COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

COLUMBIA ICSL
INTELLIGENT & CONNECTED SYSTEMS LAB

LegoSENSE: An Open and Modular Sensing Platform for Rapidly-Deployable IoT Applications

IoTDI'23
San Antonio, Taxas

Minghui (Scott) Zhao, Stephen Xia, Jingping Nie, Kaiyuan Hou, Avik Dhupar, and Xiaofan (Fred) Jiang

Electrical Engineering Department, Columbia University

In the era of computing, technology, and AI we need easily deployable sensing systems

Transportation

Manufacturing

Agriculture

Healthcare

Urban Safety

Supply Chain

Environment

Smart Home

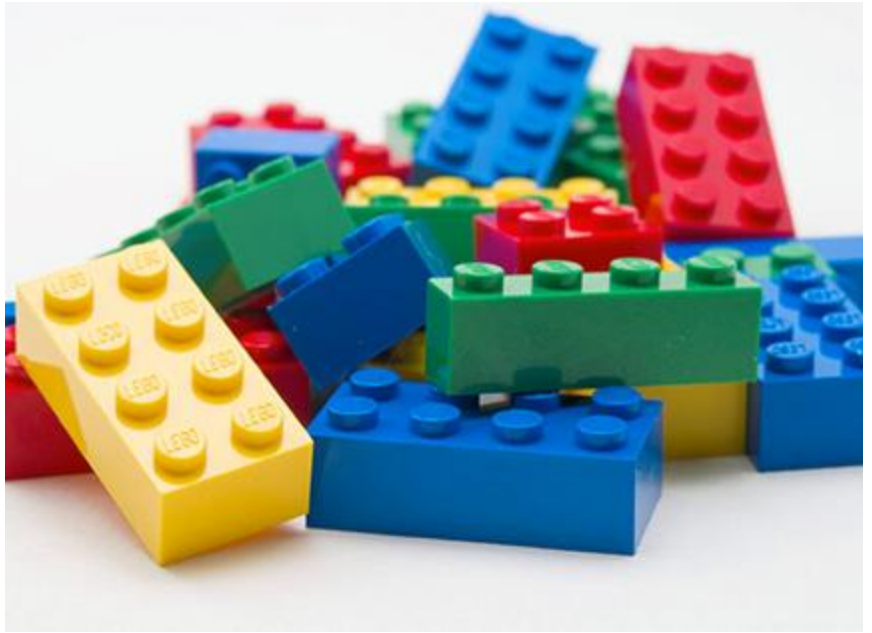
Deploying Sensing Systems Requires $$ and/or Domain-Specific Knowledge

Industrial Solutions

- Ease of Use
- High Cost
- Low Customizability

Custom DIY

- Not Easy
- Low Cost
- High Customizability



- Easy to Use
- Low Cost
- High Customizability

Deploying Sensing Systems Requires $$ and/or Domain-Specific Knowledge

Industrial Solutions

- Ease of Use
- High Cost
- Low Customizability

Custom DIY

- Not Easy
- Low Cost
- High Customizability

LegoSENSE

- Easy to Use
- Low Cost
- High Customizability

LegoSENSE makes sensing systems accessible

Carrier Board (RPi)

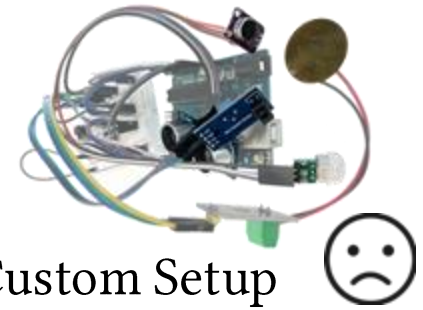
Unified **Hardware** + **Software** Interface

- No wiring or coding
- Low cost and open-source

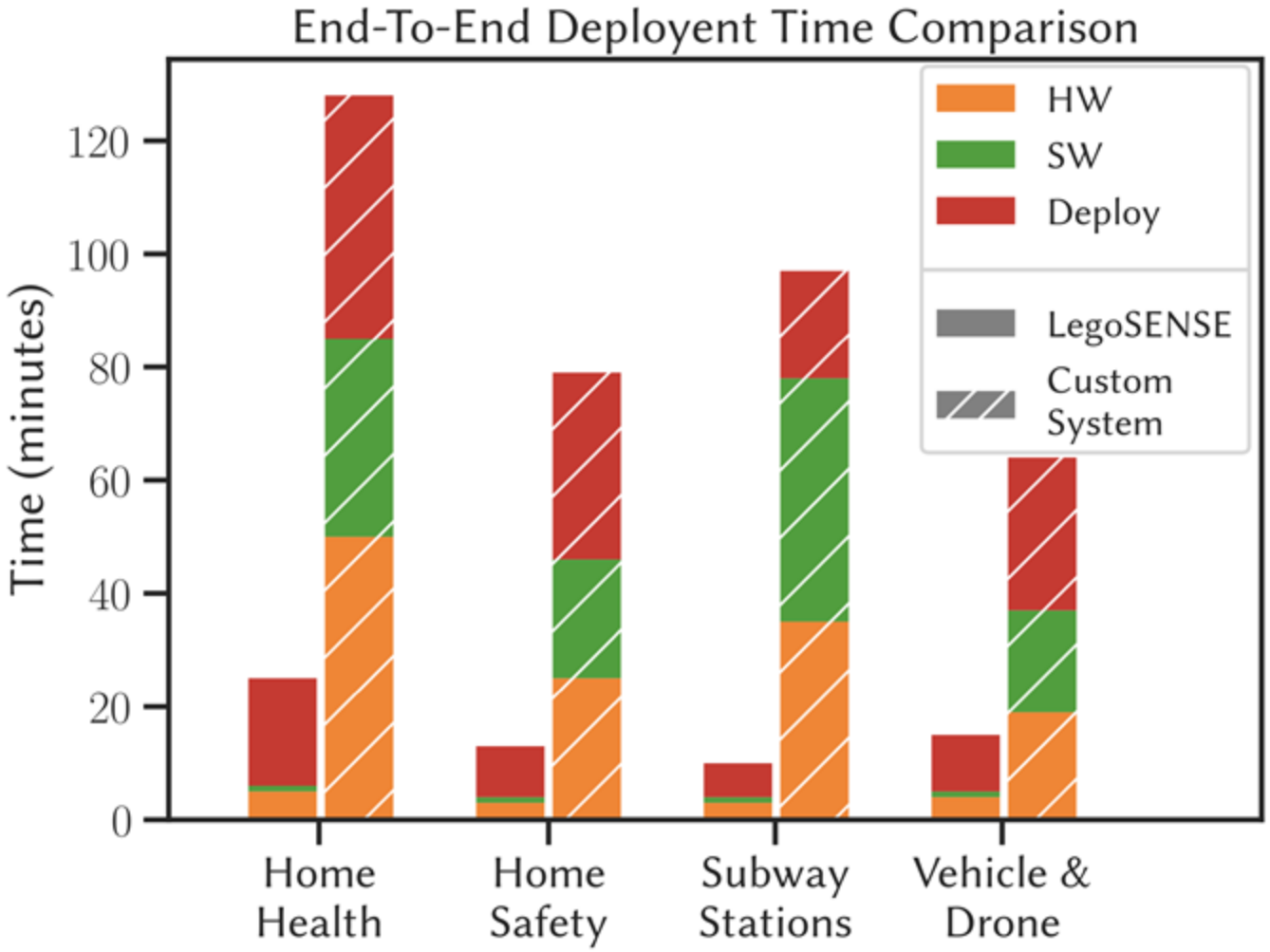
Suite of Sensors

Deployment time reduced by 5.4x

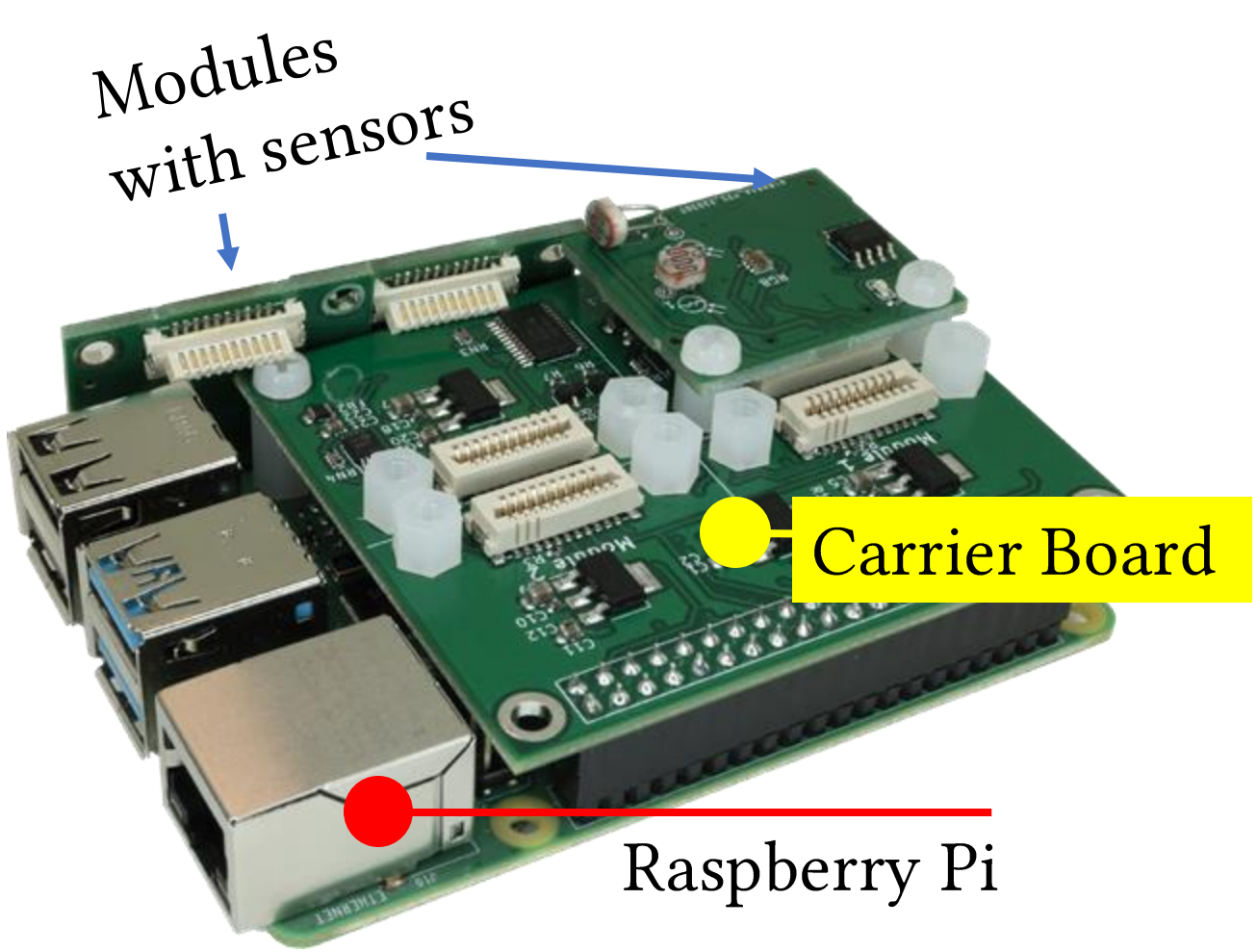
LegoSENSE

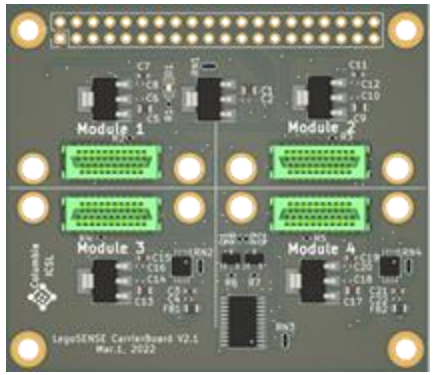


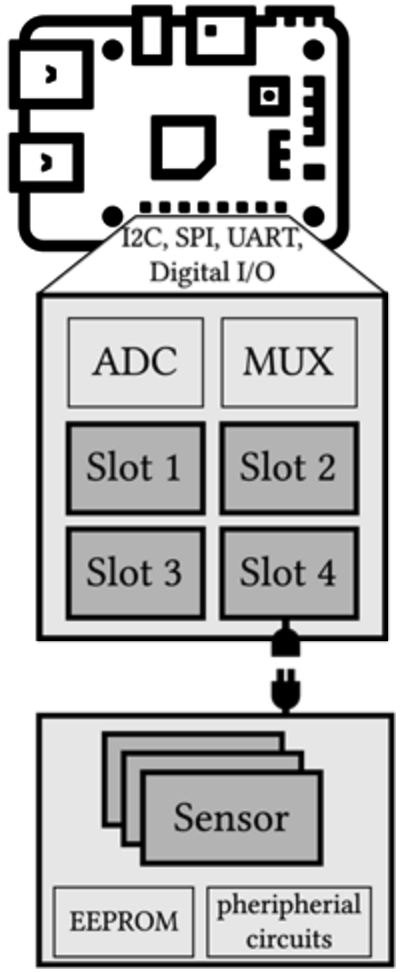
Custom Setup



Modularizing LegoSENSE's hardware architecture

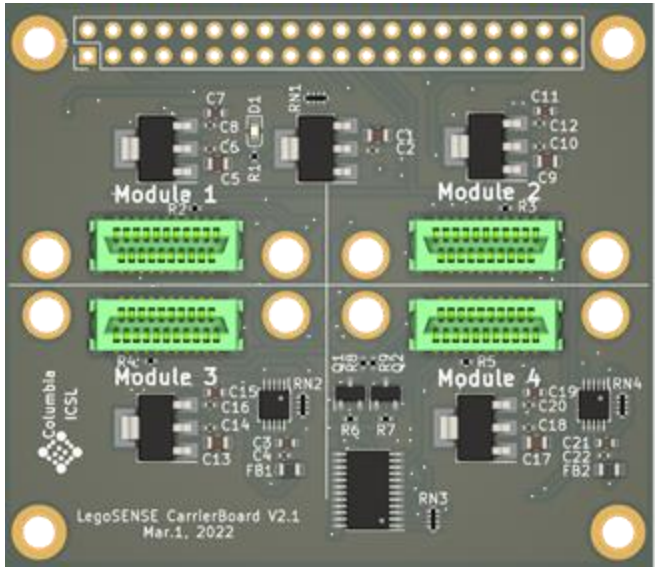


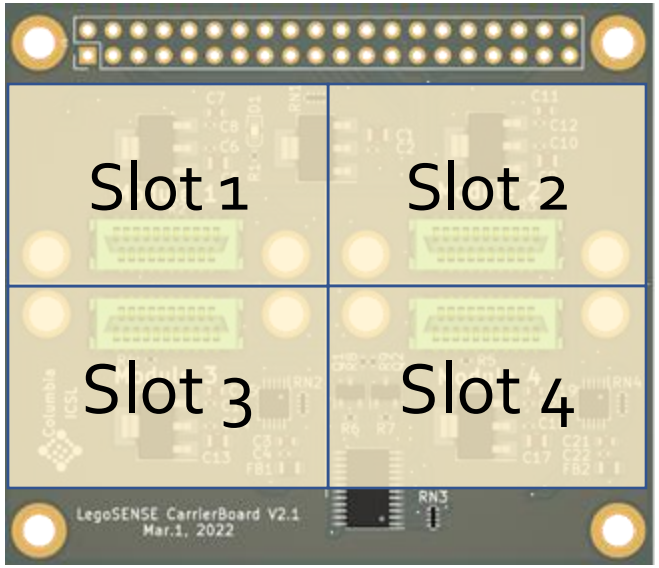




Modularizing LegoSENSE's hardware architecture

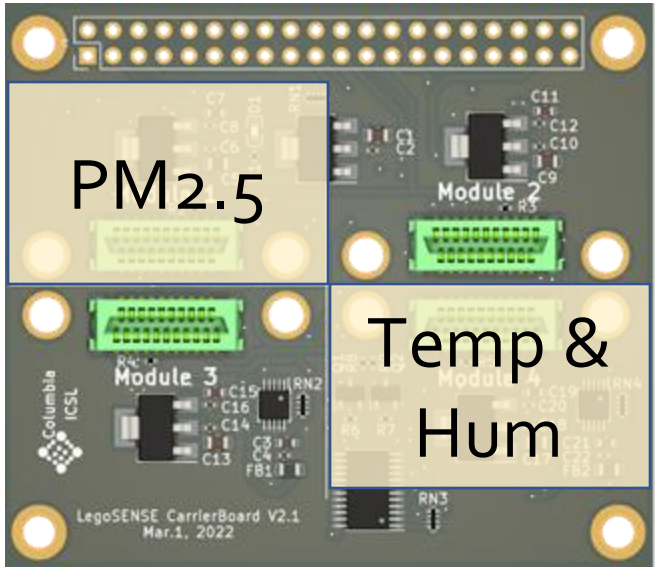
- Four slots on carrier board

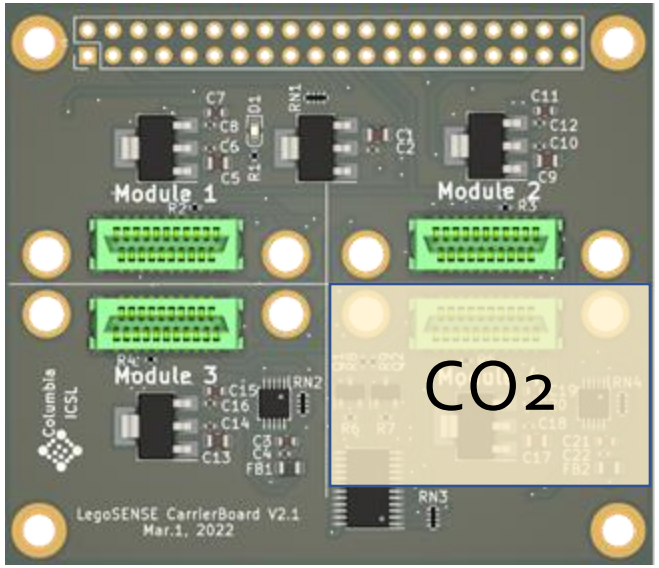


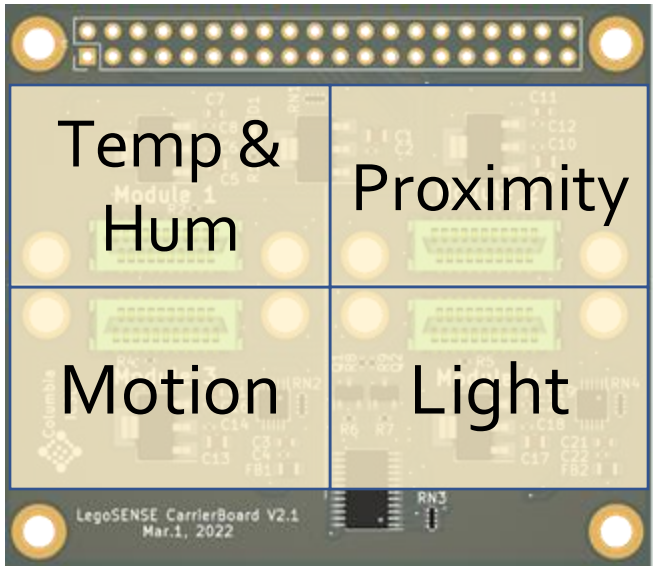


Modularizing LegoSENSE's hardware architecture

- Four slots on carrier board
- Easily reuse and mix & match different modules

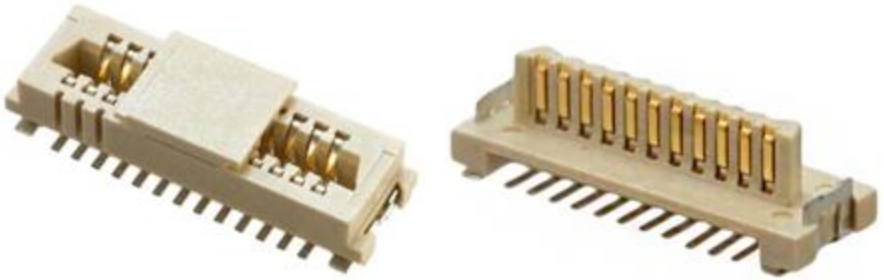


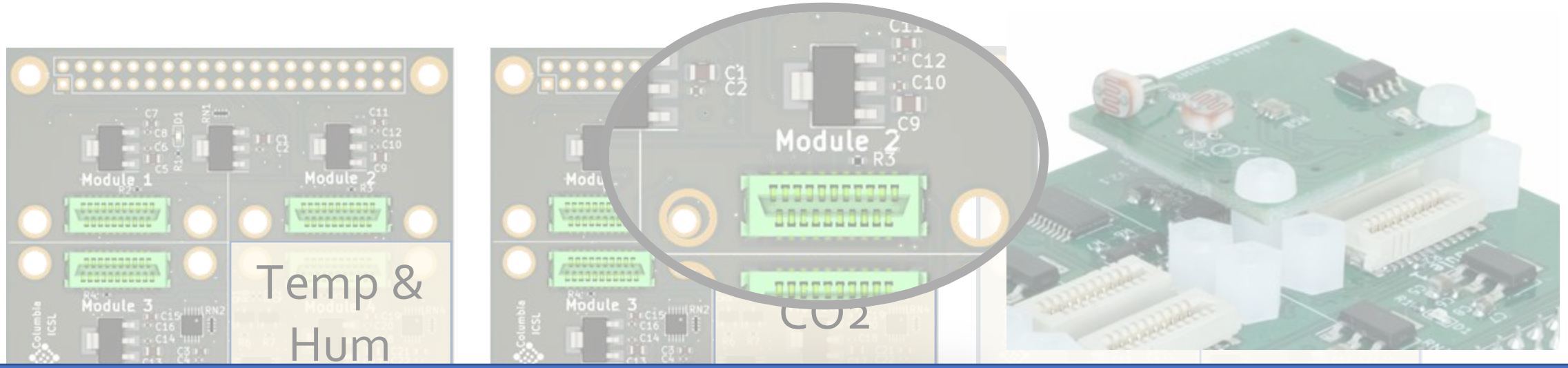




Modularizing LegoSENSE's hardware architecture

- Four slots on carrier board
- Easily reuse and mix & match different modules





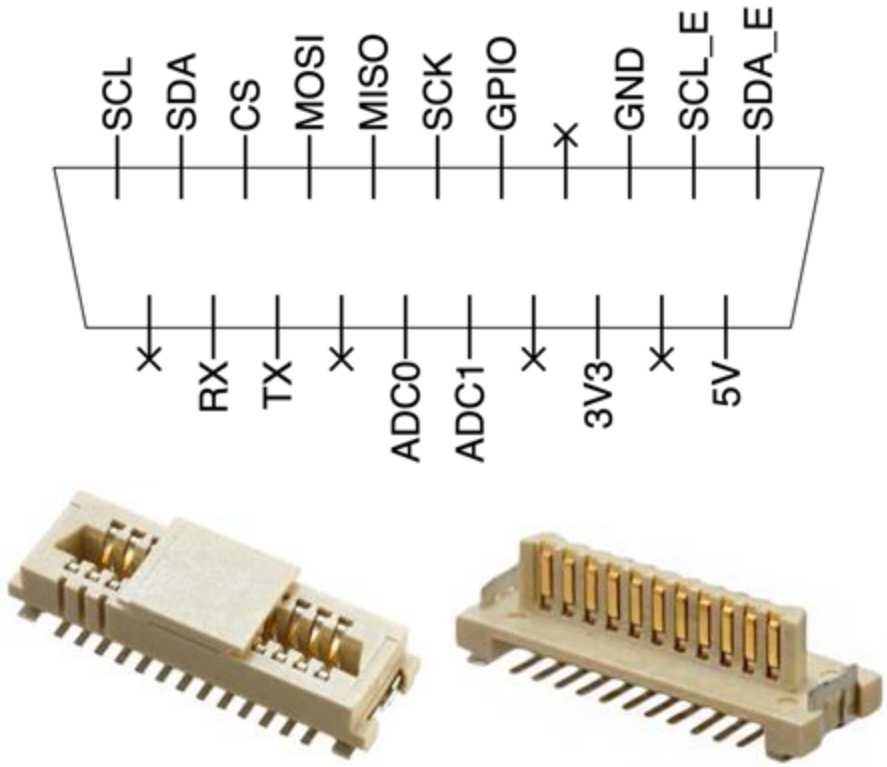
How to make everything plug-and-play?

Making modules plug-and-play

- EEPROM stores module info
  - Module type (what driver to use)
  - Factory calibration constants
  - …

Making modules plug-and-play

- Compact connector with most common protocols
  - I2C
  - SPI
  - UART
  - Analog
  - Digital I/O
- Carrier Board
  - ADC
  - Multiplexers
  - Voltage regulators



Now, onto the software framework

Streamlined experience for non-technical users

Easily customized and expandable by developers

Designing the software framework

Traditionally:

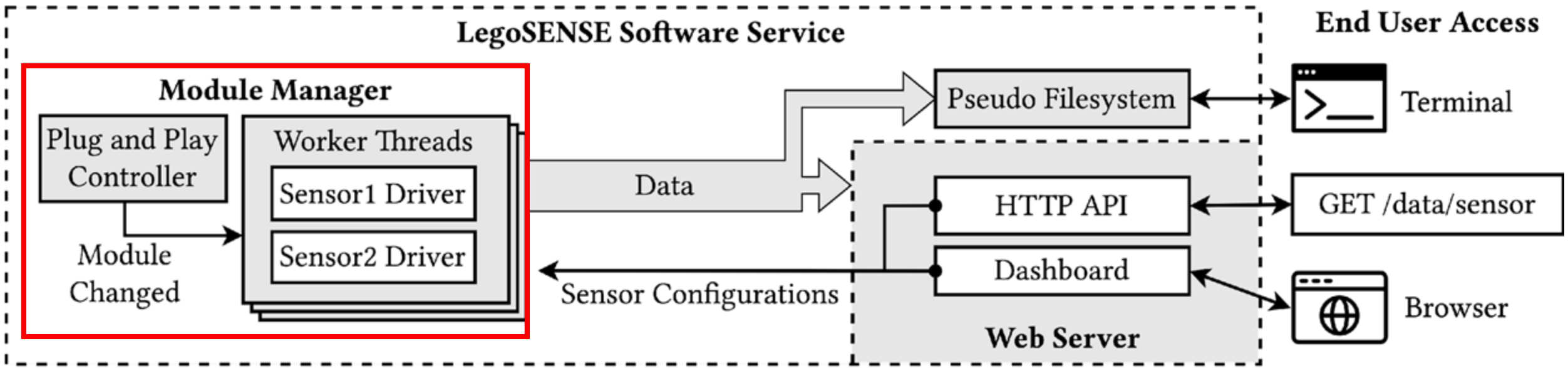
- Install driver
- Configure sensors
- Run each sensor's driver separately

Goal:

- Automated and streamlined plug-and-play user experience

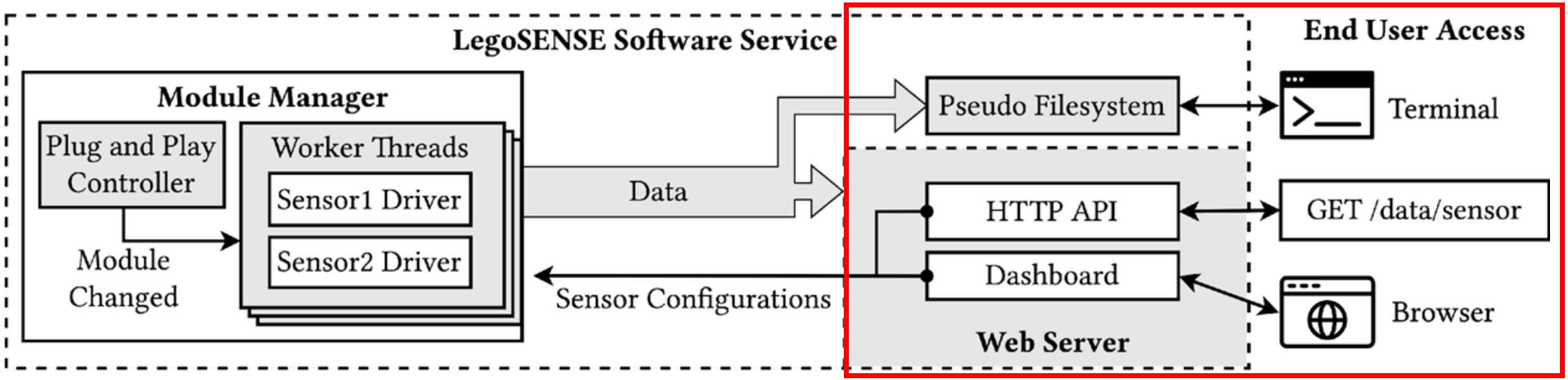
Creating a streamlined experience

- **Module Manager**: Acquire the sensor data



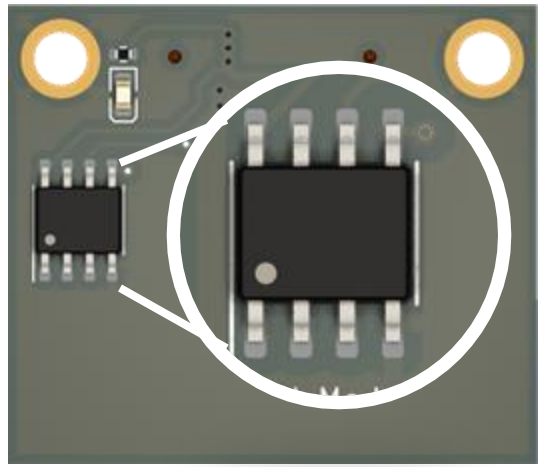
Creating a streamlined experience

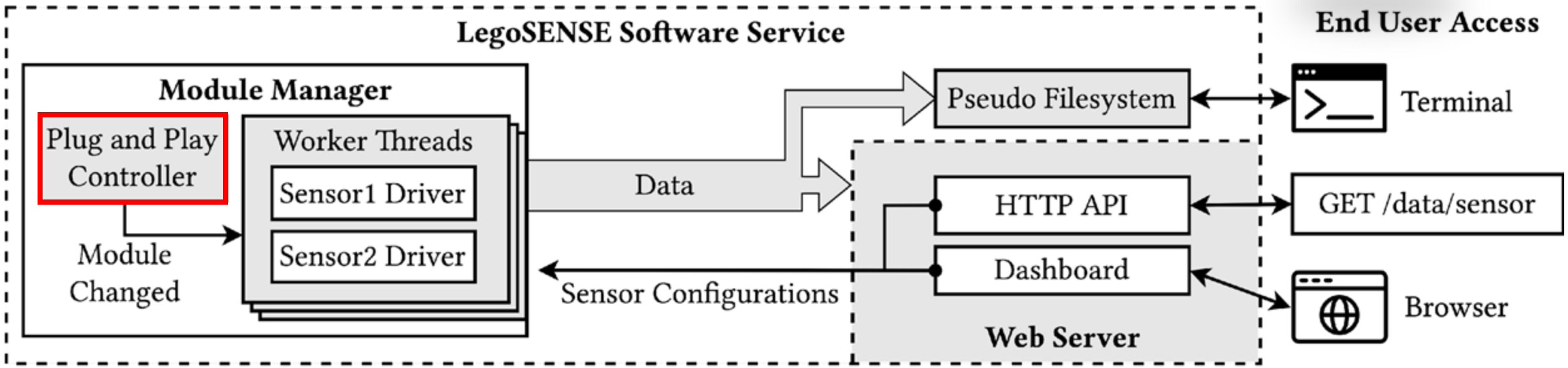
- **Module Manager**: Acquire the sensor data
- **Sensor Access**: Present the data and allows user configuration



Automatically detecting sensors

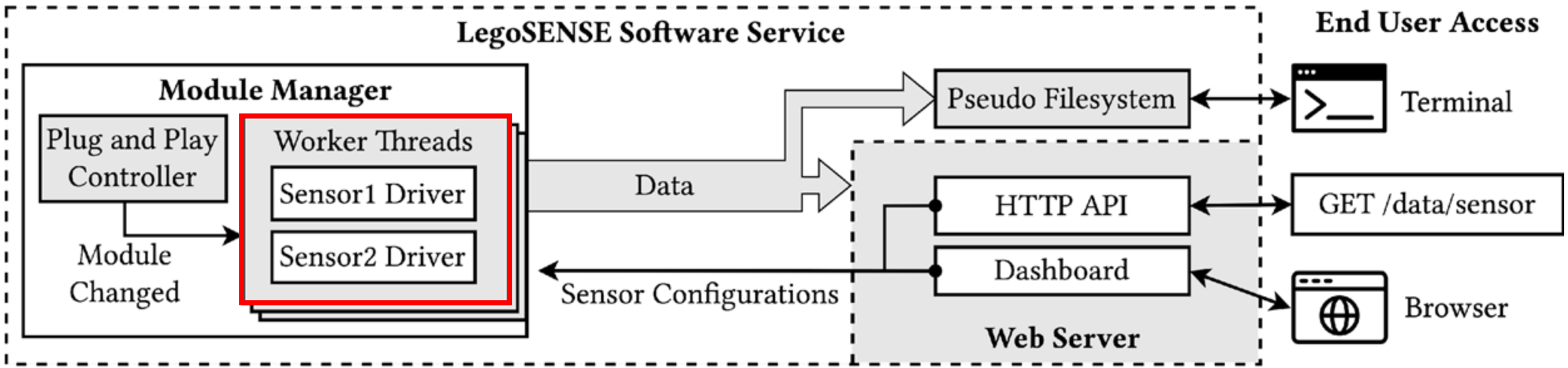
- Multiplexer on carrier board
- Pulls each module's EEPROM in round-robin





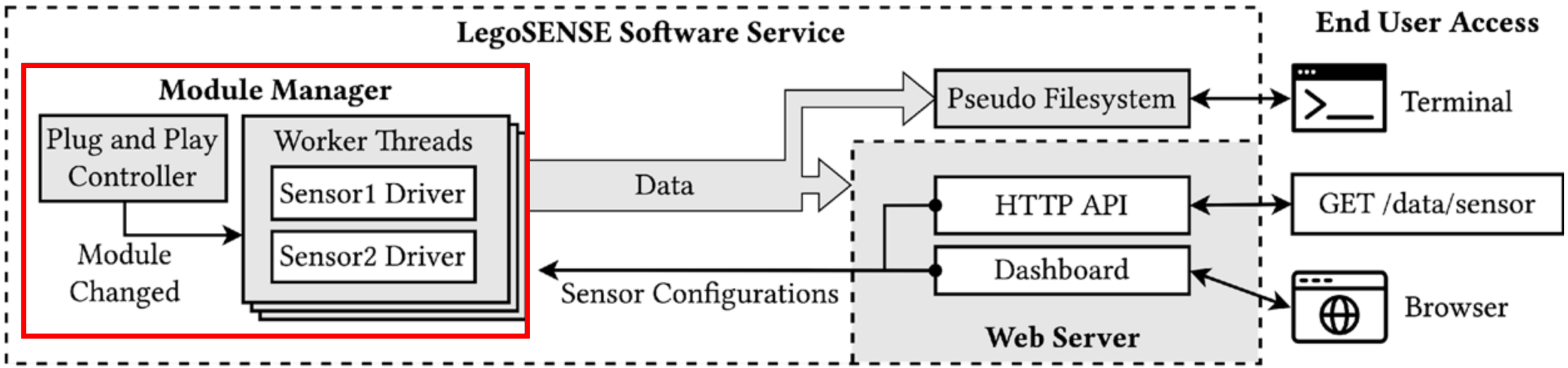
Automatically run sensor drivers

- Load sensor driver automatically upon connecting
- Open-source design - download / update sensor driver automatically



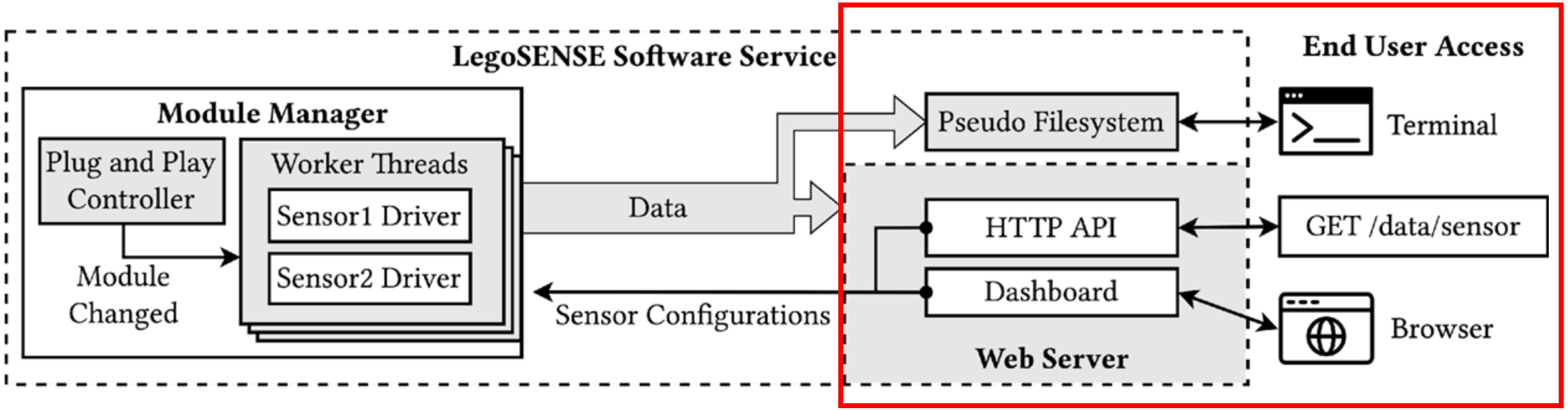
Managing sensor drivers

- Watches for sensor failures and recovers automatically
- Passes sensor data into next layer
- Performs sensor configuration updates from the user



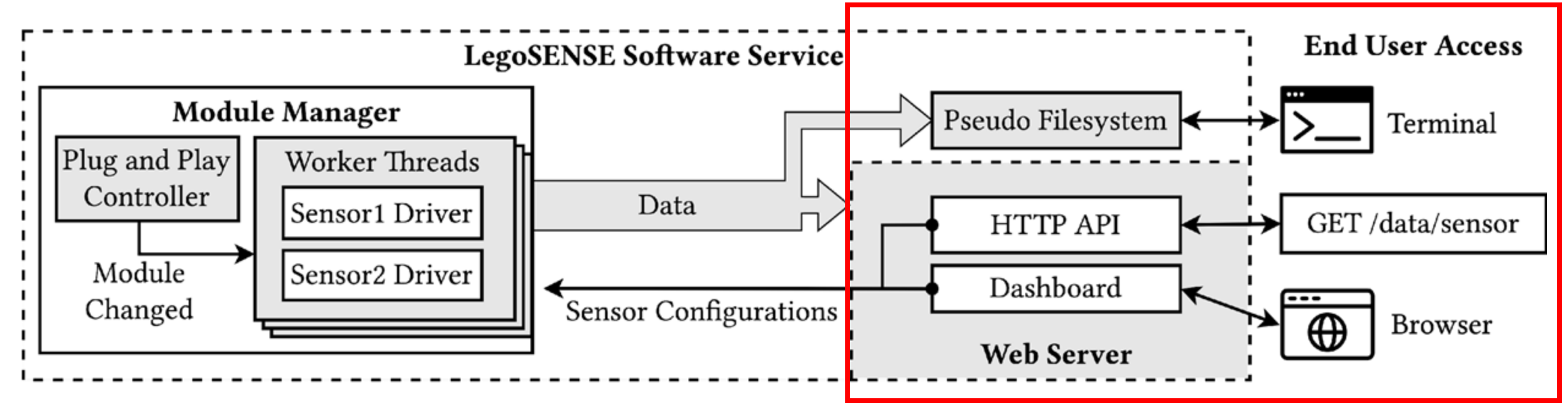
Accessing the sensors: design goals

- Easy to access data
- Seamless integration with other programs / tools



Accessing the sensors: methods

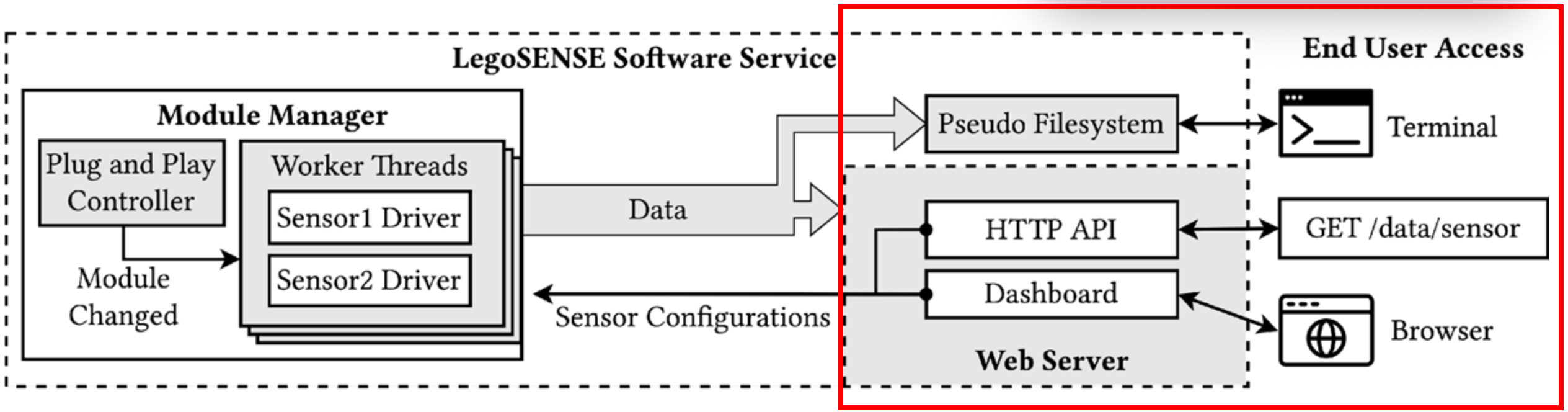
- Linux pseudo filesystem



Accessing the sensors: methods

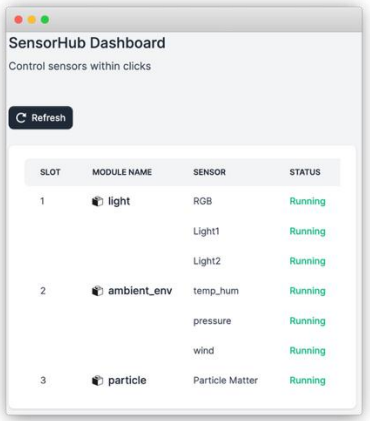
- Linux pseudo filesystem
- HTTP API

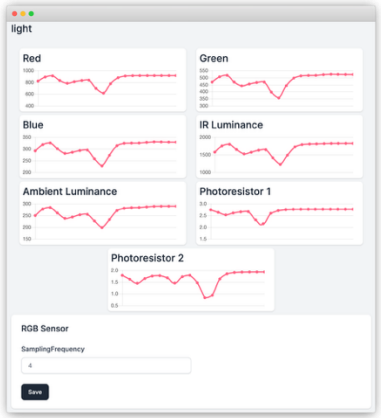
```
> GET /sensors/temp_hum

> { time: 5:20:25
    temperature: 24,
    humidity: 35 }
```

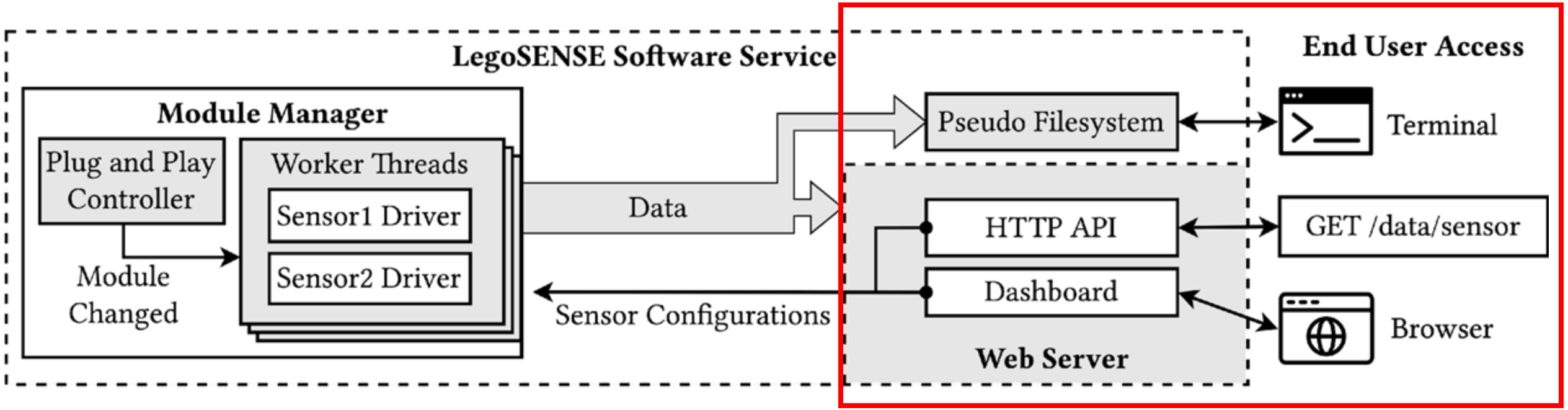


Accessing the sensors: methods

- Linux pseudo filesystem
- HTTP API
- Web dashboard

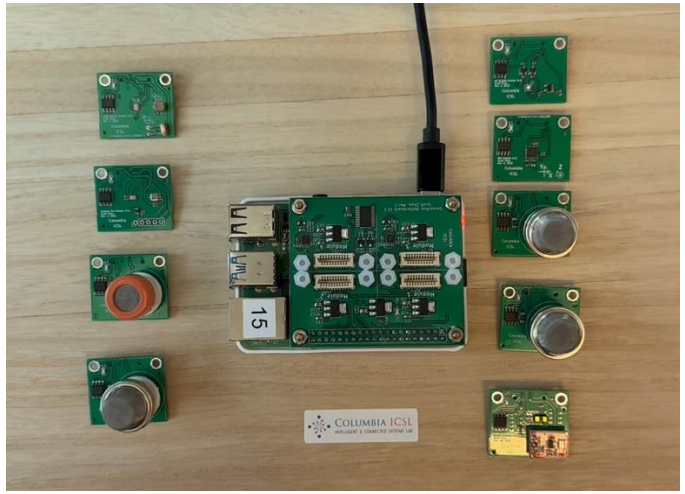


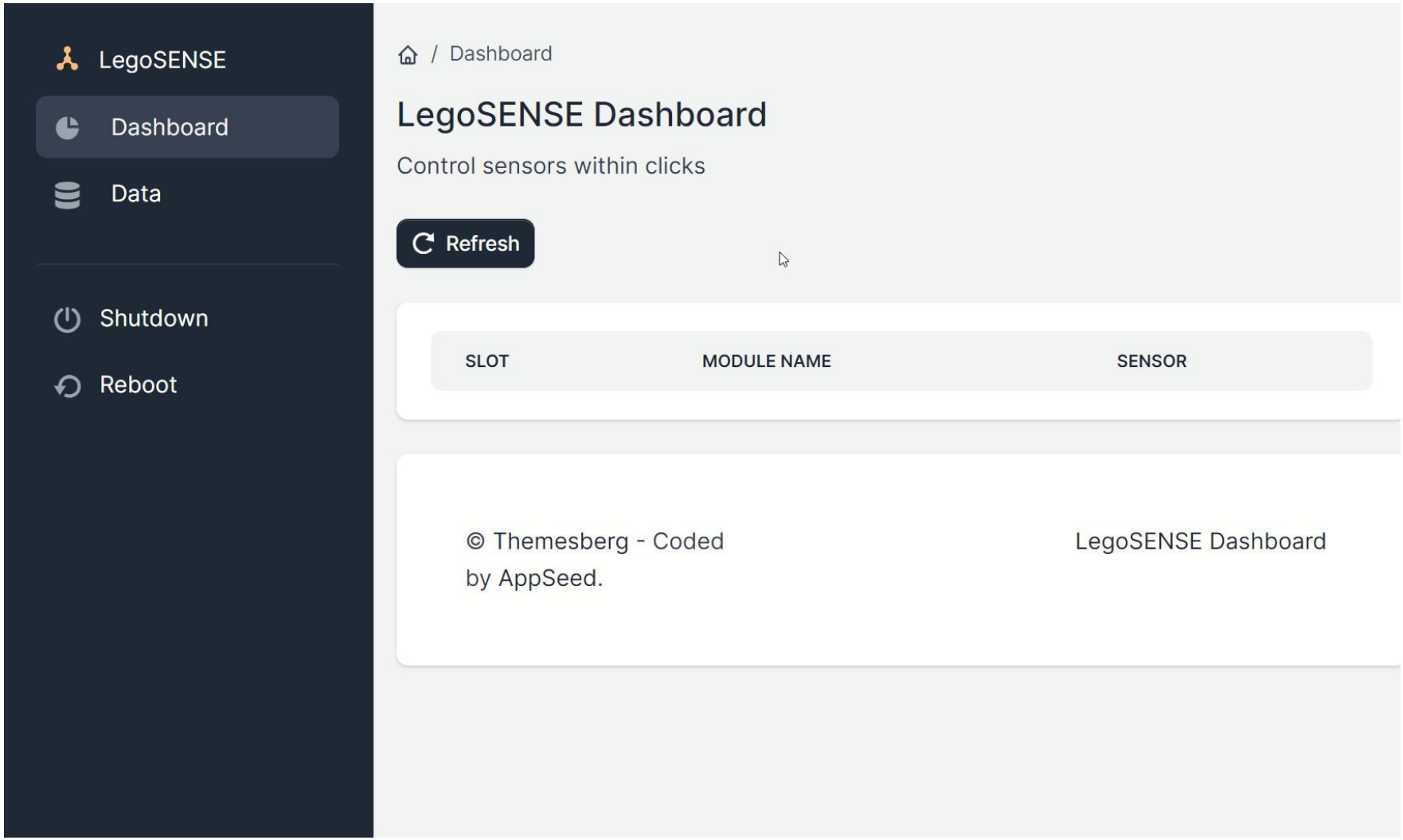




Web dashboard

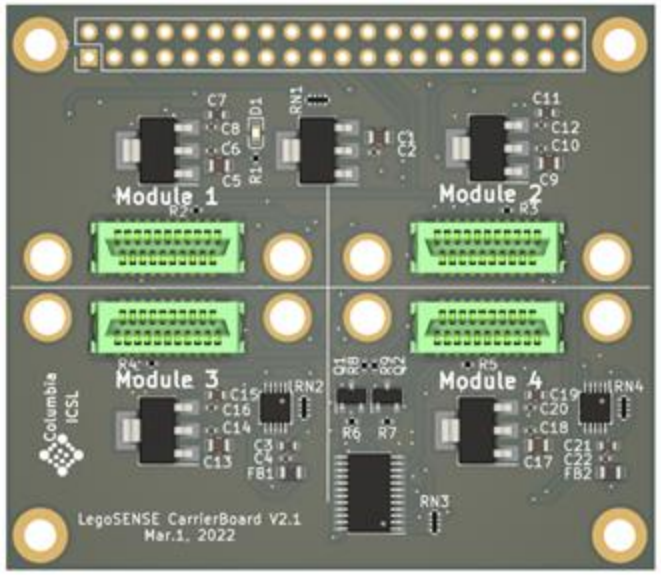
- List of modules
- Real-time graphs
- Configurations
- Download data



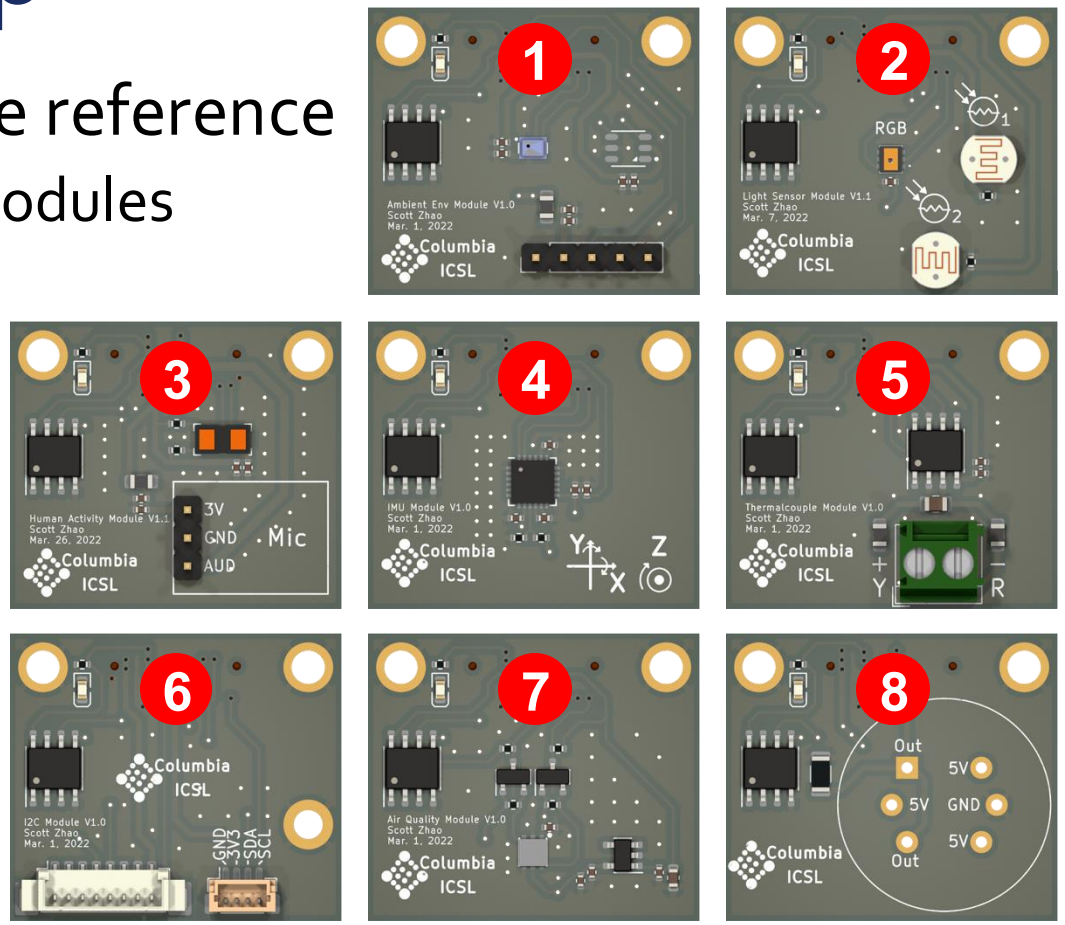


Easy to develop

- Open source hardware reference
  - Carrier board and 8 modules



Carrier Board

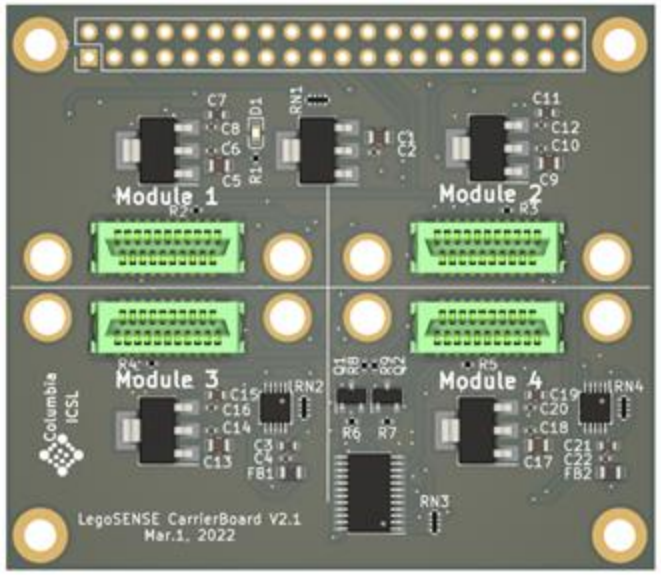


Sensor Modules

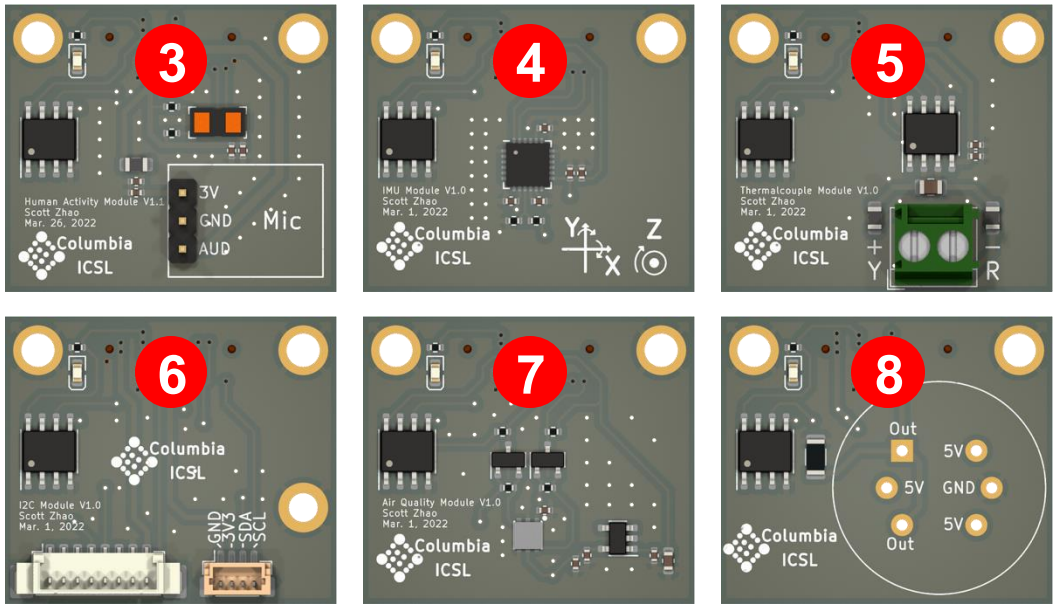
1) Temperature, Humidity, Pressure, Wind Speed
2) Ambient Light Color, Light Intensity
3) Proximity, Microphone
4) Motion (IMU)
5) Thermocouple (Temperature)
6) Particulate Matter (PM1, PM2.5, PM10); Any I2C Sensor with STEMMA QT / Qwiic
7) CO_2, Volatile Organic Compound
8) Gas Density (MQ-X Series)

Easy to develop

- Open source hardware reference
  - Carrier board and 8 modules



Carrier Board



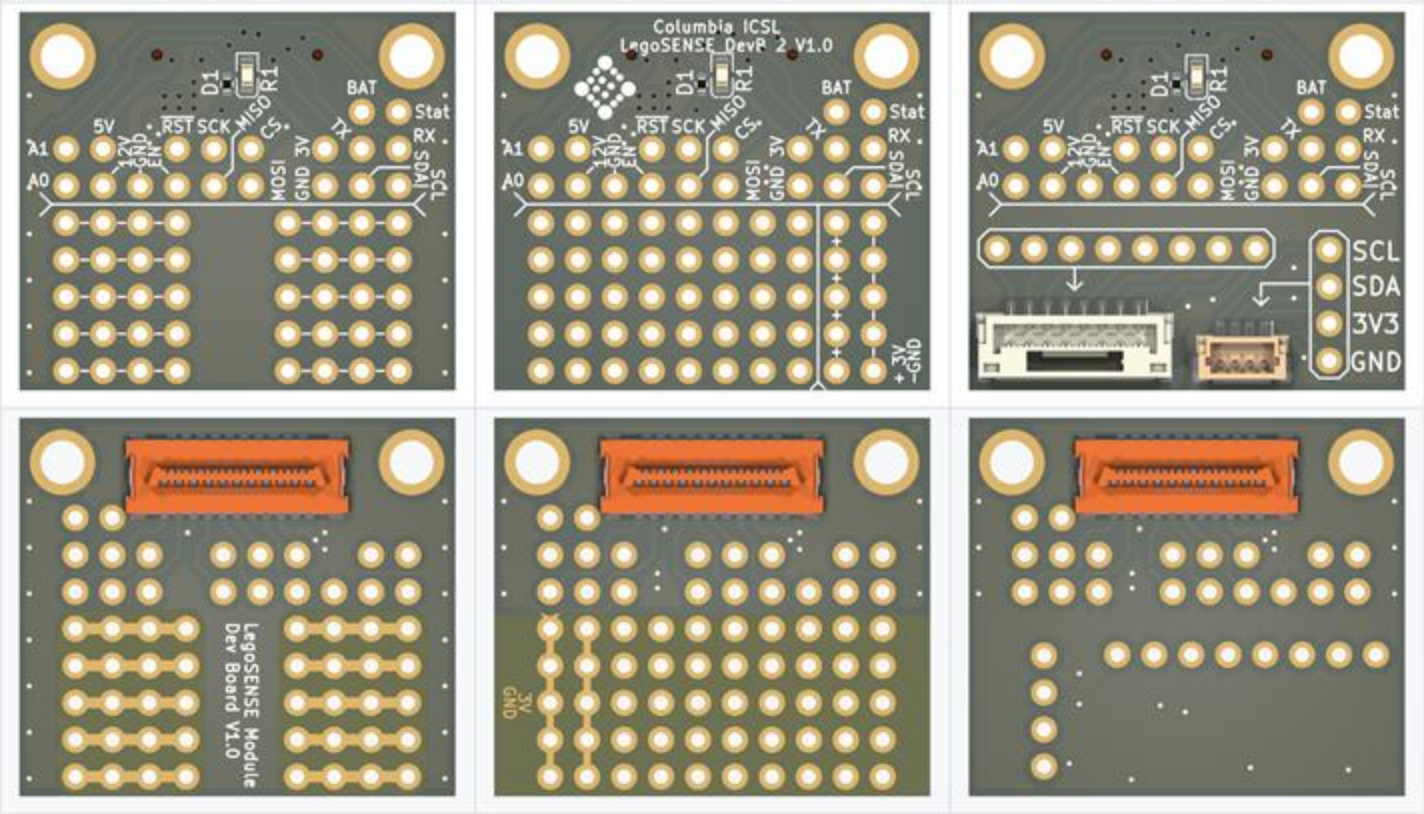
Sensor Modules

1) Temperature, Humidity, Pressure, Wind Speed
2) Ambient Light Color, Light Intensity
3) Proximity, Microphone
4) Motion (IMU)
5) Thermocouple (Temperature)
6) Particulate Matter (PM1, PM2.5, PM10); Any I2C Sensor with STEMMA QT / Qwiic
7) CO_2, Volatile Organic Compound
8) Gas Density (MQ-X Series)

https://github.com/Columbia-ICSL/LegoSENSE/tree/main/hardware

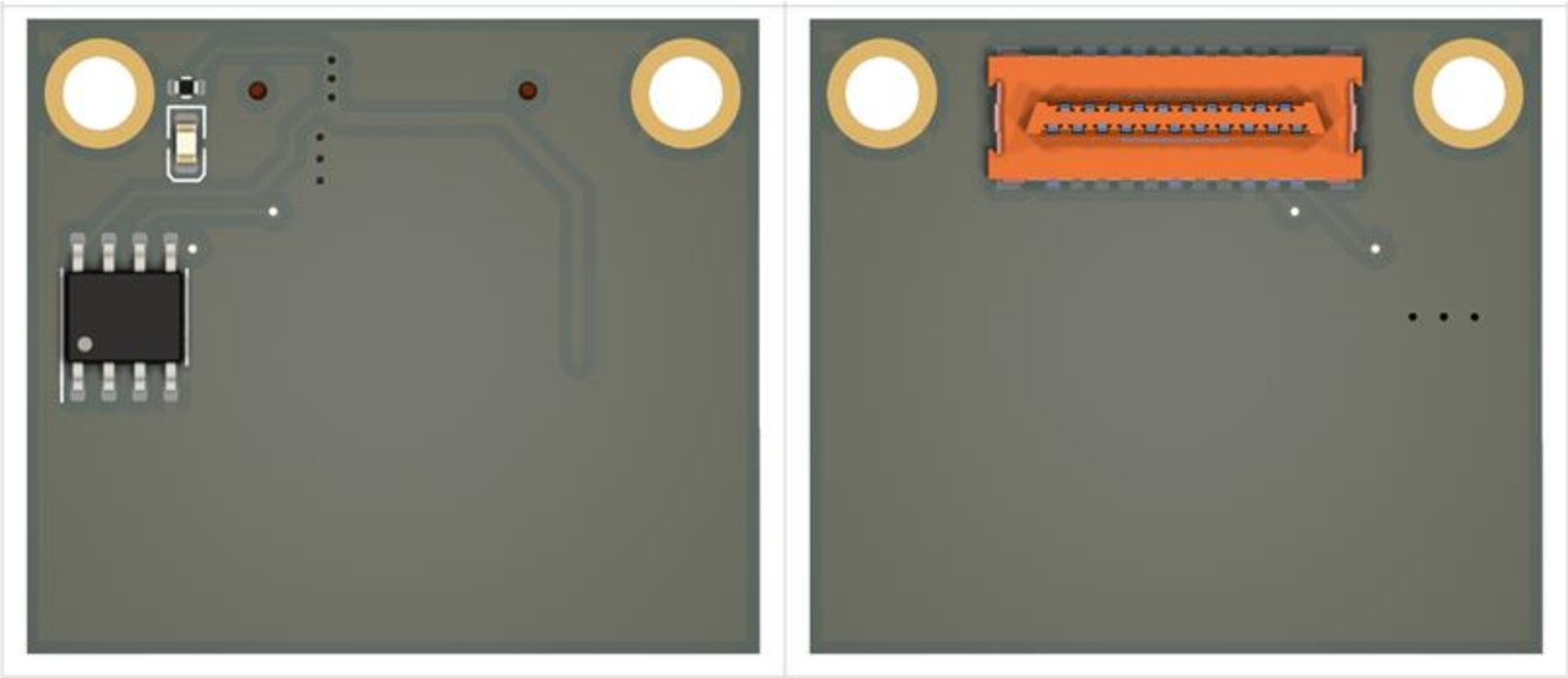
Easy to develop

- Three development boards for prototyping

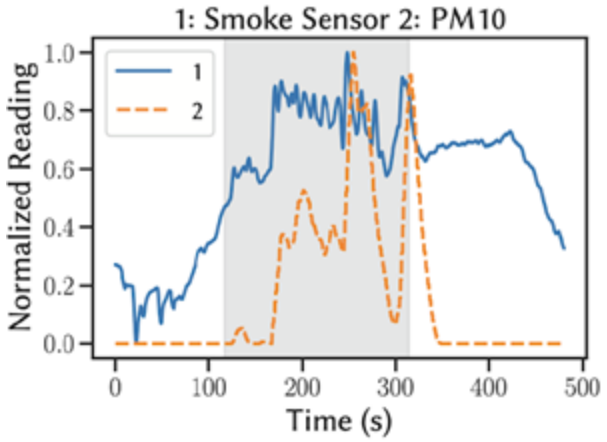


Easy to develop

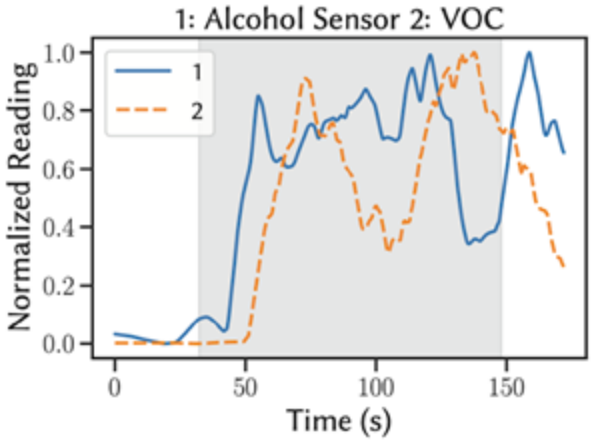
- Template module board and driver



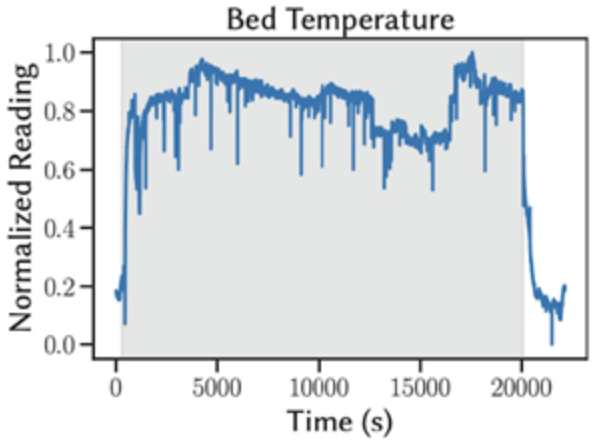
Evaluation – Home activity detection



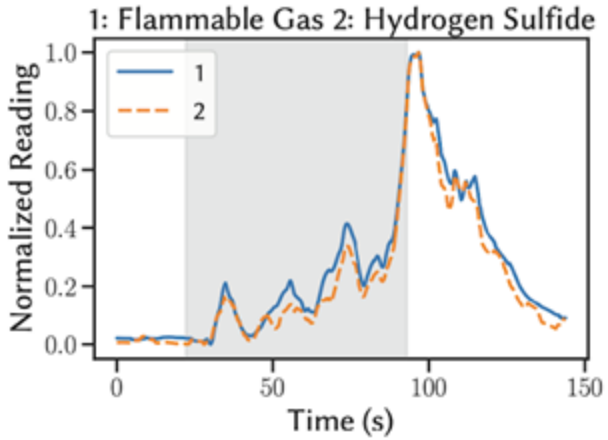
(a) Smoke



(b) Alcohol



(e) Sleep Temperature

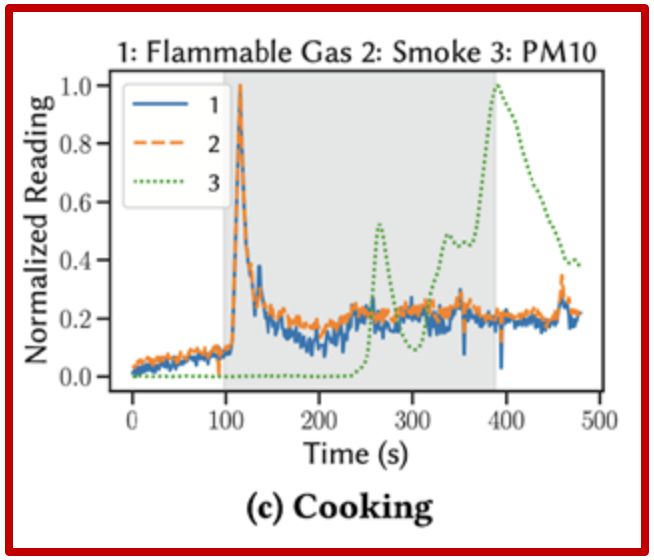


(f) Gas Leak

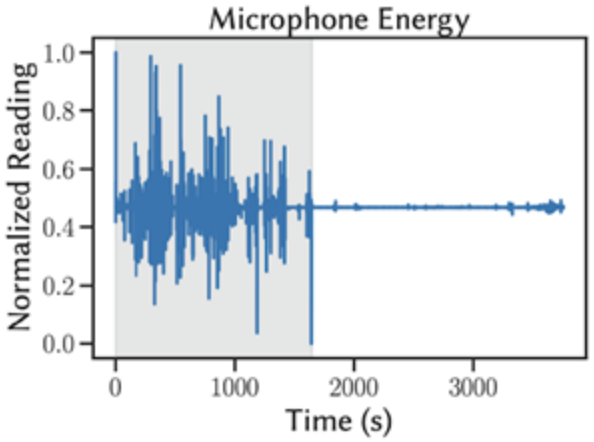
(g) Intrusion

(h) Fall

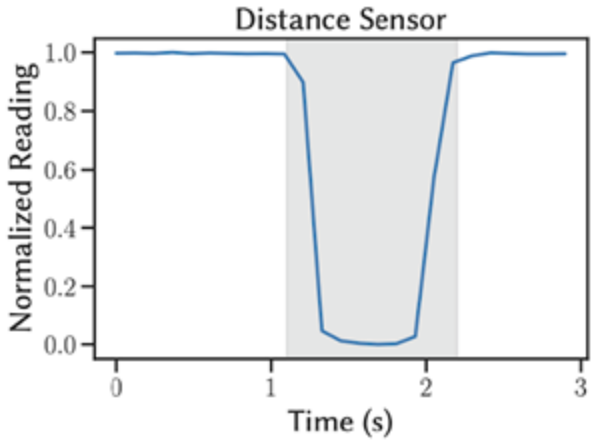
Evaluation – Home activity detection



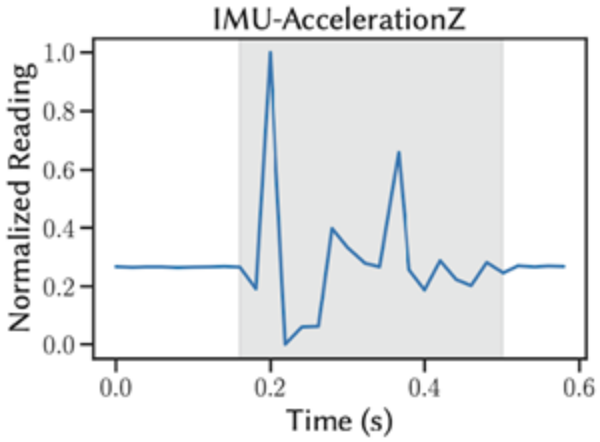
(c) Cooking



(d) Social Activity

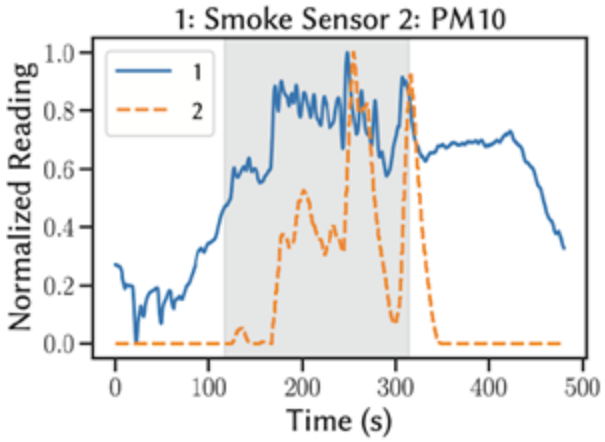


(g) Intrusion

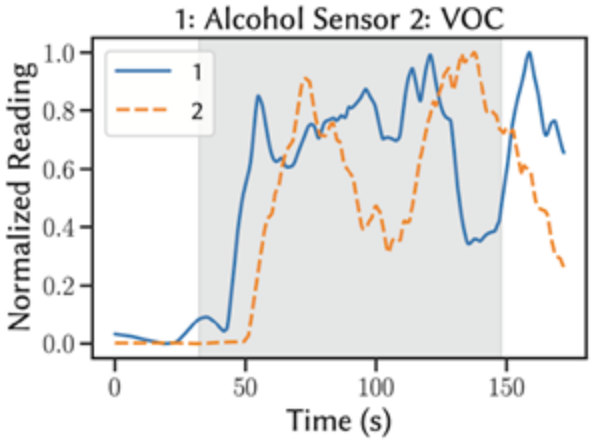


(h) Fall

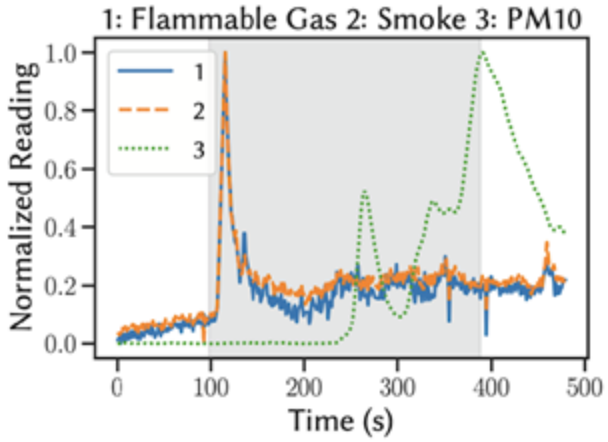
Evaluation – Home activity detection



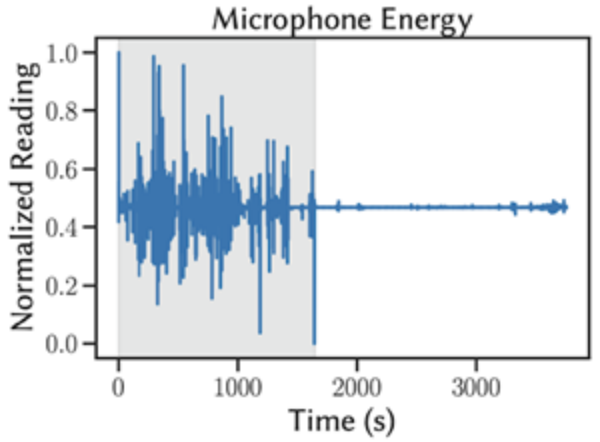
(a) Smoke



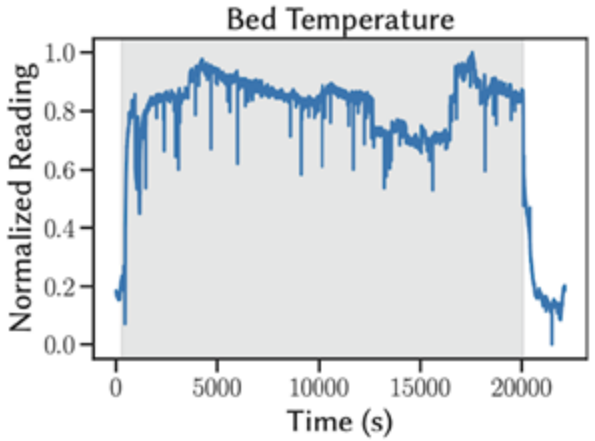
(b) Alcohol



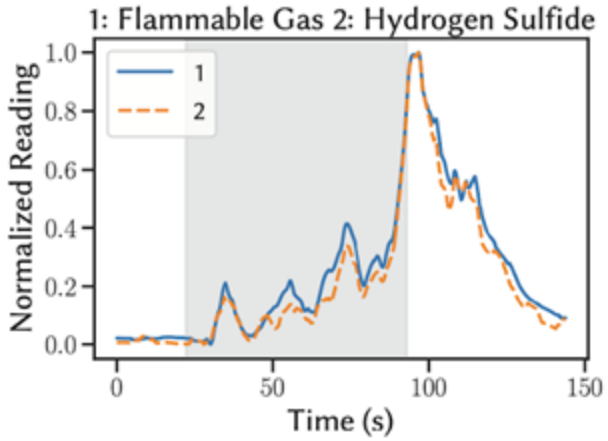
(c) Cooking



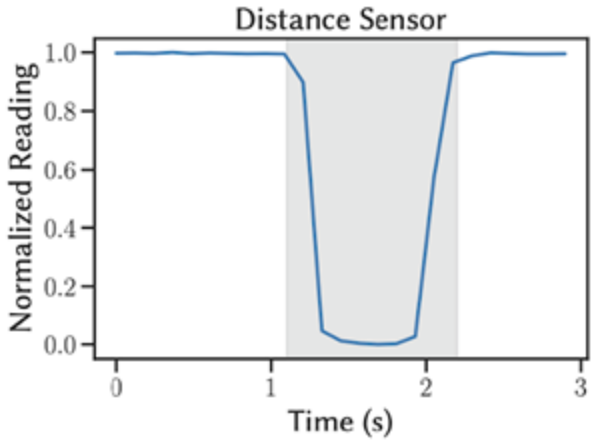
(d) Social Activity



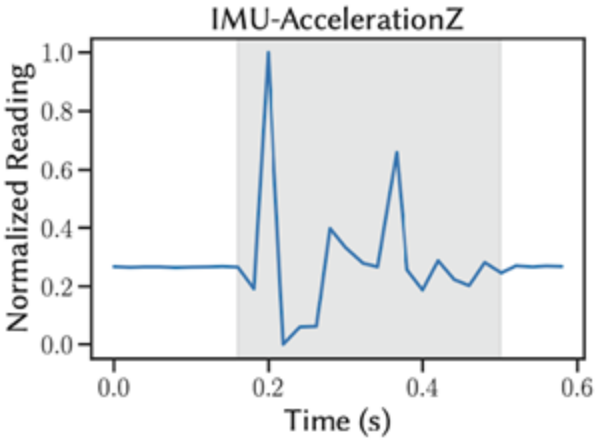
(e) Sleep Temperature



(f) Gas Leak

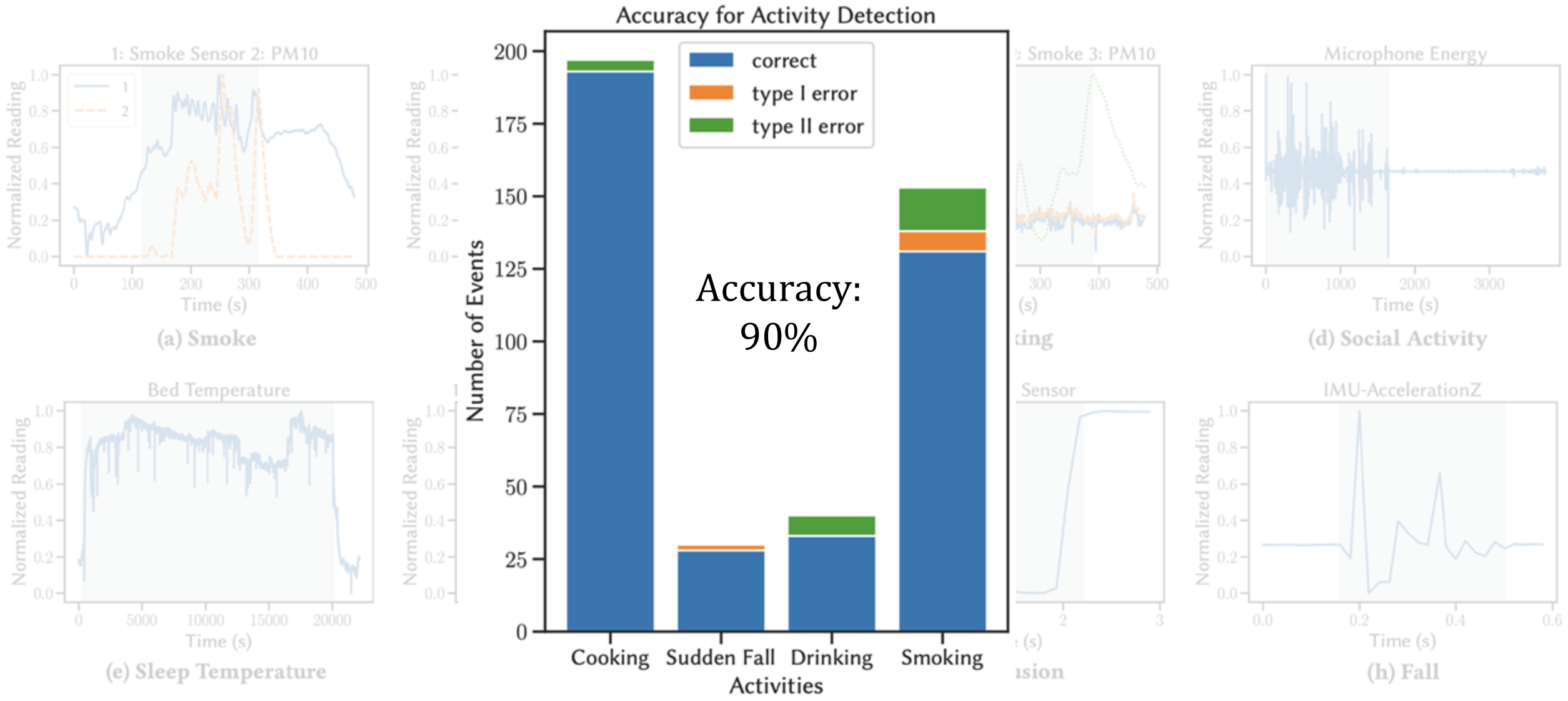


(g) Intrusion



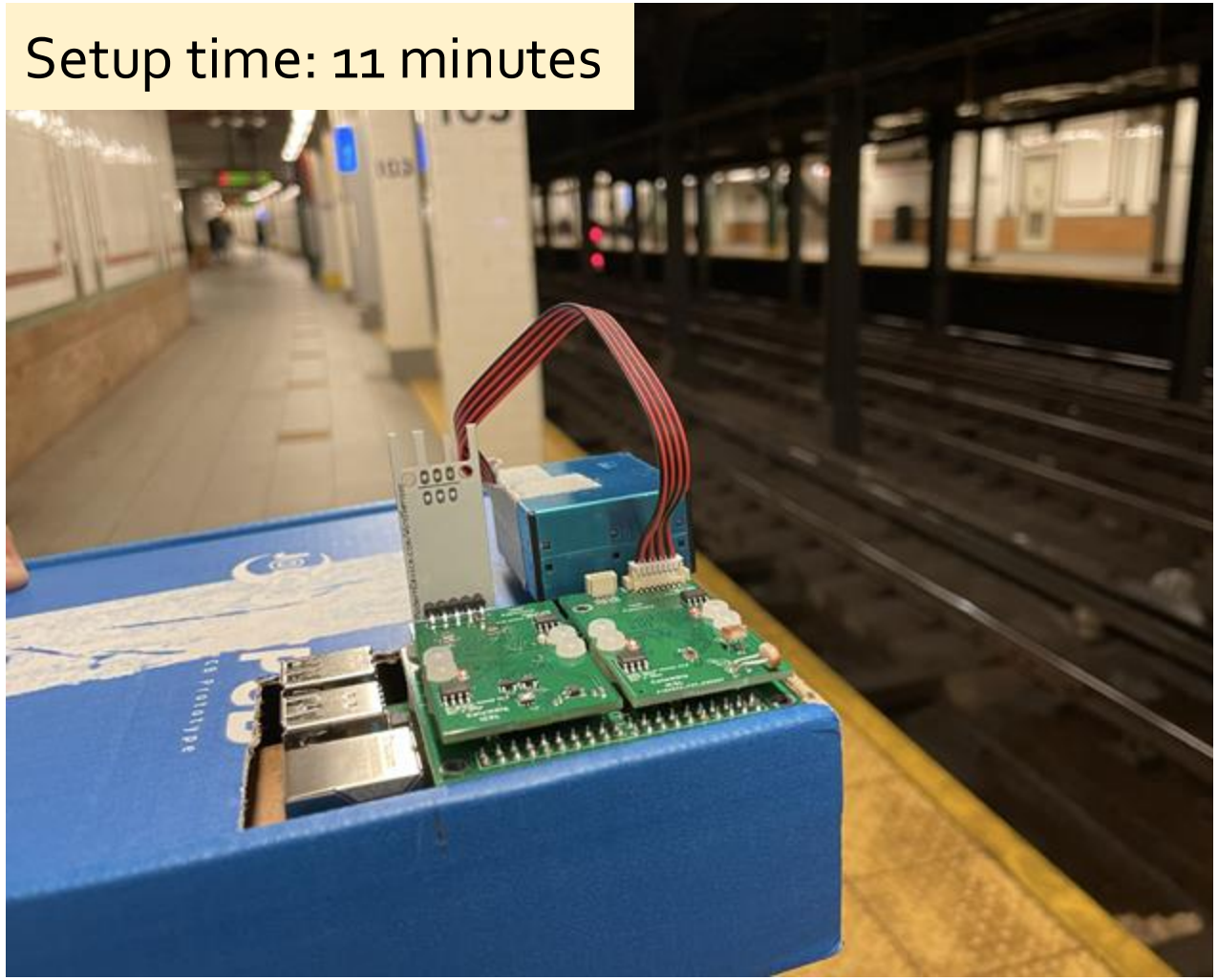
(h) Fall

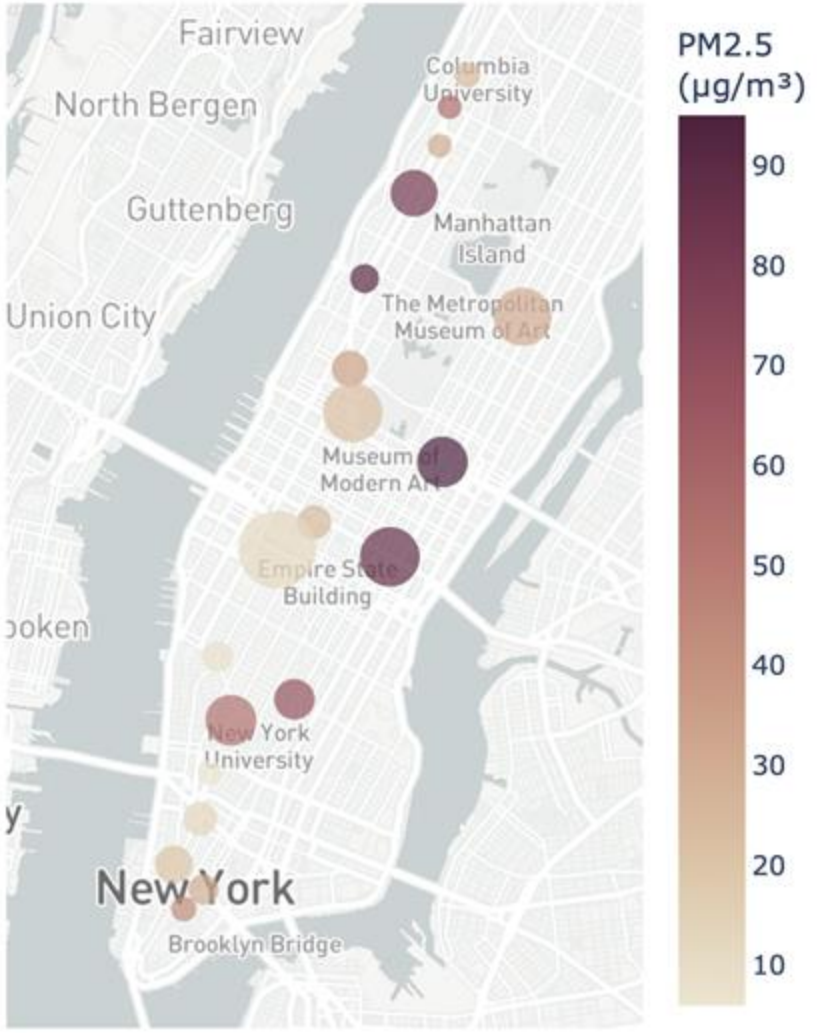
Evaluation – Home activity detection



Evaluation – Subway air quality

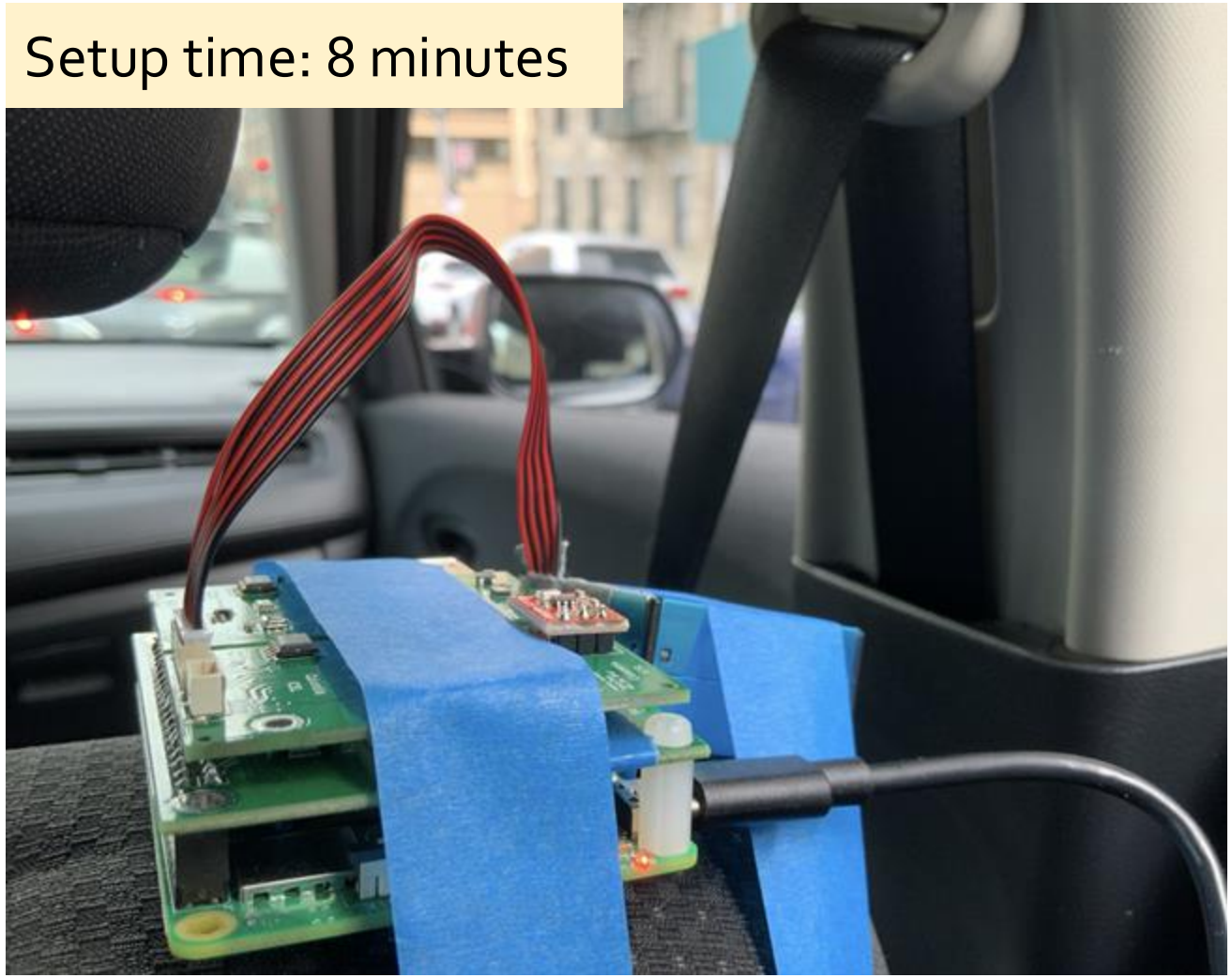
Setup time: 11 minutes

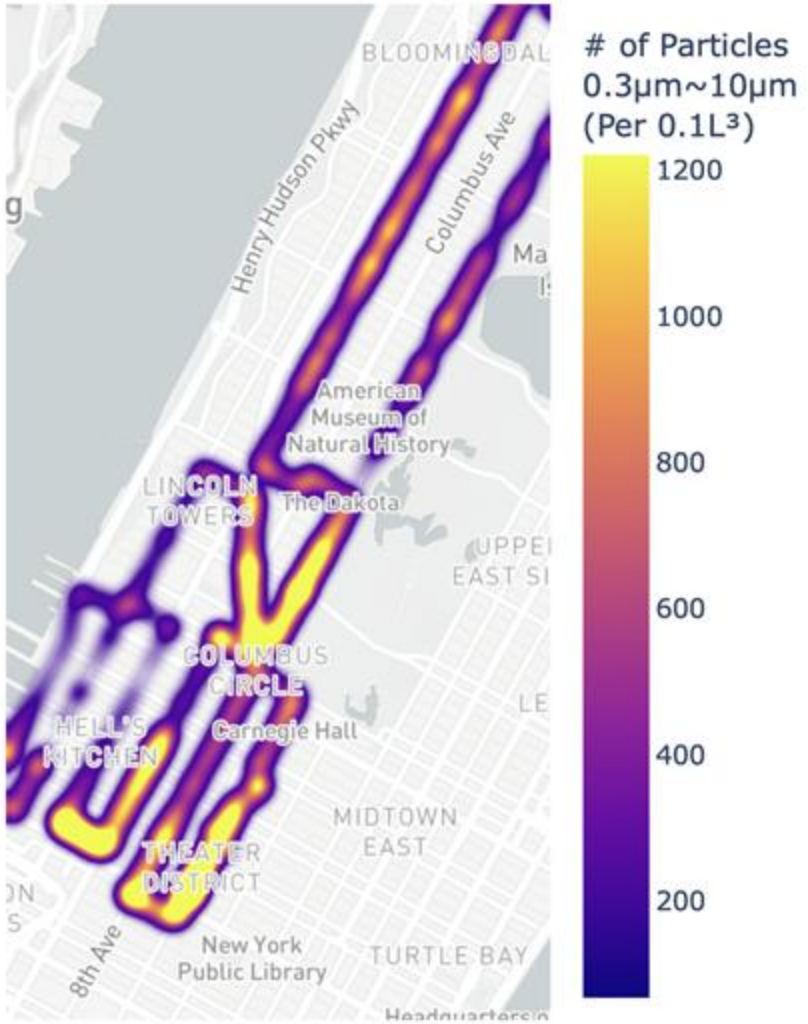




Evaluation – Manhattan air quality

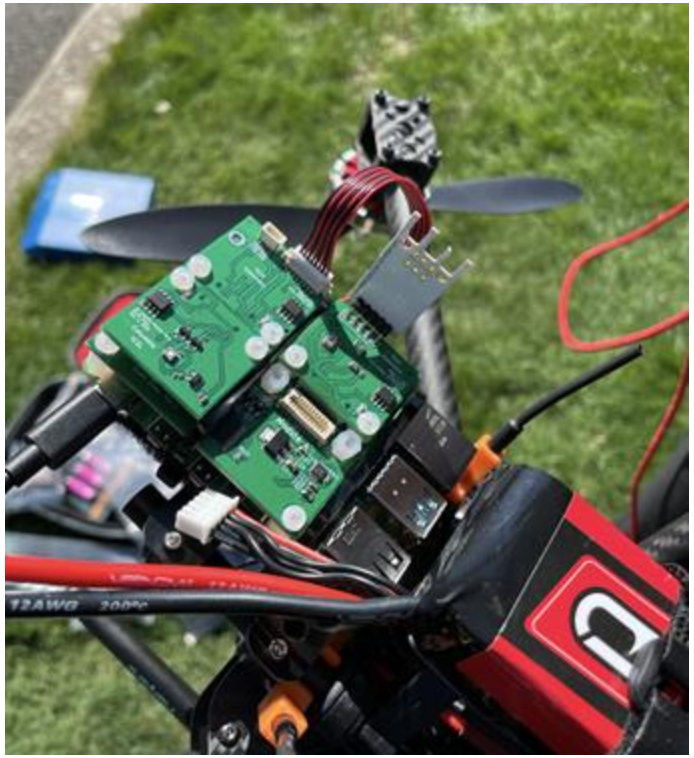
Setup time: 8 minutes



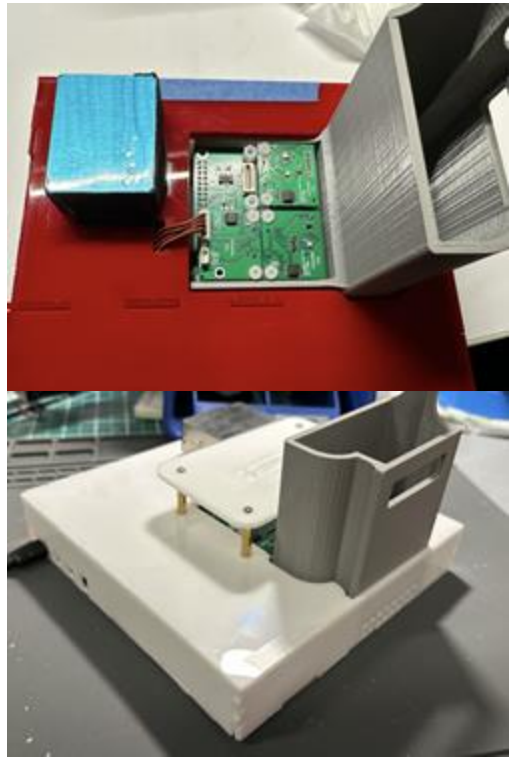


Outreach and additional deployments

K-12 outreach



LegoSENSE on drones

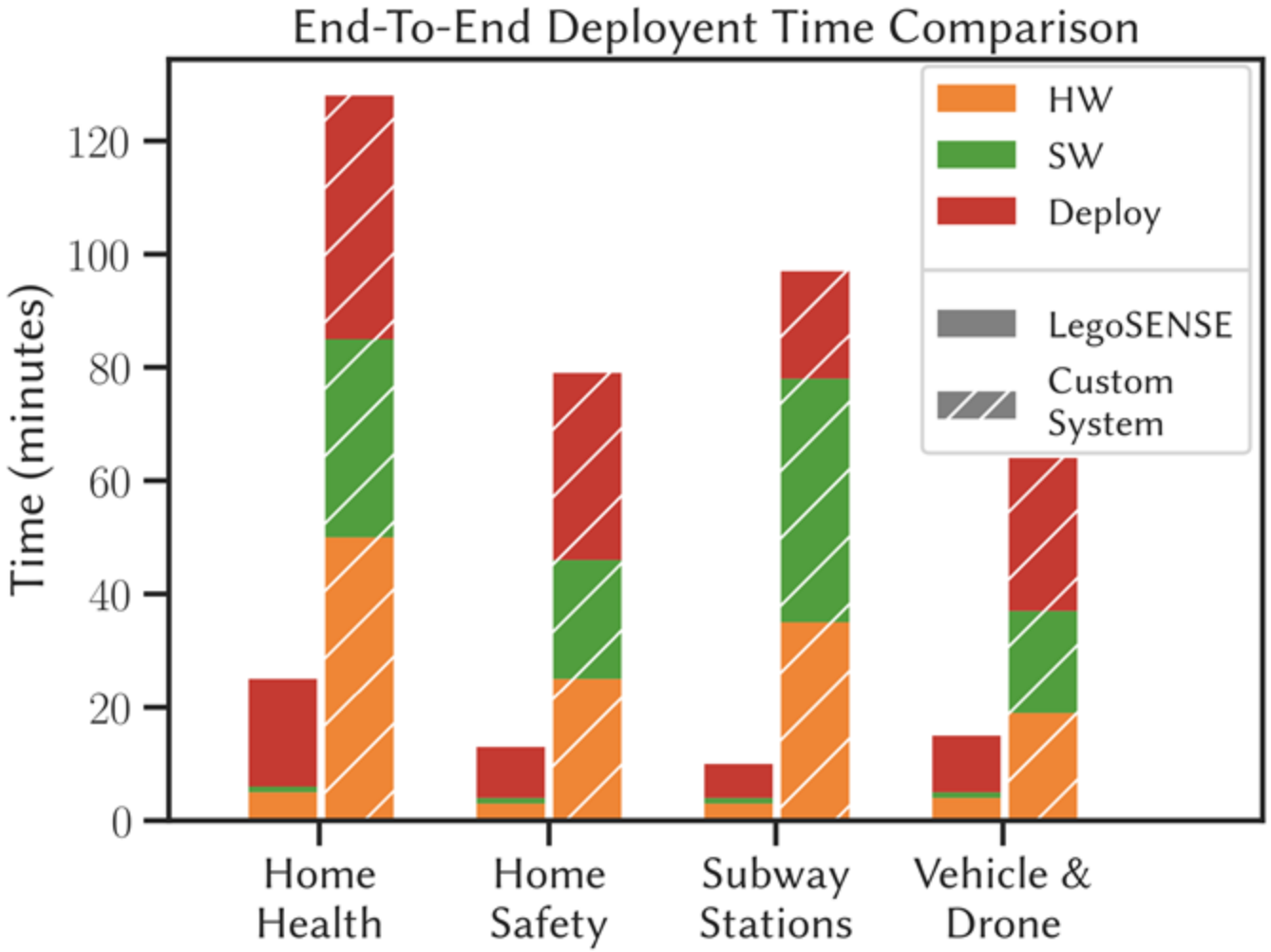


Asthema Patient Home

Evaluation – Deployment time

LegoSENSE

Custom Setup



LegoSENSE: An Open and Modular Sensing Platform for Rapidly-Deployable IoT Applications

mz2866@columbia.edu

jiang@ee.columbia.edu

LegoSENSE

Custom System

Hardware and software open source at https://github.com/Columbia-ICSL/LegoSENSE