

Introduktion till Internet of Things

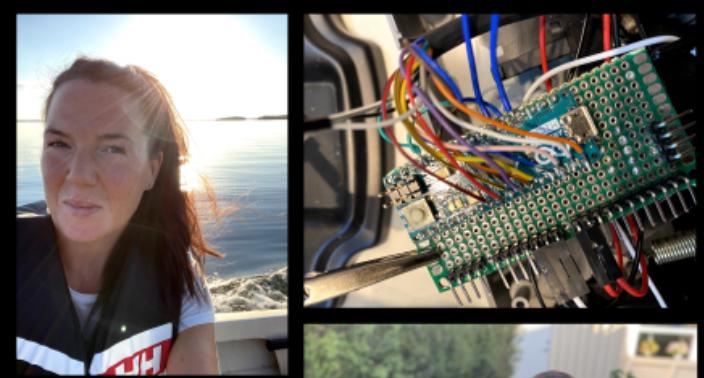
Sensorer, Mjukvara och Plattformar

Agenda

- Intro
- Hårdvara
- Mjukvara
- Plattform
 - Home Assistant
- Lab01-1
- Lab01-2
- Lab01-3
- Plattform
 - ThingsBoard
- Lab02
- Uppgift

Intro

```
    .on("click", function() {
        if (!this.paused) {
            this.paused = true;
            this.pause();
        } else {
            this.paused = false;
            this.resume();
        }
    });
});
```

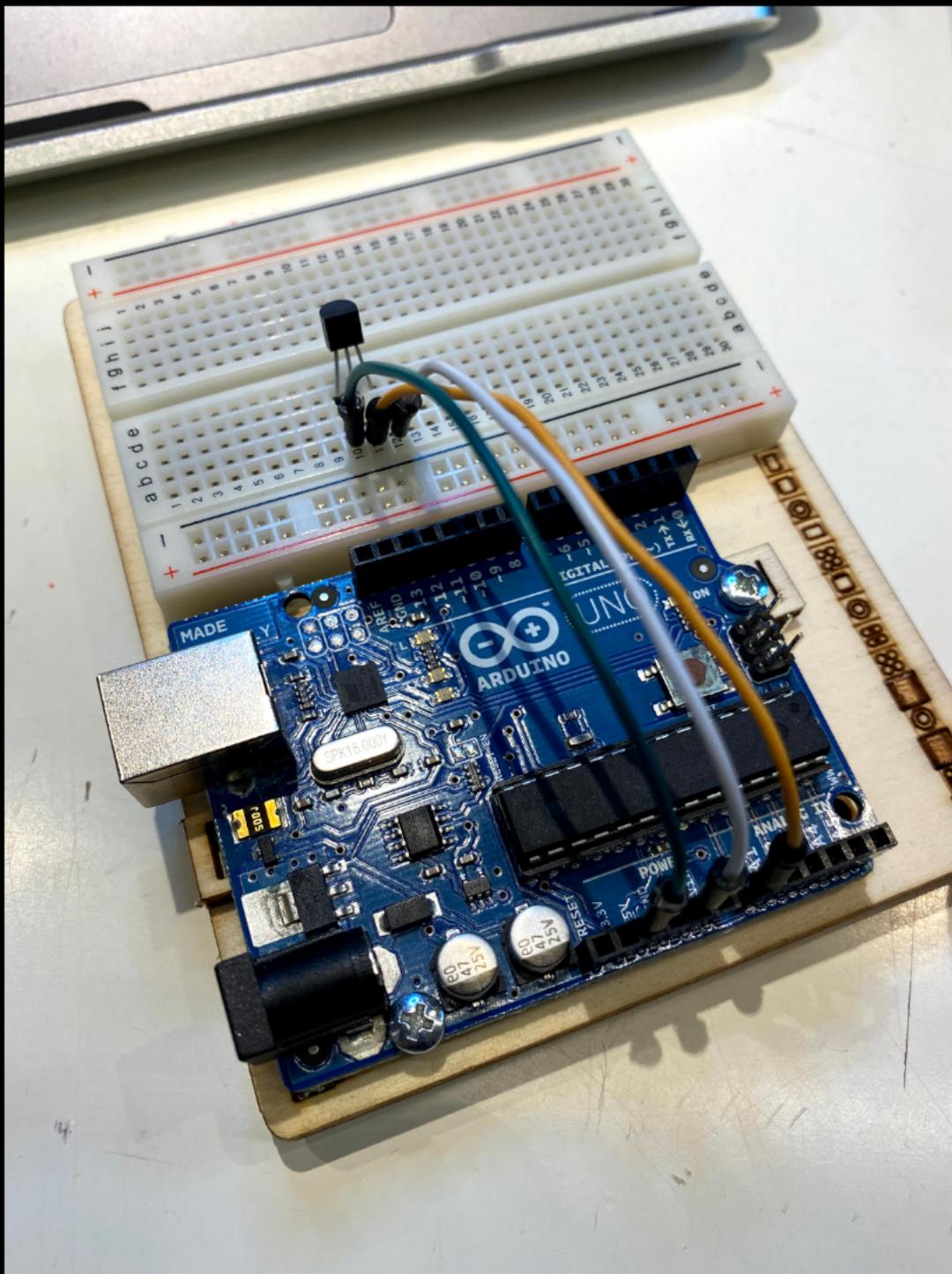


Emil Nilsson

Tinkerer, Coder, Father, Husband, Flyer

Hårdvara

Sensorer, Microkontroller, Gateways, osv.



Mjukvara

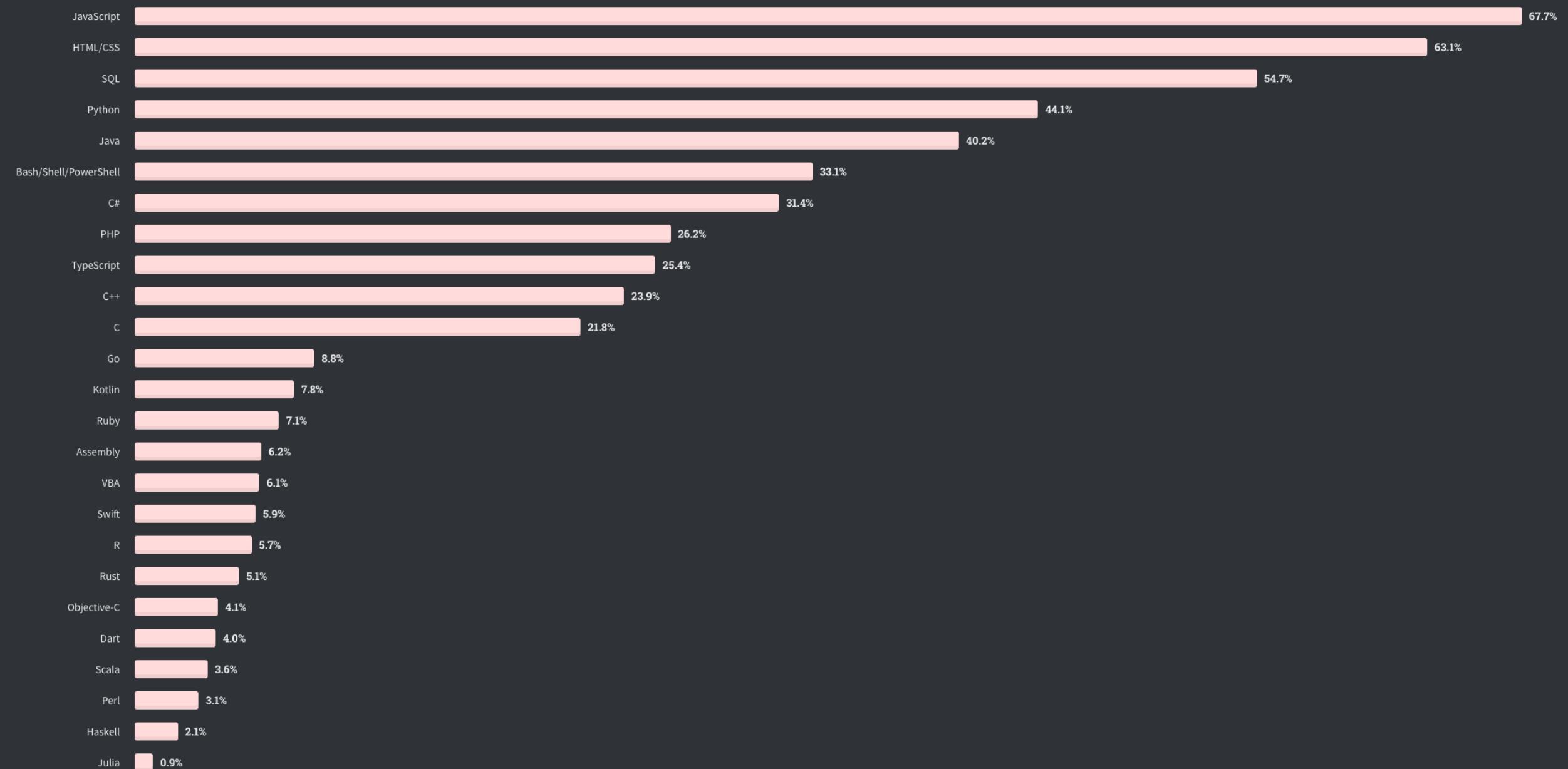
Firmware, Converters, API, SDK, Protokoll, Programmering

Fråga:

Hur många har programmerat tidigare?

Fråga:

Vilka programmeringsspråk känner ni till?



source: <https://insights.stackoverflow.com/survey/2020#technology>

Programmering inom IoT

- Firmware
 - Converters
 - Bryggor
 - Filter
 - API Anrop
-
- Python
 - JavaScript
 - Java
 - C

Exempel

<https://github.com/voxic/thingsboardTrafikverket/blob/master/trafikverketBridgeMQTT.py>

Diskussion:

Vad är Firmware?

- Användningsområden
- Miljö/Runtimee
- Programmeringsspråk
- Skillnad mellan sketch och firmware?

```
#include "Joystick.h"

Joystick_ Joystick;

int xAxis_ = 0;
int RxAxis_ = 0;
int yAxis_ = 0;
int RyAxis_ = 0;
int throttle=0;

void setup()
{
    //Setup all buttons as inputs with PULLUP
    for(int i = 3;i < 13;i++) {
        pinMode(i, INPUT_PULLUP);
    }

    Joystick.begin();
    //Serial.begin(9600); //Start serial monitoring
}

void loop()
{
```

Sketch: Arduino Joystick, C

Sketch implementerar funktionen

Firmware är bryggan mellan hårdvaran och sketch

Introduktion: Python

Note

la behöver en Kodeditor. Välj vad ni vill. Jag kör Visual Studio Code <-- Bra val

Learn Python The Hard Way

<https://learnpythonthehardway.org/python3/>

Exercise 00

"The Setup"

Exercise 01

Input:

```
print("Hello World!")
print("Hello Again")
print("I like typing this.")
print("This is fun.")
```

Output:

```
python3.6 ex1.py
Hello World!
Hello Again
I like typing this.
This is fun.
```

Exercise 02

```
# A comment, this is so you can read your program later.  
# Anything after the # is ignored by python.  
  
print("I could have code like this.") # and the comment after is ignored  
  
# You can also use a comment to "disable" or comment out code:  
# print("This won't run."  
  
print("This will run.")
```

Exercise 03

```
print("I will now count my chickens:")

print("Hens", 25 + 30 / 6)
print("Roosters", 100 - 25 * 3 % 4)

print("Now I will count the eggs:")

print(3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6)

print("Is it true that 3 + 2 < 5 - 7?")

print(3 + 2 < 5 - 7)

print("What is 3 + 2?", 3 + 2)
print("What is 5 - 7?", 5 - 7)

print("Oh, that's why it's False.")
```

Exercise 04

```
cars = 100
space_in_a_car = 4.0
drivers = 30
passengers = 90
cars_not_driven = cars - drivers
cars_driven = drivers
carpool_capacity = cars_driven * space_in_a_car
average_passengers_per_car = passengers / cars_driven

print("There are", cars, "cars available.")
print("There are only", drivers, "drivers available.")
print("There will be", cars_not_driven, "empty cars today.")
print("We can transport", carpool_capacity, "people today.")
print("We have", passengers, "to carpool today.")
print("We need to put about", average_passengers_per_car,
      "in each car.")
```

Moduler

Där python hämtar sin styrka

```
import json # <-- Modul

# Mitt program kan nu hantera JSON
print(json.loads(jsonDataFromServer))
```

```
import paho.mqtt # <--- Modul

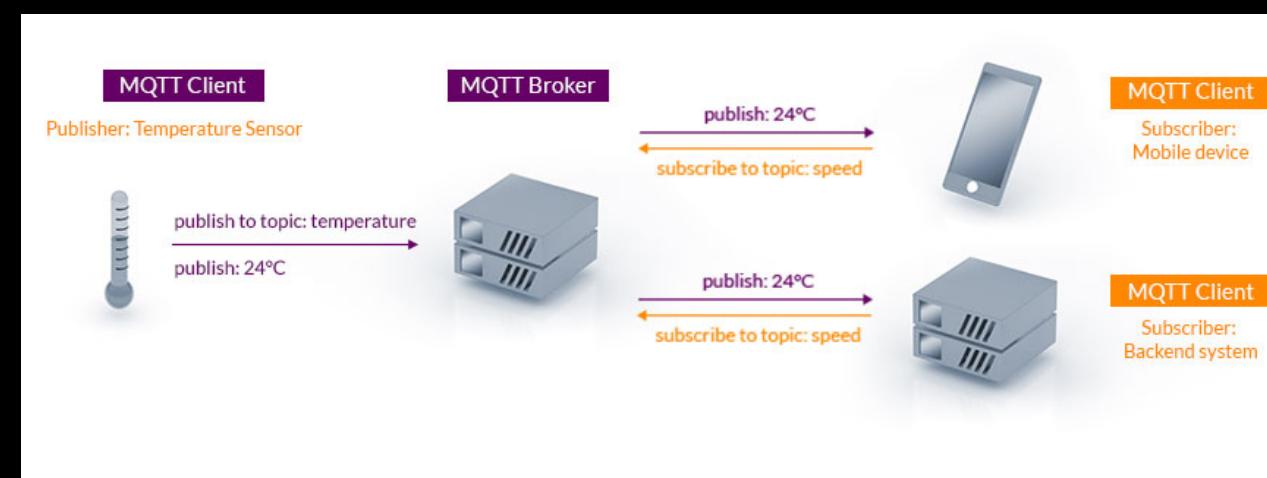
# Mitt program kan nu hantera MQTT
client = paho.Client(cfg['clientID'], True, None, paho.MQTTv31)
```

Protokoll

Vanliga protokoll inom IoT:

- MQTT
- REST
- CoAP
- SOAP

MQTT



MQTT forts.

Broker: Mosquitto

<https://mosquitto.org/>

Client: MQTTExplorer

<http://mqtt-explorer.com/>

MQTT Övning

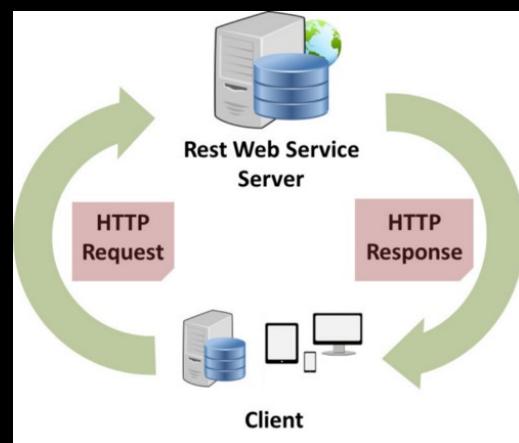
Chatta med MQTT

1. Dela in er två och två
2. Installera MQTT Explorer
3. Anslut till mqtt.eclipse.org
4. Kom överens om en topic
5. Börja chatta

REST

Representational state transfer (REST)

- En bas URI, exempel: <http://api.example.com/collection/>
- Standard HTTP metoder (e.g., GET, POST, PUT, PATCH and DELETE)
- Vanligt med svar i JSON-format



Vanligt protokoll för att interagera med API:er

```
{  
    "name": "pikachu",  
    "order": 35,  
    "species": {  
        "name": "pikachu",  
        "url": "https://pokeapi.co/api/v2/pokemon-species/25/"  
    },  
    "stats": [  
        {  
            "base_stat": 35,  
            "effort": 0,  
            "stat": {  
                "name": "hp",  
                "url": "https://pokeapi.co/api/v2/stat/1/"  
            }  
        }  
    ]  
}
```

Verktyg

Postman

<https://www.postman.com/>

cURL

<https://en.wikipedia.org/wiki/CURL>

Övning:

Använd postman för att göra ett API anrop

Launchpad

GET https://pokeapi.co/api/v2/poke... ● + 000

No Environment

Untitled Request

GET https://pokeapi.co/api/v2/pokemon/pikachu

Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

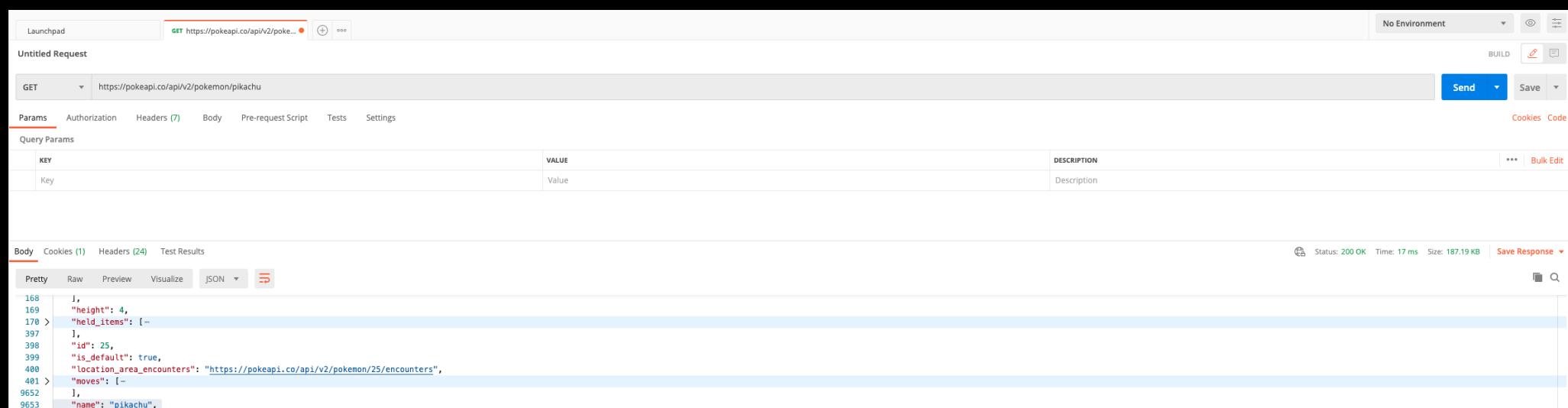
KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (24) Test Results

Pretty Raw Preview Visualize JSON

168],
169 "height": 4,
170 > "held_items": [..
397],
398 "id": 25,
399 "is_default": true,
400 "location_area_encounters": "https://pokeapi.co/api/v2/pokemon/25/encounters",
401 > "moves": [..
9652],
9653 "name": "pikachu",

Status: 200 OK Time: 17 ms Size: 187.19 KB Save Response



API

Application Programming Interface

SDK

Software Development Kit

Diskussion:

Användningsområden för API:er/SDK

- Exempel på API:er
- Exempel på där ett API används
- Skillnader mellan SDK och API

Plattformar

"A platform to rule them all"

Diskussion:

Plattformar och Ekosystem

- Vilka känner ni till?
- Skillnader?



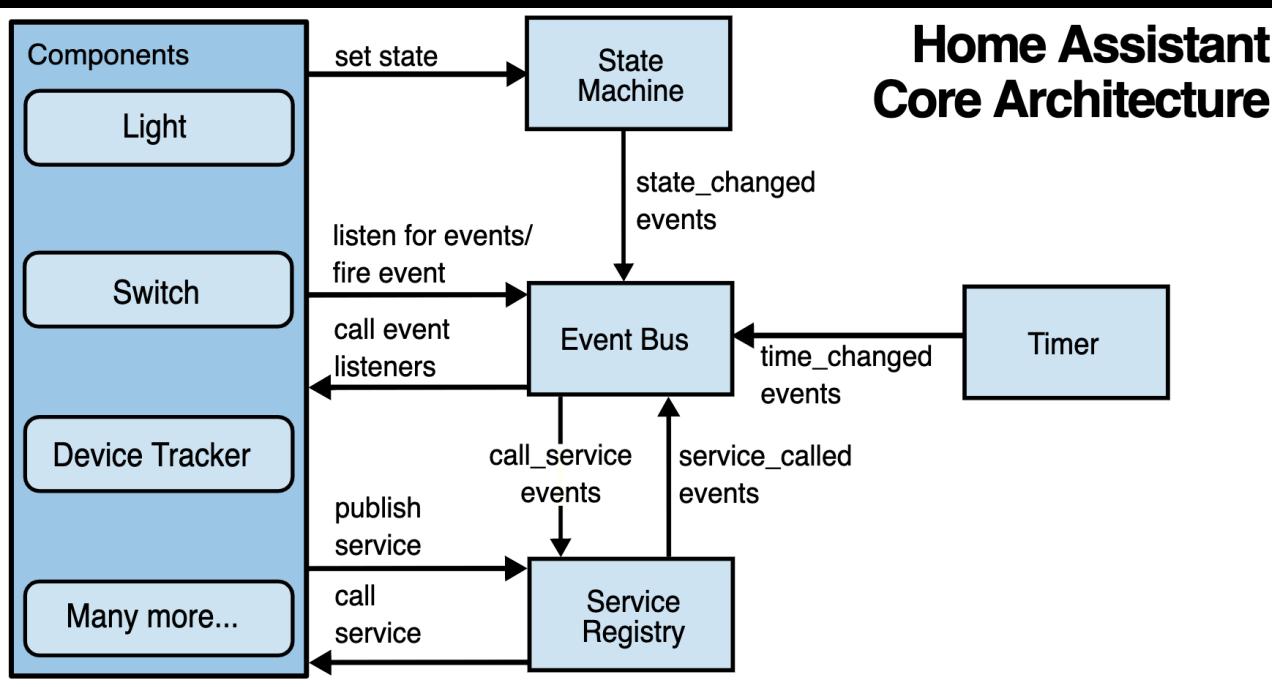
**Home
Assistant**

Home Assistant

"Open source home automation that puts local control and privacy first.
Powered by a worldwide community of tinkerers and DIY enthusiasts.
Perfect to run on a Raspberry Pi or a local server."

<https://homeassistant.io>

Arkitektur



Bra exempel på en modern event-driven arkitektur

- Abstraktionslager
- Integration
- Automation
- Logging
- NAV

Lab01-1

Installation Home Assistant

ide:

<https://github.com/voxic/Nackademin/blob/master/Dag1/Installation%20Home%20Assistant.rq>



Lab01-2

Uppsättning Arduino Sensor

uide:

<https://github.com/voxic/Nackademin/blob/master/Dag1/Upps%C3%A4ttning%20Arduino.m>



Lab01-3

Arduino sensor till Home Assistant

e:

<https://github.com/voxic/Nackademin/blob/master/Dag1/Arduino%20till%20Home%20Assistant>

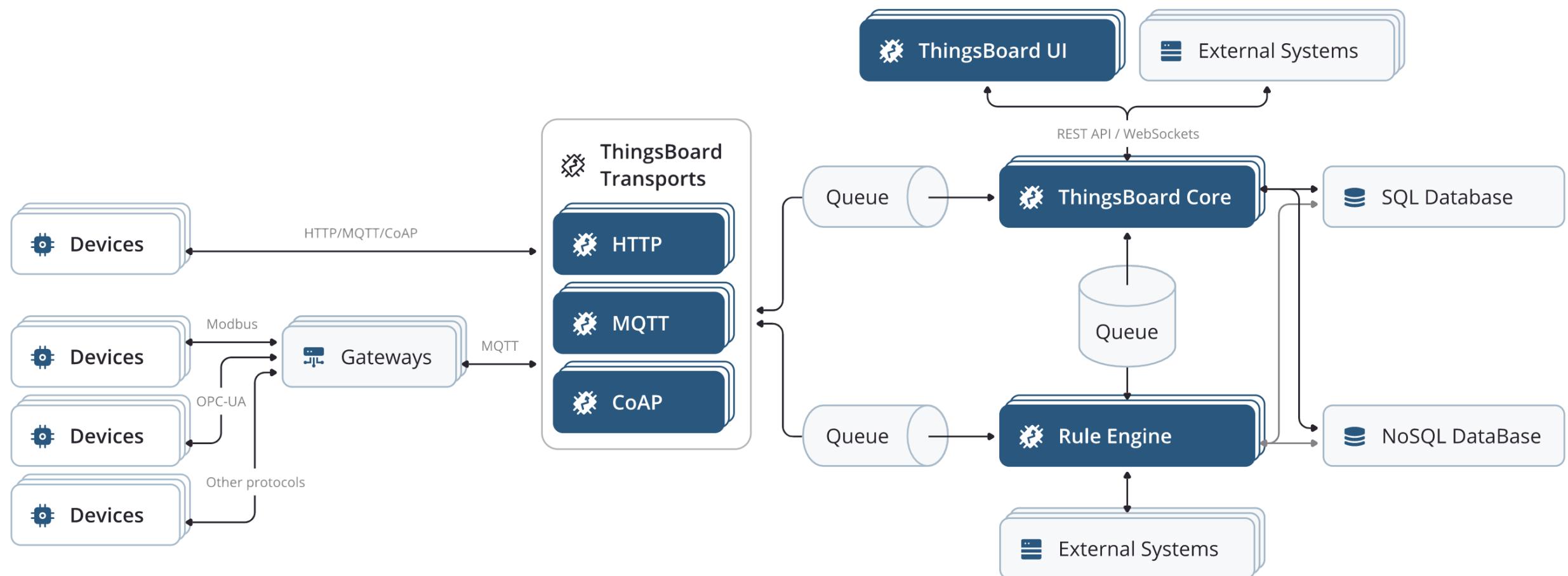


Thingsboard

Thingsboard is an open-source IoT platform that enables rapid development, deployment and scaling of IoT projects. The goal is to provide the out-of-the-box IoT cloud services solution that will enable server-side infrastructure for your IoT application.

<https://thingsboard.io>

Arkitektur



"Enterprise-grade Event architecture"

- Provision devices, assets and customers and define relations between them.
- Collect and visualize data from devices and assets.
- Analyze incoming telemetry and trigger alarms with complex event processing.
- Control your devices using remote procedure calls (RPC).
- Build work-flows based on device life-cycle event, REST API event, RPC request, etc
- Design dynamic and responsive dashboards and present device or asset telemetry and insights to your customers
- Enable use-case specific features using customizable rule chains.
- Push device data to other systems.