

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA E TELECOMUNICAZIONI

DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE

"Guglielmo Marconi"

DEI

TESI DI LAUREA

IN RETI DI TELECOMUNICAZIONI

Configurazione della piattaforma ThingsBoard per l'IoT e integrazione con la tecnologia Multi-access Edge Computing (MEC)

CANDIDATO

Ophelia Giannini

RELATORE

Chiar.ma prof.ssa Carla Raffaelli

CORRELATORE

Chiar.mo prof. Walter Cerroni

Anno Accademico 2020/2021

Sessione II – Ottobre 2021

Piattaforme IoT

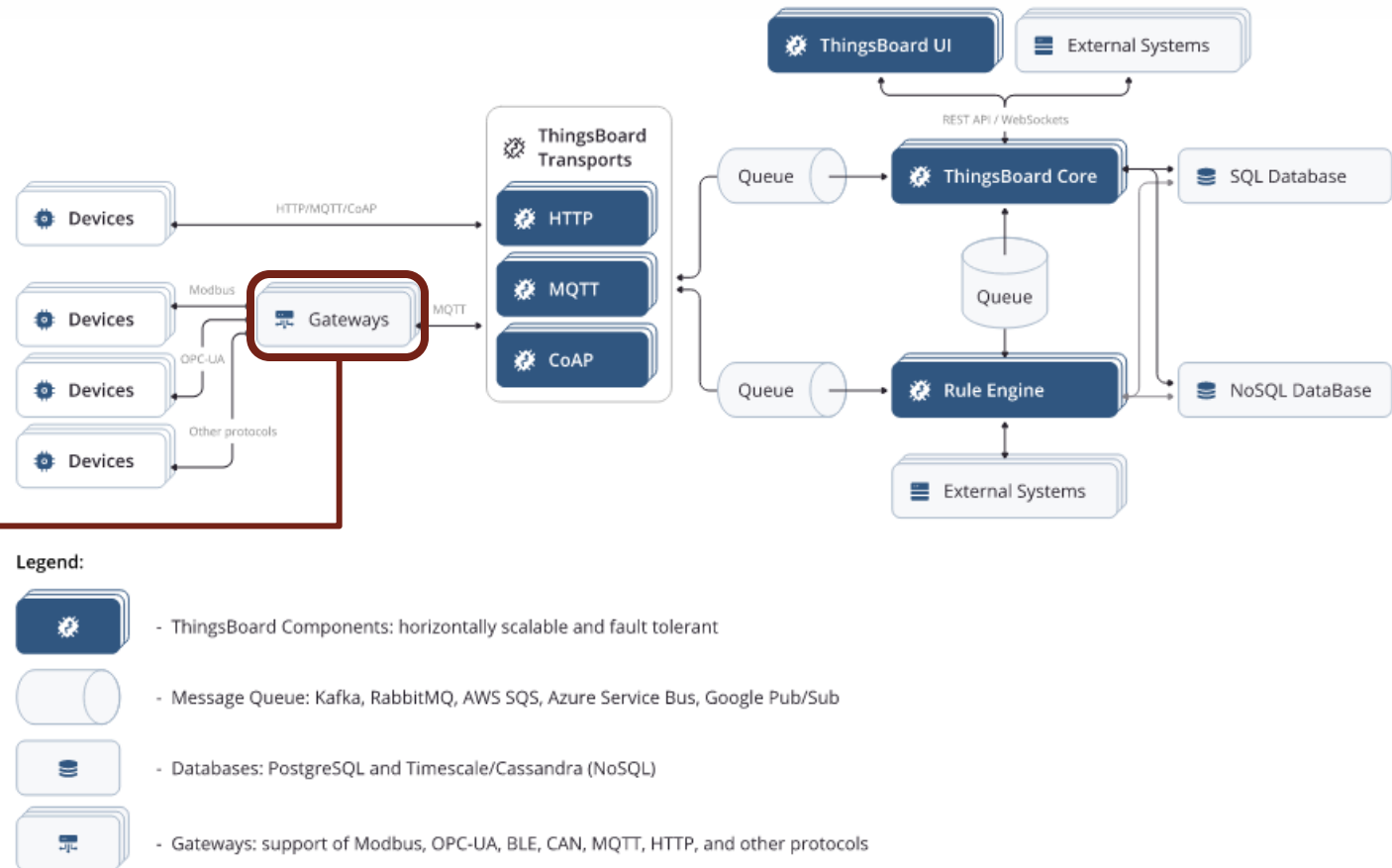
Sono strumenti che permettono l'amministrazione e la gestione di tutti gli elementi di un'applicazione IoT e permettono la rapida visualizzazione dei dati raccolti

Tra le varie piattaforme esistenti, ThingsBoard è una soluzione open-source che permette lo sviluppo, l'amministrazione e la scalabilità di progetti IoT

All'interno di una macchina virtuale Linux è stato creato e avviato un container con l'immagine Docker della piattaforma, al fine di testare le sue funzionalità

Architettura di ThingsBoard

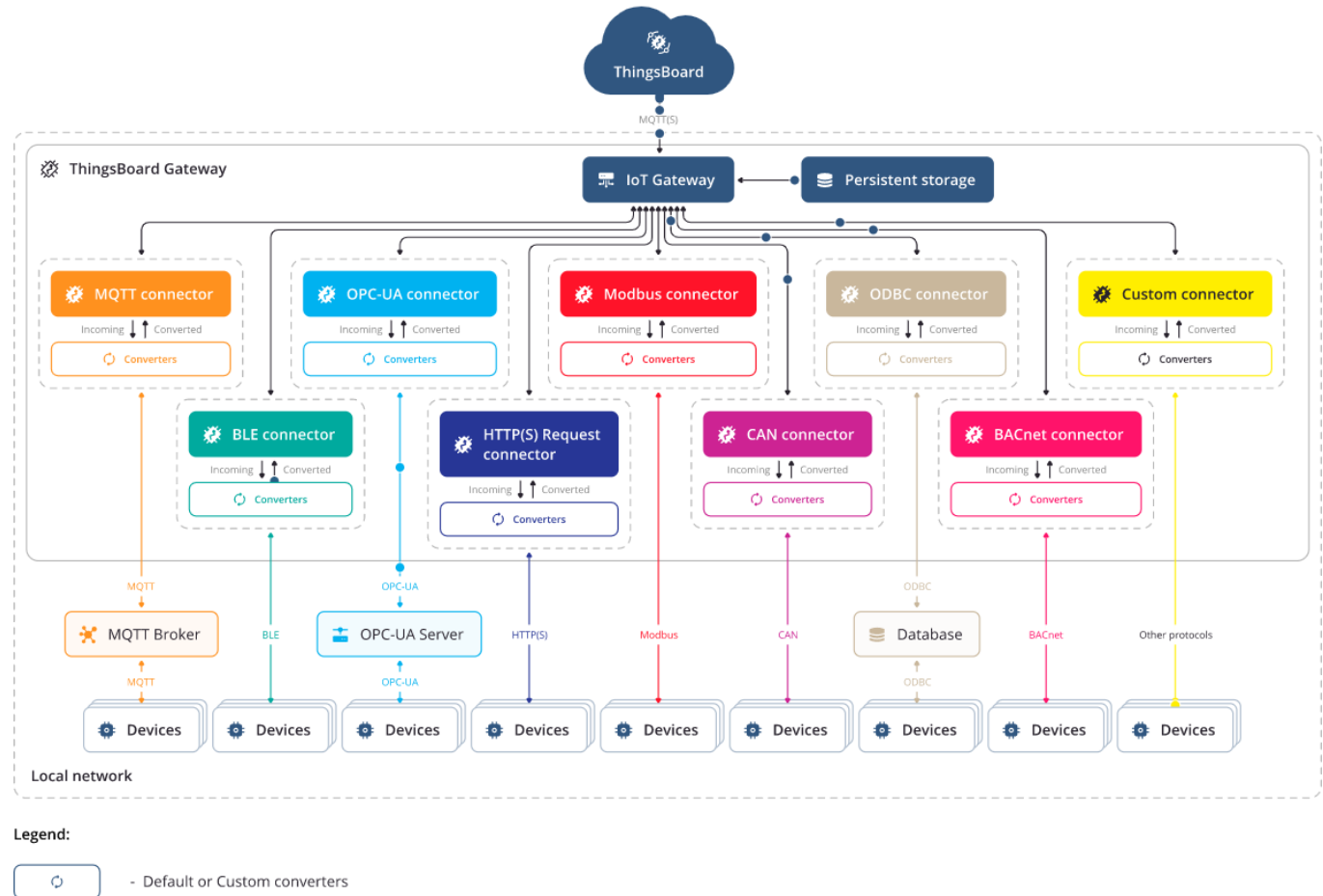
- Componente che permette di aggregare dati da più dispositivi e di inviarli alla piattaforma
- Può essere installato all'edge della rete
- ThingsBoard mette a disposizione un'immagine Docker per poter installare il gateway all'interno di un container



Architettura del gateway IoT di ThingsBoard


I componenti principali dell'architettura sono:

- Connettori: si connettono al sistema esterno e sono specifici per ogni protocollo
- Convertitori: convertono i messaggi in entrata in messaggi MQTT per poterli inviare alla piattaforma




Configurazione gateway

```
thingsboard:  
  host: indirizzo ip dell'istanza di ThingsBoard  
  port: 1883  
  security:  
    accessToken: access token del gateway  
storage:  
  type: memory  
  read_records_count: 10  
  max_records_count: 1000  
connectors:  
  
-  
  name: MQTT Broker Connector  
  type: mqtt  
  configuration: mqtt.json  
-  
  name: OPC-UA Connector  
  type: opcua  
  configuration: opcua.json  
-  
  name: BLE Connector  
  type: ble  
  configuration: ble.json
```



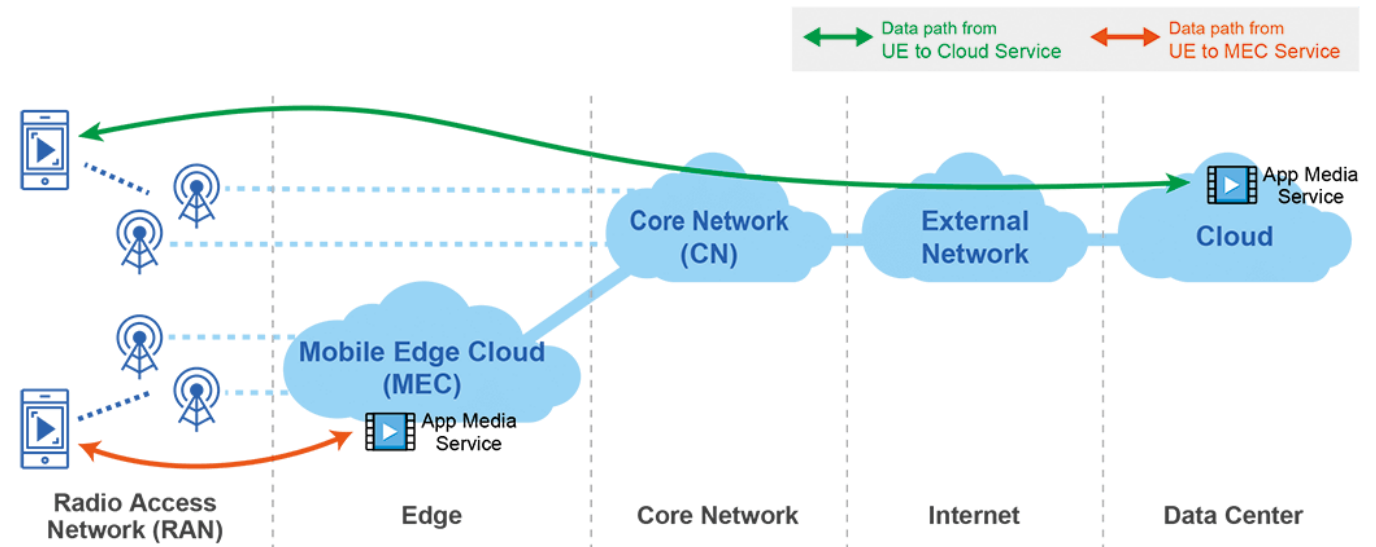
Informazioni
generali



Elenco dei
connettori da
attivare

Multi-access Edge Computing

- Sviluppato dall'European Telecommunications Standard Institute (ETSI)
- Architettura di rete che fornisce le capacità di calcolo del Cloud Computing ai margini della rete, vicino agli utenti
- L'obiettivo del MEC è quello di ridurre la latenza, garantire un funzionamento della rete altamente efficiente e migliorare l'esperienza del cliente



MEC e IoT

Nell'ambito dell'IoT, adottare un'architettura MEC è particolarmente utile perché permette di:

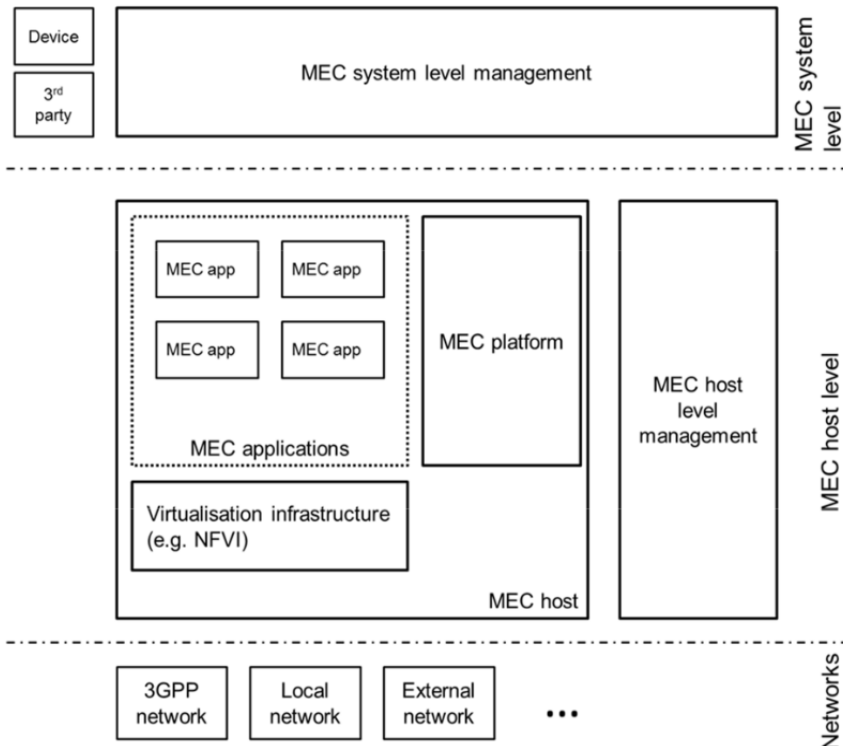
- Filtrare il traffico e quindi ridurre il traffico verso la rete centrale
- Rendere il processo decisionale più rapido perché i dati vengono elaborati localmente
- Migliorare la durata della batteria dei dispositivi IoT: diminuendo la latenza diminuisce il tempo che ciascun dispositivo impiega per trasmettere dati

In generale, un'architettura MEC è adatta per quelle applicazioni IoT che richiedono l'elaborazione dei dati in tempo reale e non possono ammettere i tempi di latenza del Cloud Computing

Applicazioni per l'loT

Dominio loT	Ruolo del MEC
Smart Home	Il MEC offre riduzione della latenza, facile istanziazione e ricollocazione. Inoltre i dati sensibili vengono elaborati localmente preservando la privacy
Smart City	Il MEC si occupa dell'elaborazione dei dati, dell'archiviazione e del recupero delle richieste ai margini della rete, fornendo così bassa latenza, alta disponibilità, gestione della mobilità e scalabilità.
Industry (IIoT)	Il MEC abilita le future applicazioni IIoT affrontando le carenze della comunicazione M2M (ad esempio latenza, connettività peer-to-peer, resilienza, costi, sicurezza).

Struttura MEC



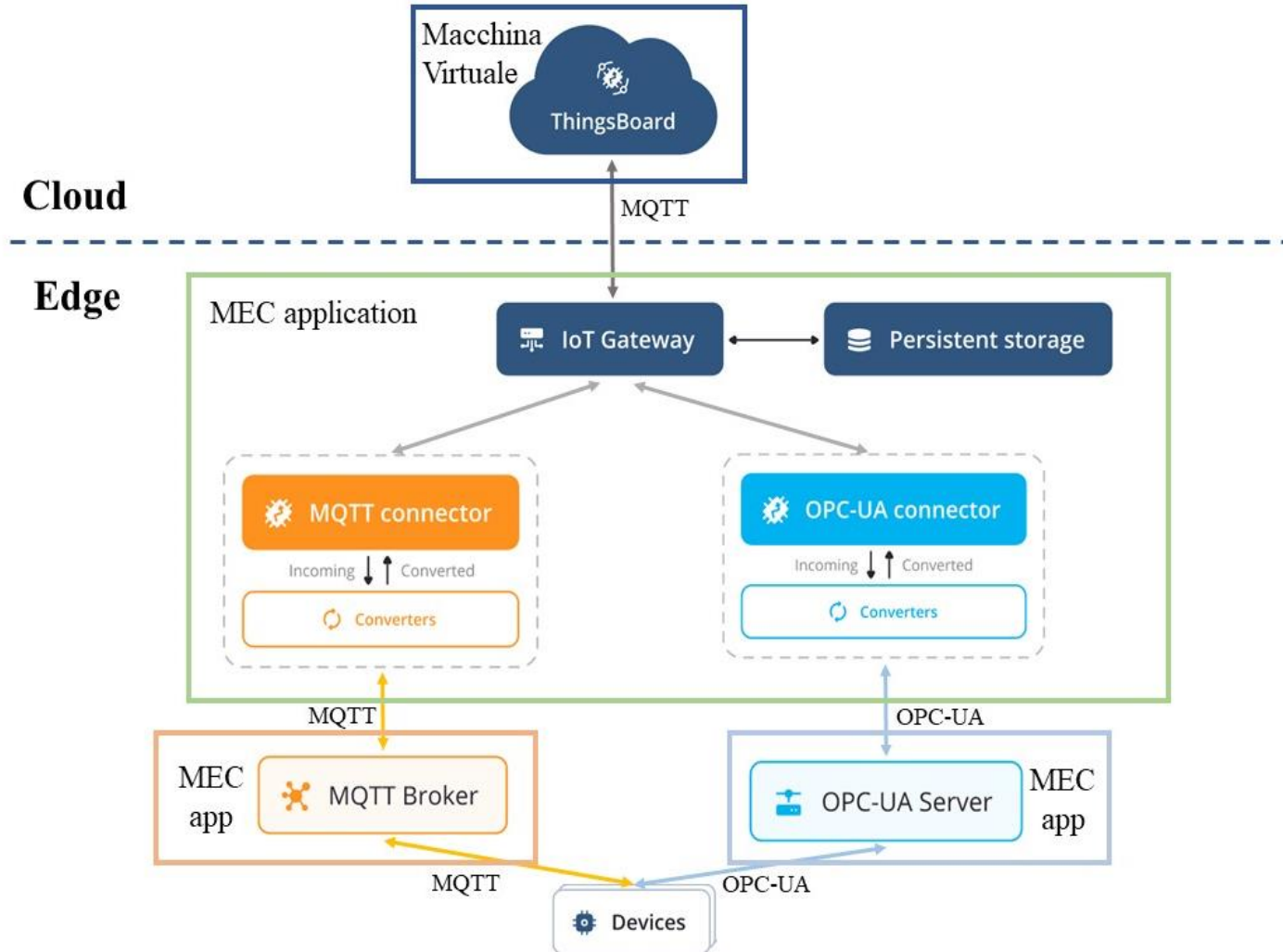
Gli elementi funzionali di un MEC host sono:

- MEC platform: l'insieme delle funzionalità essenziali per eseguire applicazioni MEC su una particolare infrastruttura di virtualizzazione e consentire loro di fornire e sfruttare servizi MEC
- MEC application: sono istanziate sull'infrastruttura di virtualizzazione del MEC host in base alle richieste del livello di gestione e possono interagire con la piattaforma MEC per fornire o sfruttare servizi
- infrastruttura di virtualizzazione

Obiettivo della tesi

L'obiettivo di questa tesi è quello di identificare possibili architetture per unire le funzionalità della piattaforma ThingsBoard con i vantaggi apportati dal Multi-access Edge Computing, al fine di garantire bassi tempi di realizzazione e facilità di riconfigurazione.

Integrazione tra TB e MEC



L'architettura realizzata è composta dai seguenti elementi:

- Una macchina virtuale dove è stata installata la piattaforma ThingsBoard
- Un'applicazione MEC per il gateway
- Un'applicazione MEC per il broker MQTT
- Un'applicazione MEC per il server OPC-UA

Strumenti utilizzati

Per realizzare le applicazioni MEC sono stati utilizzati:



- Open Source MANO (OSM), sviluppato dall'European Telecommunications Standard Institute, è un sistema di orchestrazione che serve per modellare e automatizzare servizi
- È stato utilizzato per istanziare le applicazioni MEC



- Kubernetes è una piattaforma portatile, estensibile e open-source per la gestione di carichi di lavoro e servizi containerizzati, in grado di facilitare sia la configurazione dichiarativa che l'automazione
- Un'applicazione Kubernetes è definita da un file in formato yaml chiamato Helm Chart

```
---
apiVersion: v1
kind: Pod
metadata:
  name: thingsboard-gateway
  labels:
    app: thingsboard-gateway
spec:
  containers:
    - name: thingsboard-gateway-1
      image: thingsboard/tb-gateway
      ports:
        - containerPort: 1883
      imagePullPolicy: Always
      volumeMounts:
        [...]
  volumes:
    - name: config
      configMap:
        name: gateway-config
    [...]

```

```
---
apiVersion: v1
kind: Service
metadata:
  name: thingsboard-gateway
spec:
  type: ClusterIP
  selector:
    app: thingsboard-gateway
  ports:
    - port: 1883
      targetPort: 1883
      protocol: TCP
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: gateway-config
data:
  tb_gateway.yaml: |
    [...]

```

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: gateway-mqtt-config
data:
  mqtt.json: |
    [...]
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: gateway-opcua-config
data:
  opcua.json: |
    [...]

```

Helm chart

OSM

Per poter creare un'istanza, OSM richiede:

il Virtual Network Function Descriptor (VNFD): nel file viene descritta la VNF e specificato l'Helm Chart corrispondente

il Network Service Descriptor (NSD): un file che elenca le VNF che compongono il Network Service e come si connettono tra loro. Un Network Service è il componente minimo di OSM e raggruppa in un unico oggetto un insieme di funzioni di rete interconnesse (VNF)

Test per il protocollo MQTT

Simulazione dell'invio di dati da parte del sensore SN-001 da linea di comando:

```
mosquitto_pub -d -h 10.15.253.15 -p 1883 -t /sensor/data -m '{"serialNumber": "SN-001", "sensorType": "Thermometer", "sensorModel": "T1000", "temp": 42, "hum": 58}'
```

Dove:

- mosquitto_pub è il comando per simulare un publisher MQTT
- -h 10.15.253.15 è l'indirizzo ip che il broker espone internamente al cluster Kubernetes
- -p 1883 è la porta tipicamente utilizzata per i messaggi MQTT
- -t /sensor/data è il topic relativo al messaggio



SN-001
Dettagli dispositivo

Details Attributi **Ultima telemetria** Allarmi Eventi Relazioni Log Audit

Ultima telemetria

<input type="checkbox"/>	Ultimo aggiornamento	Chiave ↑	Valore
<input type="checkbox"/>	2021-09-16 10:39:16	humidity	62
<input type="checkbox"/>	2021-09-16 10:39:16	temperature	42

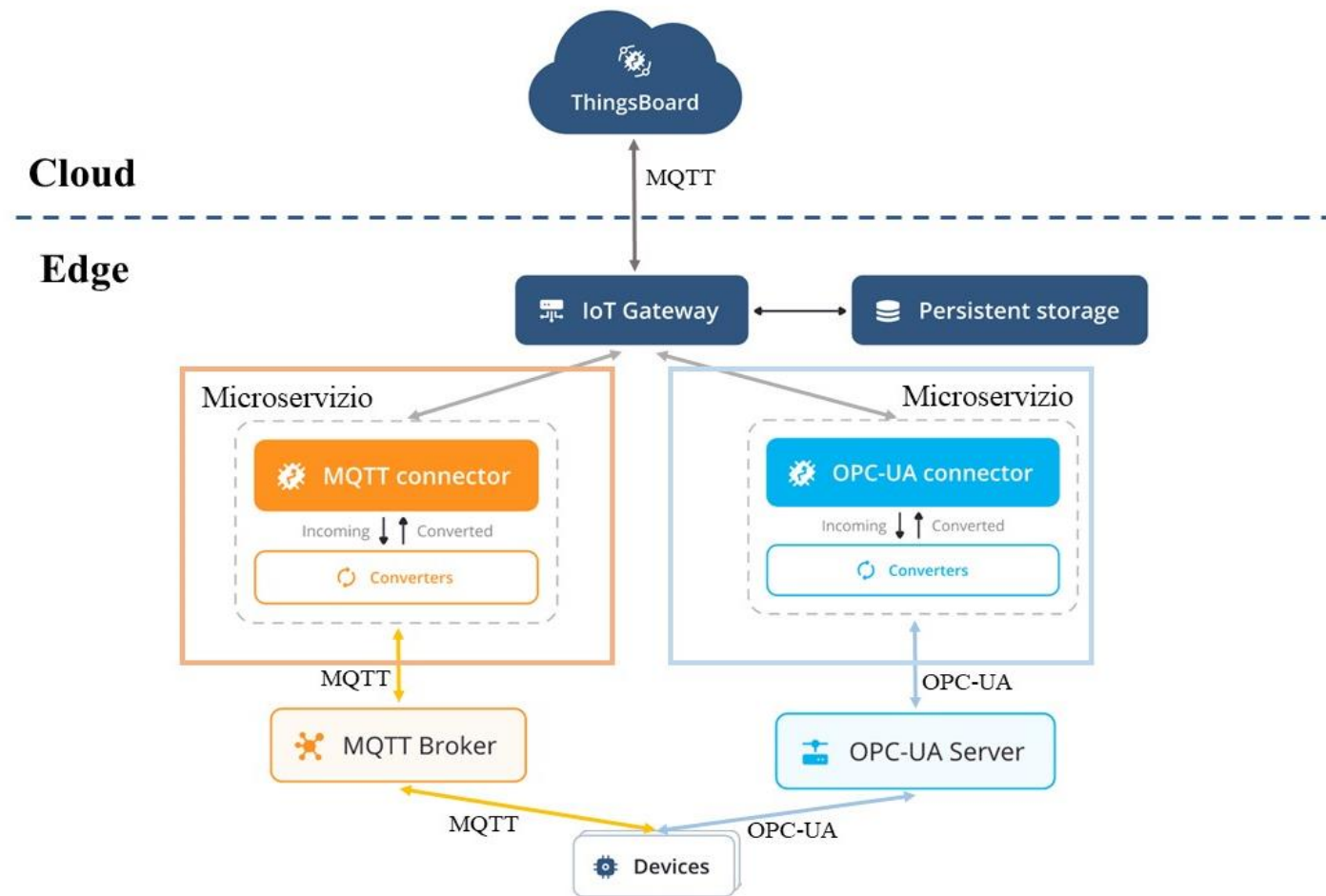


Sviluppi futuri

Possibili lavori futuri riguardano la realizzazione di un'architettura potenzialmente più flessibile di quella proposta, in cui si effettua un'ulteriore disgregazione del gateway di ThingsBoard nelle sue componenti principali: i connettori.

Ciascun connettore, infatti, potrebbe rappresentare un microservizio.

La complessità in un questo scenario risiede nella connessione tra microservizi e la piattaforma ThingsBoard.





Conclusioni



Per riassumere, è stato mostrato come un'architettura mista sia possibile



In particolare, la struttura realizzata supporta due protocolli spesso usati nelle applicazioni IoT, ovvero i protocolli MQTT e OPC-UA



Il sistema è stato testato simulando la pubblicazioni di messaggi



Grazie per
l'attenzione