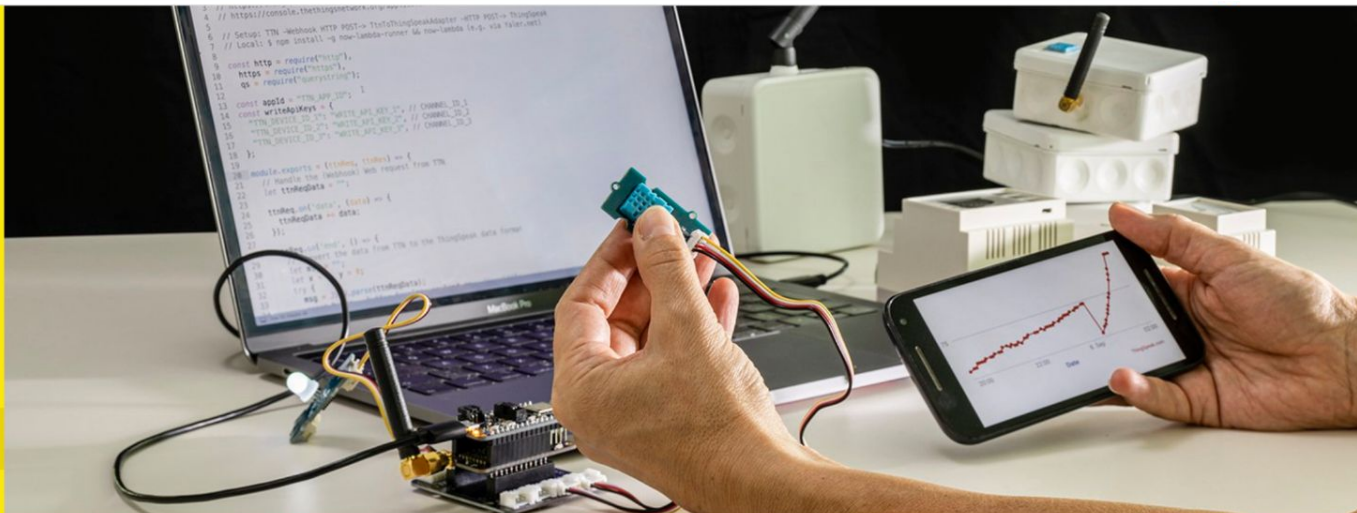


Urban IoT Prototyping with LoRaWAN

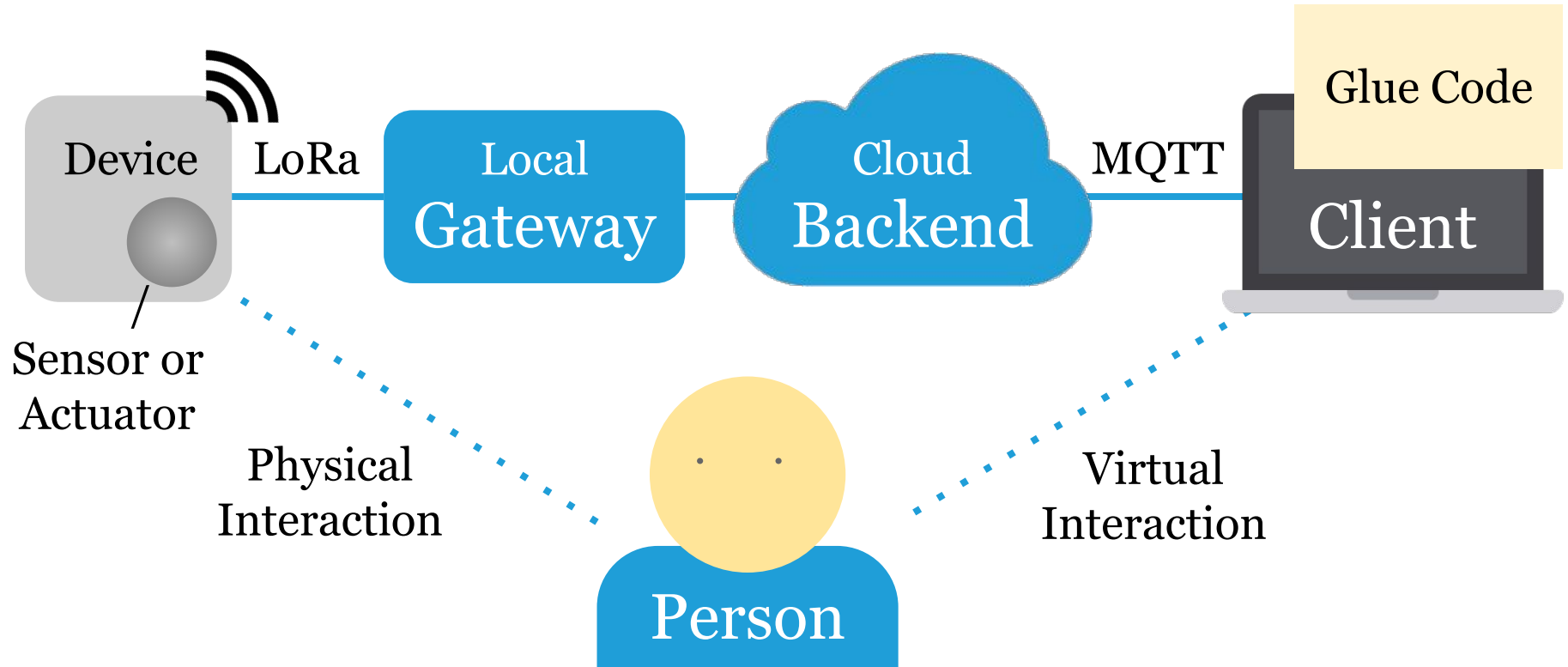
CC BY-SA, thomas.amberg@fhnw.ch
Screenshots considered "fair use"



Institute of Mobile and Distributed Systems

We are a multidisciplinary computer sciences institute specialising in the Internet of Things (IoT) and IT security.

IoT architecture



IoT bricks*

Building blocks with connectivity and long battery life

Focused on distributed use cases, no-touch hardware

Enable room-, building- and city-scale prototyping

Client-side Java SDK for writing clean glue code

*Or **FHNW IoT bricks**, to avoid confusion



Hardware
prototypes of
camera, LED
and distance
bricks



Bricks serve open data — each brick has a public URL and a unique ID

IoT bricks SDK

```
// org.eclipse.paho.client.mqttv3-1.2.3.jar  
// fhnw-iot-bricks-0.0.1.jar
```

```
String BASE_URL = "brick.li"; // as printed  
String BRICK_ID = "0000-0001"; // on the brick
```

```
Proxy proxy = MqttProxy.fromConfig(BASE_URL);  
TempBrick brick = TempBrick.connect(proxy, BRICK_ID);
```

```
double temp = brick.getTemperature(); // this brick  
int battery = brick.getBatteryLevel(); // all bricks  
String time = brick.getTimestampIsoUtc();
```

Monitoring example

```
Proxy proxy = MqttProxy.fromConfig(BASE_URL);  
TempBrick tempBrick = TempBrick.connect(proxy, TEMP_ID);  
LedBrick ledBrick = LedBrick.connect(proxy, LED_ID);  
  
while (true) {  
    double temp = tempBrick.getTemperature();  
    Color color = temp > 23.0 ? Color.RED : Color.GREEN;  
    ledBrick.setColor(color);  
    proxy.waitForUpdate();  
}
```

Doorbell example

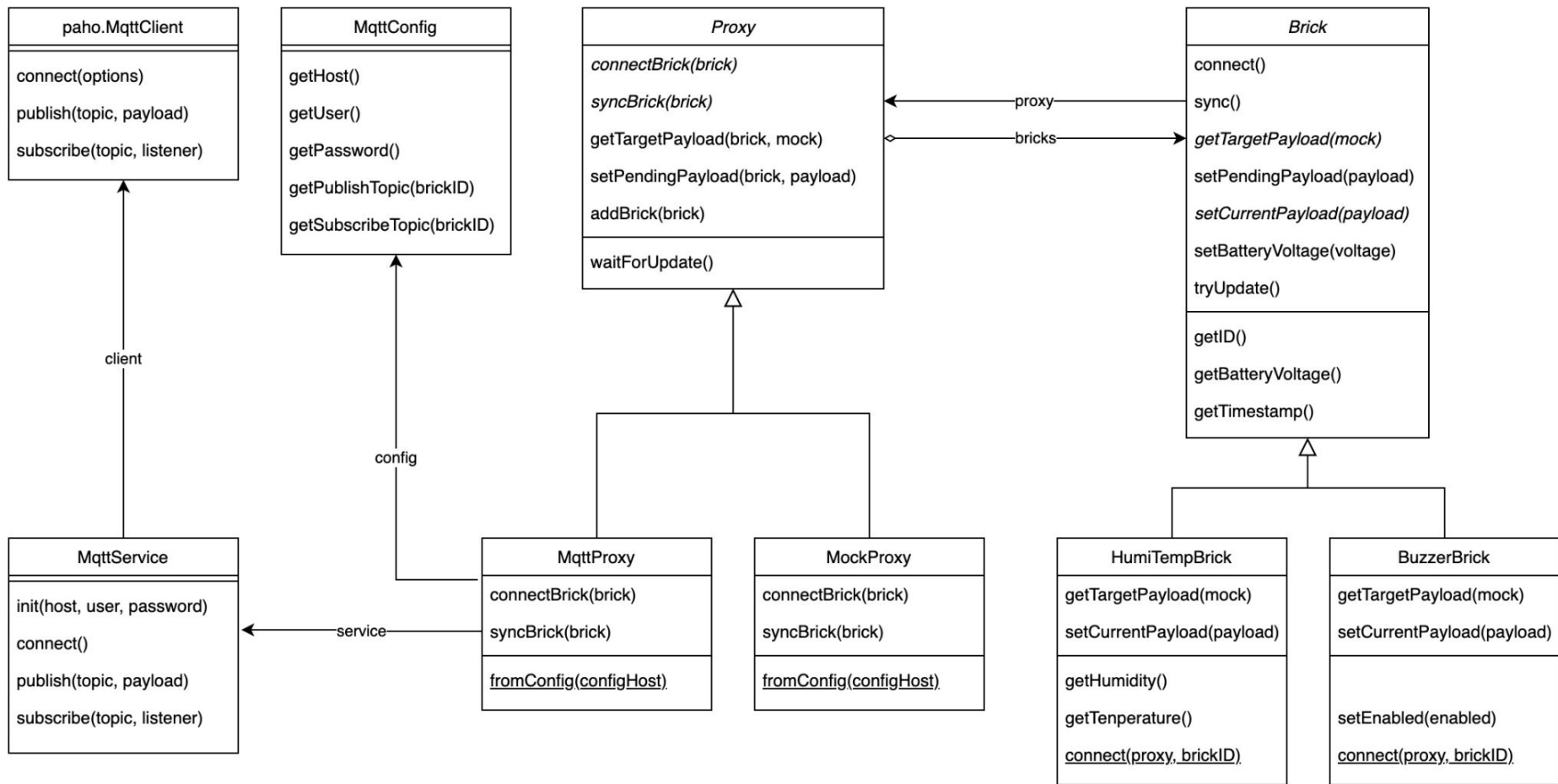
```
Proxy proxy = MqttProxy.fromConfig(BASE_URL); // fast, < 1"
// or proxy = LoRaProxy.fromConfig(BASE_URL); // slow, > 5'
ButtonBrick button = ButtonBrick.connect(proxy, BTN_ID);
BuzzerBrick buzzer = BuzzerBrick.connect(proxy, BZR_ID);

while (true) {
    boolean pressed = button.isPressed();
    buzzer.setEnabled(pressed);
    proxy.waitForUpdate();
}
```

Counter example

```
Proxy proxy = MockProxy.fromConfig(BASE_URL);
ButtonBrick button = ButtonBrick.connect(proxy, BTN_ID);
DisplayBrick display = DisplayBrick.connect(proxy, D_ID);

int count = 0; // could be stored on disk for persistency
while (true) {
    if (button.isPressed()) {
        count = count + 1;
    }
    display.setValue(count); // no overhead if same value
    proxy.waitForUpdate();
}
```

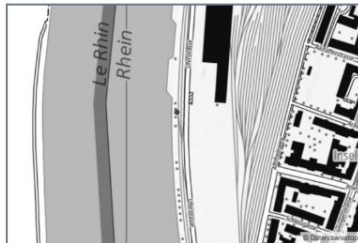


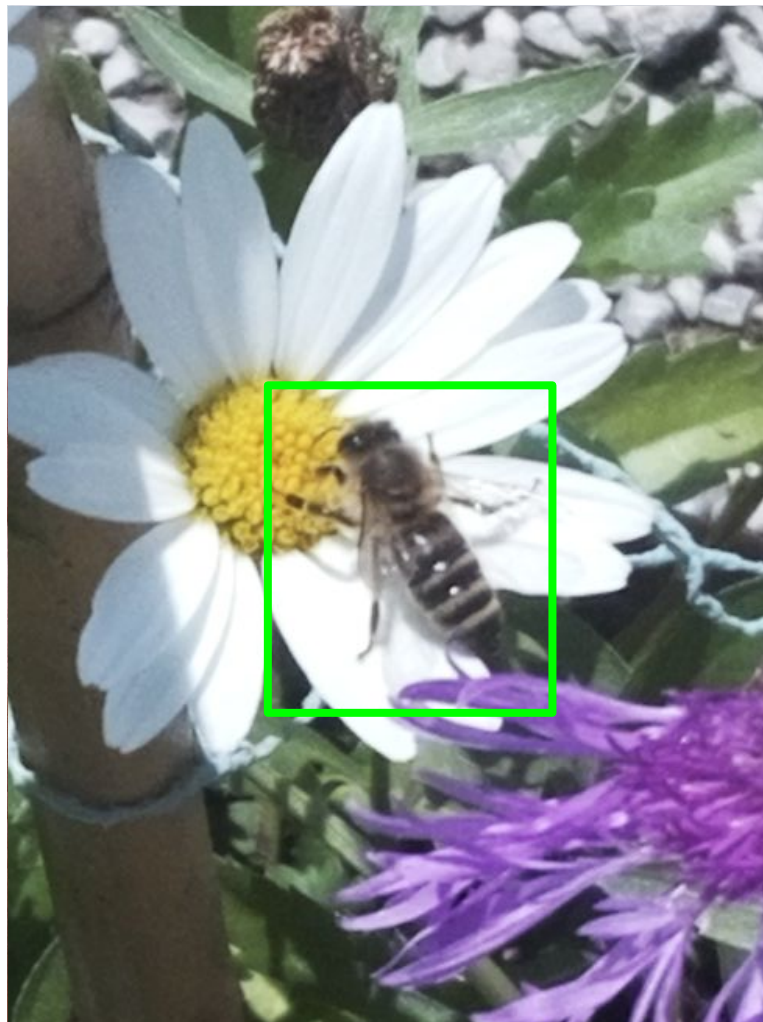
Mitwelten — Medienökologische Infrastrukturen für Biodiversität

Interdisziplinäres Designforschungsprojekt (SNF, 2020-2024)

Wie können mediale Designinterventionen auf Grundlage des Internets der Dinge ökologisch und kulturell zur Förderung von Biodiversität in lokalen Ökosystemen beitragen?

Zur Beantwortung dieser Leitfrage werden in drei Fallstudien in Mensch-Umwelt-Aushandlungsprozessen medientechnologische Infrastrukturen gestaltet und installiert: in der Kulturlandschaft der Merian Gärten bei Basel, in einem ehemaligen Hafengebiet der Stadt Basel und in einem Naturschutzgebiet in der Agglomeration Basels.



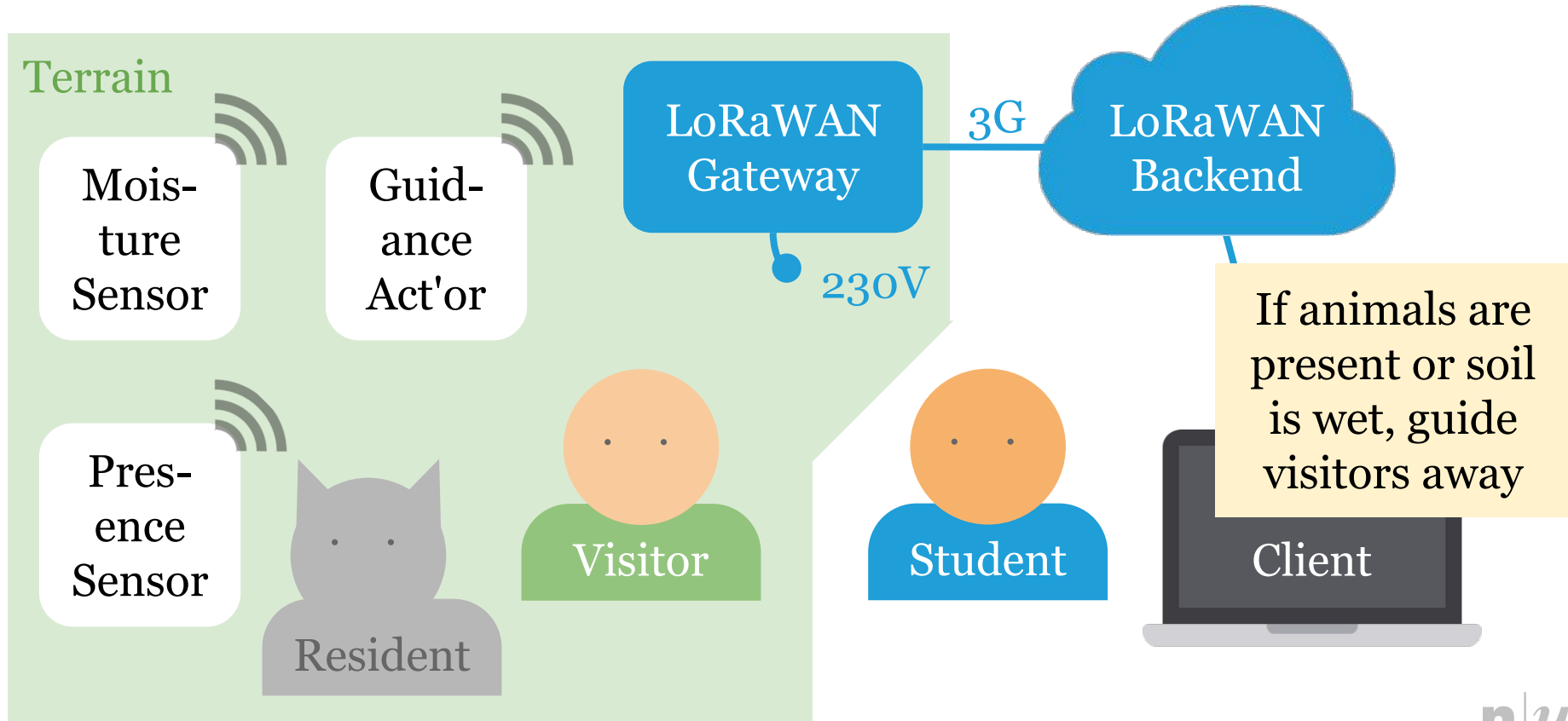


Vision
sensor,
detects
bees on
flowers

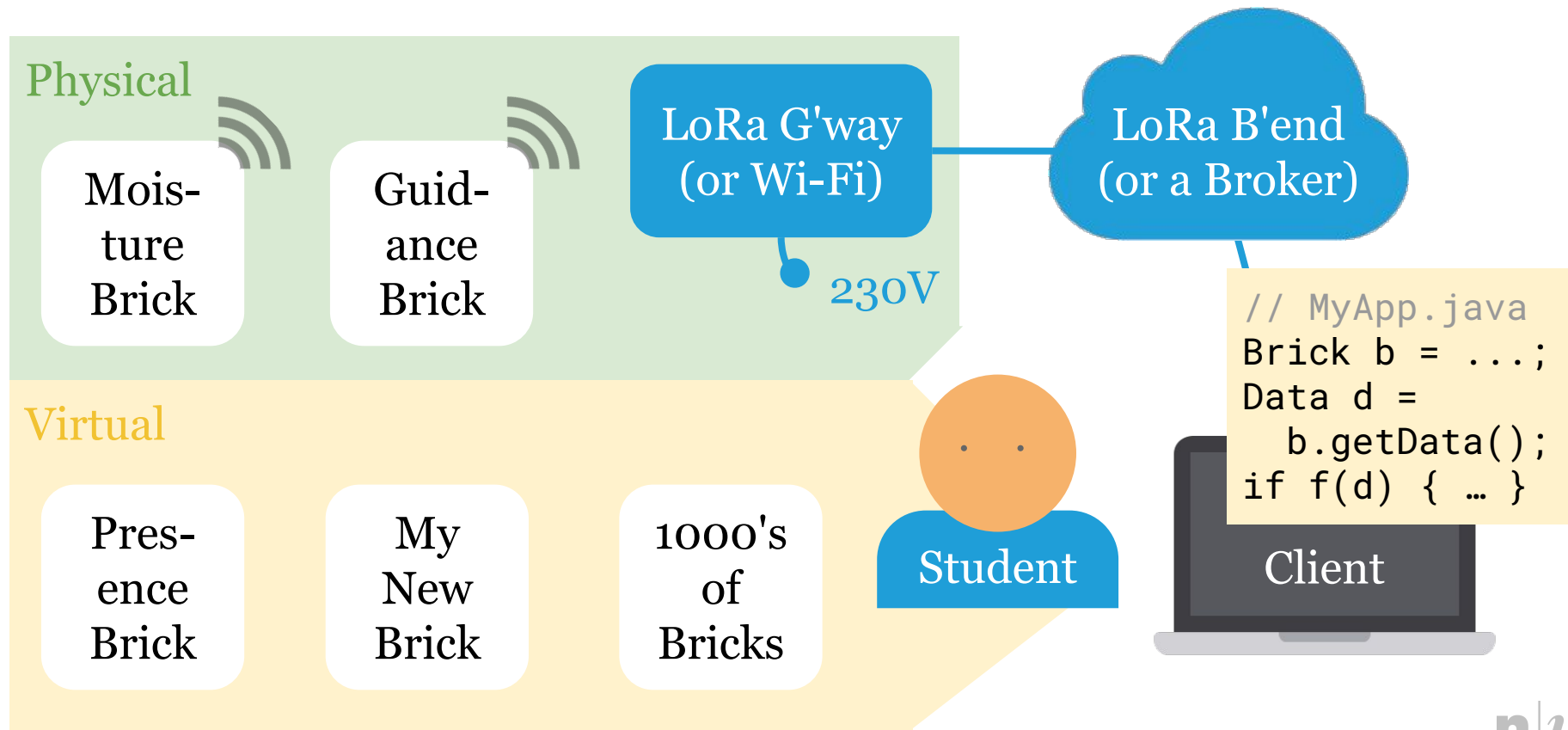


Sensors
and ac-
tuators
to guide
visitors

IoT deployment



IoT prototyping



Guidance system example

```
MqttProxy p = MqttProxy.fromConfig(BASE_URL);
PresenceBrick ps[] = { PresenceBrick.connect(p, P_ID), ... };
GuidanceBrick gs[] = { GuidanceBrick.connect(p, G_ID), ... };
BrickPlacement xs[] = { new BrickPlacement(ps[0],E,N), ... };

while (true) {
    PresenceBrick p = getMostActive(ps); // TODO
    for (GuidanceBrick g : gs) {
        pointAwayFrom(g, p, xs); // TODO
    }
    proxy.waitForUpdate();
}
```

Mixing real and mock sensors

```
Proxy mqttProxy = MqttProxy.fromConfig(BASE_URL);
Proxy mockProxy = MockProxy.fromConfig(BASE_URL);
... // create bricks, using mqtt or mock proxy
ProxyGroup proxies = new ProxyGroup();
proxies.add(mqttProxy);
proxies.add(mockProxy);

while (true) {
    ... // use bricks
    proxies.waitForUpdate();
}
```

Adding a new type of sensor / actuator

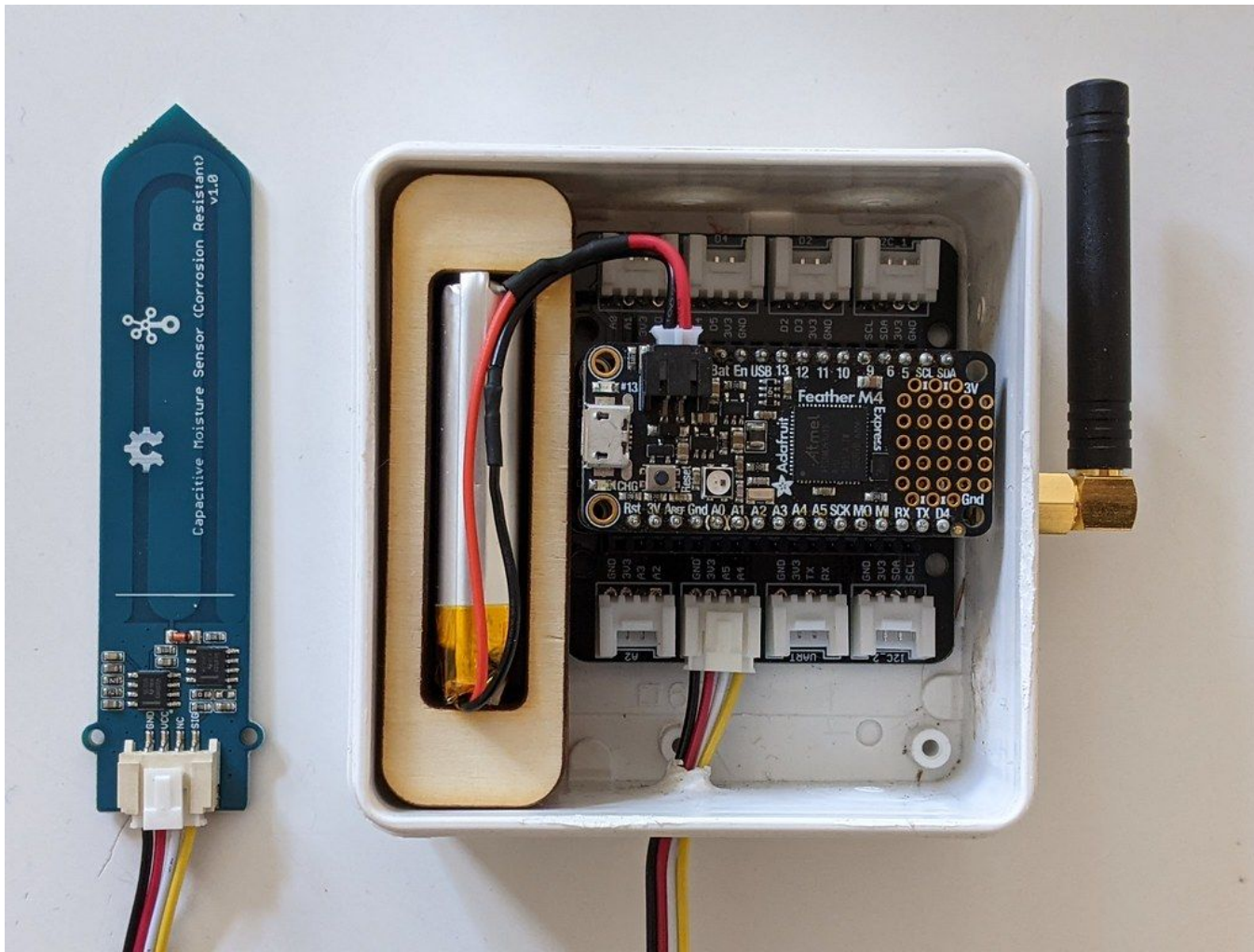
Create **Java** class MyNewBrick extending Brick

Create **Arduino** code MyNewBrick.ino

Create the physical brick design

Reuse* makes this easy

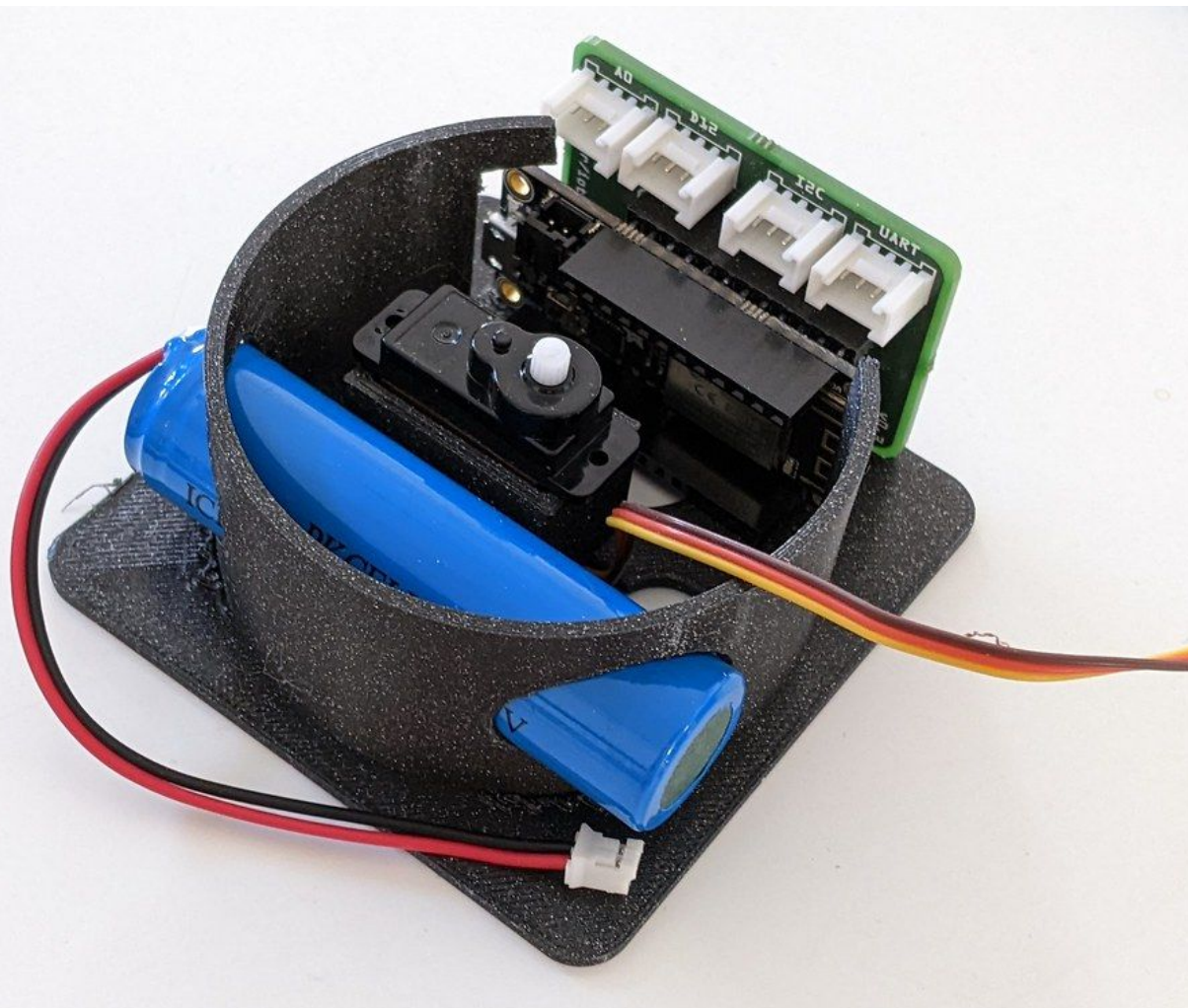
*Hard parts are abstracted away, or can be copied **n|w**



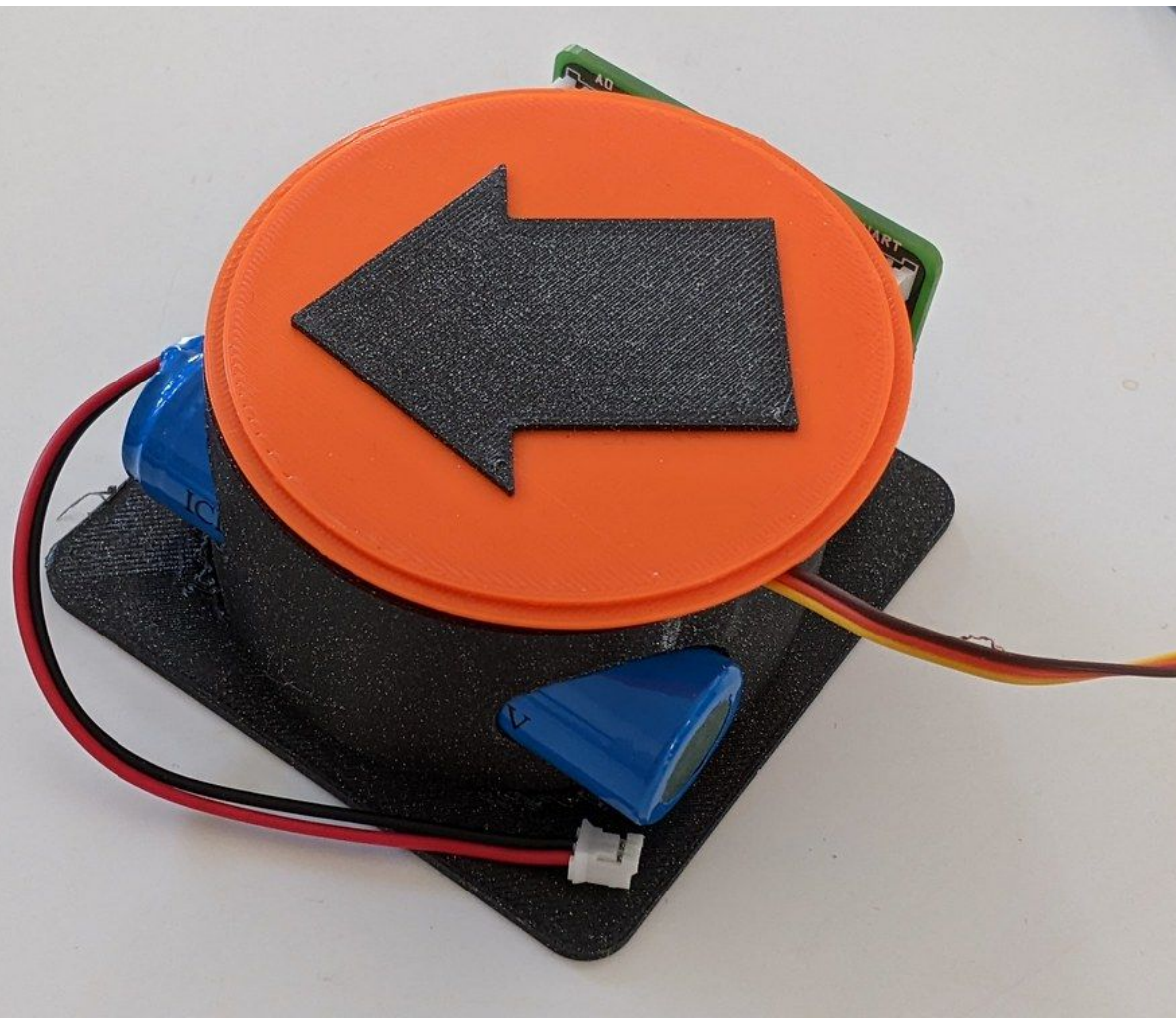
LoRaWAN
node with an
Feather M4, a
soil moisture
sensor and a
LiPo battery



Production in
the lab,
fixture
for solar
panels



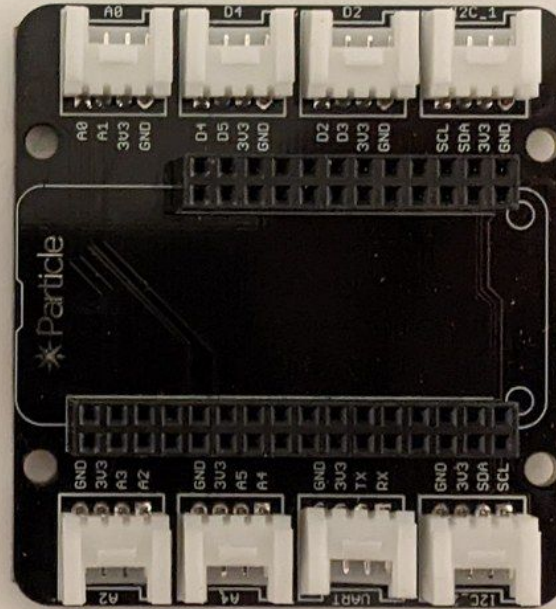
3D printed
adapter for
electronics,
servo and a
2200 mAh
Li-Ion cell



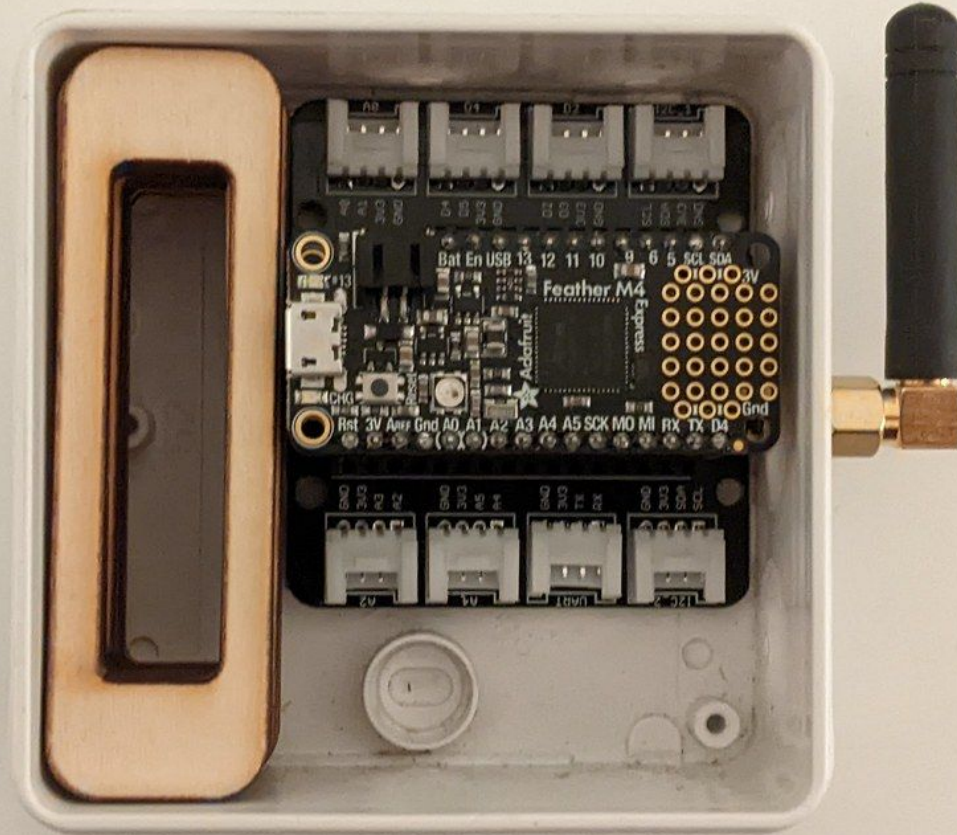
3D printed,
2-colored
arrow that
snap fits to
servo axle



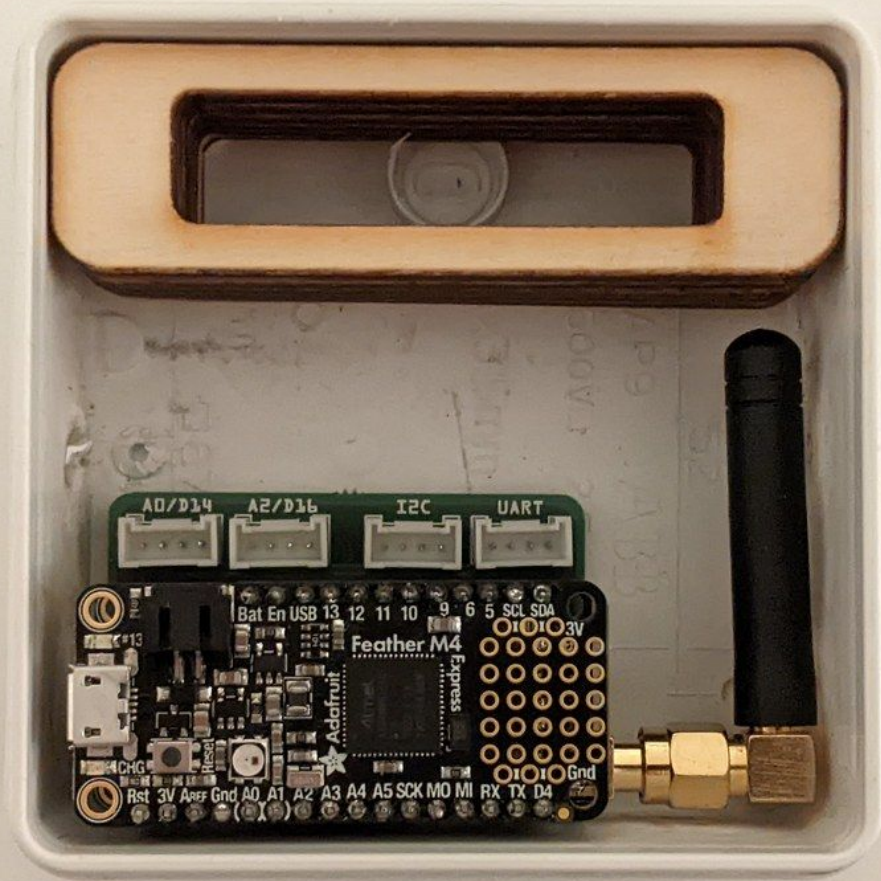
AP9 box with
laser-cut face
plate, fully
mounted



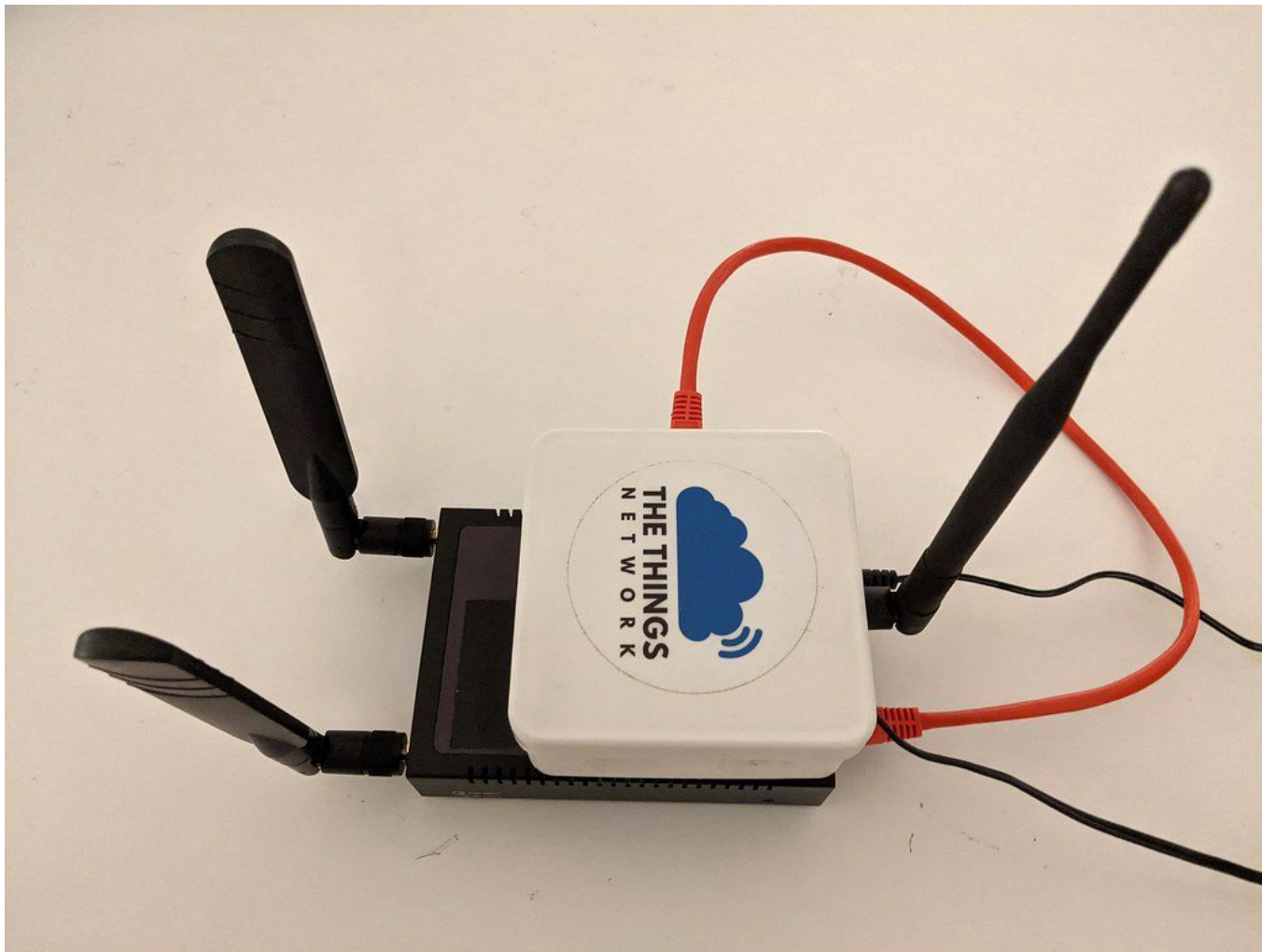
Custom Grove
adapter made
with Fritzing



Particle Grove
adapter for
reference



Smaller size
allows to keep
the antenna
in the box



LoRaWAN
gateway w/
3G uplink to
TTN backend

Make, Learn, Share

<https://github.com/mitwelten/mitwelten-iot-hardware-poc>

<https://github.com/tamberg/fhnw-iot-bricks>

Everything shown is open source

Designs are open hardware

It's work in progress

Thank you.