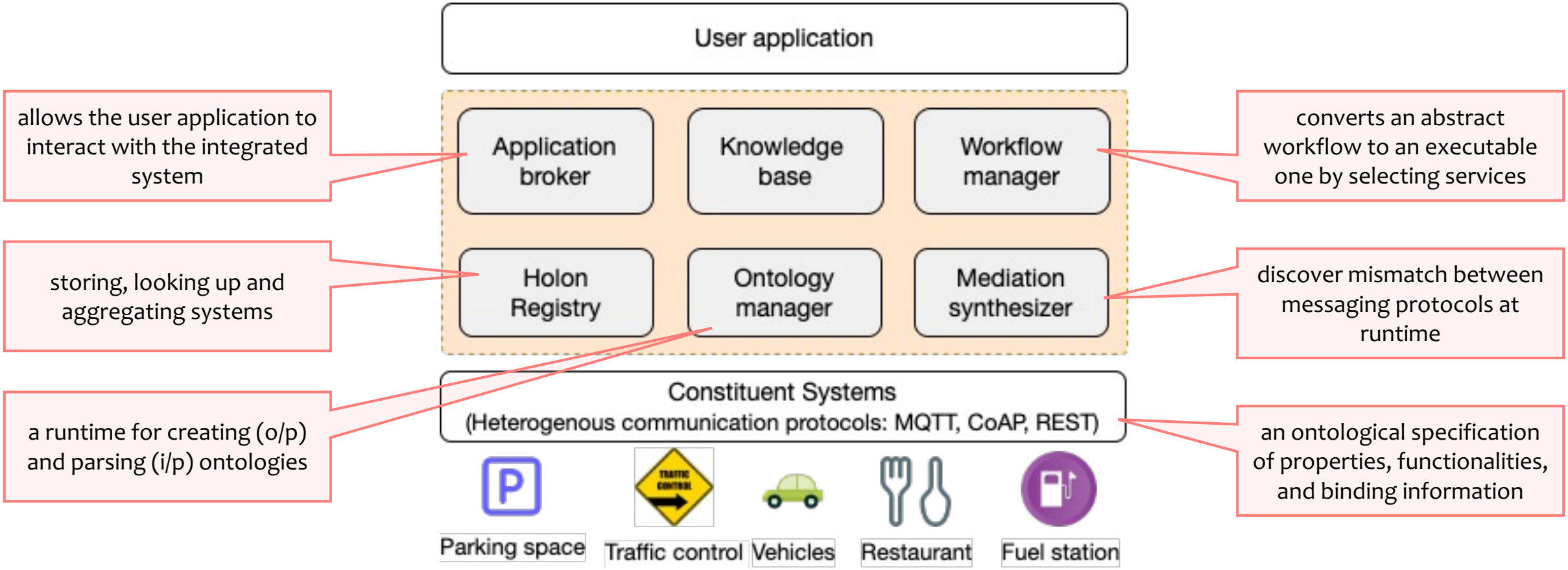
Level 1: The holon itself

Level 2: Holons that are directly reachable by the root holon

Level 3: Detected functionalities

Cost of accessing the functionality

Hetero-Genius Architecture

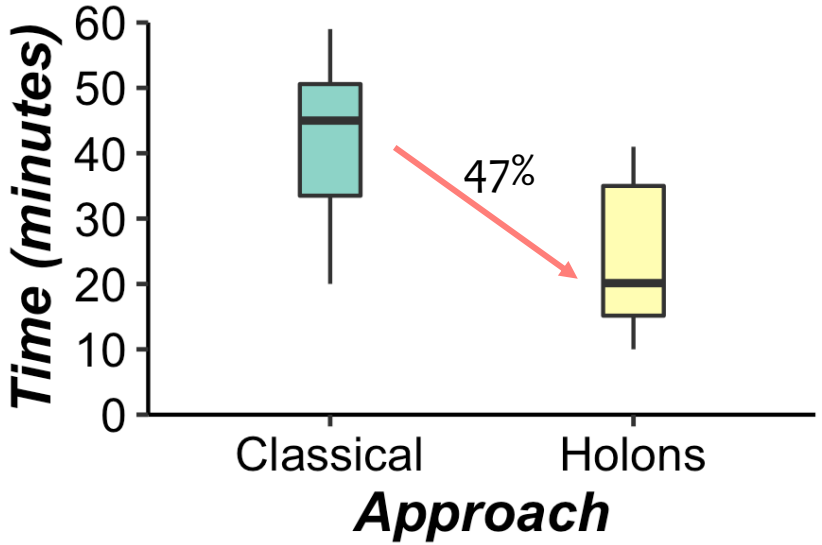


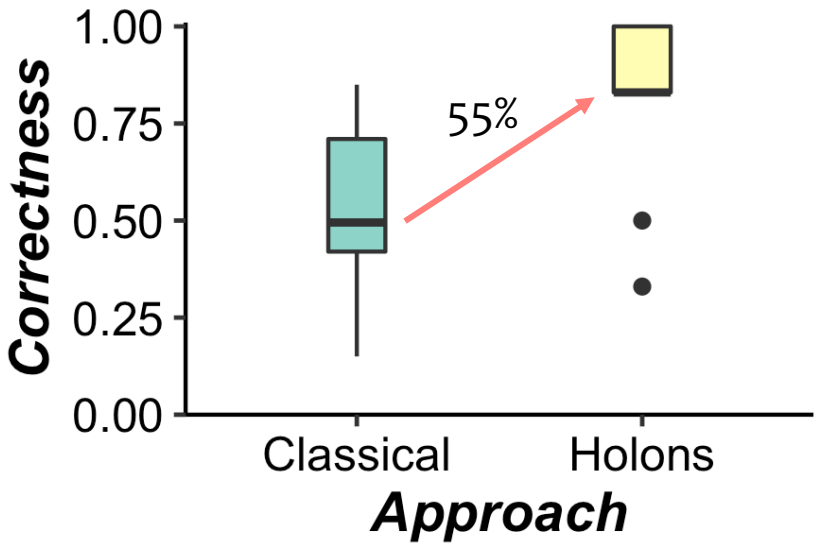
Sample Response – Classical

```
String vehicleId = "123";
int fuelThreshold = 111;
Location locationToNavigate; //assume Location data format exists
String token = ""; //authentication todo
String resultFuel = call("https://api.mercedes-benz.com/vehicledata/v2/vehicles/vehicleId/resources/rangeliquid", token);
int factFuel = JsonParser.parse(resultFuel).get("rangeLiquid").get("value");
if (factFuel <= fuelThreshold) {
    String location = "" //find location todo
    String resultStations = call("https://api/alt-fuel-stations/v1/nearest.format?parameters", location);
    double factLatitude = JsonParser.parse(resultStations).get("latitude");
    double factLongitude = JsonParser.parse(resultStations).get("longitude");
    locationToNavigate = find(factLatitude, factLongitude); //assume find function exists
}
String resultRestaurants = call("https://foodbukka.herokuapp.com/api/v1/restaurant");
String[] factResuatrants = JsonParser.parse(resultrestaurants).get("businessName");
String selectedRestaurant;
if (selectedResuarant.equals("Foodfusion") {
    String FoodfusionToken = "";
    String restaurant = "Foodfusion";
    int noOfPersons = 1;
    Date bookingFrom = Date.now;
    String resultFoodfusion = call("https://foodfusion/booking", FoodfusionToken, restaurant, noOfPersons, bookingFrom);
    String factFoodfusion = JsonParser.parse(resultFoodfusion).get("statusCode");
    if (!"OK".equals(factFoodfusion)) {
        //repeat booking
    } else {
        //locationToNavigate = this restaurant
    }
} else if (selectedResuarant.equals("Hathaway") {
    String name = "John";
    String number = 1;
    String email = "john@gmail.com";
    Date date = Date.now;
    String resultHathaway = call("https://api.hathawayfood.com/booktable", name, number, email, date);
    String factHathaway = JsonParser.parse(resultHathaway).get("statusCode");
    if (!"OK".equals(factHathaway)) {
        //repeat booking
    } else {
        //locationToNavigate = this restaurant
    }
}
//navigate to locationToNavigate
```

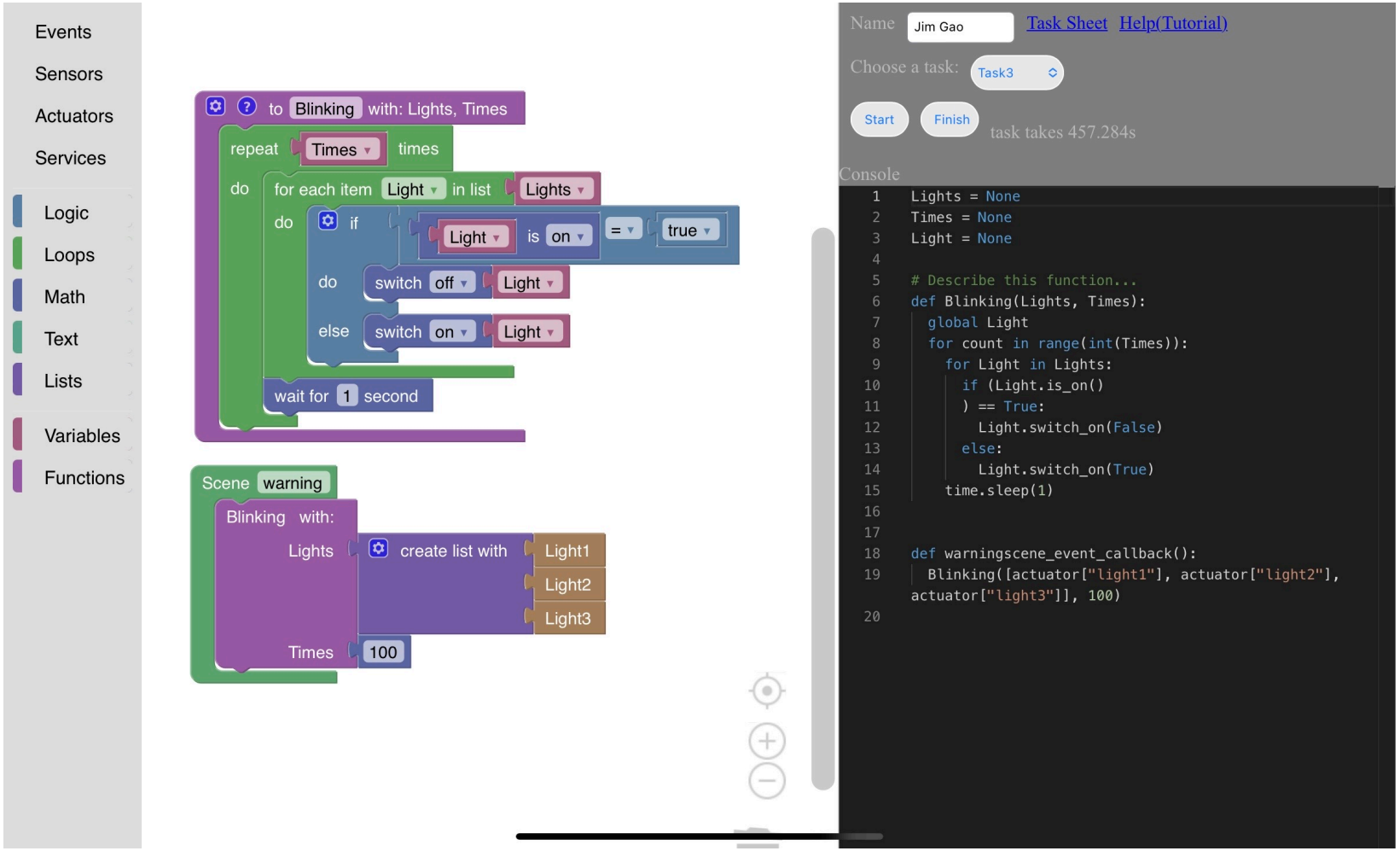
Sample Response – Holon DSL

```
String vehicleResult = call("http://holons/registry/getHolon?holonID");
vehicle= Holon.parse(vehicleResult);
connection = vehicle.call(AUTHENTICATE);
fuel= vehicle.call(CAR_RANGE);
threshold = 5;

if(fuel < threshold ) {
    String stationResult= call("http://holons/registry/getHolon?type=station&longitude=longitudeValue&latitude=latitudeValue)";
    fuelStation= Holon.parse(stationResult);
    echo "Fuel Station";
}

String restaurantResult= call("http://holons/registry/getHolon?type=restaurant&longitude=longitudeValue&latitude=latitudeValue");
restaurant= Holon.parse(restaurantResult);

restaurant.call(BOOK_TABLE);
```



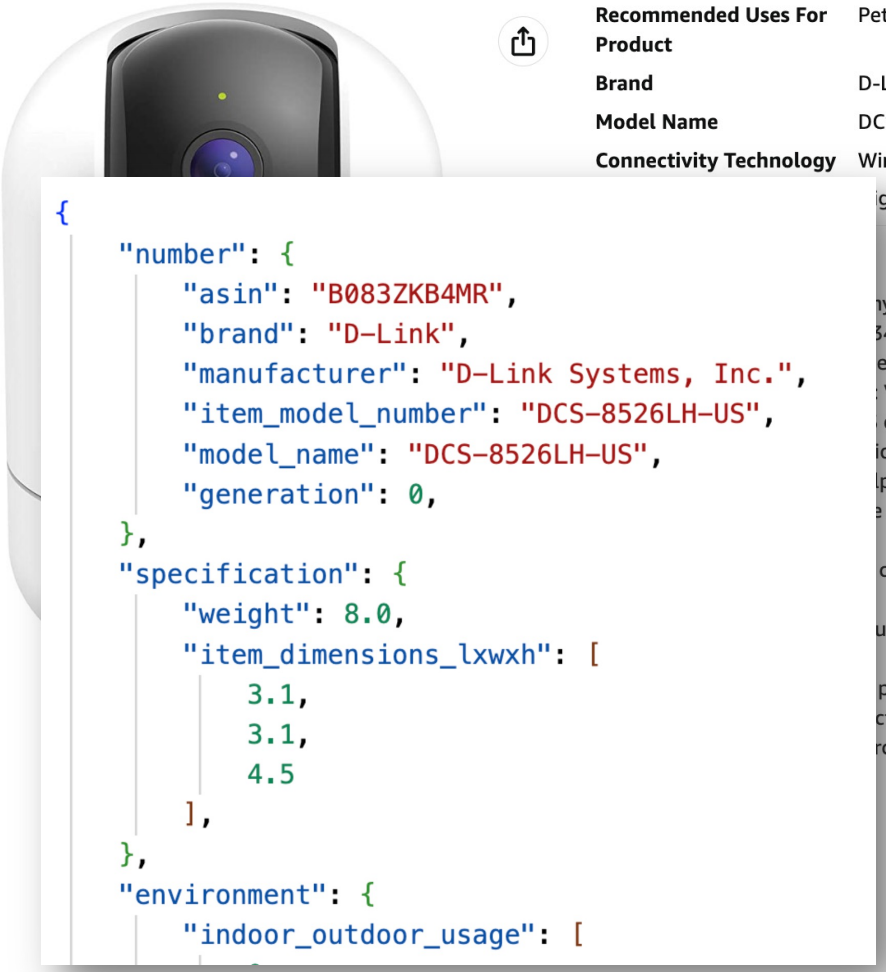


Graphical Workflow – HolonCraft



Knowledge Extraction

(2) Holon description generation



Parameters

Key-value pairs that include parameters and device properties

Functional Description

Several paragraphs of unstructured text, describing the functions of the device

(2) Holon description generation

Knowledge Extraction

- Robustly optimized BERT approach (RoBERTa), a model using the Stanford Question Answering Dataset (SQuAD.0)
- Experiment on 140 devices
  - Learning rate: 5e-5
  - Batch size 32, Epoch 20

	Precision	Recall	F1
Parameter	98.17%	94.81%	96.11%
Function description	94.64%	78.00%	84.19%
Overall Holon description	96.72%	87.53%	91.90%

#	Device Category	Number of Devices
1	Smart camera	28
2	Speaker	17
3	Smart plug	16
4	Smart bulb	15
5	TV	15
6	Sensor	11
7	Smart coffee maker	8
8	Smart cooker	8
9	Smart washer	7
10	Smart refrigerator	5
11	Smart vacuum cleaner	5
12	Smart toilet	5

References

https://yelkhatib.github.io/

- **[ARM'15]** Gordon Blair, Yérom-David Bromberg, Geoff Coulson, Yehia Elkhatib, Laurent Réveillère, Heverson B. Ribeiro, Etienne Rivière, and François Taïani, "Holons: Towards a Systematic Approach to Composing Systems of Systems", Workshop on Adaptive and Reflective Middleware, 2015.
- **[ICIoT'20]** Vatsala Nundloll, Yehia Elkhatib, Abdessalam Elhabbash, and Gordon S Blair, "An Ontological Framework for Opportunistic Composition of IoT Systems", International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), IEEE, pages 614-621, February 2020.
- **[SEAMS'20]** Abdessalam Elhabbash, Vatsala Nundloll, Yehia Elkhatib, Gordon S Blair, and Vicent Sanz Marco, "An Ontological Architecture for Principled and Automated System of Systems Composition", International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), July 2020.
- **[FiCloud'22]** Zhuo Wang, Yehia Elkhatib, and Abdessalam Elhabbash, "HolonCraft -- An Architecture for Dynamic Construction of Smart Home Workflows", International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, pages 213-220, August 2022.
- **[IoT'22]** Abdessalam Elhabbash, Yehia Elkhatib, Georgios Bouloukakis, and Maria Salama, "A Middleware for Automatic Composition and Mediation in IoT Systems", International Conference on the Internet of Things (IoT), ACM, pages 127–134, November 2022.
- **[ICWS'23]** Ziyu Zhang, Yehia Elkhatib, and Abdessalam Elhabbash, "NLP-based Generation of Ontological System Descriptions for Composition of Smart Home Devices", International Conference on Web Services (ICWS), IEEE, July 2023.
- **[FGCS'24]** Abdessalam Elhabbash, Yehia Elkhatib, Vatsala Nundloll, Vicent Sanz Marco, and Gordon S Blair, "Principled and Automated System of Systems Composition using an Ontological Architecture", Future Generation Computer Systems, Elsevier, 2024.

Challenges

- IoT-motivated ontology
  - Evaluated in contexts like smart home, IoV, etc.
  - Cross ecosystem boundaries
- Discovery through announcement
  - Architecture relies on systems advertising themselves
  - Passive discovery still possible, but more challenging
- Trust
  - Assume that it is safe to compose
  - Assume that we are not compromising inherent system

chist-era

EPSRC

University of Glasgow

Lancaster University

unine
UNIVERSITÉ DE NEUCHÂTEL

UNIVERSITÉ DE RENNES 1

TELECOM SudParis