# iottly

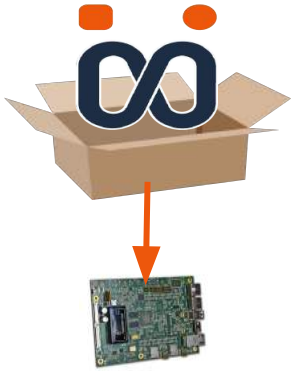
REMOTE CONTROL & TELEMETRY

## Technical description

TomorrowData

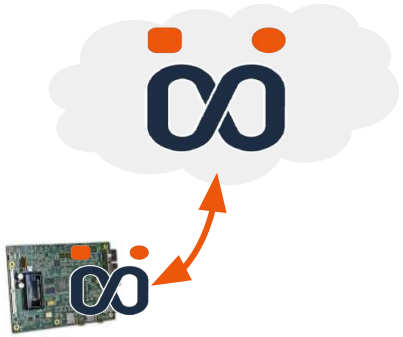
# How does it work?

1



**Instantly** install and launch iottlyAgent on devices

2

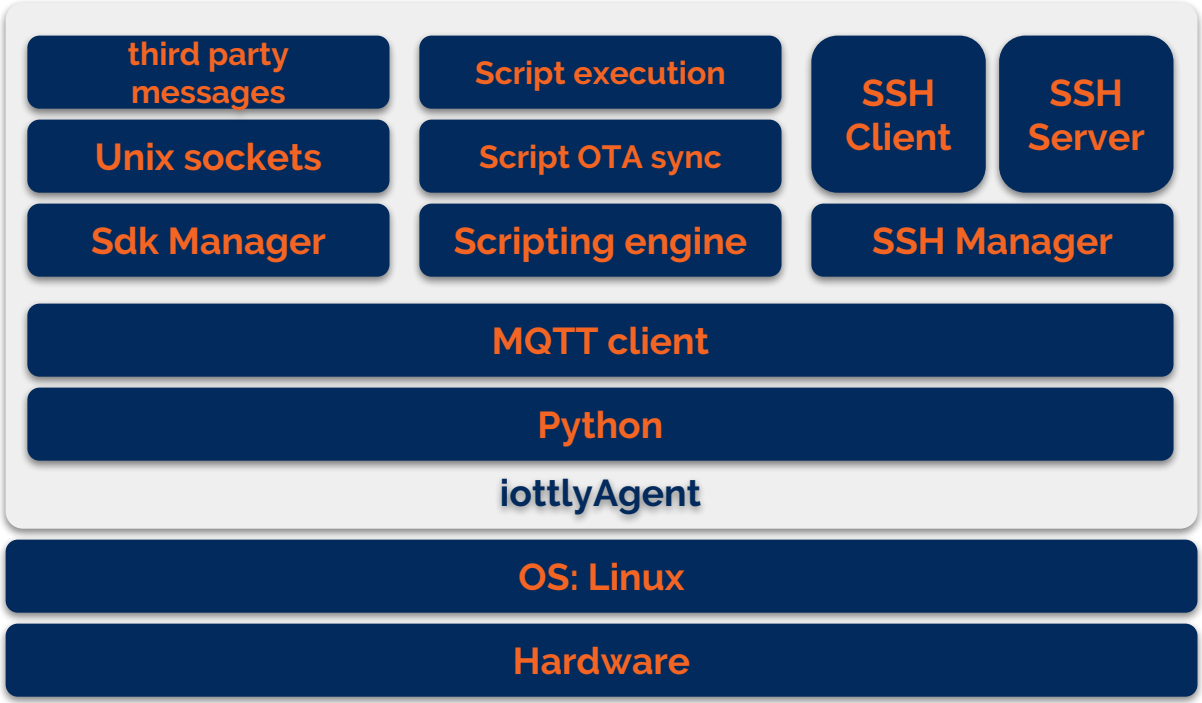


Devices **autonomously connect** to iottlyCloud (MQTT/TLS)

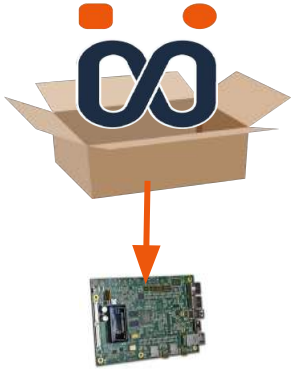
3

Technicians operate on the devices **from everywhere** via the web app

# iottlyAgent architecture



# iottlyAgent: Technical Specs



## Easy device setup

- IottlyAgent coexists without interfering with third-party firmware
- Footprint: 30Mb flash, 6Mb RAM

## Available for Linux

ARMv5 ARMv6+ AMD64 i386

- Based on static python tailored for embedded devices
- Rapid portability on any platform

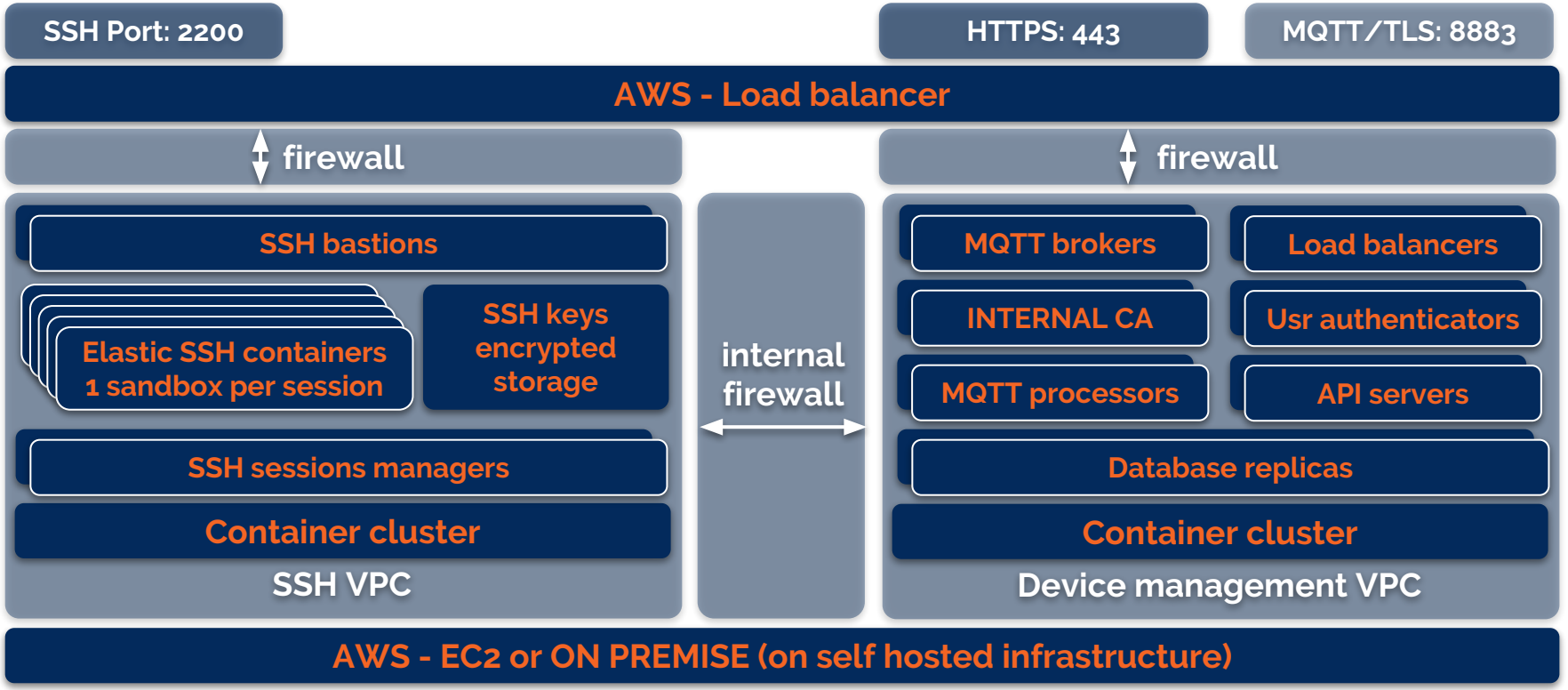
## Outbound Connections Only

- Outbound: iottlyCloud 8883 (MQTT/TLS)
- Outbound: iottlyCloud 2200 (SSH)

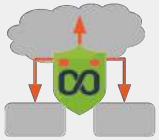
## Connectivity: LAN, WiFi, LTE, GPRS

- Lightweight MQTT protocol
- Secure communications: TLS 2048

# iottlyCloud architecture



# Core features



off the shelf **security**

on-device **SDK**

on-device python **scripting engine**

**Telemetry middleware** via cloud APIs

# Management features

iottly

everywhere **web SSH**

real-time **control panel**

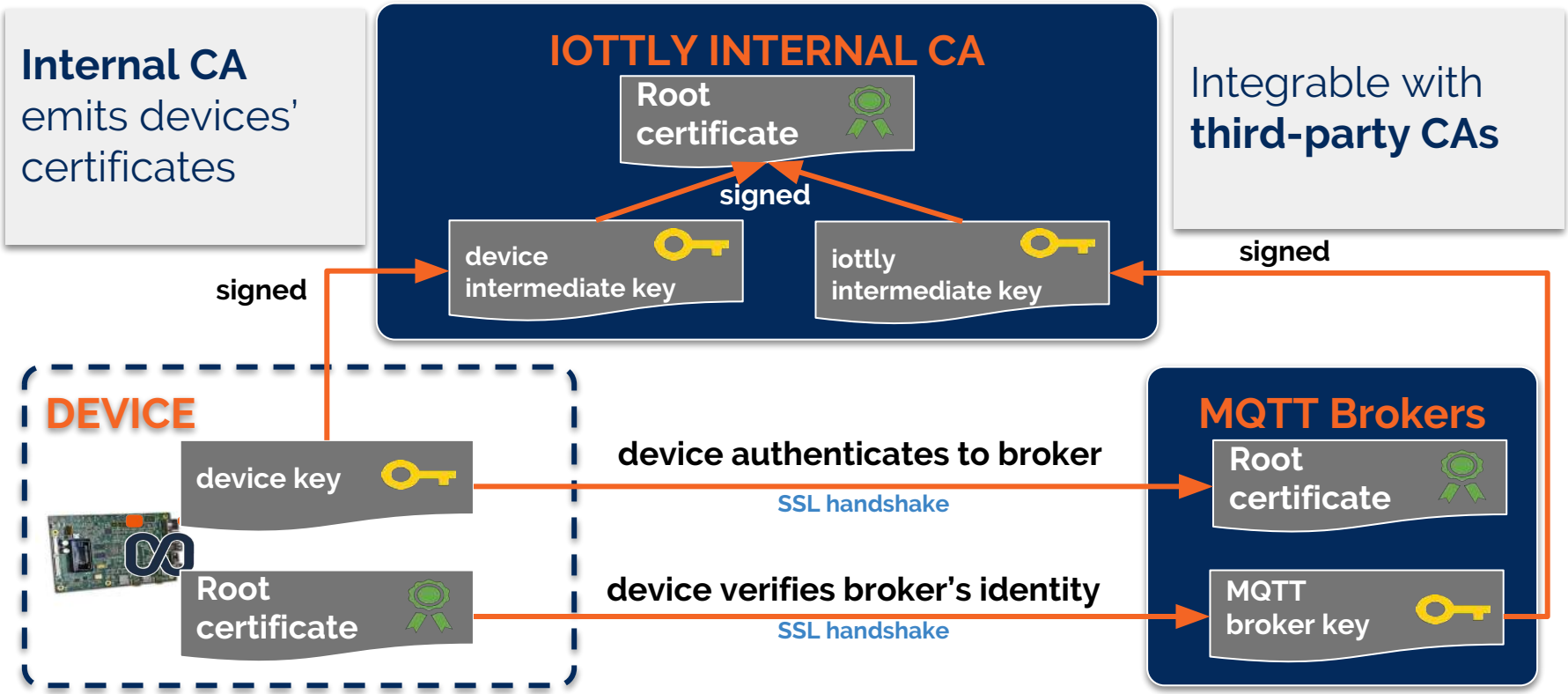
Flash over the air

third party **firmware upgrades**

fleet management with **groups**

# Off the shelf security - MQTT X.509

**Internal CA**
emits devices' certificates



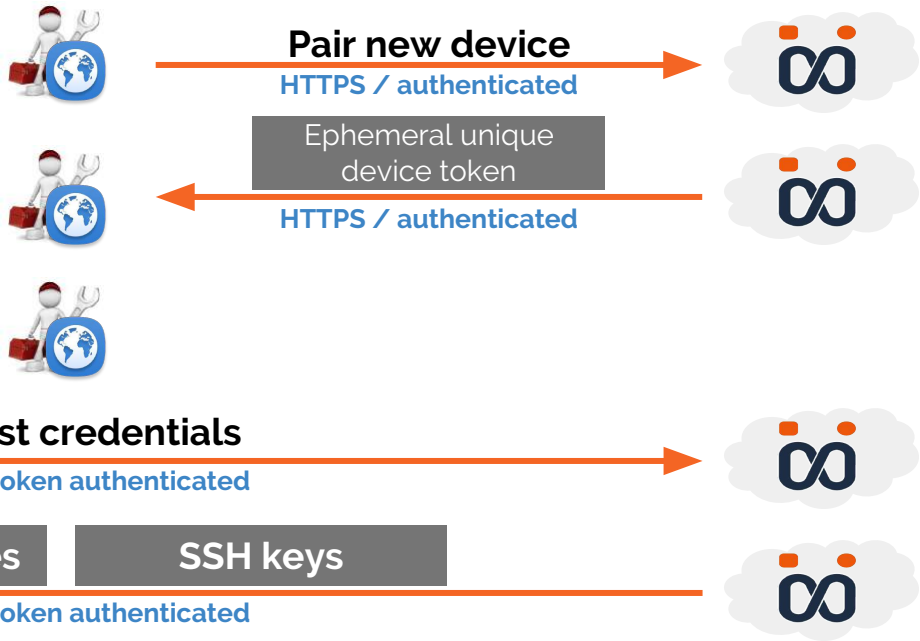

Integrable with **third-party CAs**

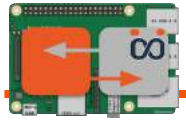
# Off the shelf security - pairing

## Device identification and registration phase.

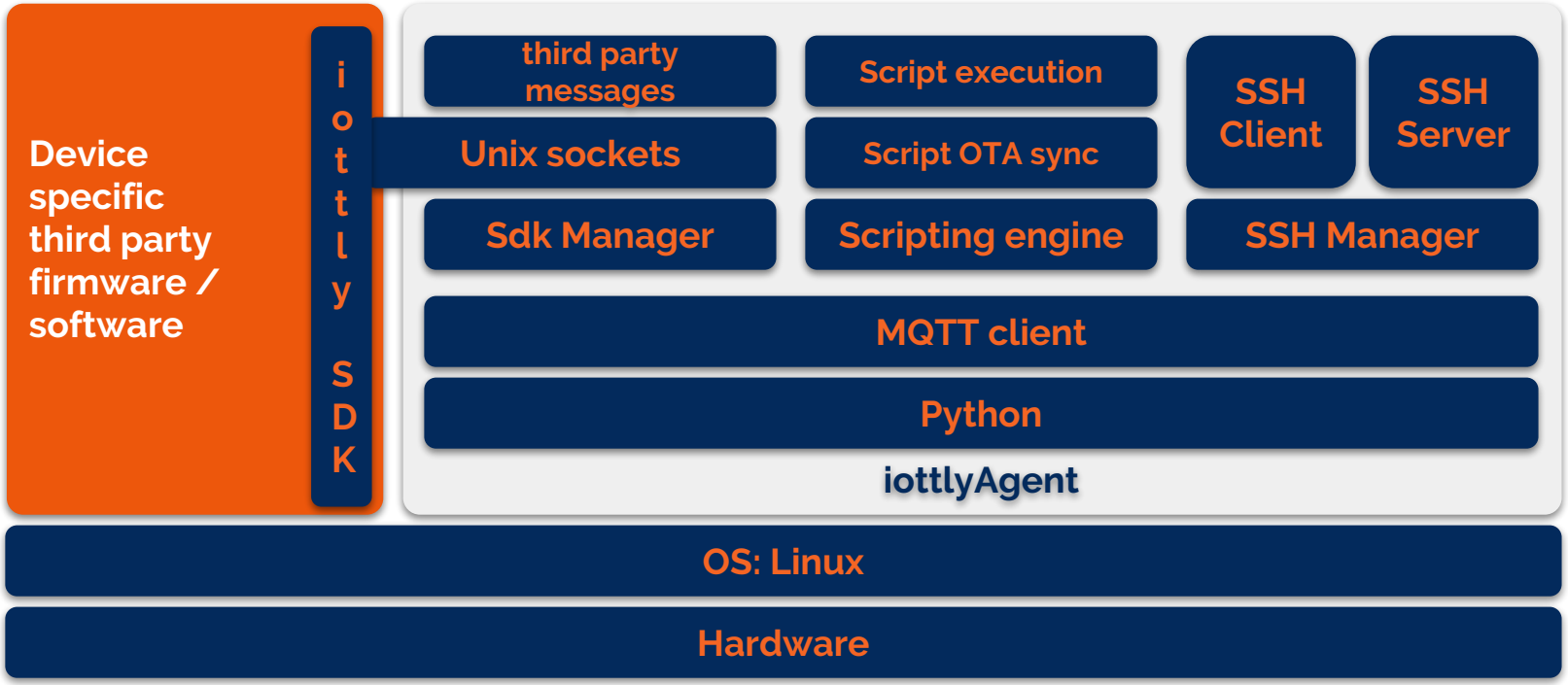
Unique and unambiguous **device identification**.

**Integrable** with third-party bulk flashing systems.

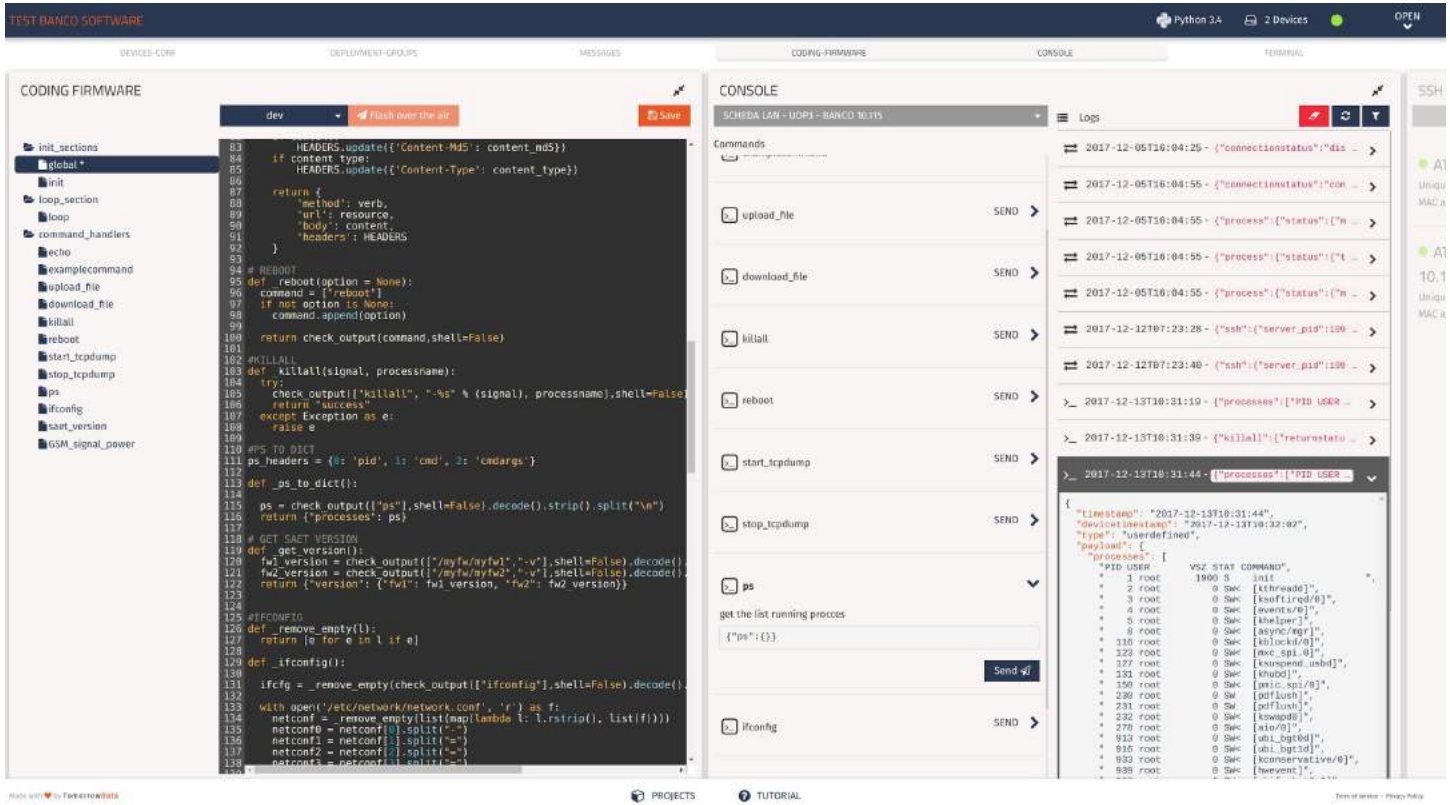




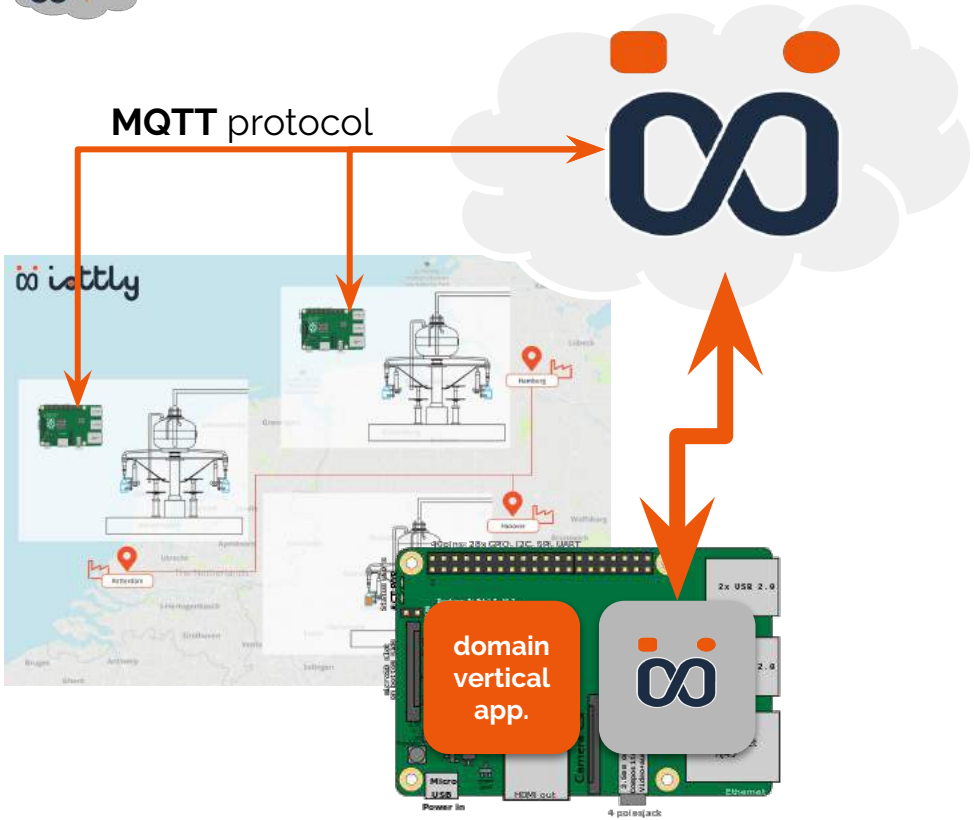
# on-device agent architecture - SDK



# on-device scripting engine - web



# Telemetry middleware via cloud APIs




**The embedded devices are connected to iottly over the internet**

both cloud or on-premise

secure connections

➔ iottly applies enterprise grade security standards
(TLS 2048 / X.509)

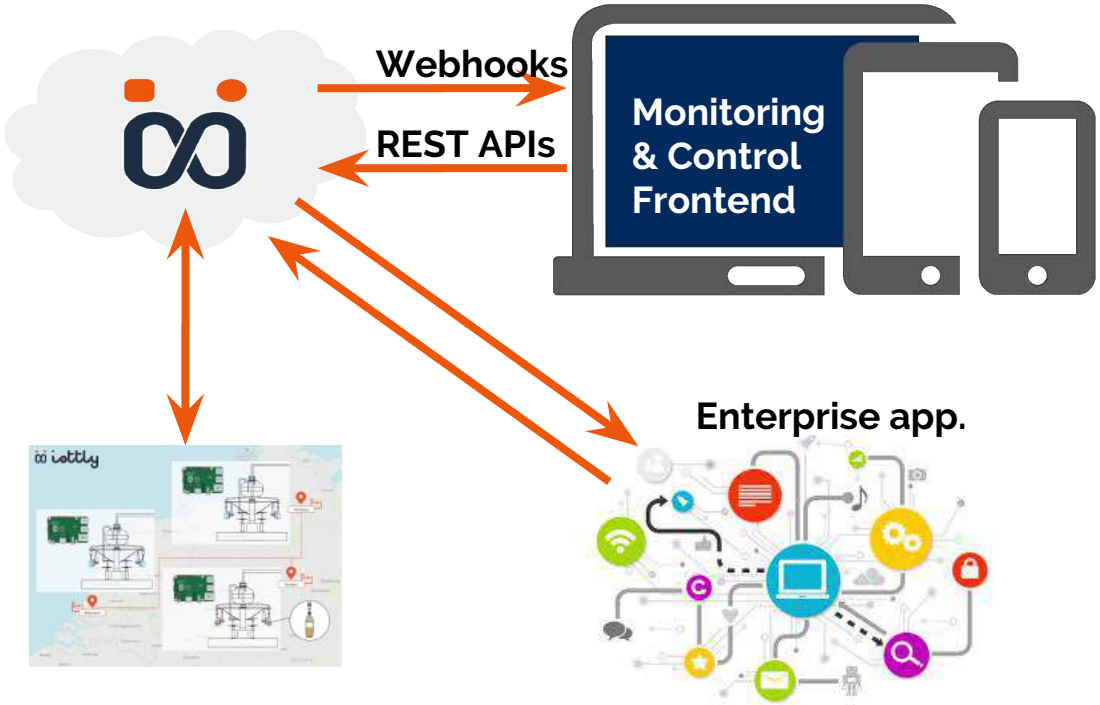
iottly management software running on the devices

➔ iottly provides a standard software to be installed on any kind of device

**edge:** domain vertical running on the devices

➔ test, debug, upgrade, maintain the custom application on the device

# Telemetry middleware via cloud APIs




**Third-party applications communicate with the devices <u>through iottly</u>**

Real-time notifications

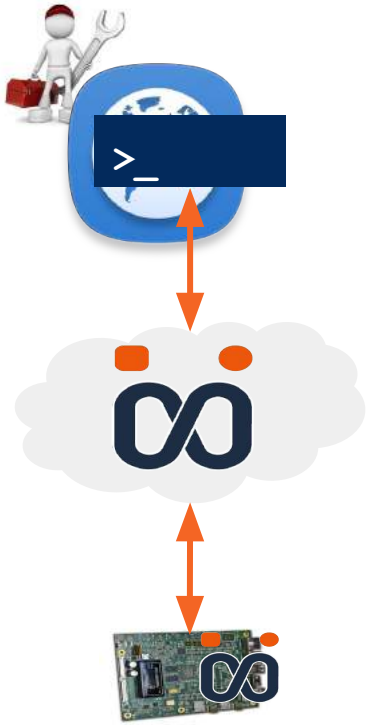
➔ iottly pushes messages from the devices to third-party applications, through webhooks

Trusted active controls

➔ third-party applications can securely interact with the devices through iottly rest apis

# everywhere web SSH

iottly



**Unique access and management console**

**100% web based**
no need to install client software

**Zero inbound connections**
no open ports on the device

**Access enabled by the device** via MQTT

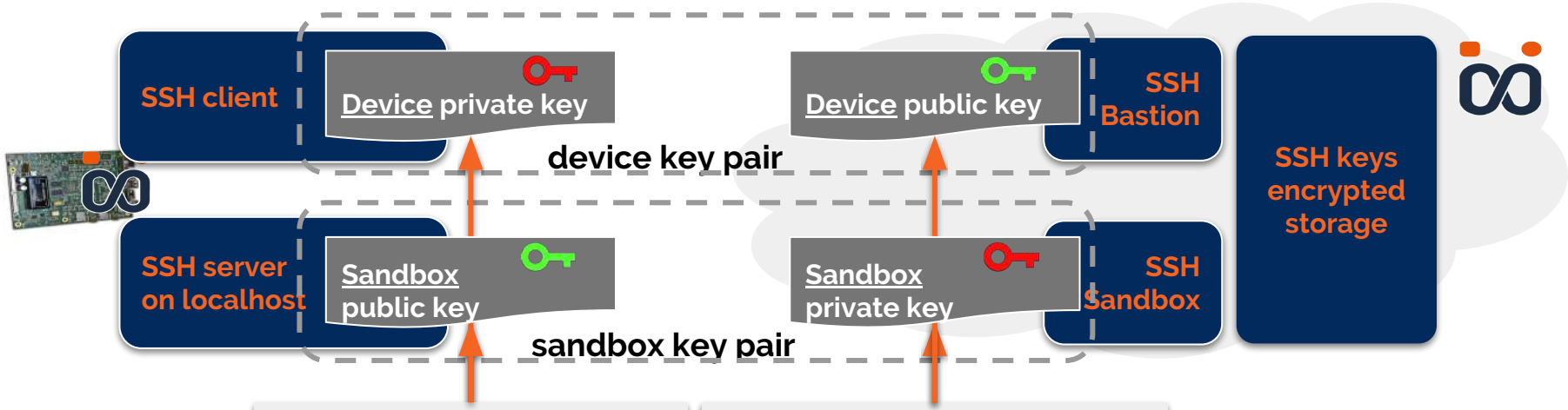
**Customizable access control**
role based policies

**Automatic activity auditing and reporting**

**Works behind NATs**
No need to maintain long IP addresses lists

**Works with devices connected to mobile networks** GPRS, 3G, 4G

# everywhere web SSH - keys




These keys are saved on the device during the pairing phase

These keys are saved in the encrypted storage and mounted in RAM at the initiation of the SSH session

Optionally, both the key pairs can be dynamically generated during the initiation of each SSH session

TomorrowData

iottly

**Tomorrowdata Srl**
*Strada Basse di Dora 42 - 10146 Torino*
tomorrowdata.io
info@tomorrowdata.io
iottly@tomorrowdata.io

# Keep in touch!

*@stefanoterna*
*stefano.terna@tomorrowdata.io*
*stefanoterna*
*stefanoterna*

*@danielegrieco*
*daniele.grieco@tomorrowdata.io*
*danielegrieco*