One Remote Control to Command them All! Building a Hypermedia API for ESP8266-based Devices

Alexey Andreev, Daniil Garayzuev, Maxim Kolchin, Nikita Chursin and Ivan Shilin,
ITMO University, Russia

Abstract

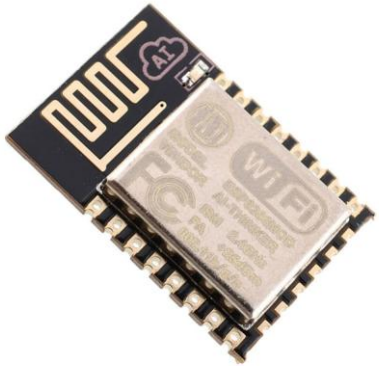
embedded systems (esp8266 wifi module)

self descriptive REST API

CoAP protocol

I ulary based appro nd gu

Hydra



Introduction

Every IoT-devices developer is trying to build full stack platform to provide access to them

Devices descpriptions, commands and observations could be shared via stardartized protocols without limiting the devices new features

Same clients could interact with the new devices if the new schema is interhited from one used by client

Requirements

Device discovery

Model-instance representation

Authentification

Publish/Subsribe pattern

On-demand configuration

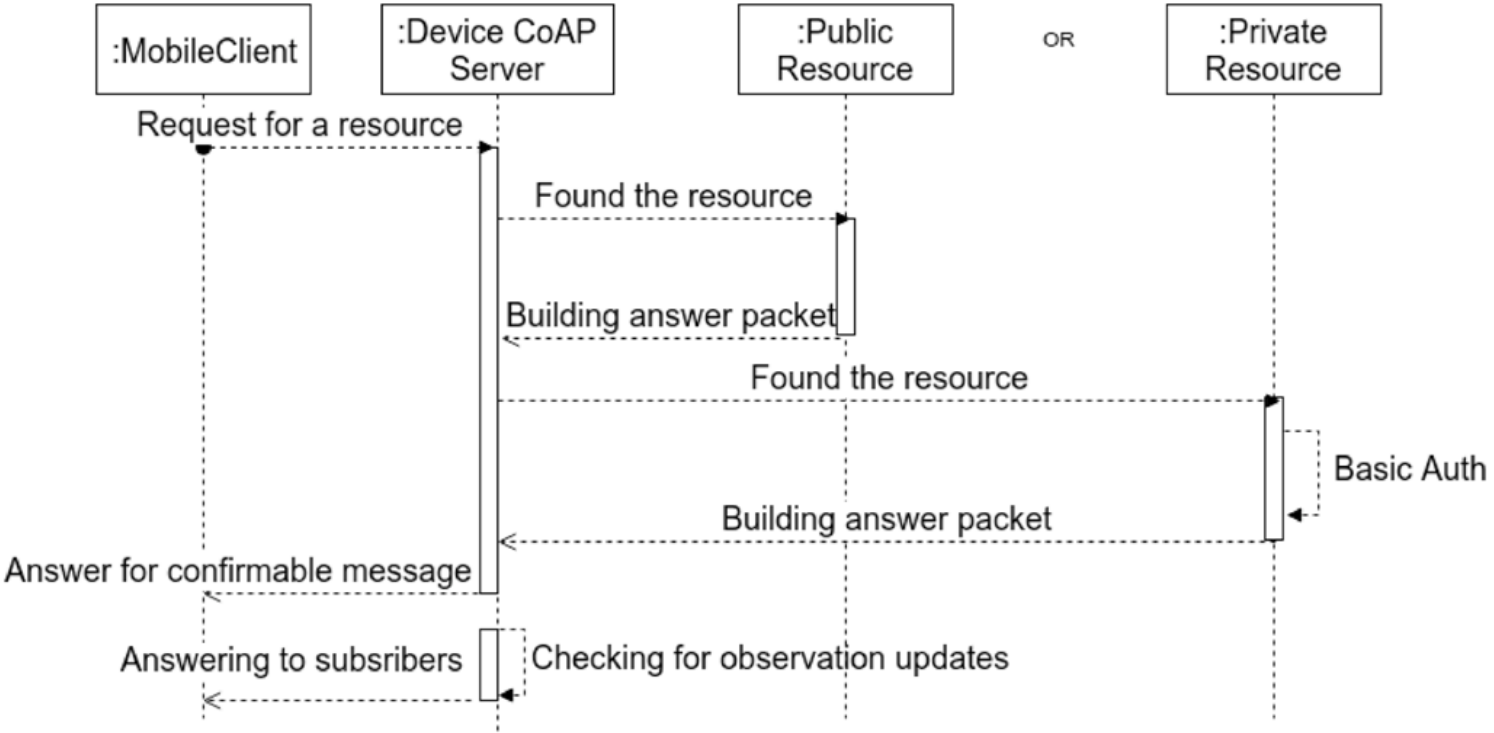
Slepping nodes support

Related work

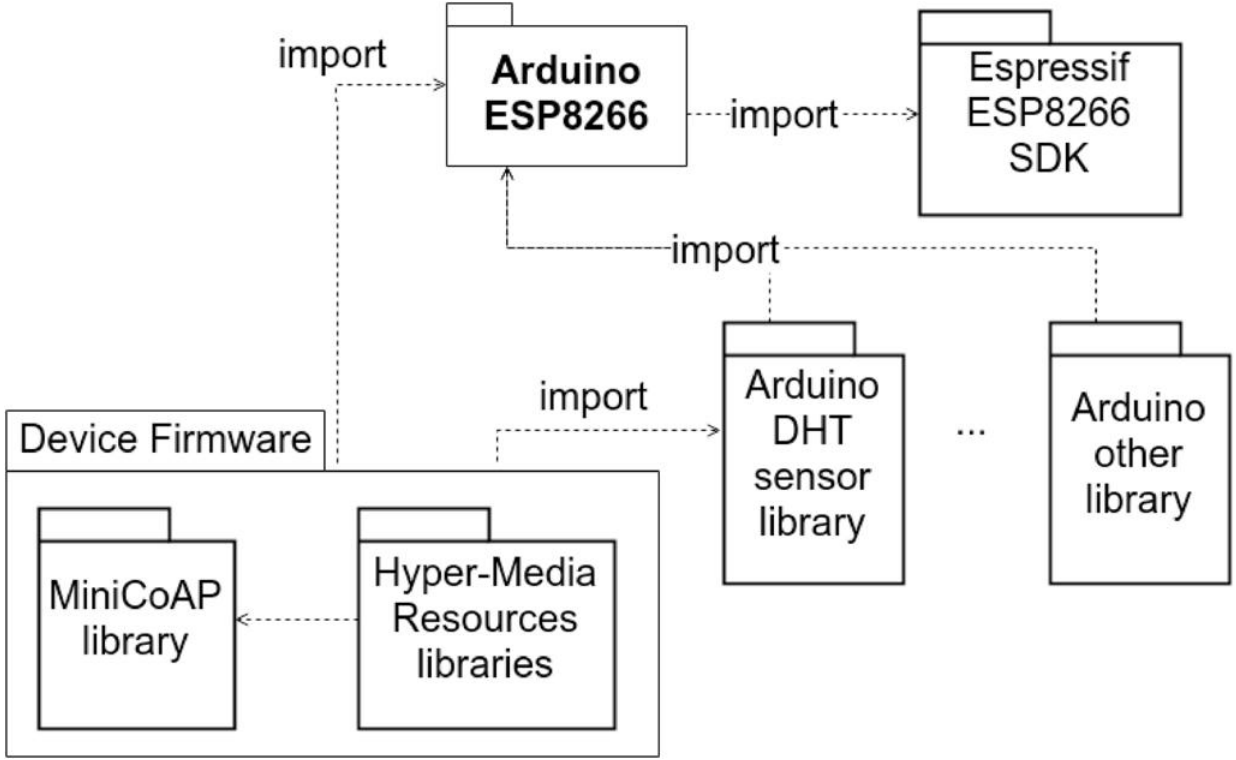
SmallHydra

Approaches based on separated custom protocols adapters

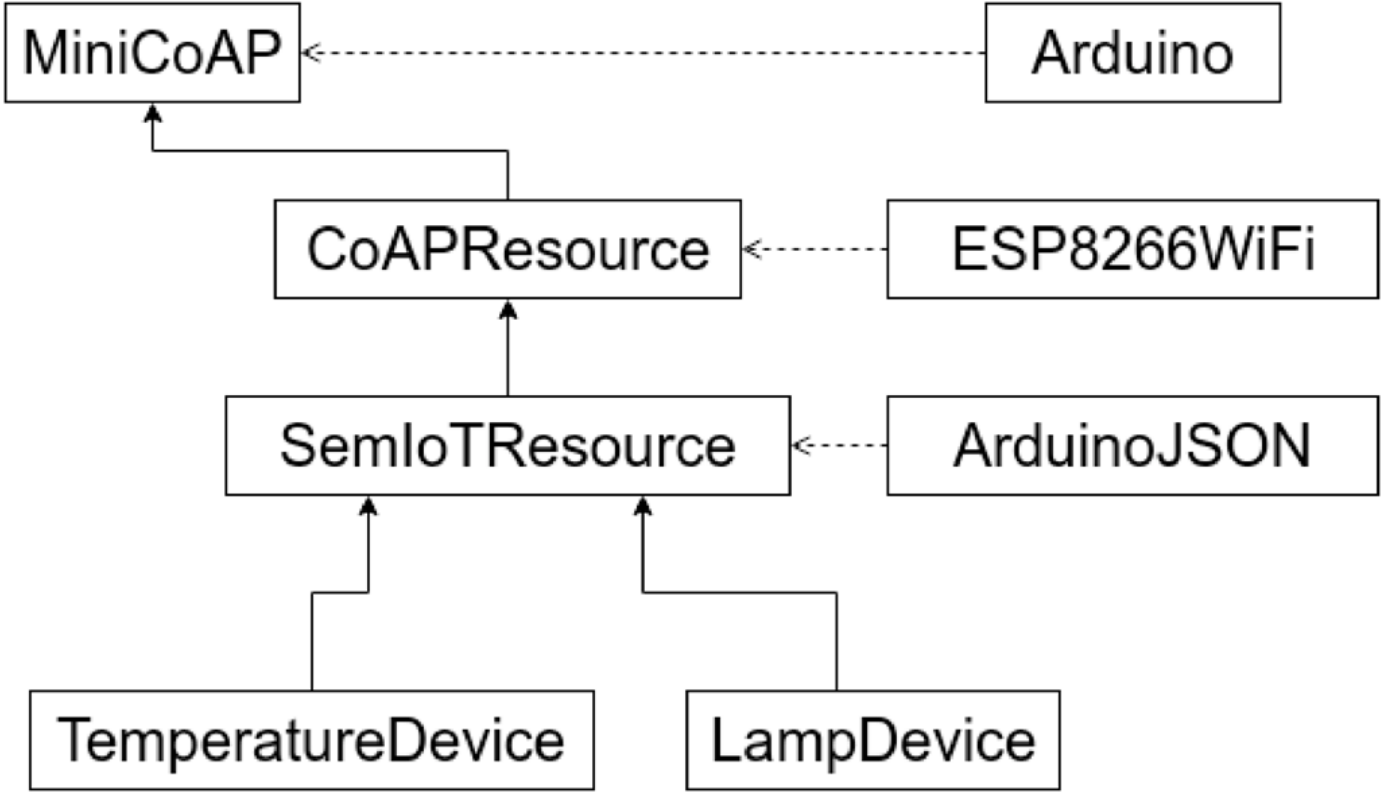
Device Software Guidelines



Device Software Guidelines



Device Software Guidelines



Device Design Guidelines

- Configuration mode for the initialization

For example, two-state hardware button to switch between configuration and regular mode.

Device Public API Guidelines

.well-known/core resource

API Documentation

Device Resource

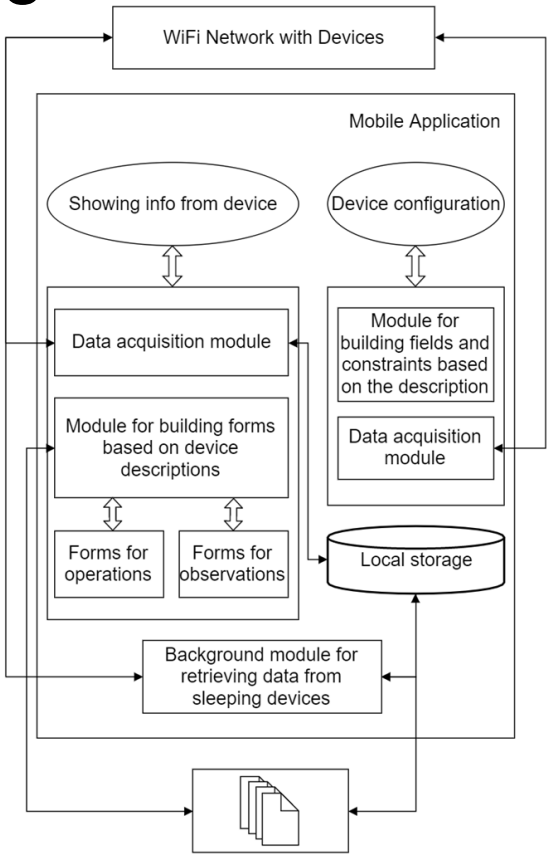
Value Resource

Action Resource

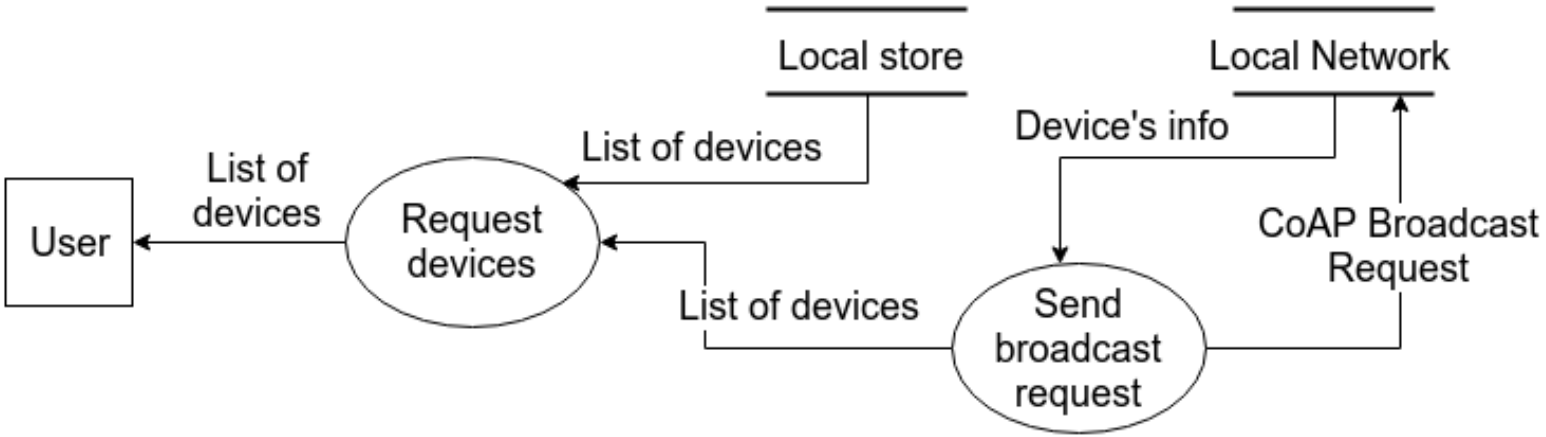
```
{
  "@context": "http://external/doc#",
  "@id": "coap://1.1.1.1/",
  "@type": "TemperatureDevice",
  "identifier": "e9704",
  "label": { "@value": "Temperature Device",
"@language": "en" },
  "location": {
    "@type": "Place",
    "label": "1010"
  },
  "temperature": "/temperatureValue"
}
```

Mobile Application Client Architecture

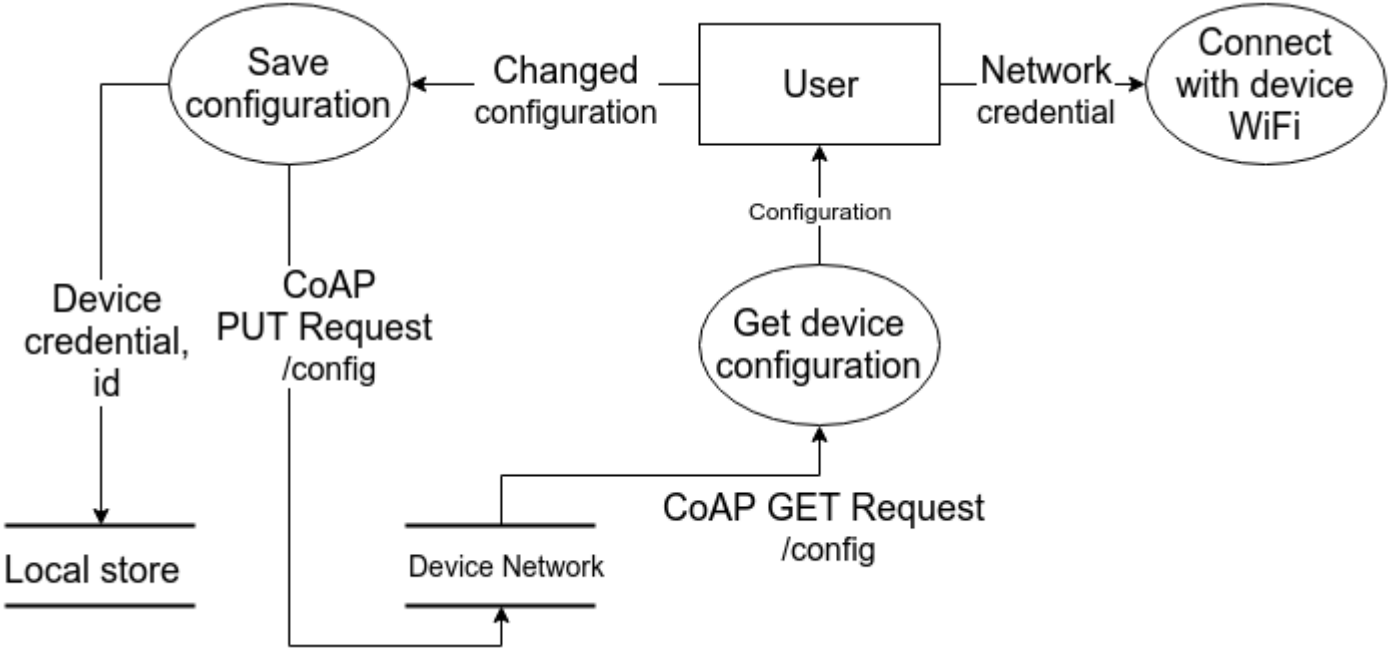
- sleeping nodes support background service
- Devices identification in configuration and regular mode
- Device discovery



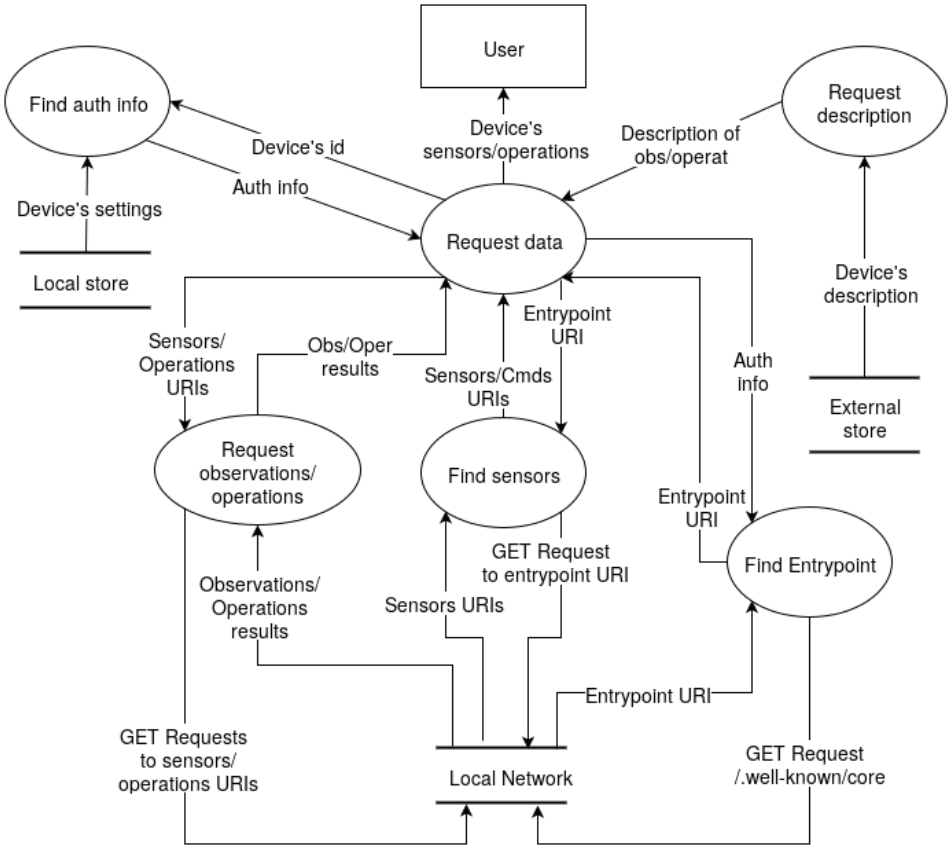
Mobile Application Client Architecture



Mobile Application Client Architecture



Mobile Application Client Architecture



Case study

Benefits

DIY devices could be easily integrated with the clients without rebuilding them

Uncompatible brands devices could be integrated to one system

Embedded devices developing and smart clients to simplify tasks and reduce the expenses