

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di Laurea Magistrale in Ingegneria Informatica

**Sviluppo e Validazione di Applicazioni Edge
su Linee di Produzione Industria 4.0**

TESI DI LAUREA

in

Mobile Systems M

CANDIDATO

Luca Lambertini

RELATORE

Chiar.mo Prof.Ing. Paolo Bellavista

CORRELATORI

Prof.Ing. Luca Foschini

Dott. Riccardo Venanzi

Anno accademico 2020-2021

Indice

1	Edge Computing in ambito Industry 4.0	5
1.1	Internet of Things	5
1.1.1	Architettura IoT	6
1.2	Edge Computing	8
1.2.1	Che cos'è l'edge computing?	8
1.2.2	Vantaggi dell'edge computing	10
1.2.3	Dispositivi di edge computing	11
1.2.4	Differenza tra Fog Computing ed Edge Computing	13
1.3	Industria 4.0	13
1.3.1	Differenza tra OT e IT	15
1.4	Container VS Virtual Machine	16
1.4.1	Che cos'è un container?	16
1.4.2	I vantaggi della contenierizzazione	20
2	Tecnologie utilizzate	22
2.1	Siemens Industrial Edge	22
2.1.1	Hub-Management-Device	25
2.1.2	Applicazioni Siemens	27
2.2	Industrial Edge App Publisher	30
2.3	FOCAS Library	34
2.4	Python e modulo ctypes	35
3	Architettura progetto	38
3.1	Progetto DEEPMON	38
3.2	Poggipolini Use Case	39
3.3	Connettore FANUC	44
3.3.1	Mapping Data Model	49
3.3.2	MQTT Client	51
3.3.3	Configuration file	52
3.4	Flow	52
3.5	Web Storage Service	54
3.6	Sistema di Visualizzazione	55
4	Validazione del sistema	57
4.1	Valutazione Siemens Industrial Edge	57
4.2	Test Connettore FANUC	58
4.3	Efficacia Data model	59
4.4	Funzionamento del sistema	60
5	Conclusione e sviluppi futuri	62

Introduzione

L'industria odierna è composta principalmente da sistemi di automazione che svolgono operazioni ripetitive e di grande precisione. Le aziende sono sempre alla continua ricerca di modi e tecnologie che permettano l'aumento dell'efficienza e l'organizzazione del processo produttivo. I sistemi tradizionali portano con loro molte limitazioni e l'introduzione di paradigmi informatici come IoT, Edge e Cloud Computing possono spingere la produttività a livelli più elevati. L'integrazione di queste tecnologie all'interno di linee di produzione, permette alle imprese di portare nel proprio stabilimento processi informatici e funzionalità che prima non erano attuabili. La raccolta e l'analisi di grandi quantità di dati, l'interazione intelligente tra macchine autonome, la digitalizzazione di componenti fisiche per svolgere simulazione. Tutte queste possibilità sono il risultato dell'evoluzione del processo industriale in quella che viene chiamata Industria 4.0. Facendo utilizzo della mole di dati prodotti dai sistemi di automazione è possibile adottare soluzione hardware e software innovative per rendere ancora più efficiente la produzione.

Questi saranno gli argomenti trattati dal seguente elaborato, in particolare concentrandosi sull'applicazione di tecniche di Edge Computing su una linea di produzione messa a disposizione dall'azienda Poggipolini. Poggipolini è un'azienda italiana specializzata in progettazione e lavorazioni meccaniche ad altissima precisione, in soluzioni tecniche per la riduzione del peso, in componenti critici e sistemi di fissaggio strutturali per i settori aerospace, automotive e motorsports.

Il lavoro effettuato presso Poggipolini si colloca all'interno del progetto DEEPMON (Dynamic Edge computing for Plant MONitoring), promosso dal consorzio Bi-Rex e cofinanziato dal Ministero dello Sviluppo Economico. DEEPMON mira alla realizzazione di una soluzione Edge scalabile per la raccolta dati da impianti industriali e più in generale cerca di definire un modello comune di riferimento per tutte quelle imprese che vogliono adottare l'Edge Computing.

Al progetto partecipano diversi casi industriali, come quello di Poggipolini,

che vengono utilizzati per sperimentare la validità delle soluzioni proposte da DEEMON. Come figura di supporto e assistenza tecnica, al progetto partecipa anche Siemens, azienda leader per quanto riguarda la fornitura e sviluppo di tecnologie IT. Siemens fornisce per il progetto l'hardware e software necessari al corretto svolgimento dell'attività, in particolare mettendo a disposizione la sua piattaforma Industrial Edge. Il software predispone un'architettura per la gestione di tutti dispositivi Edge dell'impianto industriale permettendo un'orchestrazione centralizzata di tutte le funzionalità della linea produttiva. Per il caso d'uso di questo elaborato verrà utilizzato Siemens Industrial Edge per sviluppare ed installare applicazioni custom e non, direttamente sui nodi Edge a bordo macchina. Le applicazioni eseguiranno operazioni come estrazione del dato, persistenza delle informazioni e visualizzazione grafica. Il sistema di automazione usato sarà una fresatrice connessa ad un computer a controllo numerico (CNC) sui cui verranno effettuate letture dati. Scopo dell'attività è quello di validare l'effettiva utilità di applicazioni Edge in impianti industriali e di esplorare la piattaforma Siemens Industrial Edge in tutta la sua potenzialità.

Capitolo 1

Edge Computing in ambito Industry 4.0

1.1 Internet of Things

L'avvento di Internet ha avuto un riscontro non indifferente sul progresso della nostra civiltà. Sin dalla sua nascita ha sempre continuato a crescere ed evolversi fino a diventare parte integrante della vita di tutti i giorni, diventando concetto oramai conosciuto e assimilato da tutti quanti.

Fin dagli albori di Internet, si è sempre pensato alla possibilità di connettere tra loro cose ed oggetti concreti, di trovare un modo per estendere Internet ad un qualcosa che prima non ne aveva mai fatto parte. Questo concetto è evoluto negli ultimi anni prendendo l'espressione di Internet of Things. L'Internet of Things (IoT), tradotto in "Internet delle cose" ma più propriamente detto "Internet degli oggetti", è un termine che fa riferito all'estensione di Internet ad oggetti fisicamente reali e a luoghi concreti. IoT è una tematica molto ampia che merita di essere esplorata. Lo sviluppo di un ambiente IoT porterebbe molti benefici in diversi settori, soprattutto in quello industriale, luogo in cui vi è un forte utilizzo di sensori. Le applicazioni che sfruttano l'IoT sono molteplici e non si limitano ai soli oggetti ma anche ad interi edifici o luoghi come le *Smart Home* e le *Smart City*. Queste idee e progetti sono resi possibili dall'introduzione di un contesto IoT. Ciò a cui si ambisce è l'interconnessione di più "oggetti intelligenti" (definiti smart object) permettendo loro la condivisione dati per aumentare l'efficienza delle proprie funzionalità. L'idea sarebbe quella di poter connettere, tramite un tasto on/off, un qualsiasi dispositivo a questa grande rete. Dispositivo che utilizzando i sensori in suo possesso può raccogliere dati sull'ambiente che lo circonda. Dati che poi trasmessi ed elaborati secondo un certo criterio possono essere utilizzati per decidere se effettuare determinate azioni o meno. La connettività concessa dalla rete porta quindi all'evoluzione da semplici sensori a IoT device in grado di aumentare in maniera consistente l'efficienza dell'ambiente in cui essi

vengono impiegati.

L'applicazione delle pratiche IoT prevede la raccolta e l'archiviazione di una grossa mole di dati. Nello specifico per funzionare correttamente è importante processare raccogliere e analizzare grandi volumi di dati real time, sia in azienda per migliorare sicurezza e produttività, sia in qualsiasi ambito e per qualsiasi tipo di oggetto connesso. Per questo c'è bisogno di sistemi integrati tra Big Data, database NoSQL e dati IoT.

Riassumendo un po' tutto quello che è stato detto fino a questo momento si potrebbero delineare 4 azioni principali all'interno di un contesto IoT.

- **Acquisire i dati:**

Attraverso i sensori, i dispositivi IoT catturano i dati dall'ambiente in cui sono installati. Potrebbe trattarsi di una semplice rilevazione della temperatura o di una complessa trasmissione video in tempo reale.

- **Condividere i dati:**

Sfruttando le connessioni di rete disponibili, i dispositivi IoT rendono questi dati accessibili tramite un cloud pubblico o privato, secondo le istruzioni.

- **Elaborare i dati:**

A questo punto il software viene programmato per svolgere un compito in base a tali dati, ad es. azionare una ventola o trasmettere un avvertimento.

- **Agire sulla base dei dati:**

I dati accumulati da tutti i dispositivi inclusi in una rete IoT vengono analizzati e trasformati in potenti insight su cui poggiare azioni e decisioni aziendali sicure.

1.1.1 Architettura IoT

A livello di architettura di rete vanno distinte le applicazioni radio "short range"(Personal Area Network,LAN) da quelle "long range"(WAN). Nel caso delle applicazioni "short range" la portata di trasmissione dei dispositivi è ridotta, quindi è necessario l'ausilio di un gateway che faccia da intermediario tra il dispositivo e la piattaforma IoT posta nel Cloud. Mentre nelle applicazioni "long range" i dispositivi sono connessi direttamente con la piattaforma IoT.

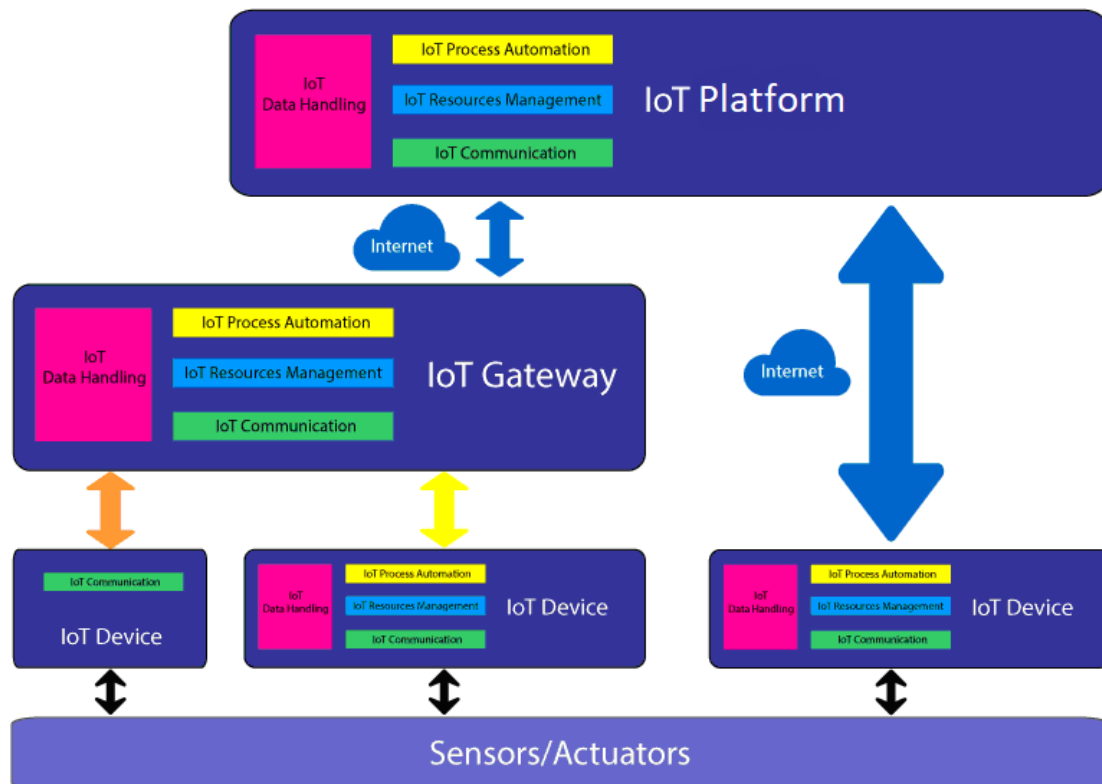


Figura 1: Architettura Internet of Things

Prendendo in esame il caso più comune delle applicazioni “short range” si possono delineare tre diverse entità:

- Una **prima entità** è composta dall’insieme di dispositivi IoT connessi tra di loro. Dispositivi che raccolgono continuamente dati utili per descrivere ciò che li circonda. La singola unità funzionale o smart object è chiamata **IoT device**.
- I dati raccolti dai dispositivi vengono poi condivisi e passati ad una **seconda entità** chiamata IoT Gateway, il cui compito è quello di fare da intermediario tra device-network oppure anche tra device-device.
- L’IoT Gateway infine trasmette i dati ad una **terza entità**, ovvero la piattaforma IoT in grado di analizzare e valutare i dati ricevuti e di decidere se compiere o meno eventuali operazioni.

Definite le entità ed assegnate a loro i vari ruoli all'interno di IoT, vanno specificate le regole con cui avvengono le comunicazioni.

1.2 Edge Computing

La crescita esponenziale dei dati acquisiti con miliardi di dispositivi IoT e mobili sta favorendo la transizione dall'invio di dati al cloud per l'elaborazione e lo storage a un modello distribuito in cui una parte dell'elaborazione avviene nell'edge della rete, più vicino a dove i dati vengono creati.

1.2.1 Che cos'è l'edge computing?

L'edge computing si riferisce all'elaborazione, all'analisi e all'archiviazione dei dati più vicino a dove vengono generati, per consentire risposte rapide o addirittura in tempo reale. Negli ultimi anni, alcune aziende hanno consolidato le loro operazioni centralizzando lo storage e l'elaborazione dei dati nel cloud. Tuttavia, le esigenze delle nuove applicazioni pratiche basate su miliardi di dispositivi distribuiti, dalle soluzioni avanzate di gestione del magazzino e dell'inventario, alle linee di produzione robotiche a visione aumentata, fino ai sistemi avanzati di controllo del traffico delle smart city, hanno reso questo modello insostenibile.

Inoltre, il maggiore utilizzo dei dispositivi edge (dai dispositivi IoT, come videocamere intelligenti, chioschi di vendita mobili, sensori medici e PC industriali, fino ai gateway e l'infrastruttura di computing) per ottenere informazioni fruibili più veloci e quasi in tempo reale alla fonte dei dati sta favorendo la crescita esponenziale della quantità di dati generati e raccolti.

Un'architettura di Edge Computing è, di fatto un'architettura distribuita di micro data center, ciascuno in grado di immagazzinare ed elaborare i dati a livello locale e trasmettere questi dati ad un data center centralizzato o a un repository di storage in cloud. I data center centralizzati hanno da tempo reso evidenti i propri svantaggi essendo costosi e poco agili. A questi limiti si è posto rimedio, negli ultimi anni, con il cloud. Spostare i carichi di lavoro nella nuvola ha permesso di ridurre sensibilmente i costi del data center, questo vale a maggior ragione se si utilizza il cloud pubblico e si ha, quindi, la possibilità di accedere a risorse storage e potenza di calcolo virtualmente illimitate e disponibili on demand. Non tutte le applicazioni, però, sono portabili nel cloud. Esistono particolari situazioni che, anzi, accentuano i limiti tipici delle infrastrutture di questo tipo. Uno dei più rilevanti è legato all'impossibilità di stabilire connessioni sempre affidabili, in grado di sostenere, per

esempio, i flussi continui di dati provenienti da sensori e oggetti connessi in ottica IoT. Un altro è quello legato alla latenza, quindi al tempo necessario per trasmettere un pacchetto dati attraverso la rete pubblica che il cloud sfrutta.

Il cloud computing è stato spinto ai suoi limiti dalle esigenze dei servizi e delle applicazioni che supporta, dallo storage e l'elaborazione dei dati alla reattività di sistema. In molti casi, la larghezza di banda o la potenza di calcolo non sono sufficienti per garantire i requisiti necessari per elaborare più rapidamente i dati da dispositivi connessi e generare informazioni e azioni immediate, quasi in tempo reale. Queste discrepanze stanno favorendo l'adozione e l'utilizzo dell'edge computing. I fattori principali che determinano le sfide del cloud sono:

- **Latenza:**

Più settori stanno implementando applicazioni che richiedono analisi e risposta rapida. Il cloud computing da solo non è in grado di soddisfare queste esigenze a causa della latenza dovuta dalla distanza di rete dalla fonte dei dati, con conseguenti inefficienze, ritardi e customer experience scadenti.

- **Larghezza di banda:**

Aumentando larghezza di banda di trasmissione o la potenza di elaborazione si potrebbero superare i problemi di latenza. Tuttavia, poiché le aziende continuano ad aumentare il numero di dispositivi edge nella loro rete e la quantità di dati che generano, l'invio dei dati al cloud può raggiungere costi insostenibili, che però potrebbero essere abbattuti se i dati potessero essere elaborati, archiviati e analizzati nell'edge.

- **Sicurezza e privacy:**

La protezione nell'edge dei dati sensibili, come la documentazione medica privata e la trasmissione di meno dati via Internet potrebbero contribuire a migliorare la sicurezza riducendo il rischio di intercettazione. Inoltre, alcune amministrazioni o clienti potrebbero richiedere che i dati rimangano nella giurisdizione in cui sono stati creati. In campo sanitario, ad esempio, potrebbero esserci anche requisiti locali o regionali a limitare l'archiviazione o la trasmissione di dati personali.

- **Connettività:**

L'assenza di connettività Internet costante può compromettere il cloud

computing, ma una varietà di opzioni di connettività di rete rende possibile il computing edge-to-cloud. Ad esempio, il 5G offre una connessione con ampia larghezza di banda e bassa latenza, consentendo un trasferimento rapido dei dati e l'erogazione dei servizi direttamente dall'edge.

- **Intelligenza Artificiale:**

Con la necessità di informazioni fruibili quasi in tempo reale, le aziende hanno bisogno dell'IA alla fonte dei dati per consentire un'elaborazione più veloce e sfruttare il potenziale dei dati precedentemente inesperto.

1.2.2 Vantaggi dell'edge computing

Spostando alcune funzioni, come storage, elaborazione e analisi, dal cloud all'edge, più vicino a dove i dati vengono generati, si possono ottenere diversi vantaggi essenziali:

- **Maggiore velocità e una latenza inferiore:**

Spostando l'elaborazione e l'analisi dei dati nell'edge si contribuisce a velocizzare la risposta del sistema, consentendo transazioni più rapide ed esperienze migliori che potrebbero essere fondamentali nelle applicazioni quasi in tempo reale, come la guida autonoma.

- **Migliore gestione del traffico di rete:**

Riducendo al minimo la quantità di dati inviati tramite la rete al cloud si possono ridurre larghezza di banda e costi di trasmissione e archiviazione di grandi volumi di dati.

- **Maggiore affidabilità:**

La quantità di dati che le reti possono trasmettere simultaneamente è limitata. Per gli ambienti con connettività Internet di qualità inferiore alla media, la capacità di archiviare ed elaborare i dati nell'edge aumenta anche l'affidabilità in caso di interruzione della connessione con il cloud.

- **Maggiore sicurezza:**

Con una corretta implementazione, una soluzione di edge computing può contribuire ad aumentare la sicurezza dei dati limitando la trasmissione dei dati tramite Internet.

L'edge computing permette tutta una serie di operazioni che fanno riferimento ad analisi che avvengono in locale, alla periferia della rete aziendale,

nella fabbrica per esempio, ovvero laddove i dati stessi si generano e vengono raccolti. L'esempio tipico è quello delle applicazioni di Industria 4.0 in cui sensori e dispositivi generano, spesso sotto forma di flussi costanti e continui, enormi quantità di record impossibili da rielaborare in efficienza nel cloud. L'Edge Computing permette di prioritizzare l'analisi dati in loco, contribuendo a ridurre il traffico di rete verso i data center aziendali. Questo avviene, generalmente, attraverso il trasferimento dei record prodotti dai dispositivi connessi a un device installato localmente, dal fattore forma particolarmente compatto, che al suo interno include risorse computazionali, storage e connettività. Alcuni dati, quelli utili per generare allarmi o suggerire interventi istantanei, sono quindi processati localmente. La quota residuale è trasferita allo storage ospitato in cloud o al data center centrale per il backup o le analisi più complesse.

Facendo riferimento ad un possibile scenario in ambito Industria 4.0, nelle aziende manifatturiere, laddove un sensore nell'impianto identifichi il possibile guasto di una componente specifico, gli algoritmi analitici che interpretano il dato a livello di edge, possono imporre il fermo macchina e l'invio degli alert al personale di manutenzione, riducendo il rischio di guasti e fermi prolungati degli impianti.

Più in generale, ci sono tutta una serie di benefici che possono essere assicurati dall'Edge computing: l'edge, innanzitutto, può aiutare gli utenti ad accedere ai dati di cui hanno bisogno rapidamente, con meno latenza e senza utilizzare gran parte della preziosa larghezza di banda. Non meno importante è l'aspetto sicurezza: l'archiviazione dei dati presenta dei rischi, ma i sistemi cloud sono più difficili da controllare per le aziende, poiché si trovano in data center remoti, gestiti tramite un sistema centralizzato. Al contrario l'edge computing può permettere di fornire l'analisi dei dati in tempo quasi reale e, poiché a differenza del cloud si basa su una rete distribuita di infrastrutture (tra cui spiccano i micro data center), è per certi versi più "al riparo" da eventuali disservizi. Che difficilmente interesseranno contemporaneamente tutti i nodi della rete.

1.2.3 Dispositivi di edge computing

Un dispositivo edge potrebbe essere visto come un punto di ingresso nelle reti principali di un'impresa o come un qualsiasi hardware che controlla il flusso di dati tra due reti. Questa definizione mette in evidenza le funzioni più importanti dei dispositivi di edge computing che servono come condotti

per l'elaborazione dei dati e la loro capacità di attraversare i data center esterni quando se ne presenta la necessità.

I dispositivi di edge computing sono progettati per svolgere ruoli specifici e sono dotati delle applicazioni di cui hanno bisogno per svolgere determinati compiti. Un esempio è l'uso di sensori per monitorare la temperatura delle macchine. In questo scenario, un sensore che può stare nel palmo della mano, raccoglie i dati di temperatura dalla macchina e li trasferisce ad un data center o ad una piattaforma IoT.

I dispositivi edge sono utilizzati dalle imprese in diverse maniere. I loro impieghi nei processi industriali sono ben noti, ma le loro applicazioni si estendono al di fuori della produzione e della fornitura di servizi. Un esempio di questo è l'uso di dispositivi di edge computing nelle iniziative di soccorso. In questo caso, un robot di soccorso funge da dispositivo di edge computing e può essere incaricato di muoversi autonomamente e di raccogliere dati visivi in aree difficili da raggiungere. I dati raccolti possono poi essere trasferiti su uno schermo visivo più grande per aiutare il processo decisionale.

I dispositivi edge sono disponibili in diverse forme, dimensioni e capacità. Un router che collega le reti pubbliche a Internet è un esempio di dispositivo di edge computing. In altre situazioni, un firewall può servire come dispositivo edge; in questo caso il firewall determina ciò che accede a una rete e quindi serve come punto d'ingresso in quella rete. Esistono anche dispositivi di edge computing specializzati, che sono la categoria in cui rientrano l'Internet of Things (IoT), l'Industrial IoT, i robot e i dispositivi smart. Questi dispositivi edge sono in grado di realizzare un'ampia gamma di funzioni. Anche macchine o apparecchiature più grandi possono fungere da dispositivi di edge computing. Un veicolo autonomo che fornisce l'accesso a una piattaforma cloud attraverso la tecnologia digital twin è anche un esempio di dispositivo edge.

I dispositivi edge sono l'hardware che guida l'applicazione dell'edge computing nei diversi settori industriali. Vengono utilizzati per svolgere compiti diversi a seconda delle applicazioni software o delle caratteristiche di cui sono dotati. Nell'industria manifatturiera, i dispositivi edge svolgono l'importante ruolo di portare l'elaborazione a bassa latenza in officina. Ciò rende possibile l'ottimizzazione della produzione attraverso politiche guidate dai dati.

Un esempio è l'uso di edge computing per raccogliere i dati macchina da apparecchiature legacy. In questo caso, i dispositivi edge sono collegati alle apparecchiature e i dati acquisiti possono essere trasferiti e utilizzati per i calcoli dell'efficienza complessiva delle apparecchiature (OEE).

I calcoli OEE(Overall Equipment Effectiveness) consentono di integrare un modello di business di ottimizzazione delle prestazioni dell'impianto guidato dai dati all'interno di strutture brownfield. L'accuratezza che i dispositivi di edge computing portano alla raccolta dei dati garantisce che il calcolo dell'OEE sia più accurato rispetto ai processi di raccolta manuale dei dati.

L'edge computing semplifica anche l'analisi dei dati all'interno delle officine di produzione. Nei casi in cui un'officina contiene centinaia di macchine e dispositivi IoT, i dispositivi edge raccolgono, elaborano e memorizzano i dati attraverso un sistema decentralizzato. Questo assicura che l'elaborazione avvenga in tempo reale senza dover sempre accedere al cloud o a data center esterni.

La natura decentralizzata dell'edge computing significa anche che la sicurezza è locale per ogni dispositivo edge. Questo serve quindi da scudo contro i cyberattacchi. In situazioni in cui si verificano violazioni di successo, le informazioni che si trovano all'interno della rete edge violata vengono influenzate ma non trasferite ad altre reti all'interno dell'ecosistema di un'impresa.

1.2.4 Differenza tra Fog Computing ed Edge Computing

1.3 Industria 4.0

Il concetto di Industria 4.0 nasce da un continuo evolversi del settore industriale, evoluzione che scaturisce dalle molteplici introduzione di nuove tecnologie innovative che permettono l'aumento dell'efficienza in ambito produttivo e organizzativo. Riguardando ai diversi cambiamenti che hanno trasformato il concetto ed il significato di industria possiamo riassumerli brevemente in questo modo:

- L'**Industria 1.0** corrisponde a una rivoluzione della manifattura rispetto all'uso dell'energia: l'invenzione della macchina a vapore consente alle fabbriche di introdurre una meccanizzazione della produzione all'insegna di una maggiore velocità e potenza.

- L'**Industria 2.0** rappresenta la seconda generazione energetica, legata all'utilizzo dell'elettricità prima e del petrolio poi, che permettono di incrementare ulteriormente i livelli di meccanizzazione e di produzione. È grazie a questa rinnovata potenza che nella manifattura si afferma progressivamente quella catena di montaggio che inaugura l'era della produzione di massa.
- L'**Industria 3.0** riassume l'ingresso in fabbrica dell'ICT (Information and Communication Technologies): informatica ed elettronica incrementano ulteriormente i livelli di automazione non solo nell'ambito produttivo ma anche organizzativo. Si diversificano le infrastrutture e si avviano nuovi processi che, all'insegna della progressiva digitalizzazione, diversificano e agevolano il lavoro delle persone migliorando la qualità della produzione.

Infine si giunge all'Industria 4.0 che sfruttando una diversa gamma di tecnologie come robotica, sensoristica, connessione e programmazione, rappresenta una nuova rivoluzione rispetto al modo di fabbricare i prodotti e di organizzare il lavoro. Tutto questo grazie a nuovi modelli di produzione sempre più automatizzati e interconnessi, asset e prodotti intelligenti comunicanti, una tracciabilità e una rintracciabilità dei processi tale da portare a una gestione delle informazioni collettiva a livello di filiera, nuove logiche di servizio all'insegna del cloud e della mobility. Il tutto incentrato su una Internet di ultima generazione (Industrial Internet), capace di portare dentro e fuori alle fabbriche più informazione, più integrazione, più interazione e più efficienza, rinnovando i processi e i sistemi ma anche portando nuove regole di comunicazione e di servizio. Software di nuova generazione da un lato e Big Data Management dall'altro, è così che la produzione riesce ad arrivare a una personalizzazione di massa. La continua evoluzione delle tecnologie sta diversificando la declinazione del 4.0 su più livelli e ambiti operativi associati all'uso dell'Intelligenza Artificiale e di tutte le derive del digitale.

L'Industria 4.0 è un processo che scaturisce dalla quarta rivoluzione industriale e che sta portando alla produzione industriale del tutto automatizzata e interconnessa. Le nuove tecnologie digitali avranno un impatto profondo nell'ambito di quattro direttrici di sviluppo: la prima riguarda l'utilizzo dei dati, la potenza di calcolo e la connettività, e si declina in big data, open data, Internet of Things, machine-to-machine e cloud computing per la cen-

tralizzazione delle informazioni e la loro conservazione. La seconda è quella degli analytics: una volta raccolti i dati, bisogna elaborarli per ricavare valore. Tali dati possono essere impiegati utilizzando tecniche di machine learning permettendo alle macchine di perfezionare la loro resa nel tempo. La terza direttrice di sviluppo è l'interazione tra uomo e macchina, che coinvolge le interfacce “touch”, sempre più diffuse, e la realtà aumentata. Infine c'è tutto il settore che si occupa del passaggio dal digitale al “reale” e che comprende la manifattura additiva, la stampa 3D, la robotica, le comunicazioni, le interazioni machine-to-machine e le nuove tecnologie per immagazzinare e utilizzare l'energia in modo mirato, razionalizzando i costi e ottimizzando le prestazioni.

1.3.1 Differenza tra OT e IT

La tecnologia operativa si riferisce alle tecnologie hardware e software utilizzate nella gestione o nel controllo di dispositivi fisici, attrezzature e sistemi. L'OT (Operational Technology) ha una componente di processo di controllo fisico che la rende un concetto progettato per il settore industriale in cui la “fisicità” domina. La linea di produzione industriale è nota come un ambiente fisico e duro; le attrezzature pesanti devono essere controllate in linea con un piano di produzione principale. L'OT fornisce gli strumenti necessari per controllare fisicamente o automatizzare le operazioni di questi macchinari pesanti.

In termini di caratteristiche, l'OT è progettato per tecnologie che includono il vecchio controllore logico programmabile e i più attuali sistemi di controllo numerico del computer (CNC). Le soluzioni di visualizzazione collegate a OT variano da semplici schermi LED digitali a HMI (Human-Machine Interface) basati sul web che sono dotati di risorse di elaborazione e software per visualizzare informazioni contestuali agli operatori nella linea di produzione.

L'OT è usato come strumento di controllo industriale o di automazione e viene applicato per controllare nastri trasportatori, macchine CNC, linee di assemblaggio automatizzate e sistemi di allarme. Tradizionalmente, i fornitori che forniscono soluzioni OT si basano su hardware e software proprietari per assicurarsi di avere il controllo esclusivo sui processi di risoluzione dei problemi e di manutenzione.

Nell'era dell'informazione, la tecnologia dell'informazione domina ed è per

questo che è considerata di maggior rilievo dell'OT. L'IT (Information Technology) si riferisce all'hardware e al software usati nella gestione dei dati. Questo include la memorizzazione, il recupero, l'analisi e la manipolazione dei dati. L'affidamento sulla tecnologia dell'informazione in ogni sfera della vita ha portato allo sviluppo di soluzioni IT avanzate che sono intuitive e facili da usare quando si gestisce qualsiasi forma di dati.

Mentre l'IT si basa su hardware e software per trattare i dati, l'OT si basa su entrambi per controllare le entità fisiche. Ma con l'introduzione di Industria 4.0, è necessario un nuovo quadro che giustapponga entrambi i sistemi tecnologici. Industria 4.0 è guidata dai dati e dall'interconnettività tra gli asset industriali. Ci si aspetta che l'IT gestisca il processo di analisi dei dati, mentre l'OT dovrebbe evolversi per gestire l'interconnettività e fare affidamento sull'IT per supportare la gestione dei dati senza soluzione di continuità. L'interconnessione tra IT e OT è un aspetto del processo di digitalizzazione che è in corso da qualche anno in alcuni settori industriali. Le aziende che hanno già la connessione online delle proprie macchine hanno oramai capito che, per ottimizzare i processi aziendali critici, i due mondi devono dialogare stabilmente ed in maniera sicura, sia tra loro che con l'esterno. Non è sufficiente avere i dati di funzionamento delle macchine in tempo reale (OT), bisogna essere in grado di trasformarle in informazioni correlandole con altri dati classicamente non gestiti dalla produzione (IT).

1.4 Container VS Virtual Machine

1.4.1 Che cos'è un container?

I container rappresentano degli spazi utente isolati in esecuzione sul sistema operativo di un server o macchina host. Si tratta quindi di funzionalità del sistema operativo Linux, che permettono di virtualizzare alcune risorse di sistema per eseguire diverse applicazioni in ambienti operativi eterogenei, all'interno di uno stesso server fisico, invece di ricreare virtualmente tutta la macchina, ovvero risorse processore, risorse storage, risorse di rete e sistema operativo, come avviene invece quando si utilizza un approccio basato sulle virtual machine. Viene quindi astratto il solo ambiente di esecuzione delle applicazioni con i suoi settaggi fondamentali. I container dunque, eseguono le istruzioni dell'applicazione direttamente sulla CPU, senza richiedere alcun meccanismo di emulazione (come, invece, avviene per le macchine virtuali), permettendo di risparmiare risorse e attuando una virtualizzazione a livel-

lo del solo sistema operativo. Similmente alle macchine virtuali, i container forniscono uno spazio isolato e separato per eseguire le applicazioni, pur sfruttando l'hardware condiviso.

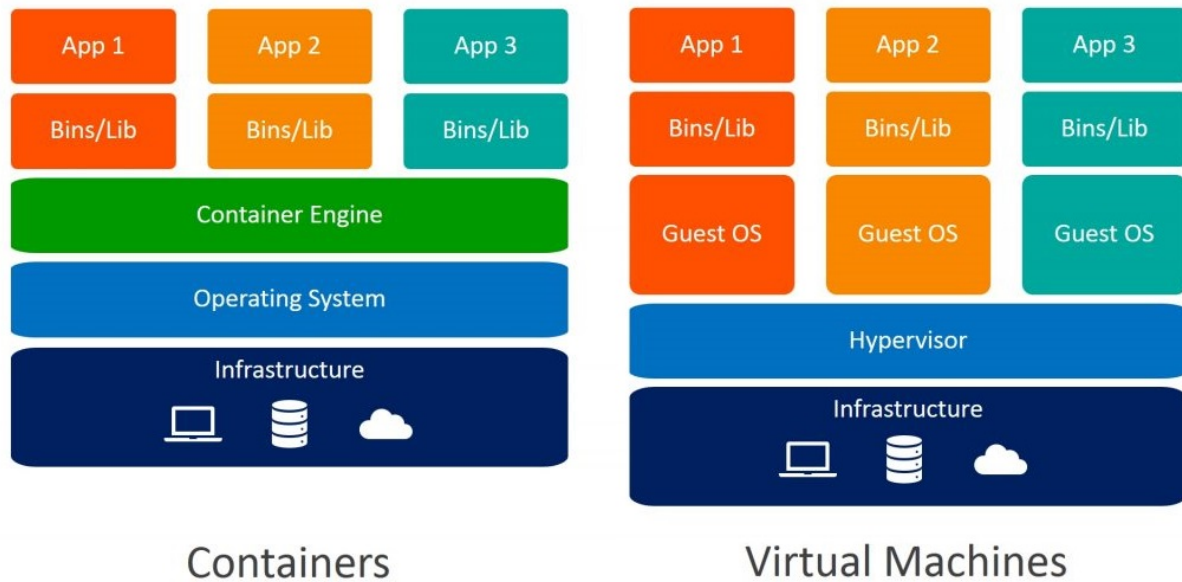


Figura 2: Virtual Machines e Containers

Il principale fattore di differenziazione è legato al ridotto consumo di risorse. Infatti non dovendo inglobare tutte le risorse di un server, i container sono più “leggeri” delle macchine virtuali e possono essere avviati in tempi molto ridotti. Questo li rende particolarmente adatti alle situazioni in cui il carico di lavoro da sostenere è variabile nel tempo e con picchi difficilmente prevedibili. I container quindi limitano il loro livello di astrazione al solo sistema operativo. C’è da aggiungere inoltre che ogni utente oltre a condividere lo stesso sistema operativo, condivide anche il kernel dello stesso, la connessione di rete e i file di base del sistema. Quindi le istanze vengono eseguite all’interno di uno spazio separato, garantendo così una notevole diminuzione di consumo della CPU e dell’overload associato, che risulta essere un fenomeno tipico dell’esecuzione di più sistemi operativi nelle macchine virtuali. Come per il caso delle macchine virtuali che hanno due tipologie di hypervisor, anche i metodi di gestione dei container si differenziano in due tipologie: Full system Container e Application Container.

- **Full System Container**

Full System Container o più comunemente Operative System Container,

condivide il kernel della macchina host, fornendo un isolamento di tipo user namespace. Questo metodo permette alla CPU di partizionare la memoria da allocare in diversi livelli di isolamento. Gli Operative System Container possono essere facilmente comparati agli Hypervisor o alle Virtual Machine in genere. Le applicazioni e le loro librerie possono essere installate allo stesso modo in come avviene in un qualunque altro sistema operativo installato su macchina virtuale. Utilizzando i Full System Container, è facile assegnare un indirizzo IP di rete statico o remoto, utilizzare diversi dispositivi di rete, o eseguire comandi di edit sui file in `/etc/hosts`.

- **Application Container**

Application Container allo stesso modo della precedente tipologia, condivide il kernel del sistema operativo della macchina host con il livello soprastante. Questa tipologia di Container è stata progettata allo scopo di poter avviare un singolo processo o applicazione all'interno di ogni container, garantendo un certo grado di isolamento, in modo tale che il processo che andrà in esecuzione all'interno di un container, abbia un proprio filesystem privato e non abbia nessuna visibilità di alcun altro processo che si trova in esecuzione sullo stesso server o macchina host. Questo tipo di isolamento viene ottenuto grazie ai namespaces. Con essi è infatti possibile gestire e isolare le risorse linux quali: Inter Process Communications (IPC), configurazione di rete, punto di mount della root, l'albero dei processi, gli utenti e i gruppi di utenti, nonché la risoluzione del nome di rete. Quindi il vantaggio principale sta nel fatto che con l'utilizzo dei namespaces, diventa possibile isolare i processi in modo efficace, però come detto in precedenza, ogni processo isolato nel container, condivide con il sistema operativo della macchina host, il kernel dello stesso. Al fine di gestire l'accesso alle risorse hardware, i container utilizzano un ulteriore modulo Kernel chiamato Control Groups, che ha il compito di impostare le priorità e di misurare diversi tipi di risorse tra cui: la memoria, l'utilizzo della CPU e gli accessi al disco. Un esempio di Application Container sono: Docker e Rocket.

Quando si parla di container, di solito ci si riferisce ai container Linux. I container Linux sono costituiti da gruppi di controllo (cgroups), che consentono al kernel Linux di limitare e isolare risorse fisiche come CPU, memoria e I/O di rete a un gruppo di processi. Questo è simile al modo in cui un hypervisor

crea risorse per un sistema operativo guest di una macchina virtuale, ma con i cgroup Linux è possibile dividere le risorse direttamente all'interno del sistema operativo Linux, indipendentemente dalla presenza di un hypervisor. Con i container, esegui le tue applicazioni con le proprie risorse separate senza dover emulare un intero sistema operativo guest.

Sia i container che le macchine virtuali semplificano l'esecuzione di più applicazioni in produzione. I container vengono eseguiti direttamente sul sistema operativo host tramite un motore di container e ogni container può avere i propri file dell'applicazione, binari e librerie. I container possono essere avviati e scalati in pochi secondi perché non necessitano di un proprio sistema operativo completo. Quando si considera il costo dei container rispetto alle VM nel cloud computing, è importante riconoscere che si sta realmente confrontando il costo dell'esecuzione dei container sulle VM con l'esecuzione delle sole VM. Molti provider di servizi cloud dipendono dalla tecnologia delle VM per fornire agli utenti i loro server personali, quindi se esegui container nel cloud, molto probabilmente eseguirai container su VM di cui è stato effettuato il provisioning. Nel cloud computing, il fattore chiave per ridurre i costi è sfruttare la tecnologia dei container migliorando la densità di distribuzione e rendendo le applicazioni più leggere e portatili in modo che più applicazioni possano essere eseguite su una singola VM, anziché utilizzare più VM.

- **Deployment semplificato:**

La tecnologia dei container permette di semplificare il deployment di qualsiasi applicazione poiché essa viene “impacchettata” in un singolo componente distribuibile e configurabile con una sola linea di comando, senza doversi preoccupare della configurazione dell'ambiente di esecuzione.

- **Disponibilità rapida:**

Virtualizzando solamente il sistema operativo e le componenti necessarie all'esecuzione dell'applicazione, invece che l'intera macchina, i tempi di avvio si riducono notevolmente rispetto a quelli di una VM.

- **Ampia portabilità:**

I container possono essere creati e replicati in qualsiasi ambiente in modo semplice e veloce. Si tratta di un grande vantaggio a livello del ciclo di vita di un software per esempio, poiché i container possono essere copiati molto velocemente per creare gli ambienti di sviluppo, test e produzione senza la necessità di operare configurazioni.

- **Controllo più granulare:**

I container possono “pacchettizzare” un’intera applicazione ma anche solo un singolo componente. In questo modo consentono agli sviluppatori di suddividere ulteriormente le risorse in microservizi, garantendo così un maggiore controllo sull’eseguibilità delle applicazioni e un miglioramento delle performance dell’intera infrastruttura.

- **Leggerezza:**

Un grande punto di forza dei container è che sono molto più “leggeri” delle virtual machines perché non devono attivare un loro sistema operativo dedicato. Di conseguenza sono molto più veloci da attivare e disattivare e ideali per quegli ambienti in cui il carico di elaborazione varia sensibilmente e in maniera imprevedibile.

Ma come per ogni cosa vi sono pregi e difetti, anche i container hanno le loro debolezze. Una di queste è rappresentata dalla difficoltà di gestione quando il numero di container è molto elevato. Al contrario, le virtual machines sono più semplici da amministrare anche perché raramente raggiungono grandi quantità, come invece può accadere facilmente con l’utilizzo dei container. Un altro punto debole è proprio la condivisione dello stesso kernel del sistema operativo. A livello teorico, uno dei container potrebbe compromettere la stabilità del kernel, influenzando anche gli altri. Le VM possono eseguire molte più operazioni rispetto a un singolo container, ed è per questo che sono state a lungo impiegate per raggruppare i tradizionali carichi di lavoro monolitici. Tuttavia, questa funzionalità estesa e la dipendenza da sistema operativo, applicazione e librerie ne limita fortemente la portabilità rispetto ai container.

1.4.2 I vantaggi della contenierizzazione

Queste caratteristiche proprie della tecnologia dei container portano ad alcuni vantaggi indiscutibili:

- La possibilità per gli sviluppatori di possedere una miriade di container anche sul proprio PC o laptop, per avere sempre a portata di mano un ambiente di deploy o test adatto a ciascuna applicazione in sviluppo. Per quanto sia possibile eseguire anche su un laptop diverse virtual machines, questa operazione non è mai veloce e semplice e impatta non poco sulle prestazioni esecutive.

- L'amministrazione dei cicli di rilascio delle applicazioni è semplificato, in quanto distribuire una nuova versione di un container è pari al tempo speso per digitare in console una singola linea di comando.
- Le attività di testing traggono un beneficio economico da un ambiente contenierizzato. Se si effettua il test di un'app direttamente su un cloud server in un ambiente di cloud computing pubblico, sarà necessario sostenere i costi relativi alla frazione di ora (o all'ora intera) di occupazione delle risorse computazionali. Questo costo aumenta all'aumentare del numero di test che devono essere eseguiti. Con un container è possibile effettuare una serie di semplici test programmati giornalieri mantenendo costante il costo, in quanto si userebbero sempre le stesse risorse di calcolo.
- La componibilità dei sistemi applicativi, specialmente per le applicazioni open source. In pratica, invece di obbligare gli sviluppatori a installare e configurare i servizi come MySQL, MongoDB, nginx, node.js, per avere la giusta piattaforma esecutiva per le proprie applicazioni, sarebbe meno rischioso e più veloce avviare ed eseguire con piccoli script quei pochi container che ospitano queste stesse applicazioni.

Il vantaggio più importante che questa tecnologia può offrire è la portabilità e la consistenza di un formato che consente l'esecuzione applicativa su diversi host. Infatti, con la standardizzazione dei container, i workload possono essere facilmente spostati lì dove vengono eseguiti in modo più economico e veloce, evitando anche i lock-in dovuti alle peculiarità delle piattaforme dei singoli provider.

Capitolo 2

Tecnologie utilizzate

2.1 Siemens Industrial Edge

I sistemi di automazione sono generalmente progettati e sviluppati per un ciclo di vita molto esteso che supera la decina di anni. Una volta che macchinari di questa tipologia sono integrati all'interno della linea di produzione e sono costantemente in fase di esecuzione, le possibilità di poter mettere mano a tali macchinari per modificarne alcuni aspetti sono poche. In altre parole oggi esistono limitate possibilità di introdurre nuove funzionalità, come per esempio la connessione ad un sistema IT in maniera retroattiva. Siemens Industrial Edge si occupa per l'appunto di trovare una soluzione a questa problematica riscontrata da diverse aziende. Essendo una piattaforma di digitalizzazione, Siemens Industrial Edge disaccoppia le funzionalità di automazione da quelle digitali in modo da facilitare la definizione di sistemi più flessibili ed estensibili. Proprietà che vengono mantenute grazie alla possibilità di utilizzare sempre le tecnologie e software più all'avanguardia. Con Industrial Edge, si utilizza l'intelligenza e la scalabilità del Cloud direttamente nell'impianto industriale, in maniera semplice, con alte prestazioni e senza che i dati escano dall'ambito produttivo. Questo framework combina un'elaborazione dei dati locale a bassa latenza direttamente sul sistema di automazione con tutti i benefici che porta il Cloud: analisi App-based, elaborazione dati e concetto di Infrastructure-as-Service.

È una piattaforma tecnologica integrata per far crescere il mondo dell'automazione industriale, facendola convergere con le più moderne tecnologie di raccolta dati, elaborazione e programmazione colmando il divario tra sistemi IT e OT in azienda.

Siemens Industrial Edge è una piattaforma che estende le funzionalità di dispositivi di automazione, senza nessuna ripercussione, introducendo tutti i

vantaggi dell'Edge Computing in modo sicuro all'interno del processo produttivo. Siemens integra all'interno della sua strategia di digitalizzazione la possibilità da parte degli OEMs e end users di poter eseguire diversi tipi di applicazioni (diagnostica, analisi predittiva, ecc...). Definendo così un ecosistema, progettato per componenti di automazione, che integra al suo interno la connessione al Cloud e la possibilità di utilizzare applicazioni proprietarie di Siemens o sviluppate da terze parti. La flessibilità del sistema può essere fortemente incrementata attraverso le features offerte dalla piattaforma.

Fino ad adesso era possibile adattare soluzioni di automazione a nuove circostanze solo entro un certo limite e impiegando un certo costo. Inoltre una volta accettata la soluzione il sistema non veniva più toccato. Con Industrial Edge è possibile provvedere in modo continuo i dispositivi con le più recenti funzionalità in circolazione, tutto questo attraverso un Management System in modo da adattare il sistema alle diverse situazioni. Questo è ottenuto in particolare attraverso la separazione tra funzionalità Edge e quelle tradizionali dell'automazione. Funzioni di digitalizzazione possono essere eseguite sugli Edge device e, sfruttando un rapido ciclo di update, possono essere mantenute al passo con l'evoluzione tecnologica.

Con Industrial Edge, ogni operatore e tecnico digitale in azienda può collegarsi alla piattaforma e accedere ad una serie di applicazioni per l'automazione che utilizzano IoT, big data e algoritmi esclusivi. Gli impieghi operativi di queste applicazioni comprendono l'automazione della produzione, il controllo qualità finale, la manutenzione predittiva e l'anomaly detection per ogni azione. Per esempio, i sistemi IoT possono essere collegati a macchinari industriali, robot e veicoli a guida autonoma, che attraverso sensori e AI (artificial intelligence) possono essere comandati e guidati da remoto.

Siemens Industrial Edge è l'infrastruttura centralizzata che permette la gestione di tutti gli Edge Devices connessi ed installati nell'impianto industriale. L'Edge Management System concede la possibilità all'utilizzatore di:

- Installare Edge apps e software per il monitoraggio dello status di tutti i dispositivi connessi
- Gestire le diverse funzionalità sull'Edge device desiderato
- Importare e spostare funzionalità dal Cloud nel processo produttivo in esecuzione

Questa infrastruttura si concentra principalmente sulla connettività integrata e sullo scambio di dati tra il livello di automazione e il livello IT (tramite MQTT o MindSphere). Inoltre questa gestione centralizzata degli Edge devices e delle applicazioni che eseguono su di essi, garantisce l'utilizzo di algoritmi e tecniche di processing del dato, permettendo l'analisi dei soli dati rilevanti. L'interconnessione di tutti i dispositivi relativi al processo produttivo gioca un ruolo fondamentale in questo ecosistema introducendo nuove funzionalità e vantaggi:

- **Estensione delle funzionalità della macchina:** permessa dall'aggiunta di nuove ed avanzate funzionalità attraverso il Management System centralizzato
- **Migliore gestione remota:** facilitazione della gestione dei dispositivi connessi e velocizzazione del ciclo di innovazione per i software.
- **Accesso remoto ai dati macchina:** In qualsiasi luogo ed in qualsiasi momento grazie ad Edge Apps e alla connettività al Cloud sarà sempre possibile l'analisi di dati macchina per migliorare il design di quest'ultima o per implementare un nuovo business model.

In combinazione con i prodotti hardware e software esistenti, questa piattaforma aperta fornisce agli utenti una soluzione pronta all'uso per l'elaborazione dei dati a livello di produzione. Ciò è ottenuto tramite una gestione integrata dei dispositivi e dei cicli di vita delle app. La gestione centralizzata delle applicazioni minimizza o addirittura elimina completamente lo sforzo necessario alle attività di manutenzione e di aggiornamento delle applicazioni distribuite sui vari dispositivi installati. I dispositivi edge possono quindi essere monitorati centralmente con semplicità, diagnosticati e gestiti da amministratori IT e ingegneri di produzione. Questo significa che le nuove applicazioni software possono essere installate a livello aziendale in modo sicuro su tutti i dispositivi collegati. Con questa piattaforma di gestione pronta all'uso, gli amministratori IT possono distribuire le Edge Apps in modo altamente automatizzato e gestirle durante l'intero ciclo di vita.

Industrial Edge semplifica la raccolta dei dati macchina e permette la definizione di app basate su container, l'elaborazione dati basata su linguaggi di alto livello e la gestione centrale di software/dispositivi presenti sulla linea di produzione.

2.1.1 Hub-Management-Device

Siemens Industrial Edge come già detto è una piattaforma che si pone l'obiettivo di rendere più facile e agevole la comunicazione tra livello OT ed IT all'interno di realtà industriali ed aziendali. In questo modo Siemens concede alle aziende una via più rapida verso quello che è il concetto di Industria 4.0, creando un'agile cooperazione tra il sistema produttivo e tutto ciò che concerne le tecnologie d'informazione. La definizione di questo ecosistema in cui tutti i macchinari e sistemi di automazione sono accompagnati da connessioni a dispositivi ausiliari, garantisce all'azienda una gestione semplificata degli hardware e software in possesso, ed un'organizzazione delle informazioni raccolte più efficace. Inoltre mette a disposizione altre funzionalità come lo storage di dati o un'eventuale analisi su codice di alto livello.

Siemens Industrial Edge è composto principalmente da tre elementi:

- Una infrastruttura centralizzata per la gestione e la amministrazione dei dispositivi edge distribuiti nel sistema
- Edge devices che forniscono infrastrutture hardware e software per l'esecuzione di applicazioni
- Edge apps che arricchiscono le funzionalità di un dispositivo edge come data analysis e diagnostica

Quindi si avrà bisogno di un Edge Management System, un framework per la gestione di tutti i dispositivi connessi e che può essere paragonato ad una cabina di pilotaggio. L'Edge Management System permette di monitorare le condizioni di ogni Edge device connesso e inoltre concede la possibilità di spostare edge apps e funzioni software su uno specifico dispositivo direttamente dal Cloud. Quando si parla di Edge Computing si fa riferimento alla distribuzione di risorse computazionali in modo che siano in prossimità a dove i dati vengono generati. Prima di poter utilizzare questi dati però prima vanno raccolti e processati dagli Edge devices. Quindi vengono installati a bordo macchina questi piccoli PC industriali con modeste capacità di elaborazione e risorse hardware. Questi IPC(Industrial PC) ospitano un runtime environment in cui possono essere messe in esecuzione diverse applicazioni. Applicazioni che possono essere acquistate direttamente da Siemens o magari sviluppate appositamente per l'ambiente in cui andranno ad eseguire. Questi sono gli aspetti di cui si compone Industrial Edge ma adesso andiamo a defi-

nire ed a spiegare in modo pratico gli elementi che compongono e sorreggono la piattaforma Siemens.

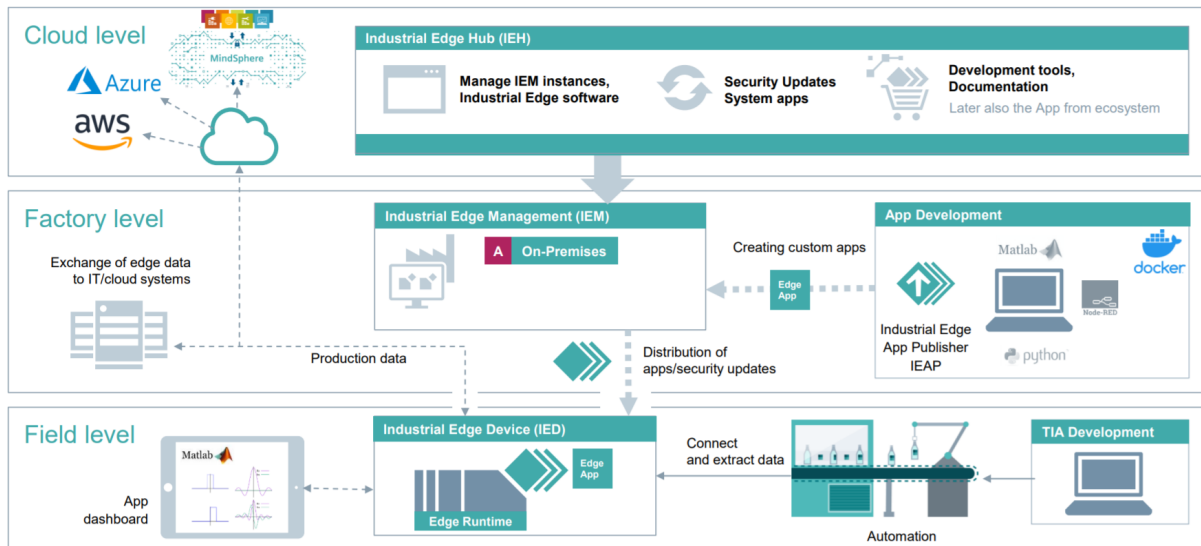


Figura 3: Architettura Industrial Edge

Industrial Edge Device

Gli Industrial Edge Devices abbreviati con IED si trovano nella parte più bassa dello schema logico e sono le entità che si trovano più vicine ai sistemi di automazione. I dispositivi vengono distribuiti su tutto l'impianto industriale, andando a formare una rete di nodi Edge, e si occupano di svolgere la parte operativa vera e propria del sistema. Siemens mette a disposizione una vasta gamma di prodotti compatibili con Industrial Edge, tutti con diverse caratteristiche e adatti a situazioni e necessità differenti. I prodotti venduti da Siemens che vengono poi installati a fianco delle macchine automatiche sono connessi alle stesse tramite protocolli industriali e fungono da architettura hardware per l'esecuzione di applicazioni e servizi. Tutti i dispositivi ospitano lo stesso sistema operativo proprietario di Siemens basato su Linux, che permette l'esecuzione di applicazioni sotto forma di Docker container. I container dovranno rispettare diversi requisiti e protocolli, necessari per poter essere integrati nell'architettura. Il sistema operativo di Siemens garantisce una connessione sempre attiva all'Industrial Edge Management System, un Docker Engine per l'esecuzione dei container Docker e una GUI di amministrazione e monitoraggio delle risorse occupate, nonché altre funzionalità come il controllo dei log delle applicazioni.

Industrial Edge Management

Successivamente ad livello concettualmente superiore abbiamo l'Industrial Edge Management(IEM), componente software installato indipendentemente on-premise o in Cloud, che svolge il ruolo di amministratore degli Edge Devices e di tutte le applicazioni e funzionalità che questi mettono a disposizione. Offre un'interfaccia per la gestione dei gruppi e degli utenti, per il monitoraggio dello stato globale, per il backup e restore degli Edge Device. L'Edge Management System è anche il sistema centrale per la gestione, l'aggiornamento e il monitoraggio degli Edge Device connessi e del software su di essi montato. Tramite l'IEM è possibile scaricare nuovi aggiornamenti e con pochi comandi fare l'update collettivo di tutti i dispositivi che sono all'interno di un impianto industriale o più. Questo aspetto offre molta scalabilità in quanto non sarà più necessario accedere ad ogni singolo Edge Device ogni qualvolta sarà presente una versione nuova di un software ma utilizzando il sistema centrale sarà possibile fare l'aggiornamento in modo trasparente e in maniera poco invasiva. L'Edge Management System garantisce anche la possibilità di accesso remoto ad ogni Edge Device connesso alla rete e facente parte del sistema. In questo modo da una posizione centralizzata è possibile accedere, modificare e monitorare lo stato interno di ogni dispositivo.

Industrial Edge Hub

L'Industrial Edge Hub(IEH) è il componente software definito come entry point quando ci si appropria a Siemens Industrial Edge. Da questo Hub centrale è possibile fare il download e la configurazione di un'istanza dell'Industrial Edge Management, scaricando il sistema operativo e tutti i software necessari per l'esecuzione dell'IEM. Una volta installato l'Industrial Edge Management on-premises sarà possibile aver accesso al catalogo di Siemens ed al suo store in modo da poter scegliere ed acquistare le applicazione desiderate per il nostro caso industriale. Oltre a questo in Industrial Edge Hub sono contenute anche tutte le documentazioni e guide necessarie per assistere i clienti nell'usufruire correttamente dei prodotti Siemens. In più sono presenti anche tutte le configurazioni iniziali da usare per avviare il sistema Siemens.

2.1.2 Applicazioni Siemens

I prodotti Siemens, da un punto di vista hardware, gettano le fondamenta su poi quello che diventerà l'ambiente di esecuzione per le Edge apps che

valorizzano così tanto l'utilizzo dell'Edge Computing in contesti di tipo industriale. All'interno di questa struttura possiamo trovare tre tipi diversi di applicazione:

- Software proprietari e sviluppati da Siemens e quindi fortemente integrati con la piattaforma
- Software sviluppati da terze parti come gli OEMs, resi poi disponibili a loro volta sul catalogo
- Software personalizzati, sviluppati indipendentemente da entità che possono far parte dell'azienda acquirente o di aziende esterne.

Il catalogo di Siemens offre già un'ampia gamma di applicazioni tutte inerenti alle funzionalità che uno si aspetta di trovare in Edge Computing. Possiamo trovare *Connector*, applicazioni che si occupano della comunicazione con i sistemi di automazione utilizzando diversi protocolli e dando la possibilità all'utente di impostare diverse configurazioni come frequenza di campionamento o dati di interesse. I Connector disponibili comprendono già e utilizzano i protocolli industriali standard più diffusi, come possono essere S7, S7 Plus, OPC-UA, Modbus TCP, PROFINET e molti altri. I connettori si occupano del ruolo di interfacciarsi alle macchine, molto spesso eterogenee tra loro, in modo da rendere accessibile all'Edge Device tutte quelle informazioni che vanno a definire lo stato di una macchina. Oltre ai connettori esistono altre applicazioni che si occupano non più della comunicazione tra macchinario e Edge Device ma di comunicazione tra applicazioni. Questo è il caso del IE Databus, software fondamentale per l'applicazione di casi pratici di monitoraggio e interazione tra applicazioni. Come altro esempio di applicazione abbiamo *Data Service* sviluppato da Siemens che non è altro che un servizio di storage che si occupa della persistenza del dato. *Performance Insight* può essere un altro software interessante permettendo all'utente la creazione di dashboard e grafici basati su flussi di dati ricavati dai macchinari stessi. Questi ovviamente sono solo alcuni esempi, che però possono già dare un'idea delle potenzialità della piattaforma.

Fino ad adesso però sono solo stati nominati software sviluppati interamente da Siemens. La possibilità di sviluppare la propria applicazione ad-hoc per una determinata situazione estende maggiormente la scalabilità e l'adattabilità del sistema. Adesso andiamo ad illustrare un po' più nel dettaglio le applicazioni cardine e principali che andranno ad essere utilizzate in questo caso di studio.

IE Databus

Come già accennato, all'interno del Edge Device è presente una rete che permette alle applicazioni di comunicare tra di loro. Questa comunicazione avviene attraverso l'IE Databus che non è altro che un'implementazione di un broker MQTT. Nello specifico si tratta del conosciuto software open source Mosquitto, concedendo la possibilità dello scambio di messaggi tramite una semantica publish/subscribe. In questo modello un client pubblica un messaggio con una certa etichetta(topic) verso il broker, che a sua volta lo distribuisce ai destinatari interessati. MQTT(Message Queue Telemetry Transport) è un protocollo che si adatta molto bene ad una comunicazione basata su eventi come può essere la lettura e l'invio di dati da sistemi di automazione.

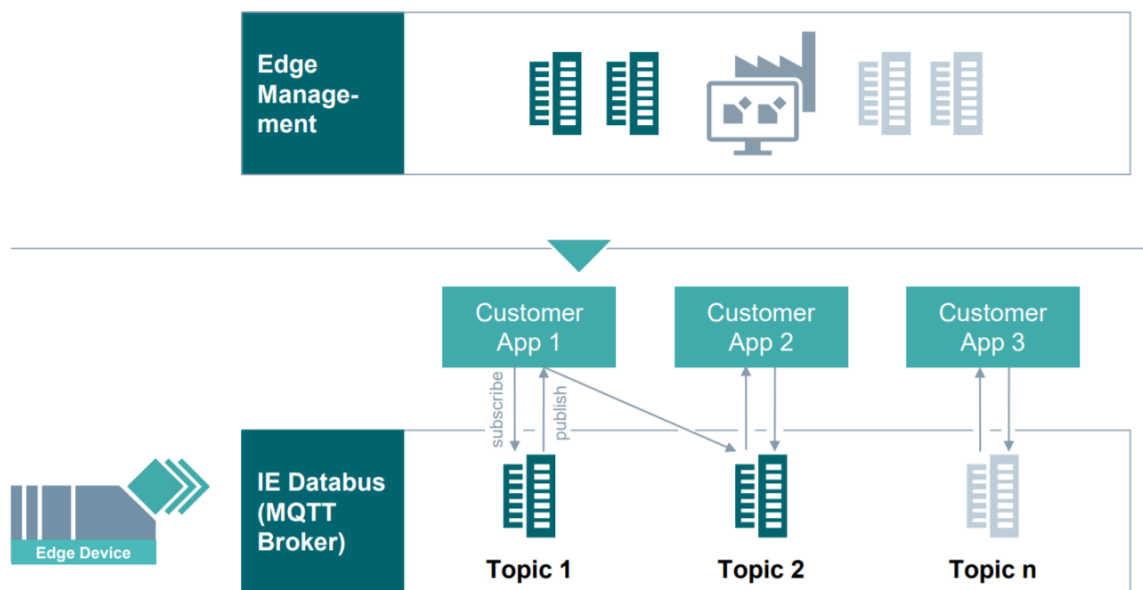


Figura 4: Industrial Edge Databus

L'applicazione in questione ovviamente può essere configurata a dovere, infatti tramite l'Industrial Edge Management è possibile lanciare il software denominato *IE Databus Configurator* per poter impostare la tipologia di topic disponibili e i diritti di accesso che gli utenti hanno nei loro confronti. Una volta definiti i settings necessari basterà fare il deploy della configurazione sul dispositivo desiderato per ottenere i comportamenti voluti.

IE Flow Creator

IE Flow Creator è un ennesimo programma offerto da Siemens che permette all'utente di aggiungere logica computazionale ai flussi di dati interni ad un Edge Device. L'applicazione è un'implementazione del già noto Node-RED e rende accessibile questo strumento attraverso la definizione di una interfaccia grafica interna al dispositivo.

Node-RED è un software open source per la programmazione tramite blocchi funzionali (chiamati nodi) di un flusso di trasformazioni dei dati. La sua natura grafica, la varietà di funzioni disponibili e la possibilità di realizzare flussi di trasformazione senza la scrittura di codice lo rendono uno strumento molto semplice e alla portata anche di persone non avvezze alla programmazione.

Caratteristica però che dà ancora più profondità a questo software è la scrittura e realizzazione di nodi capaci di eseguire codice Javascript aggiungendo varietà alle tipologie di soluzioni che si possono utilizzare.

Inoltre questa versione di Node-RED presenta anche nodi specifici all'ecosistema Siemens come blocchi che vanno a definire la lettura di uno specifico connettore. IE Flow Creator è uno strumento molto facile da approcciare ma che dà all'utente la capacità di definire diversi tipi di algoritmi, da semplici funzioni di processing del dato ad interi programmi per l'elaborazione, analisi e visualizzazione dello stato macchina.

2.2 Industrial Edge App Publisher

All'interno dell'ecosistema Siemens possono esistere ed interagire diversi tipi di applicazioni e software ma la possibilità di poter realizzare i propri programmi al solo costo di seguire determinate regole rende questa soluzione molto interessante per le aziende. Per poter assistere le aziende che decidono di utilizzare Industrial Edge e sviluppare indipendentemente le proprie Edge Apps, Siemens mette a disposizione uno strumento chiamato Industrial Edge App Publisher. L'App Publisher è un software disponibile per l'installazione su Windows e Ubuntu/Debian, e che permette di gestire la creazione e l'aggiornamento di applicazioni custom, basate su Docker, per l'esecuzione esclusiva su Industrial Edge Device marcati Siemens. Per poter funzionare correttamente, l'applicazione necessita di due cose fondamentali:

- Una connessione all'istanza dell'Industrial Edge Management, su cui poi si vorranno caricare le applicazioni custom
- Una connessione a un Docker Engine tramite TCP/IP, che non necessariamente deve risiedere sullo stesso host dove l'App Publisher viene eseguito.

La connessione all'Edge Management serve per poter inviare nuove versioni delle applicazioni sviluppate o per poter importare software già installati sui dispositivi. Mentre la connessione al Docker Engine permette all'App Publisher di poter eseguire comandi Docker in modo da eseguire operazioni come il build di immagini.

L'Industrial Edge App Publisher mette a disposizione un'interfaccia grafica che aiuta e semplifica la realizzazione di Edge Apps compatibili con le piattaforme Siemens. Il tutto viene fatto tramite la definizione di un file Docker-Compose che andrà a descrivere i comportamenti, la struttura e le caratteristiche dell'applicazione.

Inoltre da questo file l'App Publisher, utilizzando la connessione al Docker Engine remoto o locale, saprà quali immagini di Docker dovrà caricare per costruire l'App. Per rendere meglio l'idea dei passaggi che bisogna compiere nella realizzazione di una custom App basterà guardare l'immagine seguente.

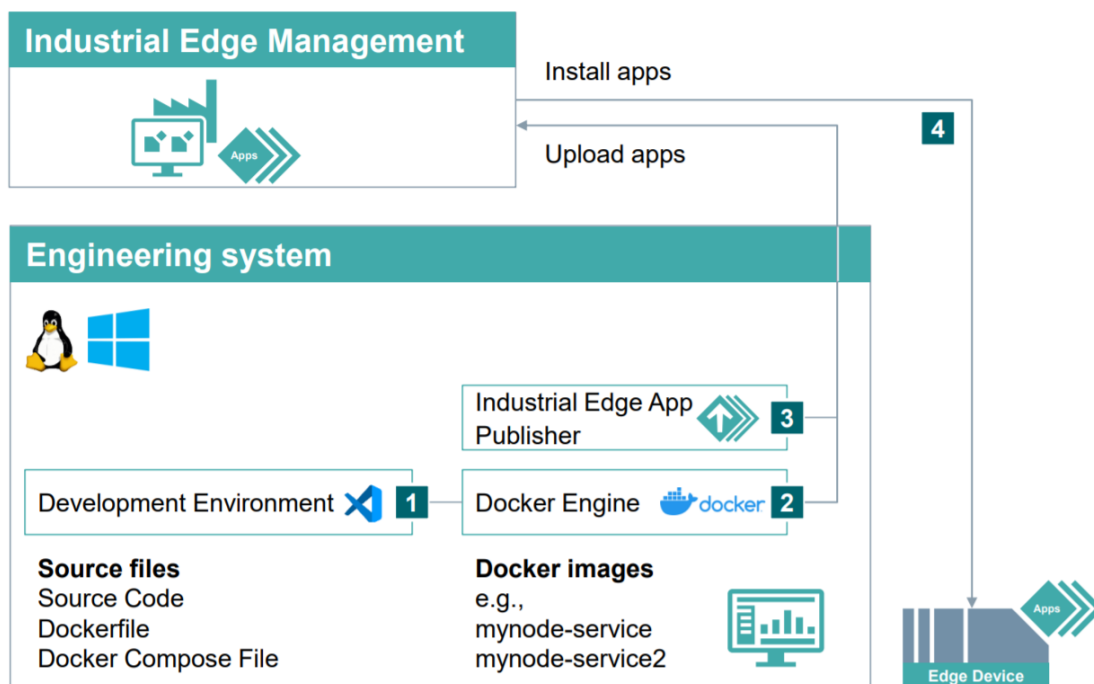


Figura 5: App Development Workflow

1. Inizialmente il tutto parte dalla scrittura del codice utilizzando un qualsiasi ambiente di sviluppo come può essere Visual Studio Code. Importante la definizione di Dockerfile e Docker Compose a seconda dei servizi interni all'applicazione.
2. Una volta ottenuti tutti i files necessari, utilizzando i comandi del Docker Engine andiamo a creare le immagini necessarie per i container. Le immagini verranno salvate poi nella repository del Docker Engine.
3. Accedendo all'App Publisher ci verrà chiesto di selezionare un file Docker Compose, file che informerà il programma quali immagini dover prendere dalla repository del Docker Engine. Successivamente si potrà avviare la creazione dell'applicazione e farne l'upload verso l'Industrial Edge Management.
4. Infine la nostra applicazione custom sarà disponibile all'interno dell'Industrial Edge Management e sarà possibile installarla su qualsiasi Edge Device connesso.

L'App Publisher, tra le funzionalità già elencate, presenta all'utente anche un'interfaccia grafica in cui si potrà modificare il file Docker Compose anche dopo essere stato caricato. In questa parte si potranno selezionare ed

impostare diversi parametri necessari per essere eseguiti su Edge Devices Siemens.

- **Volumes:**

Per ogni App esisteranno sempre due bind mount di default verso il file system del dispositivo e saranno sempre chiamate come segue:

```
- ./publish/:/publish/
```

```
- ./cfg-data/:/cfg-data/
```

La mount denominata *publish* è designata come repository su cui pubblicare risultati o altri file. Questo permette anche lo scambio di file tra applicazioni che eseguono sullo stesso dispositivo. La mount chiamata *cfg-data* invece è unicamente usata in lettura per permettere alle applicazioni di utilizzare files di configurazione all'avvio.

- **Memory limit:**

Per ogni servizio è obbligatorio indicare la memoria massima che potrà utilizzare tramite il parametro *mem_limit*. Questa informazione servirà al dispositivo per sapere quanta memoria dover riservare al momento dell'installazione dell'app. A seconda dell'Edge Devices utilizzato come hardware, si avrà un certo numero di risorse e spazio per poter installare un determinato numero di software. Diventa così cruciale la definizione di questo parametro cercando di ottimizzare al meglio le risorse disponibili. Un valore troppo basso andrebbe ad incidere sulle performance dell'applicazione, trovandosi costretta a rallentare l'elaborazione per non usare troppa memoria. Mentre un valore troppo alto andrebbe a riservare una parte della memoria che eventualmente non verrà mai impiegata ma che comunque rimarrà tolta ad altri software.

- **Networks:**

Come è risaputo, servizi facenti parte dello stesso Docker Compose e di conseguenza della stessa applicazione possono comunicare tra di loro mediante una rete interna. Su dispositivi Siemens però è sempre presente anche un'altra rete denominata *proxy-redirect*. Se un programma utilizza questa rete allora è in grado di comunicare con altre applicazioni in esecuzione sul dispositivo come può essere l'IE Databus.

La comunicazione può avvenire anche con entità esterne al dispositivo ma dovranno essere esposte porte dedicate che fungeranno da canale per lo scambio di messaggi.

Alcune di queste features diventano anche obbligatorie per l'app affinché possa eseguire correttamente all'interno dei dispositivi Siemens. Possiamo dire che ogni applicazione sviluppata debba seguire delle regole guida per poter usufruire al massimo della piattaforma.

Fare il deploy di applicazioni partendo dal codice sorgente è una procedura che richiede tempo e che non si vuole ripetere costantemente. Quindi a fronte di piccole modifiche di comportamento da parte dell'applicazione è possibile utilizzare un file di configurazione per notificare al software cosa cambiare senza dover mettere mano al codice. Ogni qualvolta si fa l'installazione di un'applicazione ci viene sempre anche chiesto di inserire un file di configurazione. Il file se esiste viene portato insieme al software sul dispositivo e successivamente salvato nel filesystem nella cartella *cfg-data*. Inoltre il file potrà essere caricato nuovamente anche se l'applicazione è già installata, anche se questo procedimento richiederà il riavvio dell'app. Il file può essere scritto anche sul momento mentre si utilizza Industrial Edge Management rendendo l'operazione più veloce e meno dispendiosa.

2.3 FOCAS Library

Prima di poter andare a parlare di FANUC FOCAS Library è necessario introdurre un paio di concetti per definire al meglio il contesto di applicazione. Va spiegato innanzitutto a cosa ci si riferisce quando si fa riferimento ad una macchina a controllo numerico abbreviato con MNC(Computerized Numerical Control Machine). Generalmente si tratta di macchine utensili che utilizzano una tecnologia a controllo numerico computerizzato, detta CNC(Computerized Numerical Control). Quindi quando si parla di CNC non si fa riferimento direttamente al macchinario industriale bensì al computer, molto spesso esterno alla macchina, che permette l'interfacciamento e la programmazione della stessa. Questo tipo di approccio oggi è ampiamente utilizzato perché permette il controllo numerico diretto da un computer esterno(CNC). Uno dei tanti produttori e venditori di questi macchinari è proprio FANUC, marchio ormai affermato in tutto il mondo. In questo caso di studio verranno utilizzate proprio apparecchiature e tecnologie proprietarie di FANUC. In particolare l'azienda ha a disposizione un'ampia gamma di sistemi CNC in grado di soddisfare qualsiasi esigenza, da controlli di qualità a sistemi di controllo con prestazioni elevate per macchine complesse. La maggior parte di questi controllori CNC utilizzano un particolare protocollo che permette tramite una semplice connessione Ethernet di accedere dall'esterno

a tutte le informazioni inerenti al macchinario. Questo protocollo standard per la raccolta dati è chiamato FOCAS ed è largamente utilizzato in ambito industriale. Una macchina a controllo numerico che supporti FOCAS mette a disposizione una porta al numero 8193, riservando un canale per la comunicazione esterna. Utilizzando un computer esterno, molto spesso PC industriali, è possibile connettersi e comunicare con il controllore numerico (CNC). Le operazioni che si possono eseguire sono molteplici dalla semplice raccolta dati alla programmazione vera e propria di programmi CNC.

Adesso che abbiamo gettato le basi su quello che sono gli elementi in gioco possiamo finalmente iniziare a parlare dell'argomento vero e proprio. FANUC FOCAS 2 Library è una serie di files (dll) che vanno a definire una libreria per interfacciarsi a CNC FANUC. La libreria mette a disposizione una serie di API per poter leggere i dati delle macchine utensili controllate dal CNC. La libreria accede al CNC sia tramite connessione Ethernet sia tramite HS-SB (High Speed Serial Bus). FANUC FOCAS è scritta interamente in C++ ed è compatibile con sistemi operativi Windows e Linux ma principalmente utilizzato con quest'ultimo in quanto maggiormente impiegato in ambito industriale. Anche l'Edge Device di Siemens utilizzato per questo caso di studio monta al suo interno un OS Linux. La libreria in questione definisce e utilizza un ampio numero di funzioni ma a seconda del modello di FANUC CNC in uso, alcune funzioni saranno compatibili mentre altre no. La compatibilità della funzione deriva anche dalle capacità e caratteristiche della macchina utensile collegata. Per ogni funzione è inoltre definita una struttura dati che va passata al momento della chiamata e una volta terminata la funzione la struttura sarà popolata con i dati letti dal macchinario.

2.4 Python e modulo ctypes

Come linguaggio di programmazione per lo sviluppo di applicazioni custom è stato scelto Python, andando poi a creare del codice eseguibile su sistema operativo Linux. Python è un linguaggio che è cresciuto notevolmente negli ultimi anni grazie alla possibilità di definire moduli personalizzati contenenti diversi tipi di funzioni. Il linguaggio è fortemente utilizzato per eseguire operazioni di Data Analytics e più in generale quando si ha da manipolare una grande quantità di dati. Il che lo rende molto adatta al caso di studio trattato visto che avrà a che fare con raccolta dati.

Tra i diversi moduli disponibili di Python, uno in particolare suggerisce una

soluzione progettuale efficace, garantendo l'utilizzo da parte di codice Python della libreria FANUC FOCAS. Il modulo in questione si chiama *ctypes* e mette a disposizione una serie di funzioni e data types che permettono di fare il wrapper di librerie C/C++ in modo da poter poi essere invocate con codice Python. Questo modulo agevola notevolmente lo sviluppo di applicazioni che fanno uso di librerie scritte in C/C++, in modo da non necessitare il supporto di nessun altro tipo di estensione. Perciò all'interno di *ctypes* esistono alcune funzioni che permettono di caricare librerie chiamate Dynamic Link Libraries (DLL) o in Linux, Shared Objects (SO). La libreria viene caricata e modellata come classe Python di cui poi se ne creerà un'istanza. L'istanza fungendo da wrapper alla libreria nativa in C/C++ permetterà l'invocazione di funzioni sotto la forma di una chiamata di metodo di classe. Non solo saranno accessibili le funzioni ma anche i dati types e strutture dati definite nella libreria. Inoltre per garantire il corretto passaggio di dati vengono definiti dei C data types mappati poi su quelli Python:

ctypes type	C type	Python type
<code>c_bool</code>	<code>_Bool</code>	bool (1)
<code>c_char</code>	char	1-character bytes object
<code>c_wchar</code>	wchar_t	1-character string
<code>c_byte</code>	char	int
<code>c_ubyte</code>	unsigned char	int
<code>c_short</code>	short	int
<code>c_ushort</code>	unsigned short	int
<code>c_int</code>	int	int
<code>c_uint</code>	unsigned int	int
<code>c_long</code>	long	int
<code>c_ulong</code>	unsigned long	int
<code>c_longlong</code>	__int64 or long long	int
<code>c_ulonglong</code>	unsigned __int64 or unsigned long long	int
<code>c_size_t</code>	size_t	int
<code>c_ssize_t</code>	ssize_t or Py_ssize_t	int
<code>c_float</code>	float	float
<code>c_double</code>	double	float
<code>c_longdouble</code>	long double	float
<code>c_char_p</code>	char* (NUL terminated)	bytes object or None
<code>c_wchar_p</code>	wchar_t* (NUL terminated)	string or None
<code>c_void_p</code>	void*	int or None

Figura 6: Mapping C-Python Data types

Aspetto però che potrebbe risultare un po' svantaggioso è quando si ha la necessità di utilizzare Array e Struct definite in C/C++. In particolare le strutture C non possono essere utilizzate direttamente dalla libreria così come sono. Anche qui si ha bisogno di effettuare un wrapping, mappando la struttura definita nella libreria su una classe Python. Operazione che sembrerebbe non inficiare notevolmente nello sviluppo ma quando le librerie presentano un elevato numero di strutture si va a perdere un po' di praticità in quanto vanno riscritte manualmente una ad una. Tralasciando questo particolare, l'impiego di questo modulo è fortemente consigliato e permette allo sviluppo di codice "cross-language" senza l'utilizzo di ulteriori estensioni.

Capitolo 3

Architettura progetto

L'idea che ha dato vita al caso di studio impiegato in questa tesi deriva da un progetto molto più ampio, in cui diverse entità aziendali collaborando con figure universitarie, hanno condotto ricerche e sperimentazioni su soluzioni di Edge Computing applicabili a situazioni pratiche e implementabili a livello industriale.

3.1 Progetto DEEPMON

Il progetto in questione è organizzato dall'ente di ricerca Bi-Rex (Big Data Innovation and Research Excellence), consorzio pubblico-privato con sede a Bologna, nato nel 2018 dalla cooperazione di alcune tra le più importanti imprese del panorama industriale del territorio emiliano-romagnolo, le università e diversi centri di ricerca. Bi-Rex ricopre un ruolo di supporto per le imprese interessate ad adottare i principi dettati dall'Industria 4.0. I servizi offerti spaziano dalla formazione e consulenza alle imprese fino a progetti di ricerca e sviluppo. Tra le diverse attività che Bi-Rex supporta non possiamo non nominare l'iniziativa che prende il nome di progetto DEEPMON (Dynamic Edge computing for Plant MONitoring). Progetto che ha come obiettivo la realizzazione di una soluzione Edge scalabile per la raccolta dati da impianti industriali e più in generale la definizione di un modello comune di riferimento per tutte quelle imprese che vogliono adottare l'Edge Computing.

L'introduzione di tecnologie IT in campo industriale sta portando notevoli incrementi delle performance produttive, riduzione dei costi, facilitazione dei processi decisionali e manutentivi. Nonostante ciò la rivoluzione tecnologica deve far fronte alla vasta eterogeneità dettata dai diversi macchinari e protocolli impiegati alle volte anche nella stessa linea produttiva. Inoltre se aggiungiamo anche la necessità da parte delle aziende di mantenere sistemi

hardware e software indicati come legacy si viene a creare una situazione complessa in cui possono sorgere diverse problematiche. Ed è qui che DEEPMON cerca di intervenire proponendo un modello comune a livello industriale per l'applicazione di Edge Computing su linee produttive. La piattaforma mira ad essere agnostica rispetto al contesto d'uso specifico e basata quanto più possibile su standard aperti. In questo modo potrà essere applicata anche qualora esistano diversi protocolli, macchinari e procedure. Altro obiettivo che si pone il progetto DEEPMON, in modo da agevolare il passaggio ad un modello comune, è la definizione di un data model da utilizzare come riferimento per applicazioni e software che si occupano della raccolta dati. A livello industriale esistono già diversi data model che vengono utilizzati come riferimento ma il più delle volte sono modelli specifici ad una determinata tipologia di industria come possono essere i produttori di packaging di plastica o la produzioni di componenti metalliche. L'idea di DEEPMON è quella di definire un data model ampio e flessibile in modo da poter far ricadere al suo interno il maggior numero possibile di casi industriali e in questo modo cercare di avanzare una proposta di standard per future applicazioni. Il progetto ha anche ricevuto sostegno da Siemens, colosso in ambito di tecnologie IT, nella forma di supporto tecnico, consulenza e fornitura di componenti hardware e software per la sperimentazione. Quindi l'attività è stata utilizzata anche per condurre l'esplorazione del sistema Industrial Edge di Siemens e di software open source per il trattamento dei dati.

Per verificare e validare l'effettiva funzionalità delle soluzioni proposte da progetto DEEPMON si sono offerte di collaborare diverse imprese di spessore del territorio emiliano-romagnolo, mettendo a disposizione degli use case come banco di prova e per migliorare ulteriormente la piattaforma. Uno di questi use case è quello utilizzato come punto di partenza per lo svolgimento dell'attività svolta in questa tesi e verrà spiegata nel dettaglio nel successivo capitolo.

3.2 Poggipolini Use Case

Fondata a Bologna nel 1950, la Poggipolini è un'azienda italiana specializzata in progettazione e lavorazioni meccaniche ad altissima precisione, in soluzioni tecniche per la riduzione del peso, in componenti critici e sistemi di fissaggio strutturali per i settori aerospace, automotive e motorsports. L'azienda si occupa principalmente della realizzazione di componenti metalliche e quin-

di ospita linee produttive incentrate su fresatura e incisione. Poggipolini ha collaborato al progetto DEEMON mettendo a disposizione una linea di produzione robotizzata Erowa per la lavorazione di leghe speciali per il settore aerospaziale, racing e hypercars. La linea è composta da tre macchine utensili per la lavorazione del pezzo mediante operazioni di fresatura, una macchina di lavaggio e infine una macchina di misura Zeiss per il controllo di qualità. Lo spostamento del materiale da un macchinario all'altro è gestito da un robot che ha accesso al magazzino. Figure principali dello use case sono le tre macchine utensili che lavorano il pezzo, su cui sono installati dei FANUC CNC Control Machine 31i model B5. Come già spiegato in precedenza i prodotti CNC di casa FANUC permettono di aver accesso a tutte le informazioni interne della macchina. Alcuni esempi possono essere lo status della macchina, il consumo elettrico, il programma in esecuzione, la coppia motore e molti altri.

Obiettivo dello use case è quello di andare ad installare un nodo Edge in prossimità delle tre macchine utensili e tramite estrazione dati riuscire a monitorare l'andamento della macchina. Il tutto reso possibile dall'esecuzione di una serie di applicazioni sul nodo Edge. L'hardware che ricopre il ruolo di nodo Edge è stato messo a disposizione da Siemens. Il dispositivo utilizzato è un PC industriale chiamato SIMATIC Nanobox PC su cui monta un sistema operativo Linux ed equipaggiato con 8GB di RAM, una CPU Intel a 4 core e 200GB di storage su HDD. L'Edge Device è stato installato a fianco delle macchine automatiche e connesso alla rete interna dello stabilimento tramite cavo Ethernet. Ulteriore scopo del progetto è quello di sperimentare l'efficacia di soluzioni Edge, come la piattaforma Siemens Industrial Edge, e di come si comporti a fronte di macchinari e protocolli eterogenei.

Per fissare visualmente come i dispositivi sono collegati tra di loro basti guardare l'immagine sottostante:

- A sinistra abbiamo le tre macchine utensili che effettuano lavorazioni su materiali passati da un robot con accesso al magazzino.
- Al centro abbiamo i FANUC CNC direttamente installati su ognuno dei tre macchinari. Il CNC permette accesso al macchinario tramite libreria FANUC FOCAS.
- Infine a destra abbiamo il SIMATIC Edge Device connesso sia alla rete interna che ai CNC.

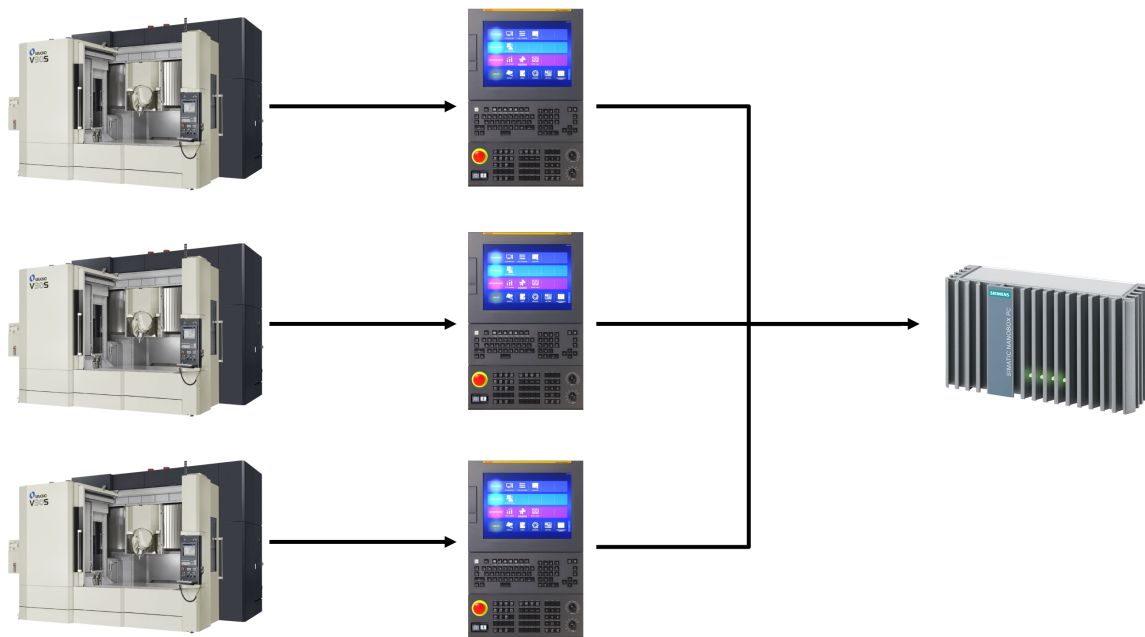


Figura 7: Collegamento Macchine Utensili

Le attività del progetto sono partite dall'effettiva installazione del SIMATIC Edge Device all'interno dell'impianto industriale. L'operazione è avvenuta mediante l'assistenza di Siemens come parte del suo supporto al progetto. Una volta ottenuto il dispositivo operativo si è passati all'installazione delle componenti software. Accedendo al portale Industrial Edge Hub è stato possibile scaricare ed installare un'istanza dell'Industrial Edge Management. Questa istanza è stata installata su una macchina virtuale predisposta sul Cloud privato aziendale. Successivamente si è passati a fare l'Onboarding del dispositivo andando a creare il collegamento tra il Management e l'Edge Device. Da questo momento in poi il PC industriale è gestibile interamente dal Management System. Il dispositivo può essere manovrato da remoto per eseguire determinate operazioni come accensione, spegnimento oppure aggiornamento del firmware.

L'ecosistema Siemens gestisce l'accesso alla piattaforma tramite la definizione di utenti e password in modo da poter garantire la sicurezza del sistema e poter monitorare le operazioni effettuate dai diversi utenti. Per ogni utente poi possono essere assegnati ruoli e gruppi definendo il tipo di operazioni eseguibili da un determinato dipendente. Ciò garantisce uno stretto controllo sulle azioni di ogni dipendente che ha accesso alla piattaforma.

A questo punto abbiamo l'Edge Device attivo e connesso correttamente alle

macchine utensili e accessibile da Management System. Ciò che rimane da fare è la definizione e sviluppo di applicazioni che andranno ad eseguire sul dispositivo in modo da leggere i dati dalle macchine utensili.

Come accennato in precedenza su dispositivi Siemens possono eseguire tre tipologie di software:

- Software proprietari sviluppati da Siemens
- Software sviluppati da terze parti come gli OEMs
- Software personalizzati sviluppati indipendentemente

Per questo progetto sono state utilizzate diverse applicazioni le quali appartengono a diverse categorie. Sono stati impiegati sia software Siemens scaricabili dal catalogo, sia applicazioni sviluppate indipendentemente come attività di tesi. Inoltre sono state anche impiegate applicazioni sviluppate da collaboratori universitari che per estremo possono essere viste come soluzioni software offerte da OEMs,

Già da questi piccoli dettagli si possono notare le potenzialità della piattaforma. Un'impresa che usufruisce di Siemens Industrial Edge può percorrere diverse strade per ottenere la soluzione più adatta e congeniale alla propria situazione. Il cliente può avere accesso al catalogo e verificare se esista la presenza di applicazioni già funzionanti e approvate da Siemens. Nel caso le Edge Apps disponibili non rispettino le necessità aziendali si può allora pensare di investire risorse nella realizzazione di un software ad-hoc. Si presentano però anche casi in cui imprese medie o piccole non hanno a disposizione reparti di sviluppo in grado svolgere compiti del genere. Come ulteriore possibilità si può pensare di delegare la produzione dell'applicazione ad una azienda esterna che si occupi di progettazione software. Quindi come possiamo vedere non esiste una soluzione unica, ma utilizzando Siemens Industrial Edge si ha la facoltà di adottare una vasta varietà di opzioni che possono essere valutate in base ai costi, tempo e risorse necessarie.

Ritornando a parlare di quello che è stato fatto per lo use case di Poggipolini, il progetto prevede la realizzazione di uno stack di applicazioni che collaborando tra di loro passano dalla lettura del dato alla sua persistenza. In questo modo si può effettuare il monitoraggio del dato, avere informazioni sullo stato della macchina e nel caso agire di conseguenza. Per avere un'idea più chiara delle entità in gioco basti guardare la figura sottostante:

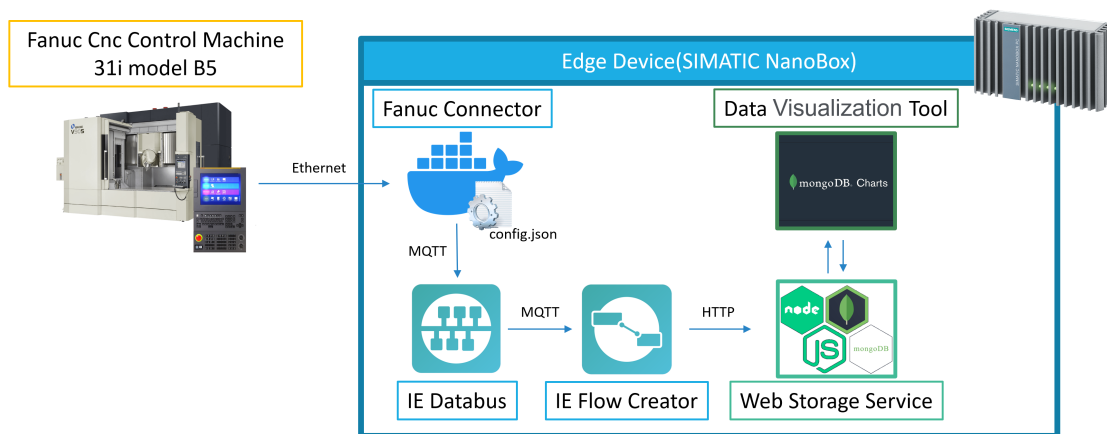


Figura 8: Software on SIMATIC Edge Device

- **Connettore FANUC:**

Applicazione custom sviluppata indipendentemente richiesta da Poggiolini per effettuare l'interfacciamento con il FANUC CNC.

- **IE Databus:**

Software di Siemens installato dal catalogo per supportare lo scambio di messaggi tra applicazioni sullo stesso dispositivo.

- **IE Flow Creator:**

Software di Siemens installato dal catalogo per la scrittura ed esecuzione di codice Javascript.

- **Web Storage Service:**

Applicazione sviluppata da collaboratore universitario per effettuare persistenza del dato. Il software prende la forma di un web service che tramite REST API permette lo storage del dato.

- **Data Visualization Tool:**

Strumento di visualizzazione grafica per la definizione di una dashboard aggiornata in real-time dai dati letti dalla macchina utensile.

Il progetto quindi ha come obiettivo la messa in pratica dei principi e idee dettate dal piano DEEPMON, comprendendo una parte di sviluppo software e un'altra di organizzazione delle applicazioni in modo da garantire la cooperazione delle stesse.

3.3 Connettore FANUC

Siemens mette a disposizione diversi tipi di connettori per comunicare con le macchine utensili, i quali utilizzano protocolli diversi. Se una azienda implementa Siemens Industrial Edge, il primo pensiero ricade su quale componente software utilizzare per accedere ai sistemi di automazione. Nella maggior parte dei casi, se il sistema di automazione impiega uno dei protocolli industriali standard più diffusi, allora è probabile che sia già presente un'applicazione adatta nel catalogo Siemens. Infatti Siemens ha selezionato un certo numero di protocolli industriali da cui poi ha ricavato dei Connector compatibili con Industrial Edge. Per citarne alcuni, esistono già soluzioni per S7, S7 Plus, OPC-UA, Modbus e PROFINET.

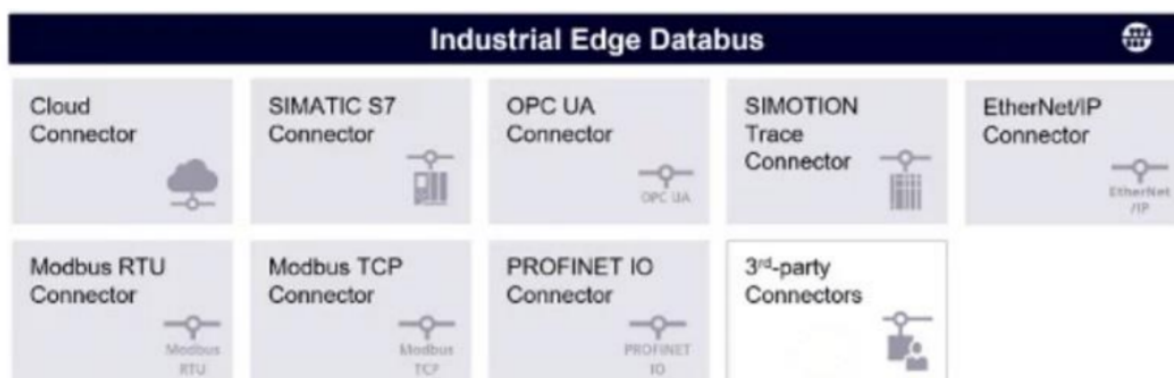


Figura 9: Siemens Connector

Sfortunatamente non è questo il caso per quanto riguarda Poggipolini. Infatti il protocollo scelto per comunicare con i FANUC CNC installati su ogni macchina utensile è FANUC FOCAS. Protocollo di trasmissione dati basato su connessione Ethernet e che predispone un canale di comunicazione sulla porta 8193.

La scelta di utilizzare un protocollo di comunicazione che non rientri tra le soluzioni offerte da Siemens ci porta alla necessità di sviluppare un'applicazione custom che faccia uso della libreria FANUC FOCAS e che sia integrabile su dispositivi Siemens.

Lo sviluppo di un'applicazione compatibile con Siemens si traduce nella realizzazione di un Docker container che eseguirà al suo interno del codice e sfruttando la libreria FANUC FOCAS invocherà funzioni sul FANUC CNC estraendo strutture dati contenenti informazioni sullo stato della macchina.

Innanzitutto quindi va definito il Dockerfile o più precisamente un file Docker-Compose che va a descrivere i servizi presenti all'interno dell'applicazione. In questo caso il Compose conterrà un unico servizio denominato *focasconnector* e utilizzerà un'immagine costruita partendo dal Dockerfile.

Nel Compose sono specificati ulteriori parametri che arricchiscono il comportamento del servizio:

- ***mem_limit*** impostato a 100 Mb. L'applicazione non potrà superare questa soglia nell'utilizzo della RAM.
- ***network*** chiamata proxy-redirect. Indica l'accesso da parte dell'applicazione alla rete intera dell'Edge Device.
- ***restart*** unless-stop permette nel caso di errore imprevisto durante l'esecuzione del servizio di fare il restart immediato.

È specificato inoltre la posizione del Dockerfile da utilizzare per costruire l'immagine che definisce il servizio focasconnector.

```
FROM python:3.8-slim@sha256:ed4bf3780773676c640a2a986bb68094afee2c6e77ae970bd5d60ddb246a5396 as base

COPY scripts/build-deps.sh /tmp/
RUN chmod +x /tmp/build-deps.sh
RUN /tmp/build-deps.sh

WORKDIR /usr/src/app

#Official FOCAS library
COPY Linux/x86/libfwlib32.so.1.0.1 /usr/libfwlib32.so
COPY Linux/fwlib32.h .

RUN pip3 install paho-mqtt

COPY ctypes/main.py .

CMD ["python3", "main.py"]
```

Figura 10: Dockerfile FANUC Connector

Il Dockerfile definisce i passaggi che devono essere eseguiti per fare il build di un'immagine, il tutto partendo da una *Base Image* dichiarata nell'istruzione che inizia con "FROM". Avendo scelto Python come linguaggio di programmazione è utile partire da un'immagine che abbia già a disposizione tutte le librerie necessarie per eseguire codice Python. Fortunatamente su Docker Hub risiede una vasta gamma di immagini con tutte le caratteristiche desiderate. Per l'applicazione è stata scelta un'immagine chiamata **python:3.8-**

slim specificando inoltre alcuni dettagli aggiuntivi.

Infatti il nome viene accompagnato da quella che si può definire una "signature" chiamata **digest** che funge da identificatore per l'immagine. In questo caso è stata utilizzata per andare a specificare il tipo di architettura desiderata ovvero una a 32 bit. La libreria FANUC FOCAS impiegata è basata su architettura a 32 bit e quindi per funzionare correttamente necessitava di un'immagine con le medesime caratteristiche.

Successivamente viene lanciato uno script che va ad aggiungere all'immagine tutte le librerie C/C++ a cui fa riferimento la libreria FANUC FOCAS. Viene poi copiato effettivamente il file contenente le funzioni FOCAS e il suo header nel filesystem. Passando all'istruzione successiva viene installato il pacchetto paho-mqtt, modulo Python che definisce le classi per realizzare un cliente MQTT e per effettuare scambio di messaggi basato sul concetto di topic. Come ultimi due passi per la costruzione dell'immagine viene copiato il file contenente il codice sorgente dell'applicazione e viene lanciato il comando per eseguirlo. Una volta illustrati i files per la composizione del Docker container si può passare al contenuto del codice sorgente che definisce il comportamento dell'applicazione. Il codice è contenuto all'interno di un file chiamato semplicemente main.py dentro al quale viene fatto il binding alla libreria FANUC FOCAS e mediante invocazioni cicliche vengono estratti i dati dalle macchine utensili per poi inviarli tramite MQTT.

Prima di esporre la logica del programma è necessario mostrare come viene fatto il binding della libreria FANUC FOCAS. Il modulo ctypes mette a disposizione una funzione *LoadLibrary* che permette, passando come argomento il path della libreria, di caricarla e di creare un oggetto Python che rappresenti FANUC FOCAS.

```
libpath = (  
    # ../../libfwlib32.so  
    Path.cwd().parents[1] / "libfwlib32.so"  
)  
  
ctypes.CDLL("/usr/lib/i386-linux-gnu/libstdc++.so.6", mode=ctypes.RTLD_GLOBAL)  
  
# loading of libfwlib32.so  
focas = ctypes.cdll.LoadLibrary(libpath)  
focas.cnc_startupprocess.restype = ctypes.c_short  
focas.cnc_exitprocess.restype = ctypes.c_short  
focas.cnc_allclibhndl3.restype = ctypes.c_short  
focas.cnc_freelibhndl1.restype = ctypes.c_short  
focas.cnc_rdcncid.restype = ctypes.c_short
```

Figura 11: LoadLibrary Code

Utilizzando questo oggetto è possibile invocare le funzioni della libreria, ricordo scritta in C++, da codice Python. Prima di poterle utilizzare però vanno impostati due parametri denominati `restype` e `argtypes` che rispettivamente vanno ad impostare il tipo di dato di ritorno dalla funzione e il tipo dei dati passati come argomento. Il tipo di dato passato deve essere uno dei C Data types definiti all'interno del modulo `ctypes` per comunicare come fare il casting del dato.

Come ulteriore preparazione precedente all'esecuzione del main program vengono definite le strutture dati passate durante l'invocazione di funzioni FANUC FOCAS. Nella libreria in C corrispondono a struct mentre qui devono essere rimappate su oggetti Python. Quindi per ogni struttura dati definita nella libreria che si ha intenzione di usare, sarà predisposta una classe secondo la politica dettata da `ctypes`. In figura possiamo avere un esempio delle classi inizializzate nel file Python:

```
class FAXIS(ctypes.Structure):
    _fields_ = [("absolute", ctypes.c_long * MAX_AXIS),
               ("machine", ctypes.c_long * MAX_AXIS),
               ("relative", ctypes.c_long * MAX_AXIS),
               ("distance", ctypes.c_long * MAX_AXIS)]

class OAXIS(ctypes.Structure):
    _fields_ = [("absolute", ctypes.c_long),
               ("machine", ctypes.c_long),
               ("relative", ctypes.c_long),
               ("distance", ctypes.c_long)]
```

Figura 12: Wrapper Python Class

Per quanto riguarda il corpo del main program il suo compito è quello di agganciarsi e connettersi ai FANUC CNC tramite indirizzo IP e invocare un set di funzione in linea con le esigenze dettate da Poggipolini.

`cnc_allclibhndl3` è la funzione invocata a inizio programma e che restituisce l'identificatore che rappresenta la libreria connessa ad un determinato terminale CNC. Quindi questa operazione deve essere ripetuta per tutte le macchine utensili. Ovviamente deve essere impostata correttamente la configurazione di ip e porta anche sul FANUC CNC.

```
lib_handles = []
names = []
for name, ip in config["ip"].items():

    port = 8193
    timeout = 10
    libh = ctypes.c_ushort(0)

    print(f"Connecting to machine at {ip}:{port}...")
    ret = focus.cnc_allcplibhnd13(
        ip.encode(),
        port,
        timeout,
        ctypes.byref(libh),
    )
    if ret != 0:
        raise Exception(f"Failed to connect to cnc! ({ret})")
    lib_handles.append(libh)
    names.append(name)
```

Figura 13: Library Binding Code

Una volta ottenute le librerie connesse ai CNC, l'oggetto in questione può essere utilizzato per comunicare con la macchina utensile ed effettuare letture dati oppure inviare comandi. Essendo lo use case interamente orientato verso il monitoraggio della macchina, in questa applicazione vengono utilizzate unicamente funzioni di sola lettura. I dati ricavati vengono poi inseriti in una apposita struttura dati per poi essere inviati. Il formato scelto per l'invio del messaggio è quello JSON quindi inizialmente i dati vengono ospitati all'interno di un dizionario Python per poi essere convertiti in un JSON Object e poi ancora in una stringa.

Le macchine utensili effettuano lavorazioni di grande precisione e utilizzano un sistema a 5 assi più un mandrino equipaggiabile con diversi strumenti. Ogni asse è controllato attraverso un motore di tipo servo e ogni motore della fresatrice è soggetto a un determinato tipo di sforzo. Il macchinario tiene anche traccia di parametri come il carico corrente di ogni motore oppure la velocità di rotazione del mandrino. Per ognuno di questi valori che si vuole conoscere e si vuole portare sull'Edge Device si necessita di invocare una funzione specifica della libreria nel modo seguente:


```
# status info read
stat_info = ODBST2()
ret = focus.cnc_statinfo2(libh, ctypes.byref(stat_info))
if ret != 0:
    raise Exception(f"Failed to read cnc stat info! ({ret})")

data_to_send["status"] = {
    "timestamp" : current_timestamp,
    "aut" : stat_info.aut,
    "run" : stat_info.run,
    "motion" : stat_info.motion,
    "mstb" : stat_info.mstb,
    "alarm" : stat_info.alarm,
    "edit" : stat_info.edit
}
```

Figura 14: Example Status Function

Si parte inizializzando la struttura dati che si vuole popolare nel caso mostrato qui sopra sarebbe "ODBST2". Successivamente viene invocata la funzione per la lettura dello stato della macchina passando per riferimento la struttura dati. Se la macchina è attiva ed è connessa alla rete allora a fine chiamata la struttura sarà popolata con i dati desiderati. I dati per semplicità vengono poi organizzati all'interno di un dizionario Python in attesa di essere convertiti in stringa JSON ed essere inviati.

3.3.1 Mapping Data Model

Uno degli obiettivi principali del progetto DEEPMON è la definizione di un Data model di riferimento per utilizzo industriale in modo da poter classificare e organizzare tutti i gli stabilimenti, macchinari e sensori di una singola impresa. La gerarchia parte dallo stabilimento andando poi ad elencare tutti i macchinari ospitati al suo interno.

Per ogni macchina è poi possibile definire tutta una serie di specifiche che vanno ad arricchire di informazioni il modello. Alcuni esempi potrebbero essere:

- **Stato funzionale:**
Riporta lo stato della macchina al momento dell'ultima lettura.
- **Gruppo funzionale:**
Elenca una lista di sensori che si occupano della misurazione di determinati valori.
- **Ricetta:**
Tipo di dato che definisce i materiali necessari ad un macchinario per poter eseguire determinate lavorazioni.
- **Contatori:**
Valori che sono pensati per mantenere traccia dell'incremento di determinati parametri, come ad esempio il numero di pezzi prodotti.
- **Eventi:**
Elenca tutti i possibili eventi che possono essere lanciati dalla macchina, pensati principalmente per gli allarmi.

Il modello è molto flessibile e garantisce sufficiente generalizzazione per poter essere usato in ambiti industriali anche differenti tra di loro. DEEPMON si occupa della validazione di tale principio e per questo ha scelto diversi use case aziendali messi a disposizione dalle imprese che partecipano al progetto. Uno di questi use case è per l'appunto quello di Poggipolini.

Quindi per questa attività è stato necessario trovare un modo per conformare i dati estratti dalle macchine utensili al Data model pensato da DEEPMON. L'idea di fondo è stata quella di mappare ogni singolo parametro letto dalla macchina, come lettura da parte di un sensore facente parte dello stesso gruppo funzionale. Questa soluzione permette di non snaturare eccessivamente la struttura dei dati letti e allo stesso tempo mantenere uniformità con il Data model di riferimento.

Il JSON Object conterrà informazioni riguardanti la macchina e una lista di letture dati per ogni valore di cui si vuole fare il monitoraggio.

Come si può notare dalla figura che segue, il file JSON contiene alcune delle informazioni che abbiamo citato nel Data model, come ad esempio lo stato funzionale della macchina(status) oppure le letture da parte degli ipotetici

sensori raccolte in "readings".

```
{
  "created" : 1634215367,
  "device" : "FANUC CNC Controller",
  "id" : "4c7b6d01-c9812975-5e2cc459-3492c97e",
  "status" : {
    "timestamp" : 1634215367,
    "aut" : 1,
    "run" : 3,
    "motion" : 1,
    "mstb" : 0,
    "alarm" : 0,
    "edit" : 0
  },
  "readings" : [
    {
      "created" : 1634215367,
      "device" : "FANUC CNC Controller",
      "name" : "Axis_1_load_meter",
      "value" : "22"
    },
    {
      "created" : 1634215367,
      "device" : "FANUC CNC Controller",
      "name" : "Axis_2_load_meter",
      "value" : "9"
    }
  ]
}
```

Figura 15: Struttura JSON

3.3.2 MQTT Client

Applicazioni installate sul medesimo Edge Device possono interagire tra di loro connettendosi alla rete interna predisposta dalla piattaforma, denominata *proxy-redirect*. In ascolto su questa stessa rete è presente un'applicazione chiamata IE Databus che si occupa di svolgere le funzione di un MQTT Broker rimanendo in attesa di ricevere messaggi. A seconda del topic a cui sono inviati i messaggi, il broker si occuperà di notificare le applicazioni interessate.

Quindi per poter partecipare allo scambio di messaggi la nostra applicazione dovrà integrare un modulo che permetta la definizione di oggetti e funzioni con cui è possibile definire un MQTT Client. Il modulo Python utilizzato è paho-mqtt che permette la creazione di un oggetto "client" in grado di effettuare operazioni come connessione al broker o invio di messaggi.

Per garantire la corretta trasmissione del messaggio l'applicazione dovrà conoscere a priori delle informazioni specifiche:

- Indirizzo e/o nome del broker MQTT

- Porta su cui è in ascolto il broker
- Topic di destinazione
- Nome utente e password per accedere ai Topic

Le informazioni appena elencate sono valori che vanno impostati in precedenza utilizzando lo IE Databus Configurator dal Management System. In questo modo è possibile definire le tipologie di Topic gestite, la loro organizzazione gerarchica e gli utenti che ne hanno accesso.

Il messaggio prima di essere inviato viene convertito in una stringa JSON e inserito nel payload. Così facendo il messaggio contenente i dati delle macchine utensili potrà raggiungere qualsiasi altra applicazione all'interno del dispositivo.

3.3.3 Configuration file

Al momento dell'installazione di un'applicazione su Industrial Edge è possibile aggiungere come opzione un file di configurazione. Il file viene letto al momento dell'installazione o più in generale ad ogni restart del servizio. All'interno di questo file possono essere contenute informazioni che pilotino il comportamento dell'applicazione. In generale viene usato questo approccio quando è possibile che in futuro alcuni parametri possano variare. Così facendo l'esistenza del file di configurazione permette di comunicare questi cambiamenti all'App senza dover mettere mano direttamente al codice. I file di configurazione possono essere scritti direttamente sulla GUI del Management System e farne il deploy sul device desiderato.

Tale dinamica permette la maggior flessibilità e adattabilità delle applicazioni Edge e in più contribuisce ad aiutare gli utenti finali ad apportare modifiche senza l'intervento di sviluppatori.

Per l'applicazione Connettore FANUC il file di configurazione è stato utilizzato per impostare frequenza di campionamento, indirizzi ip delle macchine utensili e mqtt topic.

3.4 Flow

A questo punto i dati delle macchine utensili sono già stati correttamente estratti e sono ottenibili all'interno della piattaforma Siemens. I dati sono

estratti ciclicamente con una frequenza di campionamento variabile a seconda delle esigenze, andando a formare un flusso continuo di dati. Per poter sfruttare al meglio la mole di dati prodotta è necessario implementare un qualche tipo di meccanismo di persistenza del dato. Il catalogo di Siemens offre già una valida soluzione sotto il nome di Data Service, applicazione per effettuare operazioni di storage.

Si è scelto però di sfruttare l'occasione per testare soluzioni diverse e verificare il comportamento della piattaforma a fronte di simili situazioni. Lo strumento di persistenza scelto è un'applicazione che fornisce un Web Service su cui è possibile invocare determinati comandi di persistenza. Il software verrà approfondito nella sezione successiva mentre in quella corrente verrà illustrato come inviare i dati al suddetto Web Service.

Per riuscire ad inviare correttamente i dati al Web Service è stato impiegato IE Flow Creator per definire un *Flow* di ricezione ed invio dati. Il programma è stato costruito utilizzando la semantica "a blocchi" di Node-RED, dove ogni rettangolino corrisponde ad un blocco funzionale con input ed output. Tramite l'interfaccia grafica è stato possibile definire il seguente schema:

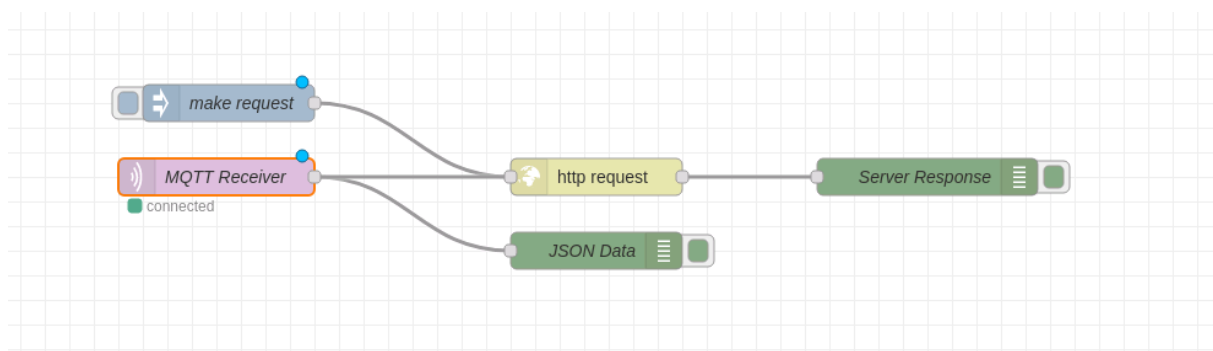


Figura 16: Workflow on Flow Creator

Sorvolando sui blocchi ausiliari che svolgono un ruolo di supporto nello sviluppo del programma, il Flow è composto da due nodi principali:

- **MQTT Receiver:**

Nodo che riceve in input tutti i messaggi che vengono pubblicati sotto un determinato topic. In questo caso il topic è esclusivamente utilizzato per l'invio di letture dati da parte de Connettore FANUC. L'MQTT Re-

ceiver esegue l'operazione di subscribe nei confronti del broker MQTT e ottiene l'accesso al flusso di dati delle macchine utensili.

- **HTTP Request:**

Riceve i messaggi ricevuti dall'MQTT Receiver e gli inserisce all'interno del body di una HTTP request. In particolare l'operazione da effettuare è una POST per informare il Web Service di voler fare la persistenza del contenuto del body.

Questo semplice strumento permette all'utente di aggiungere diversi comportamenti anche senza dover sviluppare un'applicazione ad-hoc. In questo caso ci siamo concentrati sull'elaborazione del dato in locale quindi rimanendo a livello OT, ma nulla ci vieta di voler eseguire l'analisi del dato ad un livello più alto come può essere quello IT, utilizzando il Cloud.

In generale Flow Creator può essere utilizzato anche per eseguire codice Javascript andando a definire programmi più complessi e strutturati che possono anche allontanarsi dalla semplice visualizzazione del dato.

3.5 Web Storage Service

Come già ribadito per effettuare operazioni di monitoraggio del dato è necessario avere un qualche sistema di persistenza, andando ad inserire i dati all'interno di un database. Per questo particolare aspetto è stato deciso di non far affidamento su software proprietari Siemens, bensì l'idea è stata quella di sfruttare un'applicazione sviluppata dal collaboratore universitario Riccardo Venanzi per il progetto DEEPMON.

Il software è stato sviluppato come componente del modello pensato da DEEPMON per l'applicazione di Edge Computing in ambito industriale.

Il componente è stato progettato e sviluppato utilizzando Node.js andando a definire un Web Service che mette a disposizione una serie di REST API per invocare le operazioni CRUD caratteristiche dei database. L'utilizzatore di questo servizio ha visibilità del solo punto di accesso che è definito dalle REST API potendo interagire con esso solo inviando HTTP request.

Internamente al Web Service come Database di riferimento è stato scelto MongoDB. MongoDB è un Database Management System non relazionale ma bensì orientato a documenti. MongoDB si allontana dalla struttura tradizionale basata su tabelle dei database relazionali in favore di documenti in

stile JSON venendo così definito NoSQL.

La persistenza basata su documenti JSON rende l'applicazione molto compatibile con lo use case di Poggipolini in quanto fino ad ora i dati delle macchine utensili sono stati trasmessi utilizzando lo stesso formato. In questo modo è bastato utilizzare IE Flow Creator per invocare la REST API corretta per fare l'insert del documento. L'operazione avviene in maniera trasparente senza effettivamente sapere il come viene applicata sul Database Mongo lasciando anche la possibilità in futuro di poter utilizzare un altro servizio di persistenza come base.

3.6 Sistema di Visualizzazione

All'inizio dell'attività riguardante il caso Poggipolini il progetto prevedeva la lettura di dati macchina e la semplice persistenza di tali informazioni. Durante il percorso ci si è focalizzati principalmente sull'esplorazione di Siemens Industrial Edge come piattaforma e sullo sviluppo del Connettore FANUC.

Si è pensato però di implementare un qualche tipo di strumento di visualizzazione grafica per aver un riscontro visivo dei dati. L'idea è quella di utilizzare un software sviluppato direttamente da MongoDB chiamato MongoDB Charts. L'applicazione permette di scegliere un Data source su cui poi è possibile costruire diversi tipi di dashboard. Per ogni dashboard è possibile scegliere tra un'ampia gamma di grafici da inserire, in maniera da personalizzare a proprio piacimento l'interfaccia. Inoltre si possono specificare diverse opzioni come la frequenza di aggiornamento che può essere di minuti o di ore. L'applicazione è ottenibile gratuitamente ed è sviluppata appositamente per funzionare in combinazione con MongoDB. Questi due aspetti si sono mostrati decisivi nella scelta dello strumento rispetto ad altre alternative come Grafana.

Per portare questo software all'interno della piattaforma Siemens è stato deciso di sviluppare un Docker container con relativo file Docker-Compose del quale poi fare il deploy sull'Edge Device. L'applicazione mette a disposizione un'interfaccia grafica accessibile da browser, la quale può essere utilizzata per definire grafici ad-hoc.

Al momento della stesura di questo elaborato il container è ancora in fase di sviluppo e sperimentazione e quindi non è stato possibile integrarlo completamente nel sistema, rendendo l'utilizzo di MongoDB Charts uno svi-

luppo futuro dell'architettura.

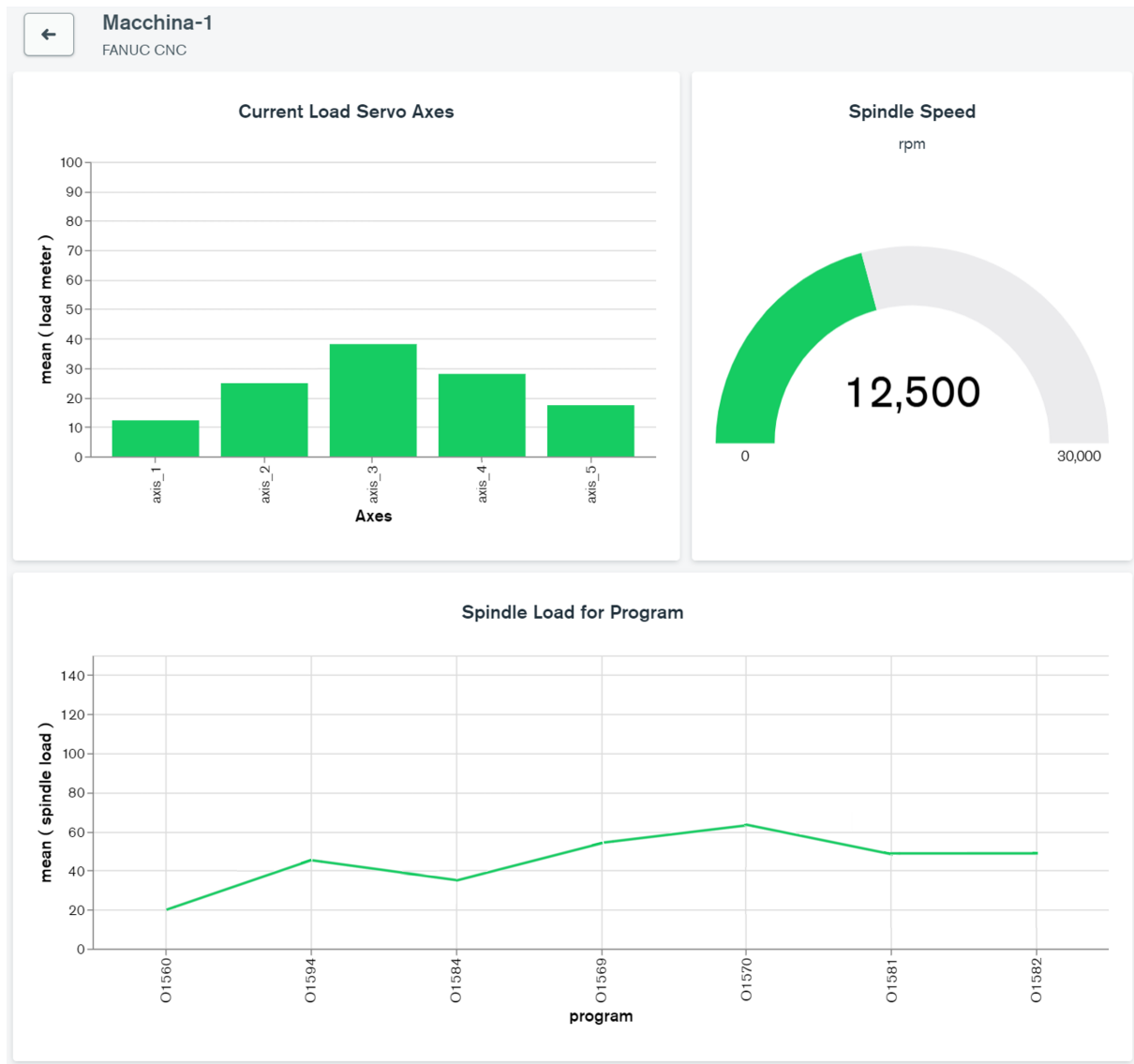


Figura 17: MongoDB Charts Dashboard

La figura qui sopra mostra comunque un esempio di dashboard realizzabile con MongoDB Charts, al cui interno sono stati costruiti alcuni grafici per il monitoraggio della macchina. Quelli mostrati sono solo alcuni esempi ma possono già dare un'idea di quello che può essere ottenuto con questo strumento.

Capitolo 4

Validazione del sistema

Finora è stata discussa e spiegata la parte che possiamo definire di sviluppo e setup dell'architettura. Sono state svolte sia operazioni di natura "meccanica" come l'installazione del nodo Edge fisico, sia di natura puramente software come lo sviluppo del connettore e la gestione complessiva della piattaforma Siemens Industrial Edge.

4.1 Valutazione Siemens Industrial Edge

Le sperimentazioni effettuate per questo use case non si concentrano unicamente sul funzionamento delle singole componenti software ma anche sul comportamento di Siemens Industrial Edge come piattaforma per l'Edge Computing. La valutazione del sistema comprende anche la parte di sviluppo software, essendo un fattore decisionale per le aziende che vogliono integrare l'Edge Computing nei propri stabilimenti.

La grande varietà di soluzioni che offre la piattaforma è sicuramente uno dei suoi punti forti. Già al momento dell'installazione esiste una vasta gamma di applicazioni scaricabili attraverso il catalogo. Inoltre nel caso si necessitino funzioni più specifiche esiste la possibilità di svilupparle indipendentemente. L'utilizzo di uno strumento come Industrial Edge App Publisher assiste notevolmente l'utente nello sviluppo di tali applicazioni occupandosi di tutti gli aspetti che riguardano la compatibilità con il sistema Siemens. In pochi minuti è possibile prendere un'immagine Docker, farne l'upload sul Management System e poi farne il deploy sui diversi nodi Edge. Procedimento che sottolinea la semplicità di sviluppo e la capacità di personalizzazione di software custom. Il fatto di poter poi gestire da un solo terminale (Industrial Edge Management) tutte le Edge Apps e dispositivi all'interno dell'impianto industriale contribuisce notevolmente alla scalabilità della piattaforma.

Sul fronte sicurezza, Industrial Edge è gestito tramite un accesso protetto da utente e password. Ogni qualvolta si vogliono effettuare operazioni o semplicemente accedere al sistema, si necessita l'inserimento delle credenziali. Credenziali definite e gestite dall'amministratore del sistema avendo anche la possibilità di definire gruppi di lavoro, ruoli e assegnare differenti diritti di accesso ad ogni utente. In questo modo l'azienda sfruttando Industrial Edge è in grado di mettere in atto uno stretto controllo sulle azioni eseguiti da parte di ogni singolo dipendente.

4.2 Test Connettore FANUC

Trattando l'argomento Connettore FANUC è possibile capire appieno le potenzialità della piattaforma agli occhi dell'usufruitore finale. L'esigenza del dover comunicare con una macchina utensile utilizzando un protocollo non coperto dalle opzioni Siemens, si è tradotto nella realizzazione indipendente del componente. Dopo avere superato la fase di sviluppo il connettore è stato installato sull'Edge Device ed è stato messo in esecuzione per verificare l'effettiva funzionalità dell'applicazione. Il software nella sua versione finale occupa su disco una dimensione di circa 500 MB che in proporzione ai 200 GB disponibili sul dispositivo è un risultato abbastanza soddisfacente, dimostrando anche la possibilità di un'esecuzione contemporanea di molteplici applicazioni. Il consumo di memoria RAM è minimo, sull'ordine delle decine di MB. Risultato in un qualche modo predicibile dato il contenuto del codice. Il connettore comunica con le tre macchine utensili contemporaneamente con la possibilità di aggiungere più unità usando il file di configurazione. Il flusso di dati prodotto dall'applicazione è consistente e stabile con i messaggi etichettati correttamente con il timestamp del momento della lettura. I valori selezionati ed estratti dalle macchine utensili sono i seguenti:

- Per ogni asse controllabile:
 - **name:** Nome e numero dell'asse
 - **abs:** Posizione assoluta dell'asse
 - **rel:** Posizione relativa dell'asse
 - **mach:** Posizione della macchina
 - **dist:** Distanza tra posizione assoluta e posizione di destinazione
 - **load meter:** Percentuale di carico sul motore servo

- **load current:** Valore in ampere del carico sul motore servo

- Per ogni spindle(mandrino) della macchina:
 - **load meter:** Percentuale di carico sul motore del mandrino
 - **motor speed:** Velocità in giri al minuto(rpm) del mandrino

- **feed rate:** Indice del consumo del materiale da lavorare
- **main program:** identificativo del programma in esecuzione
- **subprogram:** identificativo del sottoprogramma eseguito dal main program
- Status della macchina:
 - **aut:** Indica la selezione della modalità Automatica/Manuale
 - **run:** Indica lo stato funzionale della macchina (STOP, HOLD, RUN)
 - **motion:** Indica lo spostamento degli assi
 - **alarm:** Comunica lo stato in cui è stato lanciato un'allarme
 - **edit:** Comunica lo stato della macchina mentre è nella modalità editing

- **timestamp:** valore temporale per identificare il momento della lettura

Oltre ai dati appena mostrati vengono aggiunte anche informazioni relative alla macchina utensile vera e propria come ad esempio il suo nome ed identificativo. L'insieme di queste informazione viene organizzato all'interno dell'applicazione seguendo le linee guida dettate dal Data model di riferimento.

4.3 Efficacia Data model

Uno degli scopi del progetto DEEPMON è quello di utilizzare gli use case industriali per vedere l'effettiva applicabilità del Data model. Il Data model di riferimento è stato definito in maniera ampia e generale in modo da garantire l'adattabilità alle diverse situazioni industriali.

Nello use case di Poggipolini il modello non pone particolari problemi, permettendo un mapping agevole dei dati estratti dai CNC. Una criticità che

può essere mossa non nei confronti del modello in sé ma del come è avvenuto il mapping è la ridondanza di alcuni dati. Avendo mappato ogni lettura di parametro come sensore diverso facente parte dello stesso gruppo funzionale, alcuni valori come il timestamp vengono ripetuti inutilmente. La lettura dei parametri della macchina avviene nel medesimo istante il che porta ad un timestamp identico. Questo piccolo overhead del messaggio è un aspetto che può essere sicuramente migliorato in futuro.

```
{
  "created" : 1634215367,
  "device" : "FANUC CNC Controller",
  "id" : "4c7b6d01-c9812975-5e2cc459-3492c97e",
  "status" : {
    "timestamp" : 1634215367,
    "aut" : 1,
    "run" : 3,
    "motion" : 1,
    "mstb" : 0,
    "alarm" : 0,
    "edit" : 0
  },
  "readings" : [
    {
      "created" : 1634215367,
      "device" : "FANUC CNC Controller",
      "name" : "Axis_1_load_meter",
      "value" : "22"
    },
    {
      "created" : 1634215367,
      "device" : "FANUC CNC Controller",
      "name" : "Axis_2_load_meter",
      "value" : "9"
    }
  ]
}
```

Figura 18: Mapping Data Model

4.4 Funzionamento del sistema

Mettendo da parte le valutazioni riguardanti il Data model si può passare a discutere della cooperazione tra le applicazioni e la comunicazione all'interno dello stesso dispositivo. Il prodotto Siemens impiegato per il progetto è pensato per ospitare al suo interno una moltitudine di applicazioni e mette a disposizione gli strumenti necessari per supportare l'interazione tra di esse. Nell'Edge Device è presente la rete interna proxy-redirect sui cui è in ascolto l'IE Databus. Questa applicazione funge da "collante" tra le diverse Edge Apps permettendo lo scambio di messaggi veloce e senza andare ad incide-

re significativamente sulle performance della piattaforma. Quando si parla di Edge Computing la comunicazione macchina-dispositivo e applicazione-applicazione sono aspetti molto importanti, anche perché la rapidità di trasferimento dati è uno dei fattori decisionali per scegliere se utilizzare Edge o Cloud.

L'interazione tra applicazioni è stata resa possibile tramite l'utilizzo di IE Databus e IE Flow Creator permettendo al Connettore FANUC di comunicare i dati estratti dai sistemi di automazione al Web Storage Service. Il servizio installato direttamente sul dispositivo ad ogni ricezione del dato va ad inserire un documento JSON all'interno di MongoDB. Nel caso d'uso di Poggipolini le macchine utensili da monitorare sono tre e quindi il Connettore FANUC esegue un campionamento periodico su tutte e tre le macchine generando diversi file JSON per ogni lettura. Per ogni file JSON generato utilizzando IE Flow Creator viene inviata una HTTP request di tipo POST verso il Web Service. Il servizio sempre in ascolto, riceve le HTTP request e fa persistenza del dato su MongoDB. Il vantaggio di aver utilizzato un Web Service come strumento di persistenza del dato è la possibilità di accesso da parte di altre entità come possono essere software in esecuzione a livello IT. Infatti predisponendo una serie di REST API per dialogare con il servizio, qualsiasi applicazione in grado di fare HTTP request è in grado di interrogare il Database.

Ovviamente si parla sempre di applicazioni interne alla rete aziendale e quindi protette, inoltre vengono comunque messi in atto tutti i sistemi di sicurezza caratteristici della piattaforma Siemens.

Capitolo 5

Conclusione e sviluppi futuri

L'attività svolta presso Poggipolini è servita per testare ed esplorare soluzioni Edge scalabili applicate ad impianti industriali. L'integrazione di PC industriali e di sistemi come Siemens Industrial Edge hanno permesso di verificare quello che può essere ottenuto quando si fanno interagire i sistemi di automazione tradizionali con software e strumenti di digitalizzazione. Dimostrando inoltre la scalabilità e estensibilità delle funzioni applicabili ad una linea di produzione anche se in esecuzione da diversi anni.

Quello che è stato ottenuto all'interno di questa esperienza è stata l'installazione di un nodo Edge in prossimità della linea di produzione, la connessione alle macchine utensili, l'interfacciamento ai FANUC CNC e lo sviluppo di Edge Apps per arricchire il processo produttivo.

Le applicazioni sviluppate e/o semplicemente integrate nel sistema hanno reso possibili operazioni come la lettura dello stato interno della macchina, lo scambio di messaggi tra applicazioni e la persistenza del dato. Inoltre sono state gettate le basi anche per un futuro sistema di monitoraggio dati quale MongoDB Charts.

Il progetto DEEPMON ha come obiettivo la realizzazione di una soluzione Edge agnostica al tipo di piattaforma sottostante (Siemens Industrial Edge, EdgeX Foundry) e perciò è stato utilizzato il caso Poggipolini per esplorare e mettere in pratica le potenzialità di un simile sistema.

In DEEPMON è nata anche l'idea di proporre un modello per l'organizzazione dei dati interni ad un'impresa e quindi è stata sfruttata l'attività per verificare se il Data model pensato riusciva ad adattarsi alle esigenze aziendali. A fine tesi possiamo confermare che la mappatura dei dati estratti dalle macchine utensili è avvenuta senza problemi dimostrando la flessibilità del modello e il suo potere di generalizzazione.

Per quanto riguarda lo sviluppo del Connettore FANUC si è stati in grado di effettuare prove di upload e deploy di applicazioni custom attraverso l'Industrial Edge App Publisher e il Management System. Tutto questo ha permesso di confermare nuovamente quanto sia immediato installare dei software sui dispositivi Edge da remoto. L'applicazione in sé si è dimostrata funzionante riuscendo ad interfacciarsi correttamente al FANUC CNC ed ad utilizzare a dovere la libreria FANUC FOCAS. Il resto delle applicazioni sono state in grado di comunicare tra di loro portando poi alla finale persistenza del dato su MongoDB.

Come possibili sviluppi futuri, avendo predisposto già una piattaforma per l'Edge Computing, vi è sicuramente la possibilità di aggiungere nuove funzionalità sotto forma di Edge Apps. Alcuni esempi potrebbero essere algoritmi di manutenzione predittiva, elaborazione dati locale o il già nominato monitoraggio dei dati. In aggiunta si può pensare di estendere la connessione al nodo Edge, a tutta la linea di produzione e non solo alle fresatrici permettendo funzioni più avanzate come l'orchestrazione dei diversi sistemi di automazione.

Elenco delle figure

1.1	Figura 1: Architettura Internet of Things	7
1.2	Figura 2: Virtual Machines e Containers	17
2.1	Figura 3: Architettura Industrial Edge	26
2.2	Figura 4: Industrial Edge Databus	29
2.3	Figura 5: App Development Workflow	32
2.4	Figura 6: Mapping C-Python Data types	36
3.1	Figura 7: Collegamento Macchine Utensili	41
3.2	Figura 8: Software on SIMATIC Edge Device	43
3.3	Figura 9: Siemens Connector	44
3.4	Figura 10: Dockerfile FANUC Connector	45
3.5	Figura 11: LoadLibrary Code	46
3.6	Figura 12: Wrapper Python Class	47
3.7	Figura 13: Library Binding Code	48
3.8	Figura 14: Example Status Function	49
3.9	Figura 15: Struttura JSON	51
3.10	Figura 16: Workflow on Flow Creator	53
3.11	Figura 17: MongoDB Charts Dashboard	56
4.1	Figura 18: Mapping Data Model	60

Bibliografia

- [1] Mauro Bellini (2018), «*IoT (Internet of Things): significato, esempi e applicazioni pratiche*». Indirizzo: <https://www.internet4things.it/iot-library/internet-of-things-gli-ambiti-applicativi-in-italia/>
- [2] Jacob Morgan (2014), «*A Simple Explanation Of 'The Internet Of Things'*». Indirizzo: <https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/>
- [3] Margaret Rouse (2016), «*Internet of Things (IoT)*». Indirizzo: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
- [4] Paolo Subioli (2017), «*Introduzione ai protocolli per la IIOT (Industrial IOT)*». Indirizzo: <https://www.cadlog.it/2017/11/14/introduzione-ai-protocolli-la-iiot-industrial-iot/>
- [5] Annalisa Casali (2019), «*Edge Computing: cos'è, come funziona e che benefici ci sono per le aziende*». Indirizzo: <https://www.internet4things.it/edge-computing/edge-platform/edge-computing-cosa-e-benefici/>
- [6] Laura Zanotti (2021), «*Industria 4.0: storia, significato ed evoluzioni tecnologiche a vantaggio del business*». Indirizzo: <https://www.digital4.biz/executive/industria-40-storia-significato-ed-evoluzioni-tecnologiche/>
- [7] Manuale Siemens, «*Industrial Edge - App Developer Guide V1.2.1*».
- [8] Manuale Siemens, «*Industrial Edge Databus Configurator V1.1*».
- [9] Manuale Siemens, «*Industrial Edge Flow Creator V1.1.3*».
- [10] Manuale Siemens, «*Industrial Edge App Publisher V1.0.0*».
- [11] Manuale Siemens, «*Industrial Edge Management – Operation V1.0.0*».