



THE DEVELOPER'S CONFERENCE

**Descomplicando o desenvolvimento de soluções de IoT
com a meta plataforma KNoT**

João Neto

Engenheiro de Software - CESAR
Mestrando em Ciência da Computação - CIn UFPE



THE DEVELOPER'S
CONFERENCE



C . E . S . A . R

KNOT

A FRAMEWORK FOR IOT INTEROPERABILITY





C . e . S . A . R



Private innovation center that creates products, services and business using information and communication technologies (ICT).





Identify, potentialize and realize opportunities for **transformation** of **organizations** and **people's lives**

WHY ISN'T IT EASY TO CONNECT YOUR THINGS TO THE INTERNET?



WHAT IS THE INTERNET OF THINGS?



WHAT IS THE INTERNET OF THINGS?

Embed **connectivity**,
sensors and actuators for
sharing data across
multiple devices to **add more**
value.



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R

WHY EVERY ONE IS LOOKING AT IOT?

Values



WHY ISN'T IT EASY TO CONNECT YOU THINGS TO THE INTERNET?



Every "thing" has its specific connectivity needs:

- ▶ communication range
- ▶ baud rate
- ▶ power consumption
- ▶ cost



THERE IS NO STANDARD YET

“It is very **difficult to have a unique standard** for IoT as we have for the WWW, with HTML and browsers. The THINGS in IoT are so **different** and the applications as diverse that **many standards and protocols will coexist.**”



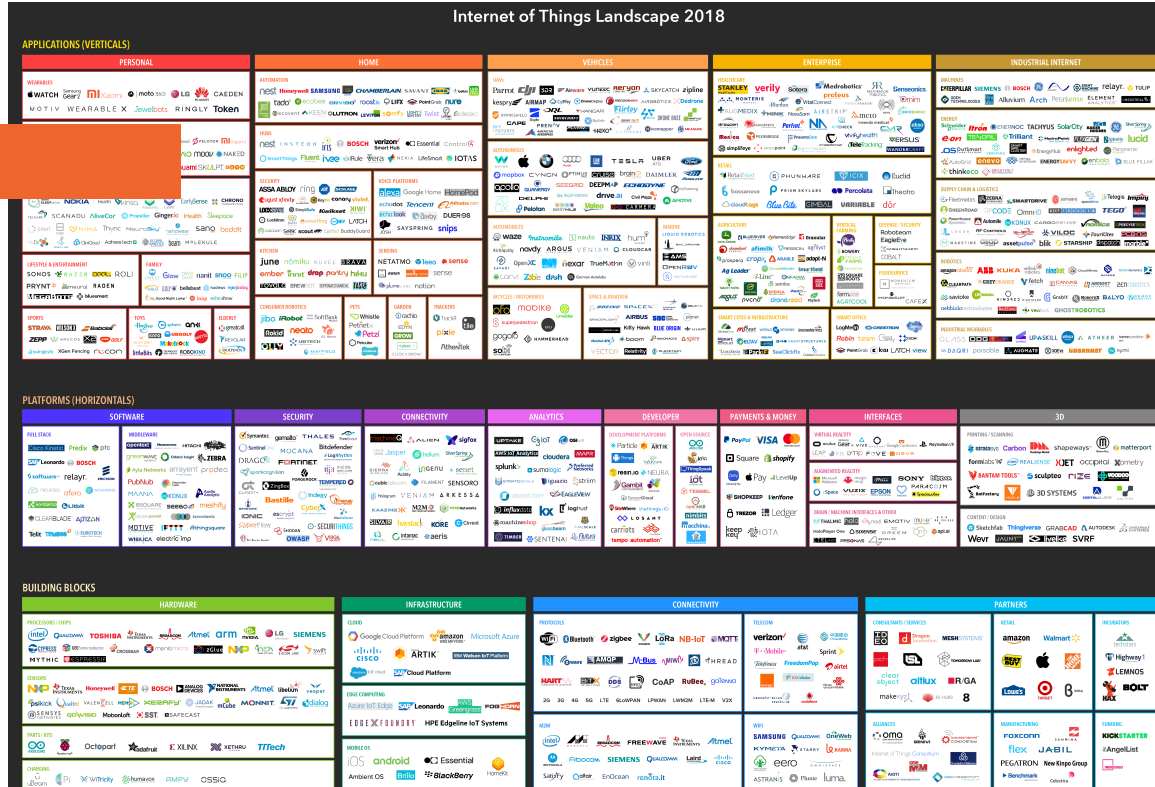
THE DEVELOPER'S
CONFERENCE



C . E . S . A . R

IT GENERATES A PROBLEM

Complexity



PLATFORMS (HORIZONTALS)

SOFTWARE	SECURITY	CONNECTIVITY	ANALYTICS	DEVELOPER	PA	
<p>FULL STACK</p>	<p>MIDDLEWARE</p>			<p>DEVELOPMENT PLATFORMS</p>	<p>OPEN SOURCE</p>	

BUILDING BLOCKS

HARDWARE	INFRASTRUCTURE	CONNECTIVITY
<p>PROCESSORS / CHIPS</p>	<p>CLOUD</p>	<p>PROTOCOLS</p>
<p>SENSORS</p>	<p>EDGE COMPUTING</p>	<p>2G 3G 4G 5G LTE 6LoWPAN LPWAN LWM2M LTE-M V2X</p>
<p>PARTS / KITS</p>	<p>MOBILE OS</p>	<p>M2M</p>
<p>CHARGING</p>		

HOW TO MITIGATE THIS COMPLEXITY?



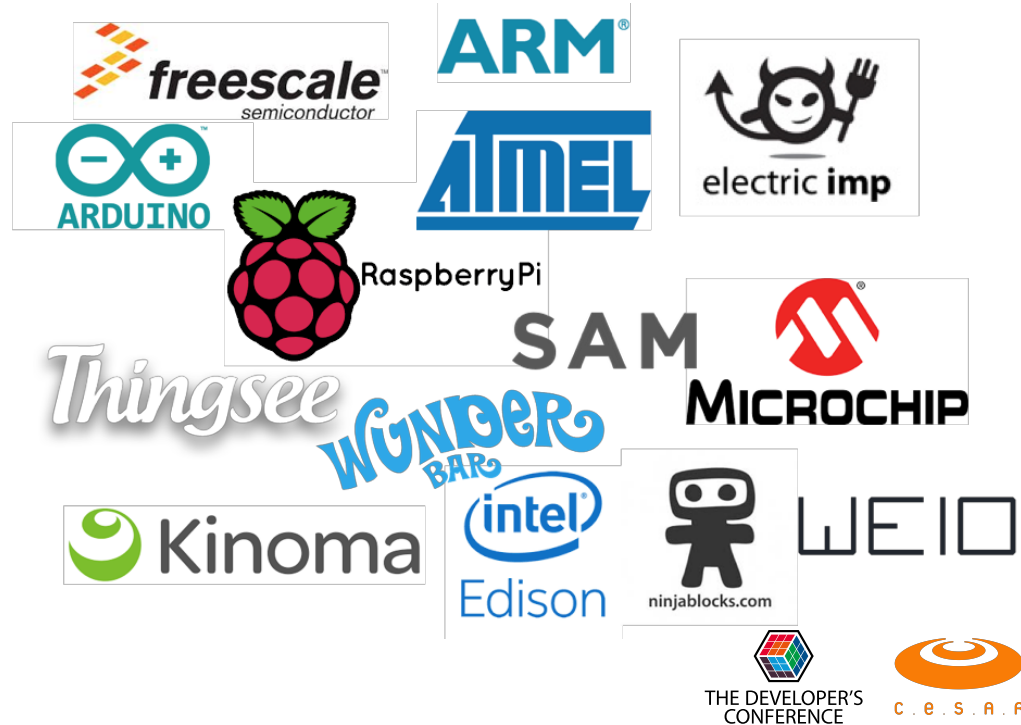
Software platforms

Cloud services to exchange data
across devices

HOW TO MITIGATE THIS COMPLEXITY?

Hardware platforms

Wireless microcontrollers
+ sensors to develop your
application





INTEROPERABILITY

Applications have its
own protocols...



...and they didn't talk to
each other.



C . e . S . A . R

What are **we** doing?



KNoT

KNoT Network of Things



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R

AVISO AOS PASSAGEIROS

ANTES DE ENTRAR NO
ELEVADOR, VERIFIQUE SE O
MESMO ENCONTRA-SE PARADO
NESTE ANDAR.

LEI ESTADUAL N° 9502/97

We are **not** creating **one**
more IoT platform!





KNoT

KNoT Network of Things

meta platform for IoT

bridging the gap between existing platforms
made on top of them



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R

IoT platforms

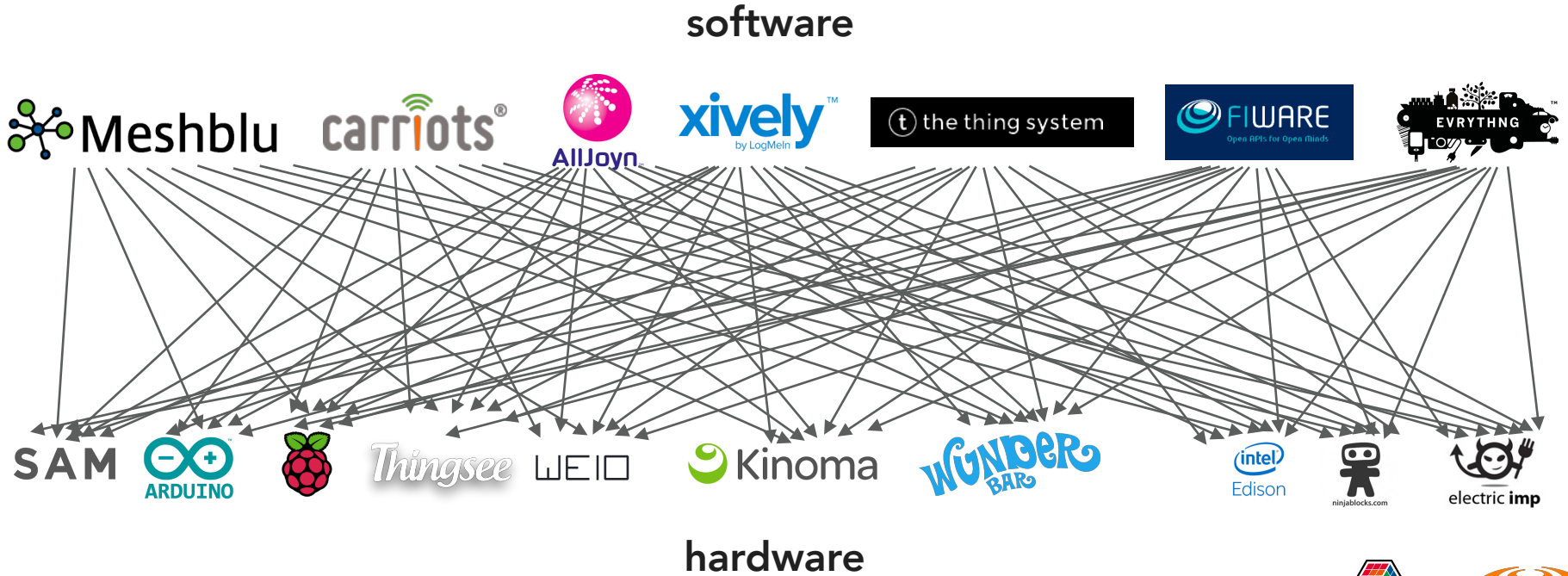
software



hardware

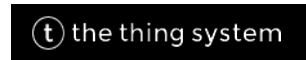


IoT platforms



IoT platforms

software



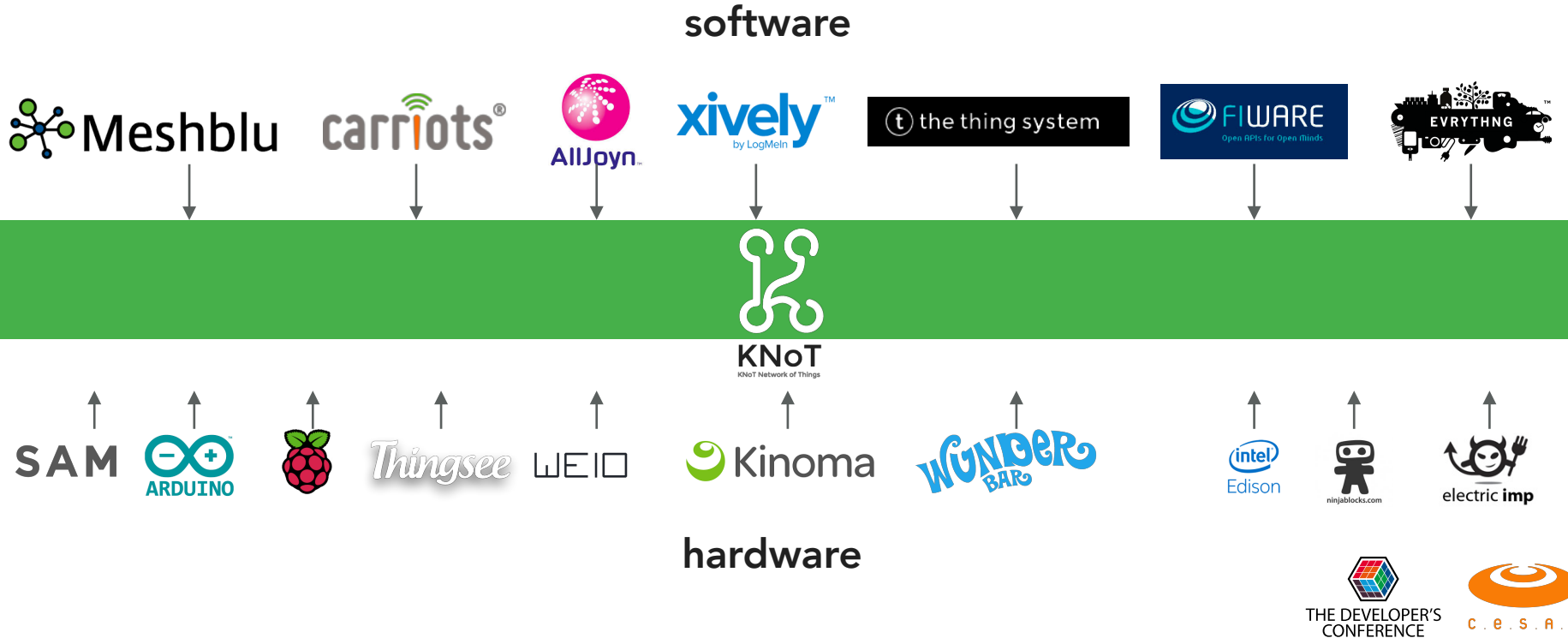
KNoT
KNoT Network of Things



hardware



IoT platforms





KNoT

KNoT Network of Things

**open source, hardware
and software.**



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R.



KNoT
KNoT Network of Things

multi protocol

for the physical (multi radios), data link, network and
transport layers

To address all communication needs.



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R



KNoT

KNoT Network of Things

end to end

from the device to the app, crossing radios, gateways
and cloud services.



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R.



KNoT

KNoT Network of Things

data sharing by space and time

"to share the data only on monday, from 2pm to 3pm."



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R



KNoT

KNoT Network of Things

semantic data model

applications will know the data type of each other.



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R.



KNoT

KNoT Network of Things

distributed cloud

a distributed system can handle billions of devices.



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R.



KNoT

KNoT Network of Things

low cost

the maximum cost of cheapest device will be \$5.



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R.



KNoT

KNoT Network of Things

easy to use

the developer only needs to implement 3 functions to have a device connected to the internet.



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R



KNoT

KNoT Network of Things

Why are we doing it?



THE DEVELOPER'S
CONFERENCE



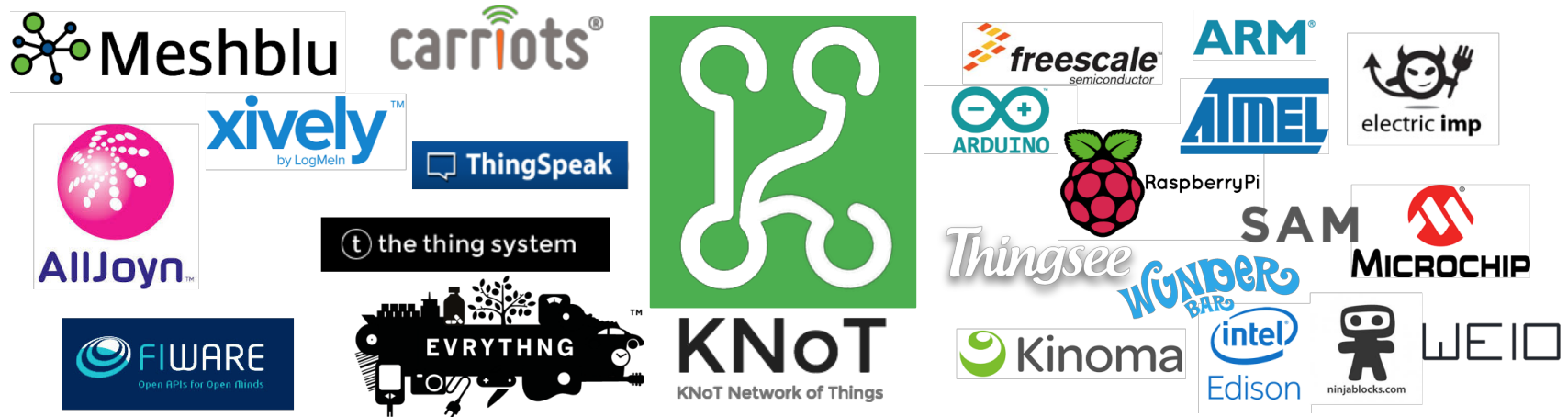
C.E.S.A.R.

Construction of applied knowledge



The group's researches turns into experiments in **observatories**, enabling CESAR to **apply** the **knowledge** acquired in IoT projects.

Integrate many platforms into a unique one



It allows **interoperability** across platforms and **across data** from many devices.

More agility in developing IoT projects

A KNoT Lamp can be simple like this:

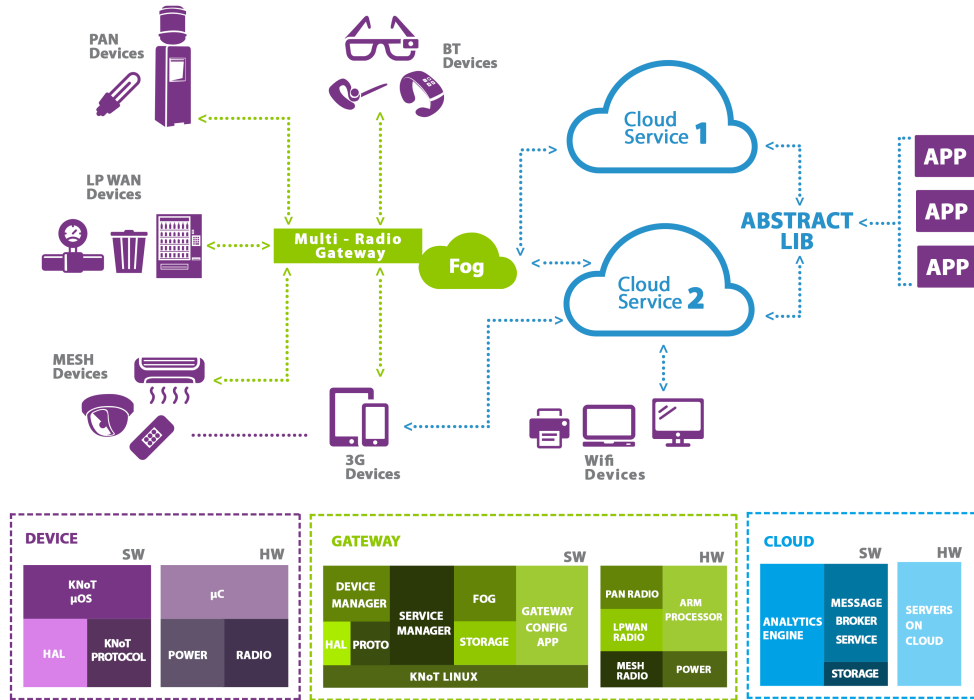
- 1. Get a KNoT device core schematic and customize it by adding a lamp actuator.**
- 2. Write 3 functions on KNoT μ OS that define your lamp behaviour.**
- 3. Write a mobile app using KNoT Cloud libraries to interact with the lamp.**



BUILDING AN IOT SOLUTION WITH KNOT



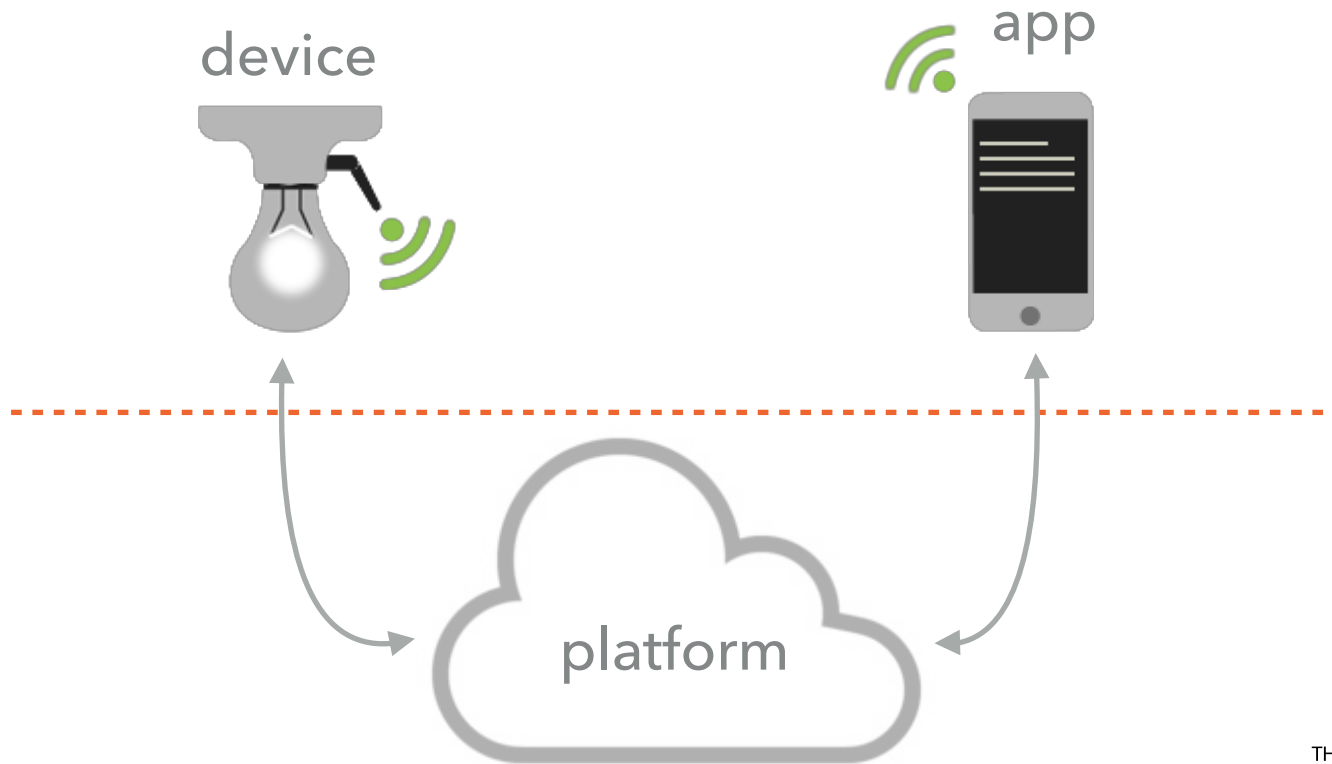
ARCHITECTURE



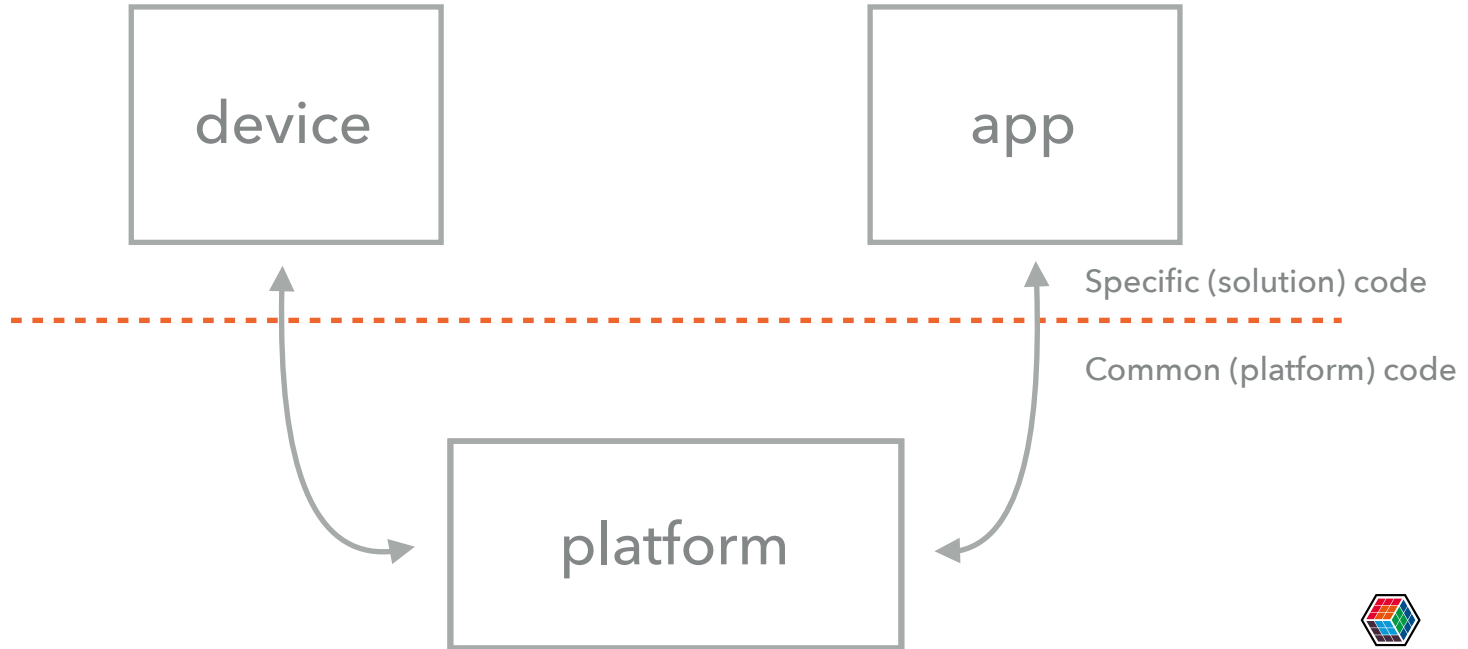
BUILDING AN IOT SOLUTION



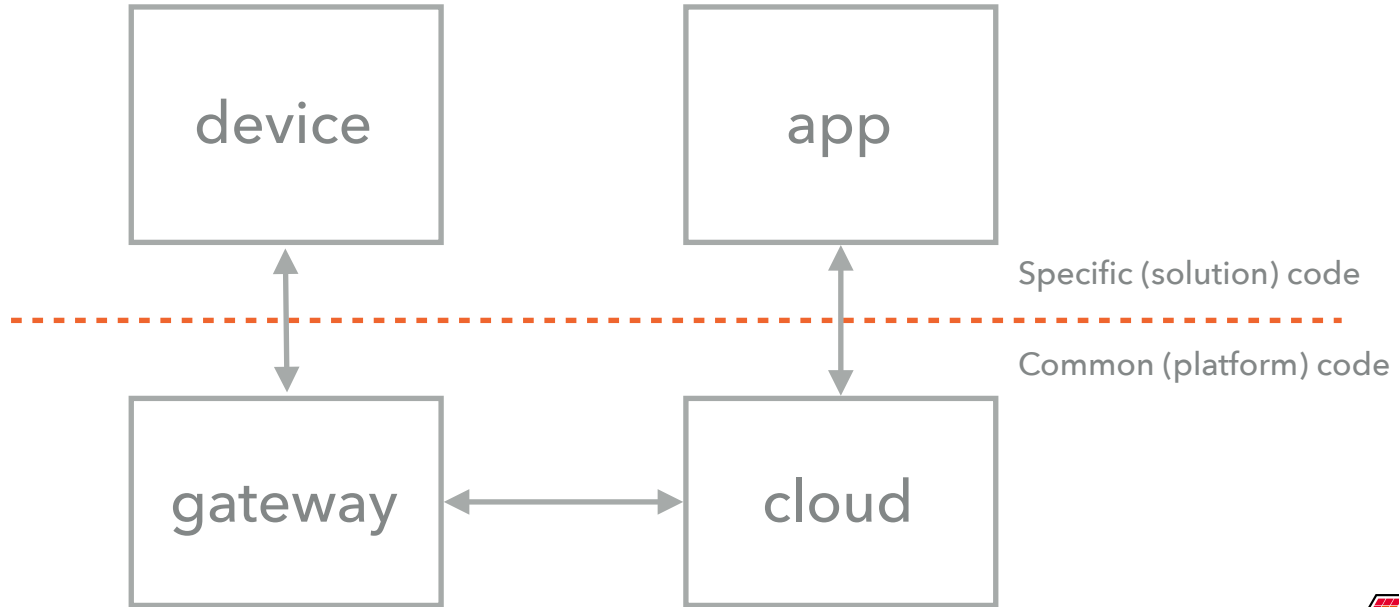
BUILDING AN IOT SOLUTION



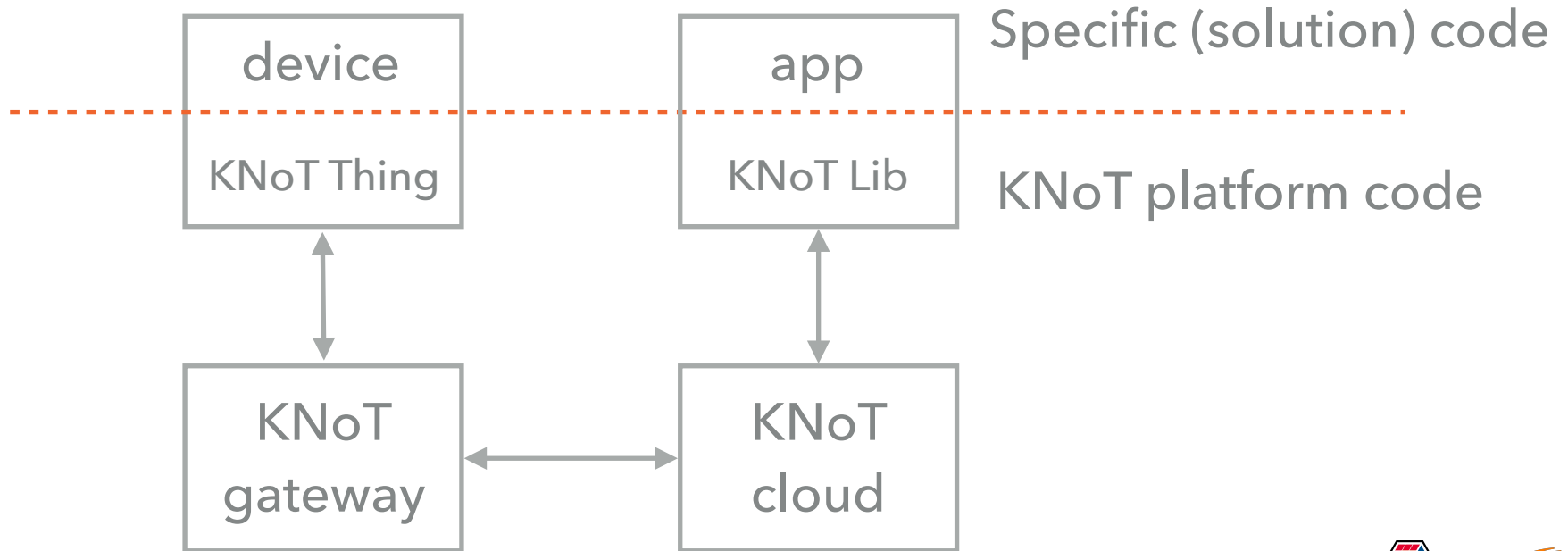
BUILDING AN IOT SOLUTION



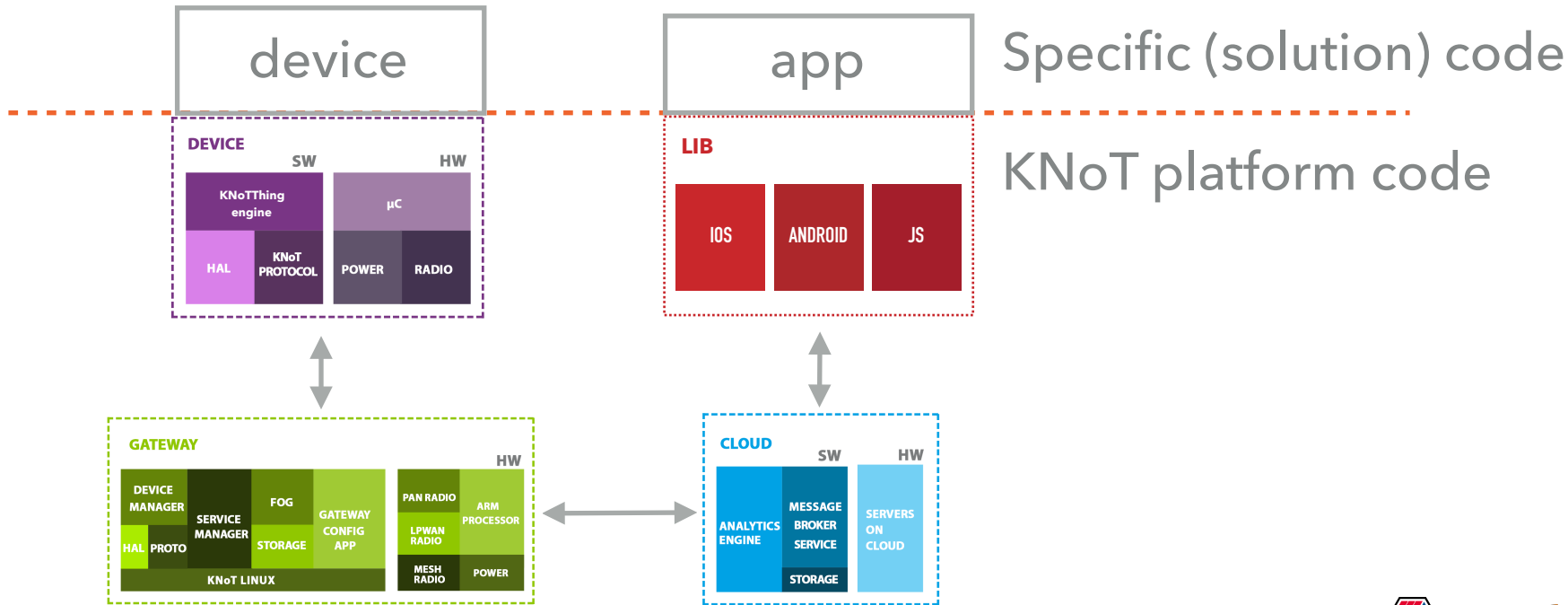
BUILDING AN IOT SOLUTION



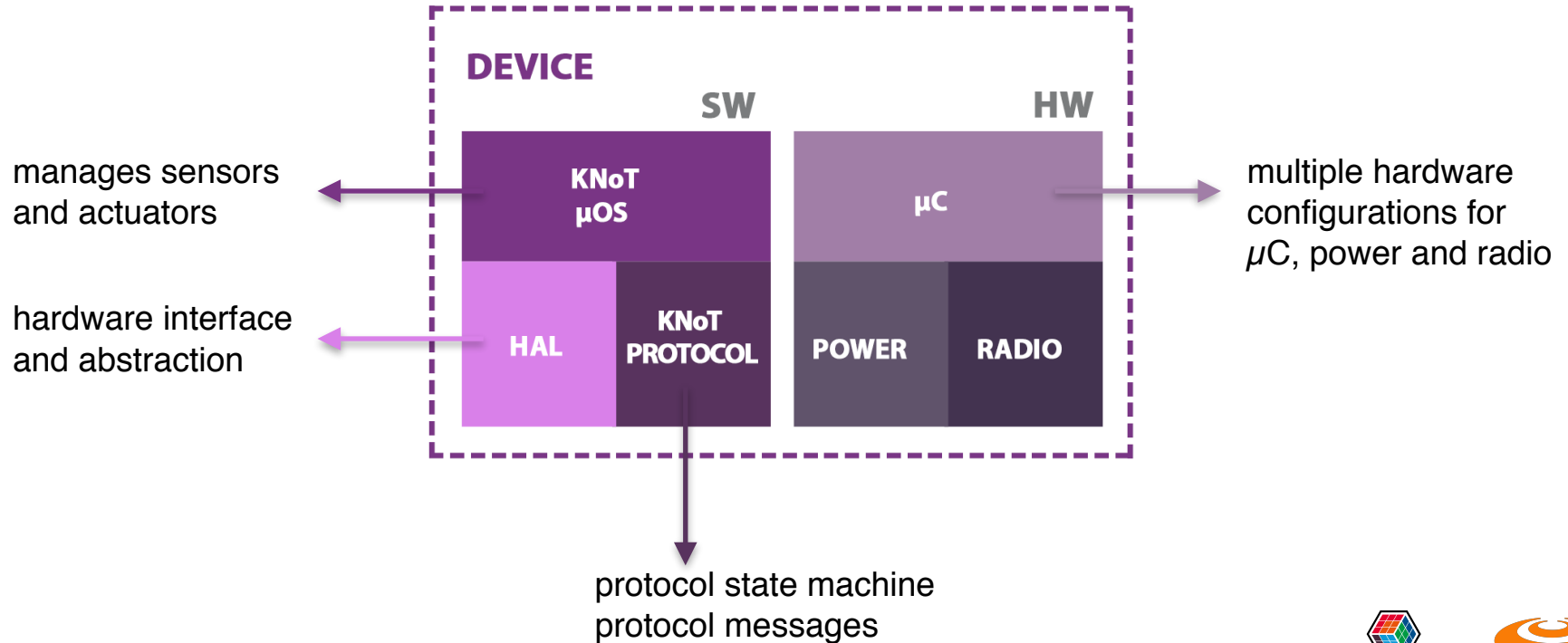
BUILDING AN IOT SOLUTION



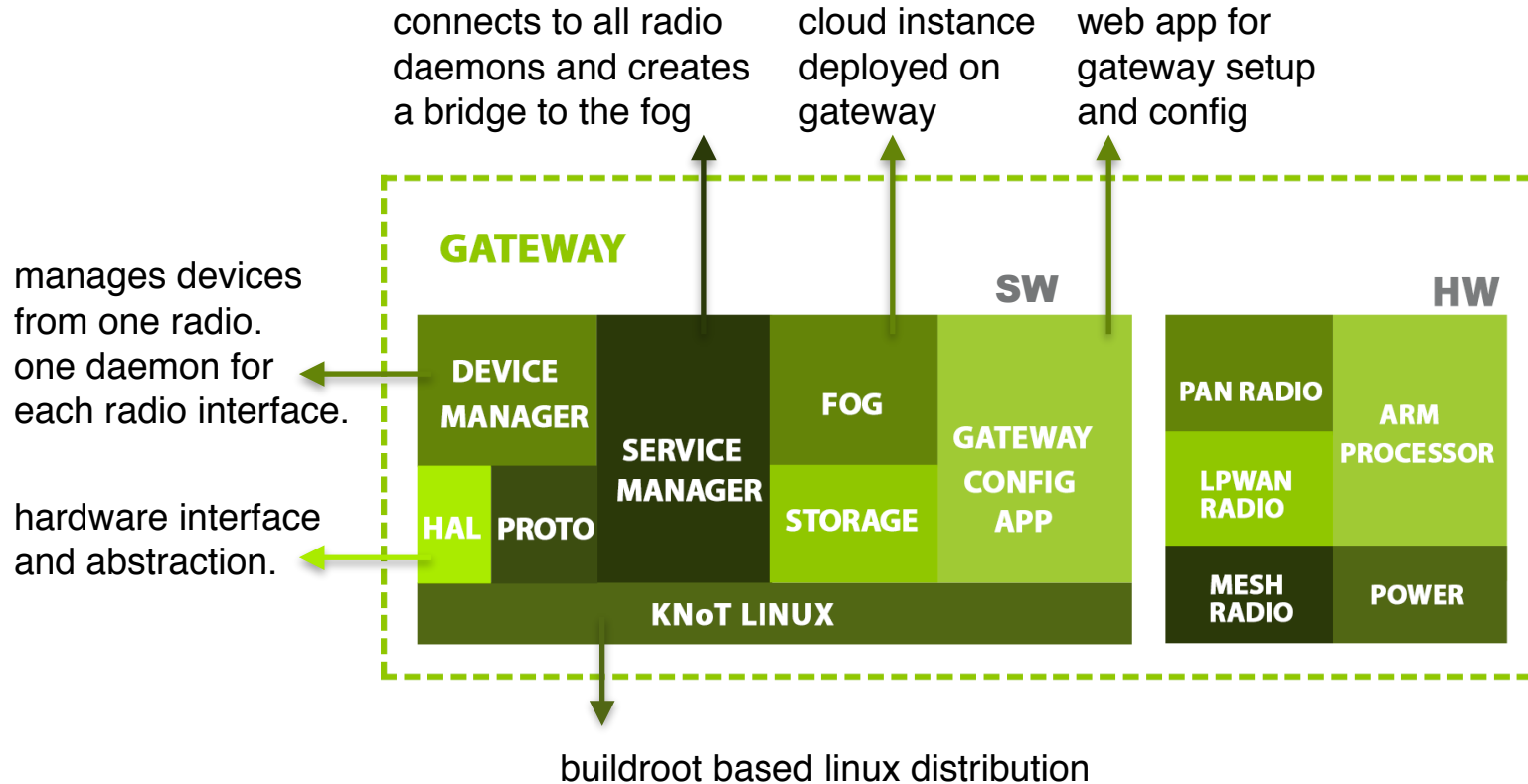
BUILDING AN IOT SOLUTION



COMPONENTS - KNOT THING

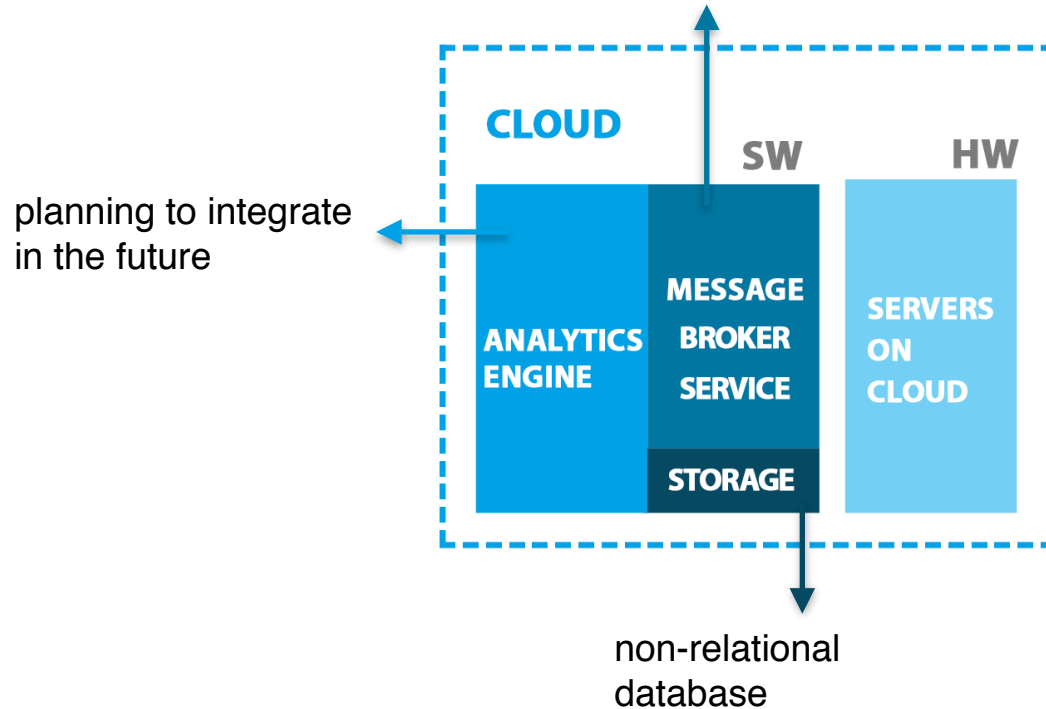


COMPONENTS - KNOT GATEWAY

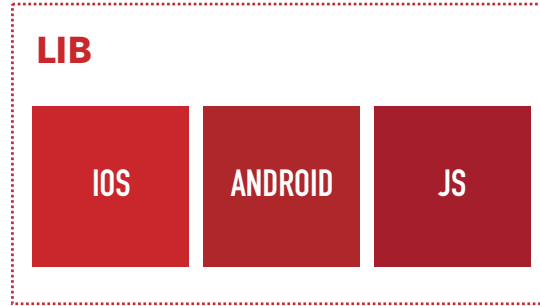


COMPONENTS - KNOT CLOUD

currently supporting KNoT Cloud and FIWARE.
planning to integrate with AWS IoT and Google Cloud.



COMPONENTS - KNOT LIB

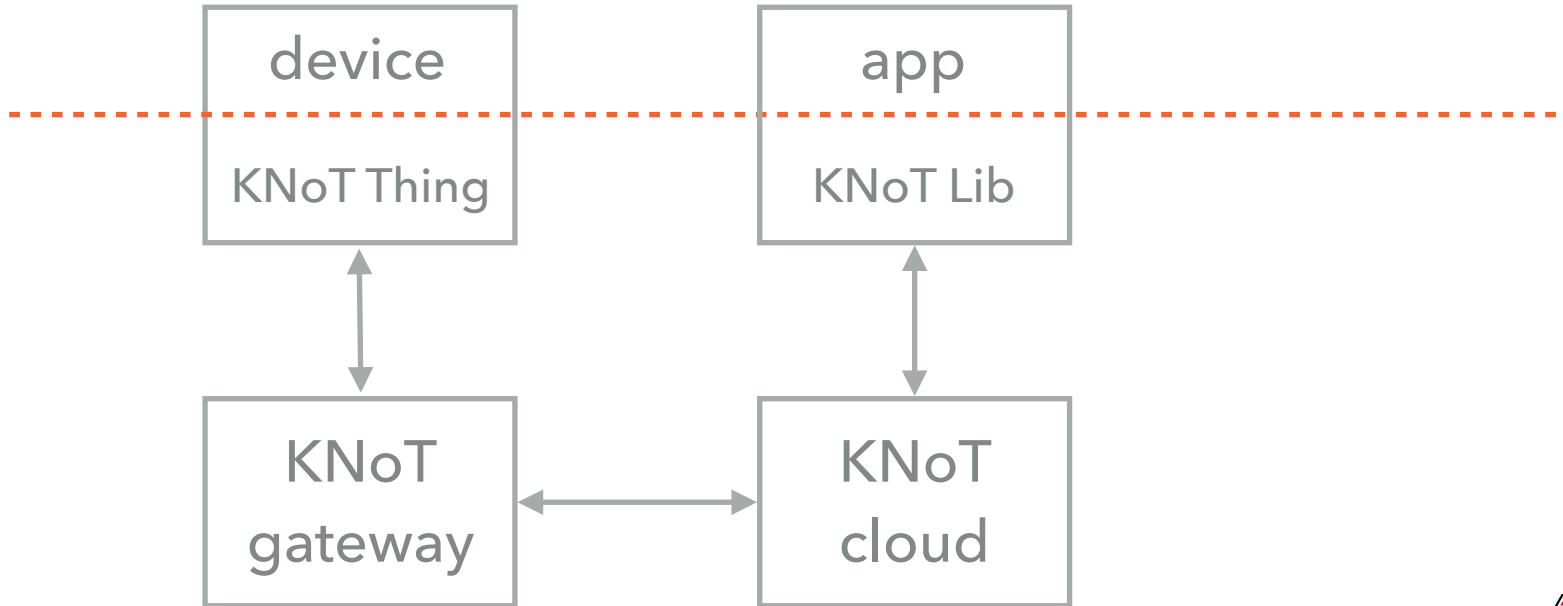


libraries that abstract the cloud services. it is used to develop applications. currently implemented for JavaScript and Android. IOS under development.

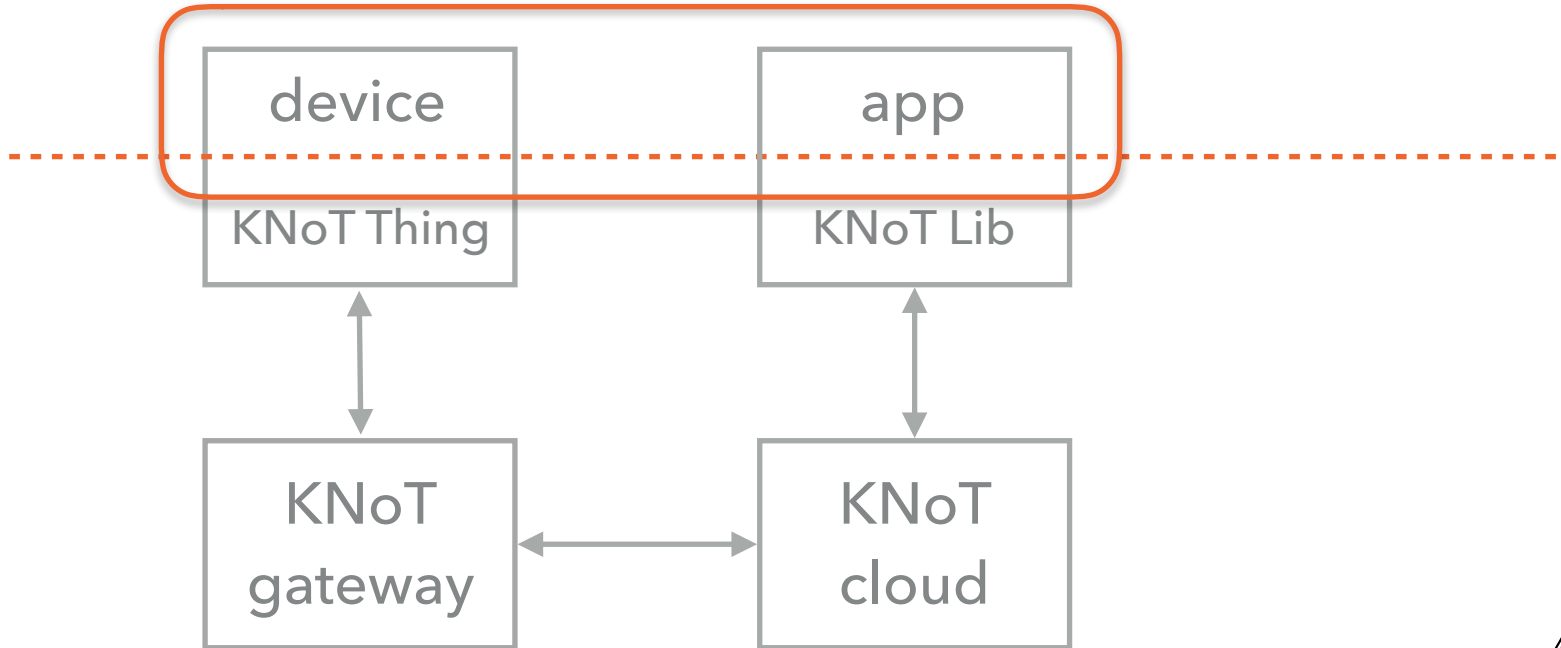
HOW EASY IS IT?



WHAT ARE WE BUILDING?



WHAT ARE WE BUILDING?



KNOT THING SPECIFICATION

nRF52840 Dongle

- System on Chip
- BLE
- Thread
- ZigBee
(802.15.4)



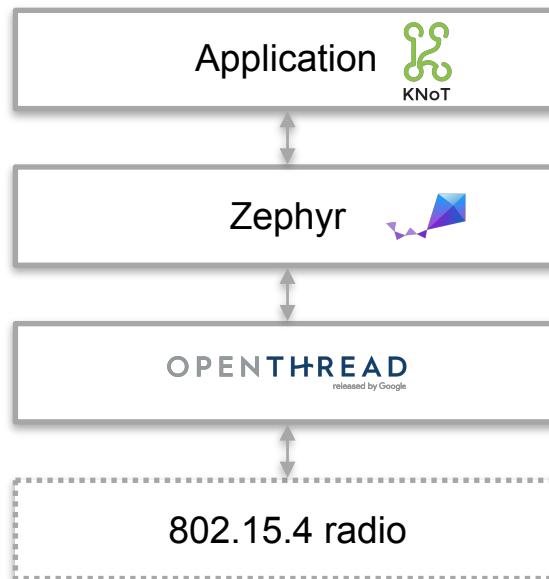
KNOT THING SPECIFICATION

nRF52840 Dongle

- System on Chip
- BLE
- Thread
- ZigBee (802.15.4)



Software Stack



KNOT THING SDK: CREATING AN APPLICATION

API Similar to Arduino



```
void setup(void);  
void loop(void);
```



KNOT THING SDK: CREATING AN APPLICATION

Virtual representation of sensors/actuators

```
struct knot_proxy knot_proxy_register(u8_t id, const char *name,  
                                     u16_t type_id, u8_t value_type,  
                                     u8_t unit, knot_callback_t changed_cb,  
                                     knot_callback_t pool_cb);
```

Example: registering a LED

```
knot_proxy_register(0, "LED", KNOT_TYPE_ID_SWITCH,  
                  KNOT_VALUE_TYPE_BOOL, KNOT_UNIT_NOT_APPLICABLE,  
                  write_led, NULL);
```



KNOT THING SDK: CREATING AN APPLICATION

Define when sending events to cloud



```
bool knot_proxy_set_config(u8_t id, ...);
```

Example: when data item has changed



```
knot_proxy_set_config(0, KNOT_EVT_FLAG_CHANGE, NULL);
```



KNOT THING SDK: STRUCTURE

Branch: master ▼

zephyr-knot-sdk / apps / toggle / src / **toggle.c**

<https://github.com/CESARBR/zephyr-knot-sdk/blob/master/apps/toggle/src/toggle.c>



THE DEVELOPER'S
CONFERENCE



C . E . S . A . R . A

KNOT THING SDK: TOGGLE APPLICATION



```
//imports...
#define TOGGLE_PORT LED1_GPIO_CONTROLLER /* General GPIO Controller */
#define TOGGLE_PIN LED1_GPIO_PIN /* User LED */

LOG_MODULE_REGISTER(toggle, LOG_LEVEL_DBG);

bool toggle = true; /* Tracked value */
struct device *gpio_led; /* GPIO device */

void write_led(struct knot_proxy *proxy)
{
    knot_proxy_value_get_basic(proxy, &toggle);
    LOG_INF("Value for toggle changed to %d", toggle);

    gpio_pin_write(gpio_led, TOGGLE_PIN, !toggle); /* Led is On at LOW */
}
```



KNOT THING SDK: TOGGLE APPLICATION

Function to get the current value of data item

```
● ● ●  
  
//imports...  
#define TOGGLE_PORT LED1_GPIO_CONTROLLER /* General GPIO Controller */  
#define TOGGLE_PIN LED1_GPIO_PIN /* User LED */  
  
LOG_MODULE_REGISTER(toggle, LOG_LEVEL_DBG);  
  
bool toggle = true; /* Tracked value */  
struct device *gpio_led; /* GPIO device */  
  
void write_led(struct knot_proxy *proxy)  
{  
    knot_proxy_value_get_basic(proxy, &toggle);  
    LOG_INF("Value for toggle changed to %d", toggle);  
  
    gpio_pin_write(gpio_led, TOGGLE_PIN, !toggle); /* Led is On at LOW */  
}
```



KNOT THING SDK: TOGGLE APPLICATION

Function to get the current value of data item



```
//imports...
#define TOGGLE_PORT LED1_GPIO_CONTROLLER /* General GPIO Controller */
#define TOGGLE_PIN LED1_GPIO_PIN /* User LED */

LOG_MODULE_REGISTER(toggle, LOG_LEVEL_DBG);

bool toggle = true; /* Tracked value */
struct device *gpio_led; /* GPIO device */

void write_led(struct knot_proxy *proxy)
{
    knot_proxy_value_get_basic(proxy, &toggle);
    LOG_INF("Value for toggle changed to %d", toggle);

    gpio_pin_write(gpio_led, TOGGLE_PIN, !toggle); /* Led is On at LOW */
}
```



Write value to GPIO

KNOT THING SDK: TOGGLE APPLICATION



```
void setup(void)
{
    /* Peripherals control */
    gpio_led = device_get_binding(TOGGLE_PORT);
    gpio_pin_configure(gpio_led, TOGGLE_PIN, GPIO_DIR_OUT);

    /* KNoT config */
    knot_proxy_register(0, "LED", KNOT_TYPE_ID_SWITCH,
                       KNOT_VALUE_TYPE_BOOL, KNOT_UNIT_NOT_APPLICABLE,
                       write_led, NULL);

    knot_proxy_set_config(0, KNOT_EVT_FLAG_CHANGE, NULL);
}

void loop(void)
{
}
```



KNOT THING SDK: TOGGLE APPLICATION

Register the led with
the write function



```
void setup(void)
{
    /* Peripherals control */
    gpio_led = device_get_binding(TOGGLE_PORT);
    gpio_pin_configure(gpio_led, TOGGLE_PIN, GPIO_DIR_OUT);

    /* KNoT config */
    knot_proxy_register(0, "LED", KNOT_TYPE_ID_SWITCH,
                       KNOT_VALUE_TYPE_BOOL, KNOT_UNIT_NOT_APPLICABLE,
                       write_led, NULL);

    knot_proxy_set_config(0, KNOT_EVT_FLAG_CHANGE, NULL);
}

void loop(void)
{
}
```

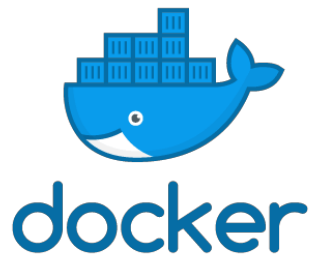


FLASHING THE KNOT THING APP

1. `git clone https://github.com/cesarbr/zephyr-knot-sdk`
2. `docker build --tag=knot-zephyr-sdk .`
3. `docker run -ti -v $(pwd)/:/workdir cesarbr/knot-zephyr-sdk:latest`
4. `container> $ knot make -b dongle`
5. `container> $ knot export /workdir/output`
6. `container> $ knot make -b dongle -flash`
(if Linux)



nRF Connect



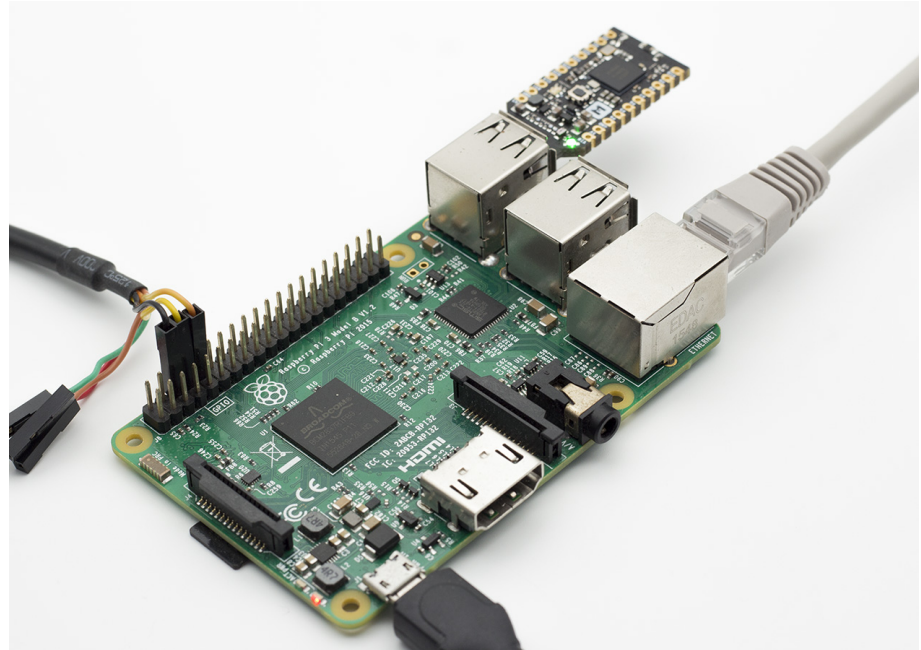
THE DEVELOPER'S
CONFERENCE



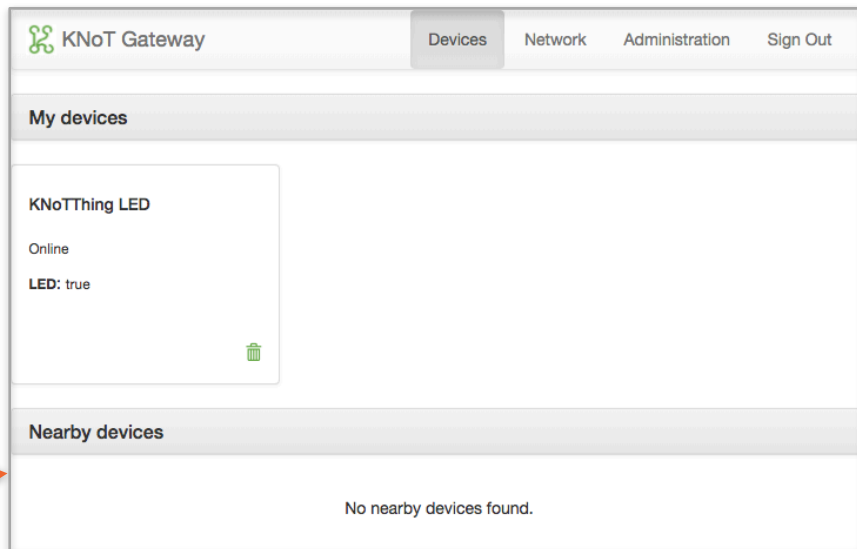
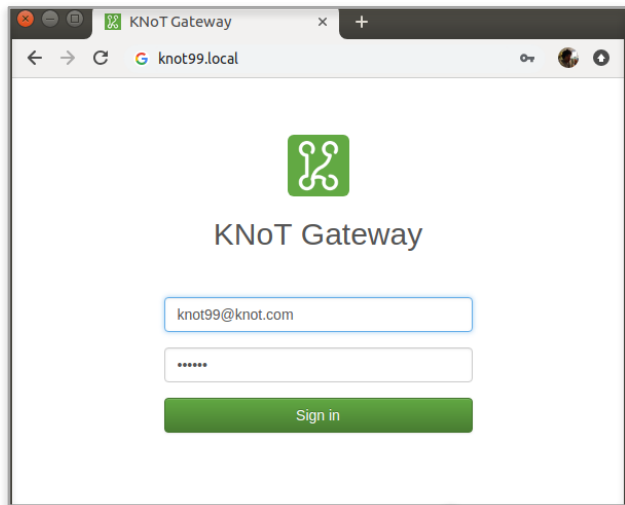
C.E.S.A.R.A

KNOT GATEWAY SPECIFICATION

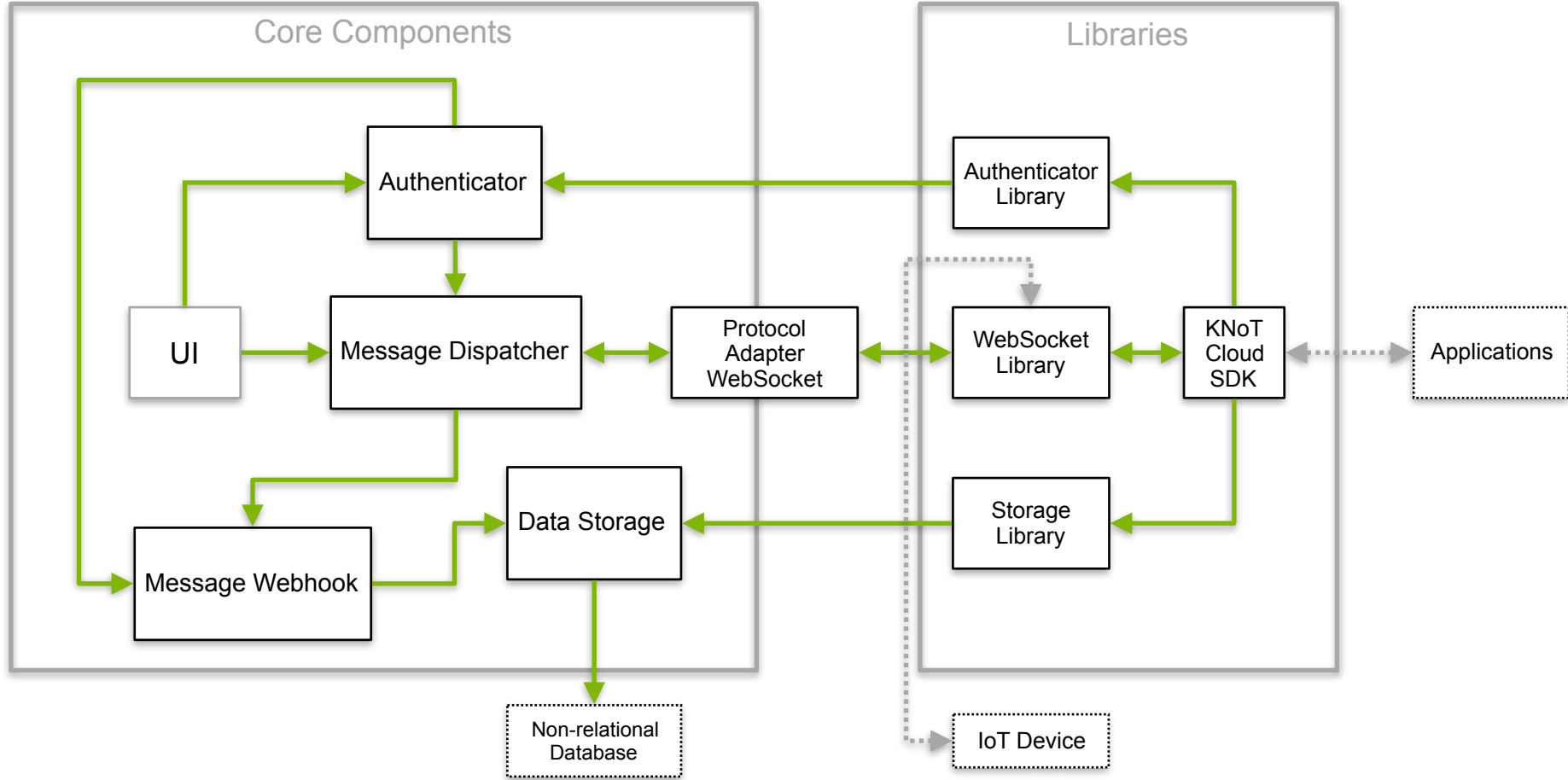
- Raspberry Pi 3 B
- NCP (Network Co-Processor)
- BLE
- Thread
- ZigBee (802.15.4)



KNOT GATEWAY INTERFACE



KNOT CLOUD SPECIFICATION



CONNECTING AN APPLICATION TO THE CLOUD

KNoT Cloud SDK for JavaScript (oficial):

<https://github.com/CESARBR/knot-cloud-sdk-js>

KNoT Cloud SDK for Go (under development):

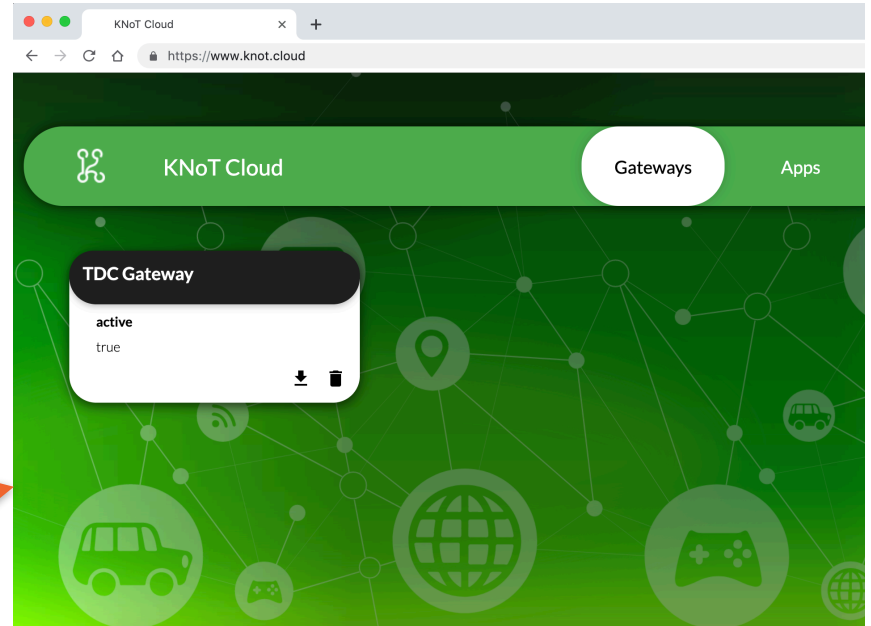
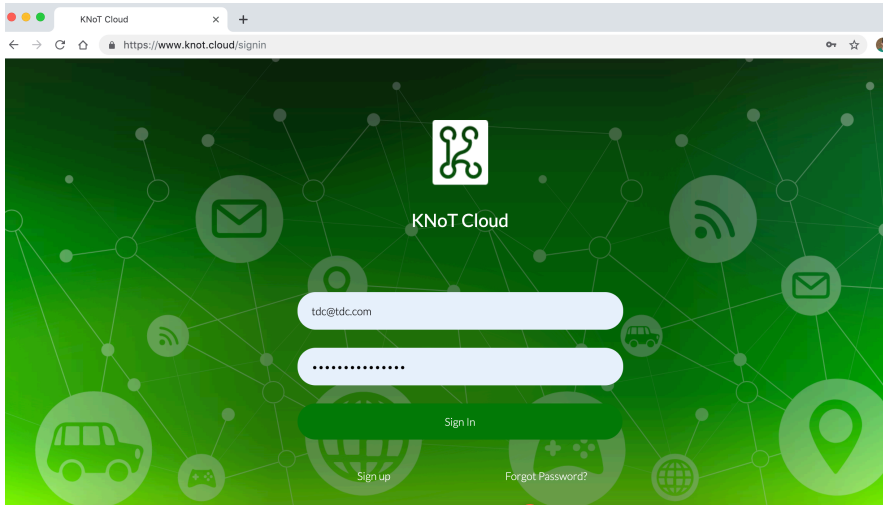
<https://github.com/joaoaneto/knot-cloud-sdk-go>

KNoT Cloud SDK for Java (under development):

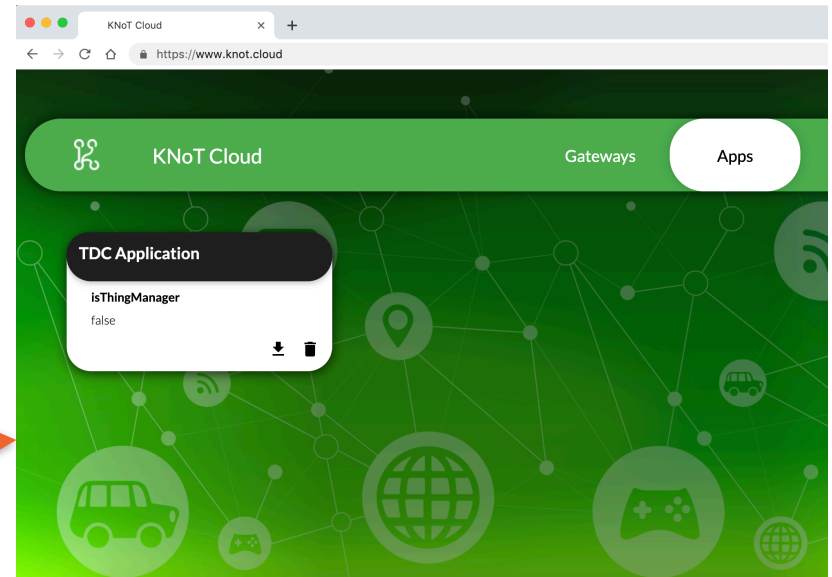
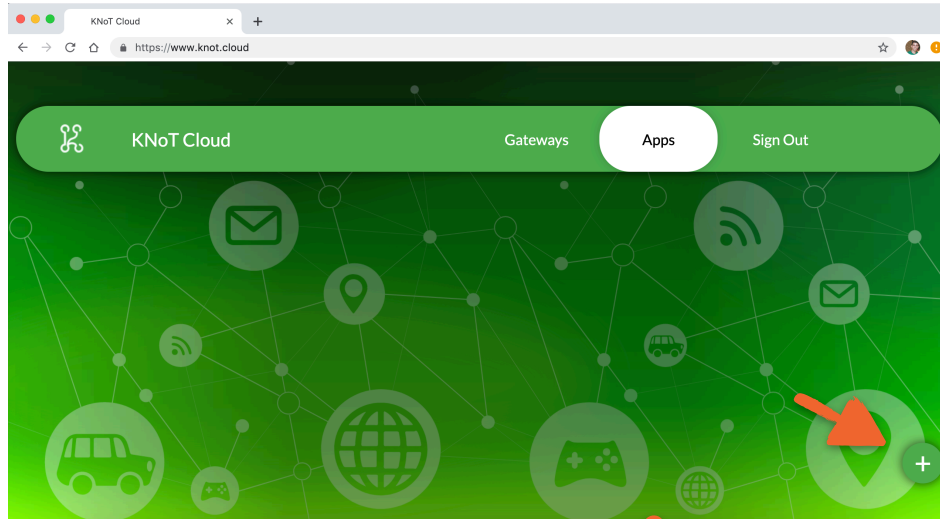
<https://github.com/joaoaneto/knot-cloud-sdk-java>



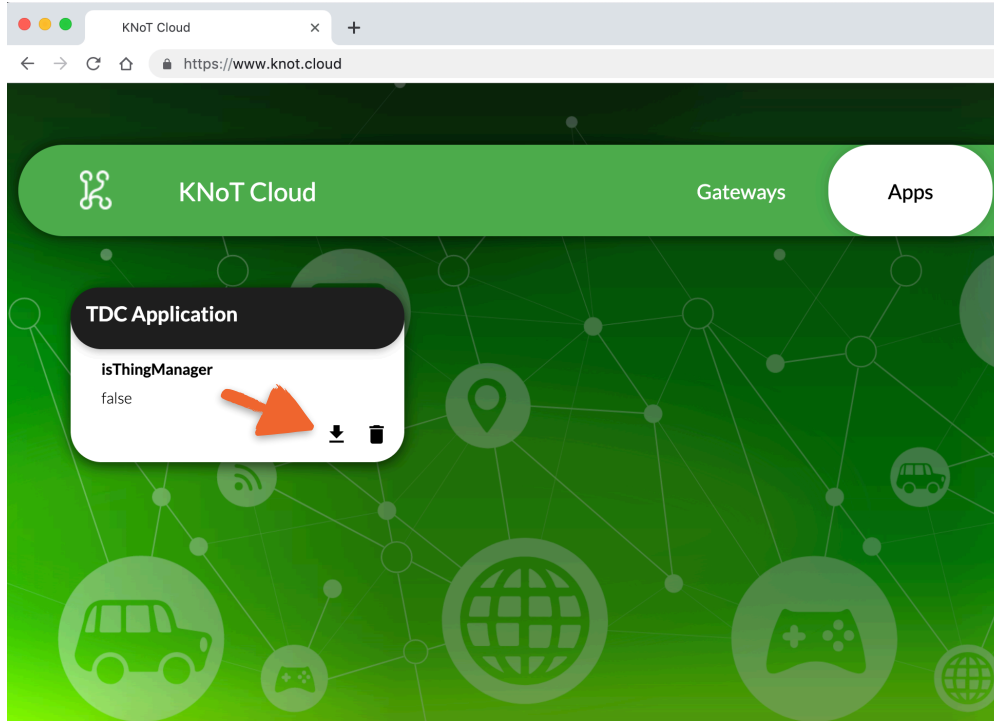
CONNECTING AN APPLICATION TO THE CLOUD



CONNECTING AN APPLICATION TO THE CLOUD



CONNECTING AN APPLICATION TO THE CLOUD



```
{
  "type": "knot:app",
  "metadata": {
    "name": "TDC Application"
  },
  "knot": {
    "isThingManager": false,
    "id": "18d390db-91f5-492e-9b79-abea21ebb262"
  },
  "token": "3d8f9d34a643de2b305d2566587c52e30192dc95"
}
```

CONNECTING AN APPLICATION TO THE CLOUD

Import SDK and create a WebSocket client instance



```
const { Client } = require('@cesarbr/knot-cloud-sdk-js');

const client = new Client({
  hostname: 'ws.knot.cloud',
  protocol: 'wss',
  port: 443,
  id: '18d390db-91f5-492e-9b79-abea21ebb262', // APP ID
  token: '3d8f9d34a643de2b305d2566587c52e30192dc95', // APP TOKEN
});
```



CONNECTING AN APPLICATION TO THE CLOUD

Connect to cloud and set device's sensor 0 value to true

```
const data = [
  {
    sensorId: 0,
    value: true
  },
];

client.on('ready', () => {
  client.setData('18571e1dcfb5dce9', data);
});

client.on('sent', () => {
  client.close();
});

client.connect();
```



EDUCATIONAL EFFORTS



Human Resources



ESCOLA
POLITÉCNICA
DA USP



UFRJ



UNIVERSIDADE
DE PERNAMBUCO



UFRN



CENTRO UNIVERSITÁRIO TABOSA DE ALMEIDA



UFAM



INSTITUTO
FEDERAL
Pernambuco



UFPB



PUCRS

<https://www.cesar.school/index.php/pos-em-iot-internet-of-things/>



THE DEVELOPER'S
CONFERENCE



C.E.S.A.R

STANDARDIZATION AND INTERNATIONALIZATION EFFORTS

It points out the conditions for improving the innovation environment (including funding sources) and leverage the internationalization of Brazilian IoT solutions.



Innovation and internationalization

- ▶ CESAR: Architectural reference models of devices for IoT applications
- ▶ MCTIC+ANATEL: Framework for Internet of things ecosystem master plan
- ▶ USP+MACKENZIE + UNESP: Framework and capabilities for smart livestock farming based on Internet of things



IEEE
Internet of Things



OBRIGADO!



João Neto | jasn@cesar.org.br

www.knot.cesar.org.br

