ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
Corso di Laurea in Ingegneria e Scienze Informatiche

STRUMENTI DI VISUALIZZAZIONE E AMMINISTRAZIONE PER IL WEB OF DIGITAL TWINS

Elaborato in
SISTEMI EMBEDDED E INTERNET OF THINGS

Relatore
Prof. ALESSANDRO RICCI

Presentata da
MARGHERITA BALZONI

Corelatore
Dott. Ing. SAMUELE BURATTINI

Dott. Ing. ANDREA GIULIANELLI

Anno Accademico 2023 – 2024

Indice

Introduzione

Negli ultimi anni, il concetto di Digital Twin (DT) ha guadagnato crescente attenzione sia in ambito accademico che industriale. I Digital Twins rappresentano repliche digitali di asset fisici, progettate per monitorare, simulare e ottimizzare il loro funzionamento nel corso del ciclo di vita. Questo paradigma ha trovato applicazione in numerosi settori, tra cui la produzione industriale, la sanità e le smart city dimostrando un potenziale significativo nel migliorare l'efficienza operativa e il processo decisionale.

Tuttavia, nonostante l'ampia diffusione e l'evoluzione tecnologica, i DT sono stati spesso sviluppati come entità isolate, con una forte dipendenza da piattaforme verticali chiuse e tecnologie specifiche. Questa frammentazione ha portato alla creazione di silos tecnologici, rendendo difficile la creazione di ecosistemi interoperabili e limitando le opportunità di sfruttare pienamente il potenziale del paradigma. A fronte di questo scenario, emerge una nuova visione: il Web of Digital Twins che punta a costruire ecosistemi aperti, distribuiti e dinamici, dove i DT possono essere connessi tra loro e navigati come un unico sistema interconnesso.

Questa tesi si colloca nel contesto del WoDT, l'obiettivo principale è implementare uno strumento che faciliti la gestione e la visualizzazione degli ecosistemi di DT, sfruttando una piattaforma Hypermedia-based Web of Digital Twins. Questo strumento, progettato per semplificare l'interazione tra utenti e DT, mira a superare le sfide legate alla complessità dell'integrazione tecnologica e alla navigazione di ecosistemi composti da DT sviluppati con tecnologie diverse.

Nei primi due capitoli si affronta la parte teorica, fornendo una solida base concettuale necessaria per comprendere il lavoro svolto. Si parte dalla definizione del paradigma dei Digital Twin, descrivendone le proprietà, le caratteristiche principali e gli scenari di applicazione. Vengono inoltre analizzati gli strumenti esistenti per la gestione dei Digital Twin, evidenziandone vantaggi e limiti. Successivamente, si amplia la prospettiva passando a una visione ad ecosistema, introducendo la teoria del Web of Digital Twins. In questa sezione, vengono illustrate le caratteristiche chiave di questa visione, che punta alla creazione di ecosistemi aperti, interoperabili e dinamici di Digital Twin

interconnessi, superando i limiti delle soluzioni verticali e chiuse.Nel terzo capitolo verrà presentato il contributo principale di questo lavoro, attraverso la progettazione di uno strumento per facilitare la gestione di DT interconnessi.In questa sezione saranno introdotti i requisiti funzionali e tecnici, insieme a un'analisi approfondita delle motivazioni che hanno portato alla scelta di intraprendere questo progetto.Nel quarto capitolo, invece, si passerà a descrivere in dettaglio l'implementazione dello strumento, evidenziando le soluzioni adottate per soddisfare i requisiti progettuali. Successivamente, lo strumento verrà testato attraverso un caso di studio specifico, progettato per mettere in rilievo gli aspetti dinamici e complessi degli ecosistemi di digital twin, dimostrando così l'efficacia e la validità del lavoro svolto.

Capitolo 1

Il Concetto di Digital Twin e le sue Applicazioni

Questo capitolo esplora il concetto di Digital Twin, partendo dalle sue origini e sviluppando una comprensione completa delle sue caratteristiche tecniche e delle applicazioni pratiche.

1.1 Introduzione al Digital Twin

Digital Twin (DT) è una tecnologia emergente che sta guadagnando sempre più attenzione in vari settori, rivoluzionando il modo in cui interagiamo con gli oggetti all'interno del mondo fisico. Il Digital Twin opera creando una controparte software di un oggetto fisico che ne riflette le proprietà e le caratteristiche principali creando così un modello realistico in grado di simulare il comportamento dell'oggetto in uno specifico contesto di applicazione.

1.1.1 Origine del concetto

Il concetto di Digital Twin è stato presentato per la prima volta nel 2003 da Michael Grieves durante un corso all'Università del Michigan, dove propose un modello costituito da tre elementi fondamentali: il mondo fisico, il mondo virtuale e un meccanismo di comunicazione che consentisse lo scambio di dati tra i due [17]. Inizialmente, questo modello venne chiamato "Mirrored Spaces Model" [17]. Tuttavia, l'idea di rappresentare digitalmente la realtà fisica risaliva già al 1991, quando David Gelernter coniò il termine "Mirror Worlds" per descrivere un concetto simile, basato sull'uso di dati reali per creare modelli virtuali [17]. Nel 2003, Kary Främling propose un'architettura basata su agenti in cui ogni prodotto fisico aveva un corrispondente gemello virtuale [17] ed ha

identificato tre livelli di integrazione: il Digital Model, Digital Shadow e il Digital Twin vero e proprio (1.1) [2].

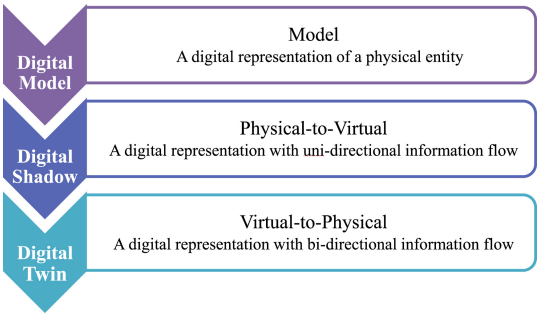

Figura 1.1: Livelli di integrazione [2]

Nel 2006, il nome del modello concettuale proposto da Grieves fu cambiato in "Mirrored Spaces Model" [17]. Il modello poneva l'accento sul meccanismo di collegamento tra i due spazi, che doveva essere bidirezionale e avere più spazi virtuali per un singolo spazio reale, dove potevano essere esplorate idee o progetti alternativi [17]. Tuttavia, a causa delle limitazioni tecnologiche dell'epoca (bassa potenza di calcolo, scarsa connettività Internet, e algoritmi poco sviluppati), il DT non aveva ancora applicazioni pratiche [17]. Il termine "Digital Twin" è apparso per la prima volta nella bozza della roadmap tecnologica della NASA nel 2010, dove era anche indicato come "Virtual Digital Fleet Leader" [17]. NASA fu la prima nel definire formalmente il concetto, descrivendolo come una simulazione multi-scala e multi-fisica di un sistema, capace di rispecchiare la vita del suo corrispettivo fisico [17]. Successivamente, l'aeronautica militare degli Stati Uniti adottò questa tecnologia per il monitoraggio e la manutenzione degli aerei [17].

Una delle prime applicazioni fu nel settore manifatturiero, dove l'idea di monitorare i macchinari e i processi produttivi attraverso una replica virtuale si dimostrò particolarmente utile [8]. Le aziende utilizzavano i Digital Twin per simulare il funzionamento delle macchine e prevedere eventuali guasti o inefficienze. Attraverso l'integrazione con i dati in tempo reale raccolti dai sensori posti sui macchinari, si poteva prevedere quando un macchinario avrebbe avuto bisogno di manutenzione o identificare difetti di produzione prima che causassero problemi [8]. Il suo utilizzo, si è poi ampliato, trovando applicazioni in ambiti come l'Internet of Things (IoT) e successivamente ai sistemi cyberfisici (CPS) [8]. Questa espansione ha portato ad una progressiva evoluzione della sua definizione. Il DT è passato, infatti, dall'essere solo una rappresentazione virtuale di un prodotto industriale ad una nozione più generale, applicabile ad una vasta gamma di oggetti fisici [8].

1.1.2 Definizione

il Digital Twin è definito come una rappresentazione virtuale di oggetti fisici che copre l'intero ciclo di vita dell'oggetto [2]. Queste rappresentazioni possono essere comprese, apprese e ragionate con dati in tempo reale o un modello di simulazione che acquisisce dati dal campo e attiva l'operazione di dispositivi fisici [2]. Ciò permette di ottimizzare le decisioni e fare previsioni.

È fondamentale mettere in risalto un primo gruppo di caratteristiche che definiscono i Digital Twins:

1. Un DT si riferisce strettamente a un Oggetto Fisico (PO).

2. Un DT contiene tutte le informazioni necessarie per caratterizzare completamente un PO e il suo comportamento previsto o desiderato.

3. Poiché il DT è inquadrato in un ciclo di vita composto da diverse fasi, può includere dati e informazioni che descrivono la "storia" del PO [8].

L'idea originale di Grieves era quella di migliorare l'intero processo produttivo attraverso una copia digitale del prodotto, che potesse riflettere in tempo reale lo stato dell'oggetto fisico e fornire supporto per analisi delle prestazioni e la previsione di problemi [8]. Questa rappresentazione virtuale non si basa su informazioni statiche, ma è dinamica, è quindi in grado di interagire con il mondo fisico e ricavare informazioni utili da esso in tempo reale, migliorando così la qualità delle decisioni e previsioni [8]. L'aspetto innovativo dei DT è, dunque, la loro capacità di evolvere insieme al prodotto fisico [8]. Questa interazione continua tra mondo fisico e virtuale permette di poter intervenire tempestivamente per migliorare le prestazioni e risolvere problemi prima che essi possano verificarsi [8].

1.1.3 Il ruolo delle tecnologie IoT

L'Internet of Things (IoT) trasforma una rete di oggetti smart connessi in sistemi intelligenti dotati di interfacce di programmazione delle applicazioni (API) ben strutturate e interfacce utente intuitive. Questi sistemi sono in grado di comprendere e rispondere agli ambienti fisici grazie alla connettività dei dispositivi intelligenti e allo scambio di dati tramite Internet. Secondo le ricerche di mercato, entro il 2025 si prevede che il numero di dispositivi IoT connessi raggiunga circa 75 miliardi, creando così opportunità per lo sviluppo di numerose nuove applicazioni e servizi in vari settori [16]. Nei Digital Twins, l'IoT rappresenta la tecnologia fondamentale per ogni applicazione, poiché consente la raccolta di dati da oggetti fisici attraverso sensori. Questi sensori trasmettono informazioni in tempo reale, che vengono utilizzate per

creare una duplicazione digitale di un oggetto fisico. L'IoT svolge un ruolo cruciale nell'aggiornare continuamente i dati, permettendo ai Digital Twin di fornire una rappresentazione virtuale accurata e in tempo reale dell'oggetto fisico. Grazie ad esso, le applicazioni dei Digital Twins possono monitorare costantemente le prestazioni degli oggetti e fornire informazioni preziose per l'ottimizzazione e la manutenzione predittiva [2].

1.1.4 Altre teconologie

Oltre all'IoT ci sono altre tecnologie che sono utilizzate per raccogliere e gestire dati:

- **Cloud Computing:** Consente ai Digital Twins di memorizzare e accedere a grandi quantità di dati tramite la rete. Grazie al cloud, è possibile archiviare dati nel cloud virtuale e accedere facilmente alle informazioni da qualsiasi posizione, riducendo i tempi di calcolo e semplificando la gestione di sistemi complessi.

- **Intelligenza Artificiale (AI):** L'AI fornisce ai Digital Twins strumenti analitici avanzati, come apprendimento automatico e reti neurali, per analizzare dati in tempo reale, fare previsioni, e offrire suggerimenti su come ottimizzare processi e prevenire problemi.

- **Extended Reality (XR):** Tecnologie immersive come Realtà Virtuale (VR), Aumentata (AR) e Mista (MR) permettono di creare rappresentazioni digitali con cui gli utenti possono interagire in tempo reale. L'XR consente di visualizzare e manipolare versioni digitali degli oggetti fisici, offrendo un'interazione più intuitiva e coinvolgente tra il mondo reale e quello virtuale [2].

1.2 Caratteristiche fondamentali nel contesto IoT

Il concetto di DT si distingue per una serie di caratteristiche fondamentali che ne determinano il funzionamento, le capacità l'adattabilità a diversi contesti applicativi. queste caratteristiche rendono il DT uno strumento utile per la simulazione, previsione e l'ottimizzazione dei sistemi complessi. In questa sezione, esploreremo le principali caratteristiche del DT [8]. .

Identità: Un aspetto fondamentale del DT è la necessità di avere un'identità univoca sia per l'oggetto fisico sia per la sua controparte digitale. un DT deve essere identificabile tramite un codice che colleghi l'oggetto fisco al corrispondente oggetto logico. La relazione tra PO e DT non sempre è una relazione di cardinalità 1 a 1 ma può anche essere 1 a N nel caso in cui esistano diverse repliche dell'oggetto fisico, in questo caso è necessario che ciascuna replica del PO abbia un identificatore unico e un puntatore all'identificatore del PO. Il DT rappresenta il PO nel tempo e nello spazio, consentendo di avere repliche che riflettono lo stato passato, presente o futuro dell'oggetto fisico, ognuna con una propria identità e contesto temporale o spaziale specifico. Queste informazioni, legate a tempo e posizione, sono fondamentali per identificare e gestire ciascuna replica [8].

Rappresentatività e Contestualizzazione: Un aspetto cruciale del Digital Twin è quanto accuratamente la replica digitale riesca a rispecchiare l'oggetto fisico. Sebbene possa essere difficile raggiungere una rappresentazione perfetta e completa del PO in tutte le sue sfaccettature, è importante che il DT catturi almeno le proprietà, le caratteristiche e i comportamenti fondamentali che definiscono il comportamento dell'oggetto fisico nelle condizioni di interesse [8].

La rappresentatività può essere valutata secondo tre parametri principali:

- **Somiglianza:** Rappresenta il grado in cui il DT riproduce accuratamnete l'oggetto originale , il suo stato e le sue caratteristiche.
- **Casualità:** Indica la probabilità che il DT possa mostrare uno stato diverso o comportamenti divergenti rispetto al PO.
- **Contestualizzazione:** l'accuratezza della rappresentazione deve essere valutata nel contesto specifico in cui il DT è utilizzato. Non tutte le caratteristiche del PO sono rilevanti per ogni applicazione: alcuni attributi possono essere omessi se non influenzano direttamente il comportamento dell'oggetto nell'ambiente virtuale considerato [8].

Riflessione: Per garantire una rappresentazione completa del PO, il DT deve essere in grado di descrivere con precisione tutti gli attributi, gli stati, i comportamenti, i dati e gli eventi ad esso associati. La proprietà di riflessione implica che il DT offre una rappresentazione accurata del PO in relazione al contesto di utilizzo. Il PO, attraverso il DT, viene descritto come un insieme di valori che possono evolversi nel tempo, riflettendo i cambiamenti nei suoi stati e comportamenti. Ogni valore del PO deve essere esattamente corrispondente all'oggetto virtuale che lo rappresenta [8].

Replicazione: La replicazione si riferisce alla capacità di replicare un oggetto all'interno di un diverso ambiente, creando una versione virtuale del PO. Questo processo consente che il PO sia virtualizzato e riprodotto più volte all'interno di uno spazio di simulazione, e il DT stesso può essere replicato più volte in contesti diversi, adattandosi a specifiche necessità operative [8].

Entanglement: L'entanglement descrive la connessione tra il PO e il suo gemello digitale, garantendo che ogni modifica nel PO si rifletta istantaneamente nel DT, e viceversa, permettendo alle applicazioni e ai servizi di accedere rapidamente alle informazioni [8]. Le principali caratteristiche di questa connessione sono:

- **Connettività:** È necessario che esista un canale di comunicazione, diretto o indiretto, che consenta il trasferimento dei dati e delle modifiche tra il PO e il DT. Se il PO ha le capacità di trasmettere e ricevere dati, queste informazioni possono essere inviate direttamente al DT. Se il PO non ha tale capacità, la comunicazione avviene tramite altri oggetti che monitorano lo stato del PO e inviano i dati al DT [8].

- **Prontezza:** Il tempo di trasmissione delle informazioni tra il PO e il DT deve essere ridotto al minimo per garantire che i cambiamenti siano riflessi in tempo utile rispetto alle necessità delle applicazioni. Alcune applicazioni potrebbero richiedere aggiornamenti in tempo reale, mentre altre potrebbero tollerare ritardi più lunghi [8].

- **Associazione:** La relazione tra PO e DT può essere unidirezionale, con il PO che invia dati al DT, o bidirezionale, in cui i due oggetti scambiano informazioni in modo continuo. Nei casi in cui il PO sia dotato di sensori o attuatori, una comunicazione bidirezionale diventa particolarmente vantaggiosa, poiché il DT può non solo ricevere dati ma anche inviare comandi per migliorare le prestazioni del PO [8].

Persistenza: Questa propietà implica che il DT debba rimanere attivo e funzionale per tutta la durata del ciclo di vita dell'oggetto fisico [8]. Poiché il PO potrebbe essere soggetto a limitazioni che ne compromettono il corretto funzionamento nel mondo reale, è fondamentale che il DT possa intervenire per sopperire a queste difficoltà [8]. Ciò significa che, in caso di malfunzionamenti o problemi del PO, il DT deve garantire la continuità del servizio e mantenere la disponibilità in modo costante [8]. Per raggiungere questo obiettivo, possono essere adottate strategie come la replicazione a più livelli e funzioni avanzate di gestione all'interno del sistema DT [8]. Questo approccio permette al DT di

mitigare le carenze del PO e di agire come una fonte affidabile per ripristinare e sincronizzare il PO in caso di necessità [8].

Memorizzazione: Il concetto di memorizzazione all'interno di un DT si riferisce alla sua capacità di memorizzare e rappresentare tutti i dati rilevanti presenti e passati. Ciò permette di rappresentare il comportamento dell'oggetto all'interno di un ambiente complesso [8]. Un DT dovrebbe essere progettato in modo da mantenere un ricco dataset, utili per studiare il comportamento dell'oggetto nel suo contesto applicativo sia per fare previsioni più accurate su eventuali comportamenti futuri [8].

Componibilità: La componibilità è la capacità di un DT di raggruppare più oggetti all'interno di una singola entità composta, controllando il comportamento sia dell'oggetto composto sia dei suoi componenti individuali [8]. Gli oggetti vengono visti come aggregazione di parti più piccole, e dunque tutti gli oggetti logici relativi a queste componenti devono essere uniti e interagire come un'unica entità [8].

Responsabilità e Gestibilità: Questa proprietà riguarda la gestione dei Digital Twin [8]. A differenza del PO che possono subire danni e malfunzionamenti, i DT non devono "rompersi", ma recuperare informazioni importanti anche in caso di problemi, fungendo da registratori affidabili del PO [8]. Devono essere gestibili come entità virtuali in sistemi distribuiti, con capacità di ripristino rapido in caso di guasti e supportare l'uso condiviso (multitenancy) [8]. Possono far parte di sistemi più complessi e devono adattarsi a diversi livelli di servizio, a seconda del dominio applicativo [8].

Augmentation: I Physical Objects hanno funzionalità e servizi fissi durante il loro ciclo di vita, ma possono essere limitati da vincoli di produzione e materiali [8]. Al contrario, i Digital Twin possono sfruttare la dematerializzazione software, consentendo di aggiornare e migliorare le loro funzioni nel tempo [8]. Grazie a tecnologie avanzate come i big data e la realtà aumentata, i DT possono integrare nuove funzioni tramite API, rendendo i PO interoperabili in ambienti complessi [8].

Ownership: La proprietà di Ownership nei Digital Twin si articola in due aspetti principali [8]. Il primo riguarda la proprietà dei dati generati dai DT, che producono una grande quantità di informazioni [8]. È fondamentale determinare chi possiede e ha il diritto di utilizzare questi dati [8]. Il secondo aspetto riguarda la proprietà del DT stesso [8]. Mentre l'oggetto fisico ha un

proprietario definito, la replica digitale può generare DT con proprietà diverse [8]. Inoltre, quando un DT viene modificato o combinato per creare un nuovo artefatto digitale, la proprietà di questo nuovo oggetto può essere diversa, mentre quella dell'oggetto fisico rimane invariata [8].

Servitizzazione: La proprietà di Servitizzazione nei Digital Twin riguarda la trasformazione di un prodotto fisico in un insieme di servizi offerti attraverso interfacce e strumenti software [8]. Grazie ai DT, un oggetto non è più visto solo come una "merce", ma come un prodotto arricchito da funzionalità, dati e processi che possono essere controllati e migliorati digitalmente [8]. Questo crea nuove opportunità di mercato, permettendo alle aziende di offrire servizi aggiuntivi e di stabilire un legame duraturo con i clienti [8].

Scalabilità: La scalabilità si riferisce alla capacità di gestire efficacemente un numero crescente di oggetti fisici e i loro corrispondenti logici, su diverse piattaforme e ambienti [8]. I DT devono poter crescere e adattarsi a una varietà di scenari. Per garantire questa scalabilità, è necessaria un'architettura flessibile e distribuita [8].

Evoluzione: Questa proprietà del DT si riferisce alla loro capacità di adattarsi e crescere nel tempo per rimanere rilevanti rispetto ai cambiamenti dell'oggetto fisico. Poiché i PO possono subire modifiche nel loro stato operativo, composizione o requisiti, i DT devono evolversi di conseguenza, aggiornando funzionalità, dati e metriche [8]. Questo processo include aggiornamenti software, nuove integrazioni di sensori o cambiamenti nei processi aziendali e normativi [8].

1.3 Applicazioni del Digital Twin

Il concetto di Digital Twin ha guadagnato una notevole attenzione negli ultimi anni, emergendo come una soluzione innovativa per affrontare problemi complessi in diversi settori [8]. Originariamente sviluppato per l'industria manifatturiera, il DT rappresenta una replica digitale di oggetti fisici, consentendo la simulazione e l'analisi del loro comportamento in vari contesti operativi [8]. Questa sezione esplorerà le applicazioni del DT, evidenziando come questo strumento possa migliorare la progettazione, la produzione e la gestione di sistemi complessi, oltre a favorire nuove opportunità di servitizzazione [8].

Determinare quali problemi possano trarre il massimo beneficio da un sistema basato su Digital Twin è una sfida complessa [8]. L'esempio del modello manifatturiero evidenzia come il DT possa affrontare problematiche intricate,

replicando il comportamento di oggetti in ambienti software per individuare difetti e opportunità di miglioramento [8]. Ci sono due aspetti fondamentali: la rappresentazione adeguata degli oggetti e la caratterizzazione dell'ambiente in cui operano. Questa necessità di una modellazione accurata richiede una profonda comprensione dei vincoli legati agli oggetti fisici e del loro contesto operativo [8]. Inoltre, è essenziale che la descrizione degli oggetti e del loro ambiente sia completa, poiché una rappresentazione carente potrebbe compromettere l'intero sistema, generando dati inadeguati [8]. L'implementazione del DT implica una visione del mondo in cui l'ambiente e le interazioni tra gli oggetti sono rappresentati in modo preciso [8]. La modellazione di sistemi complessi, che richiedono aggregazione di diversi oggetti, comporta notevoli sforzi. Non è raro, ad esempio, che dati raccolti da sensori si riferiscano a più oggetti o all'intero ambiente, complicando ulteriormente la comprensione delle interazioni [8]. È importante considerare anche quando applicare l'approccio DT, poiché alcuni problemi possono essere risolti efficacemente senza di esso [8].

1.3.1 Esempi concreti di applicazione

Sebbene il DT non rappresenti una soluzione universale per ogni problema, la sua implementazione si rivela vantaggiosa in situazioni dove può portare a innovazioni concrete. Ogni scenario sarà analizzato per evidenziare le peculiarità e i benefici specifici offerti dal Digital Twin, sottolineando la flessibilità e l'adattabilità del concetto a contesti diversificati. Attraverso l'esplorazione di queste applicazioni, si intende chiarire il potenziale trasformativo dei DT.

Applicazione nel settore manifatturiero: Il Digital Twin ha svolto un ruolo chiave nel settore manifatturiero, rivoluzionando il modo in cui i prodotti vengono progettati, monitorati e gestiti lungo tutto il loro ciclo di vita. Originariamente introdotto per fornire una rappresentazione virtuale di un prodotto fisico, il DT ha permesso ai produttori di creare copie digitali che replicano fedelmente le caratteristiche e i comportamenti del corrispondente oggetto fisico. In questo contesto, i DT non solo simulano i prototipi durante le fasi di progettazione e testing, ma supportano anche la produzione e il monitoraggio durante l'uso del prodotto [8].

Il concetto di DT permette, infatti, di effettuare simulazioni avanzate, riducendo la necessità di costruire costosi prototipi fisici. Inoltre, grazie alla raccolta continua di dati tramite sensori, i DT offrono un monitoraggio costante, utile per prevedere guasti, migliorare la manutenzione e ottimizzare l'efficienza operativa del prodotto. Questa capacità di rilevazione e analisi

dati ha portato a una riduzione dei costi di manutenzione e a una maggiore affidabilità dei prodotti [8].

Nel settore manifatturiero, i DT sono spesso integrati con le piattaforme Industrial Internet of Things, dove sensori e dispositivi connessi trasmettono dati al DT, consentendo un flusso di informazioni bidirezionale tra l'oggetto fisico e il suo gemello digitale. Questa sinergia tra DT e IIoT è stata particolarmente vantaggiosa per applicazioni che richiedono elevata precisione e tempi di risposta rapidi, come la produzione automatizzata e la gestione di impianti industriali complessi [8] .

Sanità e Virtual Patient: Nella sanità moderna, una delle sfide più rilevanti è il monitoraggio continuo e l'osservazione dei pazienti, che implica la raccolta e la memorizzazione costante di dati clinici. Questi dati sono fondamentali per mantenere registri medici completi, che includono non solo parametri clinici attuali, ma anche informazioni storiche e sullo stile di vita del paziente. Con l'avanzare dell'IoT e l'adozione di sensori corporei e ambientali a basso costo, si presentano nuove opportunità per migliorare la qualità della vita dei pazienti, permettendo il monitoraggio da casa piuttosto che in ospedale. L'idea del Digital Twin trova applicazione in questo contesto, trasformandosi in un "Paziente Digitale". Questo concetto si traduce in una rappresentazione digitale e personalizzata dello stato di un paziente, in grado di monitorare parametri fisiologici, interazioni ambientali e anche espressioni psicologiche.

Il concetto di paziente digitale si divide in due categorie principali:

- **Pazienti Monitorati:** Questi pazienti richiedono un monitoraggio continuo tramite dispositivi specifici. Possono necessitare di osservazione a lungo termine o solo per periodi limitati, mantenendo un forte entanglement con il loro gemello digitale.

- **Pazienti Non Monitorati:** Questi pazienti, pur non richiedendo un monitoraggio costante, necessitano di controlli sporadici dei parametri generali. La loro connessione con il gemello digitale è più debole, permettendo una raccolta dati meno intensa.

Il DT svolge un ruolo chiave nel cercare, accedere e memorizzare informazioni rilevanti, permettendo un monitoraggio accurato della salute. Può raccogliere dati su abitudini alimentari, attività fisiche e fattori ambientali, sempre nel rispetto della privacy del paziente. Inoltre, può analizzare dati psicologici attraverso l'osservazione delle interazioni sociali, contribuendo così a una visione più completa della salute del paziente [8].

La Città Digitale: Il concetto di DT si è dimostrato particolarmente rilevante negli studi sulle città intelligenti, grazie alla sua capacità di rappresentare, memorizzare, contestualizzare e gestire oggetti semplici fino a grandi aggregazioni. Il DT consente di monitorare, analizzare e prevedere fenomeni urbani come il traffico, offrendo servizi e applicazioni agli abitanti. Un approccio pratico è applicare il DT a piccole parti della città e poi espanderlo, permettendo un controllo capillare su diverse scale urbane, dai singoli oggetti fino all'intero ambiente cittadino. Nel caso della città digitale, è anche importante notare che i DT, grazie alle capacità di aumento delle funzionalità e servitizzazione, costituiscono la base per soluzioni aperte che non creano domini di applicazione verticali isolati e chiusi. Così, un oggetto che rappresenta un incrocio può essere utilizzato per applicazioni basate sulla logistica, per il controllo del traffico, così come per la sicurezza. Se vengono registrate e analizzate misurazioni dettagliate delle sue caratteristiche, ad esempio, gli impatti dell'introduzione di una nuova strada a senso unico, nuovi semafori, nuovi edifici, ecc., potrebbero essere simulati e studiati prima di decidere la loro implementazione. Un ulteriore elemento da considerare è che una città intelligente potrebbe essere interconnessa con Digital Twins che rappresentano altri grandi ambienti. Per esempio, esistono casi in cui città "gemelle" in diversi paesi, anche se geograficamente distanti, mantengono forti legami e collaborano strettamente. Queste relazioni consentono di analizzare e comprendere meglio gli effetti che eventi o cambiamenti in una città possono avere sull'altra, facilitando studi e soluzioni condivise, specialmente in settori come quello dei trasporti o della pianificazione urbana [8].

Agricoltura Intelligente: L'industria agricola è fondamentale per il funzionamento di qualsiasi economia, essendo una fonte chiave di cibo, materie prime e occupazione. Negli Stati Uniti, agricoltura e settori correlati generano oltre 750 miliardi di dollari all'anno. Con la crescita della popolazione mondiale e una crescente domanda di prodotti di qualità superiore, le questioni di sicurezza alimentare, sostenibilità e redditività sono sempre più cruciali. Allo stesso tempo, l'agricoltura deve affrontare pressioni economiche, oltre ai problemi legati ai cambiamenti climatici e ambientali, che rendono essenziale migliorare produttività e innovazione. I processi agricoli sono altamente complessi e dipendono da condizioni naturali come il clima, le malattie e la stagionalità. La tecnologia dei Gemelli Digitali può rivoluzionare la gestione delle aziende agricole separando gli aspetti fisici da quelli informativi. Questo approccio virtuale può migliorare l'efficienza e la produttività, riducendo al contempo i costi energetici. Sebbene il concetto di Digital Twin nell'agricoltura intelligente sia ancora agli inizi, molti agricoltori stanno valutando l'integrazione di tecnologie intelligenti per ottimizzare il processo agricolo e rispondere a queste sfide. Un

DT in agricoltura può supportare la modellazione meteorologica e la previsione degli effetti a lungo termine del cambiamento climatico. Inoltre, consente agli agricoltori di individuare aree in cui il sistema agricolo è soggetto a stress, causato da fattori come qualità del suolo, inquinamento o piante e animali invasivi. Questa tecnologia aiuta a monitorare e gestire le risorse, ottimizzando le operazioni e aumentando la sostenibilità delle coltivazioni, migliorando la capacità di rispondere a condizioni ambientali variabili [2].

1.4 Architetture Software di Supporto

Le architetture di supporto per i Gemelli Digitali rivestono un ruolo cruciale nel facilitare la loro integrazione e operatività nell'industria moderna. Queste architetture sono caratterizzate da un approccio stratificato che consente la gestione efficace di dispositivi fisici, risorse edge e cloud, creando un ecosistema altamente reattivo e versatile. I livelli inferiori dell'architettura interagiscono direttamente con i dispositivi di raccolta dati e i sistemi di controllo, garantendo che le informazioni vengano elaborate e virtualizzate in tempo reale. Questa interazione è fondamentale per ottimizzare le operazioni e garantire che i DT possano rispondere rapidamente alle variazioni delle condizioni operative [8]. Un aspetto fondamentale di queste architetture è la capacità di supportare una comunicazione ottimizzata tra i vari componenti del sistema. Le architetture devono essere progettate per gestire la complessità delle interazioni tra diversi dispositivi e sistemi, minimizzando i ritardi e migliorando l'efficienza nella trasmissione dei dati. Questo approccio è essenziale per applicazioni che richiedono decisioni in tempo reale, come nel monitoraggio della sicurezza in ambienti industriali o nella gestione delle risorse agricole. Inoltre, la piattaforma deve garantire una grande flessibilità nell'esecuzione di software eterogeneo. Ciò implica che le architetture devono essere in grado di supportare vari linguaggi di programmazione, sistemi operativi e strumenti di sviluppo, permettendo ai programmatori di implementare soluzioni senza essere vincolati a tecnologie specifiche. Questa flessibilità è cruciale per prevenire la formazione di "silos tecnologici", che possono limitare l'interoperabilità e la capacità di adattamento delle soluzioni nel tempo. La possibilità di "lanciare" software nella piattaforma senza richiedere risorse specializzate facilita anche la rapida innovazione e l'adozione di nuove tecnologie [8].

Le architetture devono anche tenere conto della gestione e della sicurezza dei dati. In un contesto in cui i DT raccolgono e analizzano informazioni critiche, la protezione dei dati diventa una priorità. Ciò implica l'adozione di pratiche di sicurezza informatica robuste, tra cui crittografia, autenticazione e controlli di accesso, per garantire che i dati siano protetti da accessi non

autorizzati e attacchi informatici. La trasparenza nella gestione dei dati è altrettanto importante, in particolare in settori regolamentati, dove la conformità a normative specifiche è fondamentale [8]..

Infine, è essenziale che le architetture di supporto per i Gemelli Digitali siano progettate con un focus sulla scalabilità. Con la crescita continua dei volumi di dati e la complessità delle operazioni industriali, le piattaforme devono essere in grado di adattarsi facilmente all'aumento della domanda senza compromettere le prestazioni. Questo richiede un attento bilanciamento tra la capacità di elaborazione e le risorse disponibili, nonché l'implementazione di strategie di archiviazione e gestione dei dati che garantiscano un'efficienza operativa continua [8].

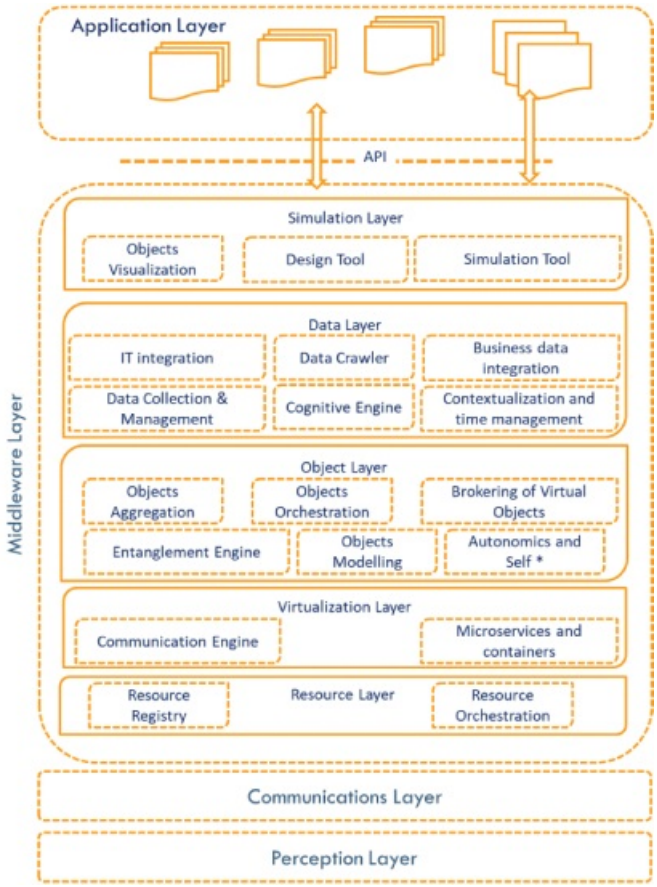

Figura 1.2: architettura di supporto Dt [8]

La 1.2 illustra un modello architettonico dettagliato per le piattaforme di Digital Twin, adottando principi di stratificazione che permettono di gestire in modo efficiente diverse funzioni e risorse. Gli strati inferiori si concentrano sull'interazione con dispositivi, edge e risorse cloud, virtualizzando le funzioni

necessarie e ottimizzando la comunicazione per rispondere a esigenze stringenti di alcuni oggetti logici. Gli strati superiori si occupano delle proprietà chiave del DT. Il livello Object Layer comprende funzioni per il ciclo di vita dei LO, come la modellazione, gestione autonoma, orchestrazione e entanglement. Il Data Layer raccoglie e contestualizza le informazioni, supportando analisi e inferenza di informazioni; qui possono essere incluse anche semantiche e ontologie per domini specifici. Il Simulation Layer agevola la visualizzazione e simulazione dei DT, includendo strumenti per progettare e definire i DT stessi. L'intera infrastruttura si basa su API aperte, che permettono l'interazione con le funzioni della piattaforma in modo strutturato e programmabile a vari livelli di astrazione, facilitando l'integrazione di applicazioni attraverso dati e API ben definiti [8].

1.5 Esplorazione degli Strumenti Attuali per i Digital Twins

In questa sezione, ci concentriamo sull'analisi delle esigenze e delle soluzioni esistenti per la gestione e visualizzazione di Digital Twin. Faremo un'analisi dei vari strumenti disponibili, valutando le loro funzionalità, punti di forza e limitazioni. Questa analisi fornirà una base solida per il nostro progetto, evidenziando la necessità di un'interfaccia utente più intuitiva e di strumenti che consentano l'esecuzione di query avanzate, come SPARQL, per interagire efficacemente con i dati. Questo ci permetterà di comprendere meglio come le attuali implementazioni affrontano le sfide associate alla creazione di Digital Twins complessi.

AWS IoT TwinMaker: AWS IoT TwinMaker [1] è una piattaforma che mira a rendere più agevole la creazione e gestione di Digital Twin per rappresentare sistemi fisici complessi, come edifici, macchinari industriali o impianti produttivi. La sua forza risiede nella capacità di combinare dati provenienti da fonti diverse con modelli 3D esistenti, offrendo una visione aggiornata e integrata dell'intero sistema operativo. Le principali funzionalità possono essere riassunte come segue:

- **Connettori per Dati:**La piattaforma dispone di connettori predefiniti per accedere ai dati di vari servizi AWS, come IoT SiteWise e Kinesis Video Streams, oltre a supportare fonti esterne attraverso un framework API. Ciò consente agli utenti di centralizzare le informazioni provenienti

[1] https://docs.aws.amazon.com/iot-twinmaker/

da diverse origini, semplificando l'accesso ai dati necessari per i Digital Twin.

- **Costruzione e Modellazione di Ambienti Fisici:**AWS IoT TwinMaker permette di rappresentare elementi del mondo reale come entità digitali, definendo relazioni personalizzate tra di esse. Queste relazioni sono visualizzate sotto forma di un grafo digitale, aggiornato dinamicamente per riflettere le variazioni nei dati provenienti dall'ambiente fisico.

- **Strumenti per la Visualizzazione 3D** Grazie al "Scene Composer", gli utenti possono importare modelli 3D esistenti e arricchirli con dati in tempo reale provenienti da sensori o flussi video. Questa funzione è utile per creare un'interfaccia visiva interattiva che consente di monitorare e analizzare lo stato operativo.

- **Sviluppo di Applicazioni Personalizzate**: La piattaforma offre strumenti per realizzare applicazioni web su misura, integrate con visualizzazioni 3D e dashboard interattive, sfruttando plugin come Grafana per la gestione dei dati e l'analisi visiva.

Azure Digital Twin: Azure Digital Twins [2] è una piattaforma di tipo "Platform as a Service" (PaaS) che permette di modellare e rappresentare digitalmente ambienti fisici complessi, come edifici, fabbriche e intere città. Con una struttura modulare, la piattaforma consente di simulare il funzionamento di entità fisiche attraverso grafi, offrendo agli utenti la possibilità di monitorare e migliorare i processi operativi.

Con Azure Digital Twins, è possibile progettare un'architettura di digital twin che simula dispositivi IoT reali all'interno di una soluzione cloud più ampia. Questa architettura si collega ai dispositivi gestiti da IoT Hub, consentendo l'invio e la ricezione di dati in tempo reale. È importante notare che i twin di dispositivo di IoT Hub e i digital twin offerti da Azure Digital Twins hanno funzioni distinte: mentre i primi sono gestiti dall'Hub per ciascun dispositivo connesso, i digital twin di Azure possono rappresentare una varietà di entità definite da modelli digitali personalizzati.

- **Modellazione Ambientale:** Consente di definire qualsiasi ambiente e di attivare i digital twin in modo scalabile e sicuro. Gli utenti possono creare modelli personalizzati che rappresentano entità del loro contesto aziendale, creando così un linguaggio specifico per descrivere le proprie operazioni.

[2]https://learn.microsoft.com/en-us/azure/digital-twins/

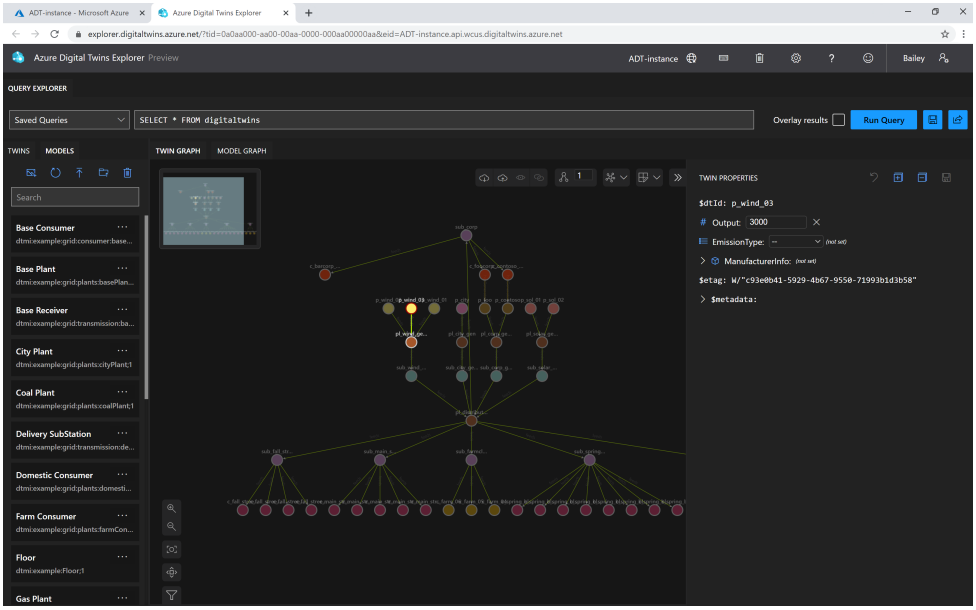

Figura 1.3: rappresentazione di un knowledge graph con Azure DT

- **Collegamento di Asset e Sistemi:** Gli asset fisici e i dispositivi IoT possono essere collegati ai Digital Twin, grazie a un sistema di eventi che garantisce il flusso continuo di dati. Questo approccio facilita l'aggiornamento dinamico del grafo digitale.

- **Analisi e Interrogazione dei Dati:** La piattaforma offre strumenti avanzati per effettuare query in tempo reale, consentendo di ottenere informazioni utili sull'ambiente operativo e migliorare la qualità delle decisioni.

- **Visualizzazione in 3D:** Grazie al 3D Scenes Studio, gli utenti possono monitorare i dati operativi attraverso una rappresentazione tridimensionale, integrando logiche aziendali per migliorare l'esperienza visiva.

- **Integrazione con Altri Servizi Azure:** I dati generati possono essere integrati con altri servizi Azure per analisi, archiviazione o ulteriori elaborazioni, garantendo un'ampia interoperabilità all'interno dell'ecosistema cloud.

Un'architettura di soluzione tipica che utilizza Azure Digital Twins può comprendere i seguenti componenti:

- **Servizio Azure Digital Twins:**Memorizza i modelli di twin e il grafo dei twin 1.4 con le relative informazioni di stato, gestendo l'elaborazione degli eventi.
- **Applicazioni Client:** Queste applicazioni interagiscono con il servizio di Azure Digital Twins per configurare modelli, definire topologie ed estrarre informazioni dal grafo.
- **Risorse di Calcolo Esterne:** Possono essere integrate risorse esterne, come Azure Functions, per elaborare eventi generati dai digital twin o dai dispositivi connessi.
- **IoT Hub:**Fornisce gestione dei dispositivi e capacità di flusso dati IoT.
- **Servizi Disposti a Valle:** Offrono integrazioni per flussi di lavoro (ad esempio Logic Apps), archiviazione (come Azure Data Lake) o analisi (come Azure Data Explorer).

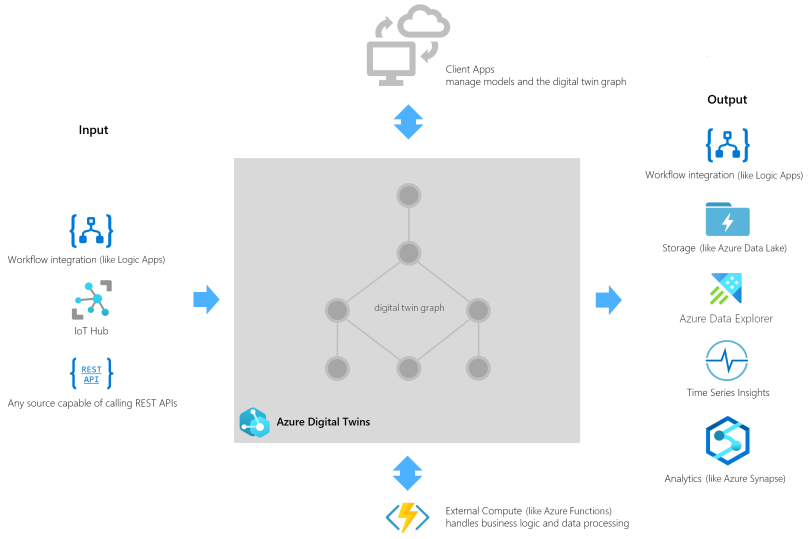

Figura 1.4: Diagramma che illustra il ruolo di Azure Digital Twins all'interno di una soluzione IoT Azure completa.

Azure Digital Twins si presenta come un insieme completo di strumenti e funzionalità dedicate alla visualizzazione e alla gestione di ecosistemi di digital twin. La piattaforma facilita la creazione di rappresentazioni digitali ricche e interattive di ambienti fisici complessi, consentendo alle aziende di ottimizzare i processi e di ottenere informazioni utili per l'innovazione e il miglioramento continuo. Queste funzionalità forniscono una base robusta per lo sviluppo

del nostro strumento di visualizzazione, integrando le migliori pratiche e le tecnologie avanzate disponibili nel campo dei digital twin.

Azure Digital Twins utilizza un linguaggio di definizione dei modelli chiamato **Digital Twins Definition Language** (DTDL), un formato JSON-LD versatile che consente la creazione di rappresentazioni digitali di entità del mondo reale. Il DTDL definisce ogni modello come una classe, descrivendo la struttura dei dati per concetti specifici nell'ambiente operativo, con attributi come proprietà, relazioni e componenti. Questa flessibilità permette di personalizzare i modelli con termini specifici del proprio business, creando un vocabolario su misura per descrivere i digital twin in modo intuitivo e conforme al contesto aziendale. Con il DTDL v3, Azure Digital Twins supporta nuove funzionalità avanzate come l'uso di array, l'incremento dei limiti di ereditarietà e l'integrazione di estensioni semantiche tramite i QuantitativeTypes, migliorando la capacità di rappresentazione.

Eclipse Ditto: Eclipse Ditto [3] è un framework open source progettata per realizzare Digital Twin di dispositivi IoT. Ditto permette di gestire milioni di DT, semplificando lo sviluppo di soluzioni IoT e offrendo API che facilitino l'interazione con i dispositivi già connessi. Eclipse Ditto, opera con un approccio "agnostico" nei confronti di dati, infatti, non pretende di sapere con esattezza la struttura che hanno le "cose" nell'IoT. Gli unici due elementi definiti sono:

- Attributi: riguardano la gestione di metadati statici di una cosa. Descrivono l'oggetto in modo più dettagliato e possono essere di qualsiasi tipo.
- Caratteristiche: riguardano la gestione di dati di stato di una cosa.

Le "**cose**" sono delle entità generiche caratterizzate da un identificatore univoco, e possono contenere anche una definizione (che può essere un URL HTTP) che serve per collegare la cosa ad un modello che ne definisce le capacità e e caratteristiche.

La comunicazione tra l'oggetto fisico e il suo corrispettivo digitale è bidirezionale e avviene attraverso l'invio di **Messaggi** che contengono un payload personalizzato.

Eclipse Ditto accetta messaggi tramite **HTTP** e **protocollo Ditto**, verifica se le parti interessate attualmente connesse possono ricevere un messaggio, se la risposta è affermativa instrada il messaggio e i messaggio di risposta tra i client connessi. Ditto non offre nessuna reiterazione dei messaggi. Se al momento dell'invio un dispositivo non è connesso non riceverà più quel messaggio.

[3] https://eclipse.dev/ditto/intro-overview.html

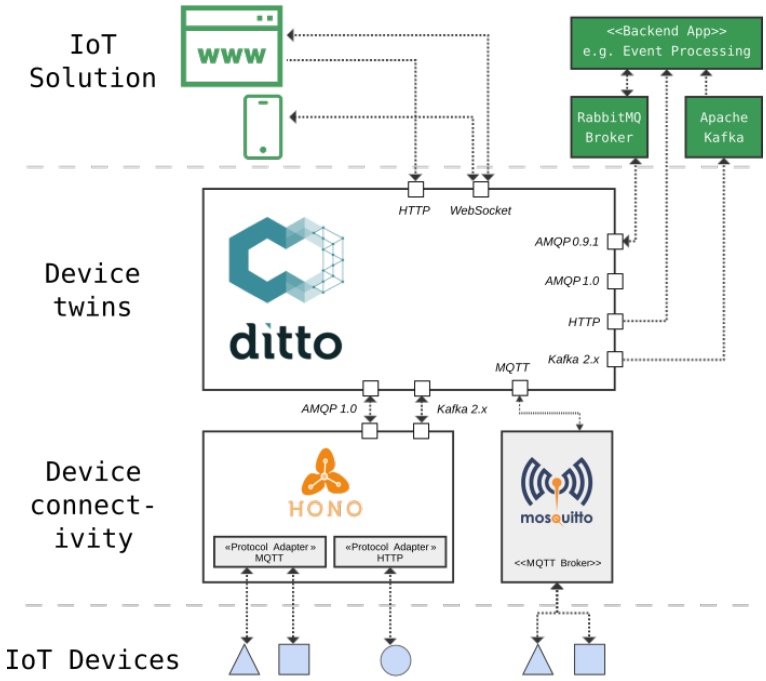

Figura 1.5: Architettura di Eclipse Ditto.

1.6 Sfide e Prospettive Future

Il concetto di Digital Twin è in continua evoluzione e si prevede che il suo impatto diventerà sempre più rilevante nel prossimo futuro. Nel 2019, un sondaggio di Gartner ha rivelato che i Digital Twins stavano entrando nell'uso mainstream da parte delle organizzazioni. Si prevedeva che il 75% delle organizzazioni dell'Internet of Things (IoT) avrebbe utilizzato la tecnologia Digital Twin o pianificato di utilizzarla entro il 2020. Gartner stima inoltre che entro il 2027 oltre il 40% delle grandi aziende a livello mondiale utilizzerà il Digital Twin nei propri progetti per aumentare i ricavi [2].

La varietà di scenari rilevanti dimostra che il concetto di DT è già ben accettato sia negli ambienti accademici che industriali. Bisogna comprendere però che, il dominio applicativo dei Digital Twin è stato caratterizzato da prototipi e soluzioni proprietarie, spesso parzialmente sviluppate. Molte di queste tecnologie si concentrano su ambiti specifici, risultando limitate in termini di flessibilità e standardizzazione e quindi non facilmente utilizzabili in contesti più ampi. Dunque è necessari ala definizione di standard aperti per superare questa carenza.

Dal punto di vista tecnico, restano numerose sfide da affrontare e convalidare per i Digital Twin. Tra queste vi sono la capacità di entanglement, la

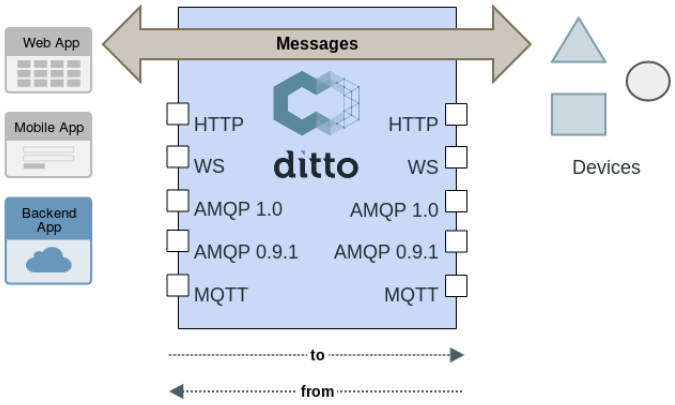

Figura 1.6: Invio di messaggi attraverso Ditto.

scalabilità delle piattaforme fino a milioni di oggetti, l'aggregazione di dati, l'autonomia gestionale con approccio "zero-touch", la raccolta e l'analisi dei dati catturati, la capacità di contestualizzare i DT. Un'altra area critica riguarda la sicurezza rappresenta una questione fondamentale, richiedendo ulteriori ricerche accademiche e industriali per garantire sistemi robusti e protetti.

Queste sfide non si limitano agli ambiti individuali e settoriali, ma richiedono anche una visione sistemica più ampia, che permetta di integrare e interconnettere i diversi Digital Twins in un ecosistema complesso. È in questo contesto che il concetto di Web of Digital Twin emerge come una soluzione promettente per superare le limitazioni dei modelli autonomi e isolati di DT. Nel prossimo capitolo, esploreremo come il WoDT possa trasformare il panorama dei Digital Twin, offrendo una visione che va oltre la replica statica di singole entità fisiche, e includendo le dinamiche interattive tra asset diversi provenienti da vari settori e organizzazioni.

Capitolo 2

Web of Digital Twin

Come descritto nel precedente capitolo, negli ultimi decenni, l'approccio del digital twin si è diffuso in diversi ambiti, dando vita a modelli virtuali che riflettono il comportamento di oggetti fisici. Questi modelli software consentono di offrire una varietà di servizi, che spaziano dalla semplice raccolta e visualizzazione dei dati relativi allo stato attuale dell'entità fisica, fino alla capacità di analizzare e prevedere eventi critici, ottimizzando così le prestazioni e permettendo una gestione più efficace delle risorse nel lungo periodo [12]. Nonostante le applicazioni promettenti, l'approccio attuale ai DT è spesso limitato a una prospettiva verticale, dove ogni gemello digitale è un'entità autonoma, isolata e dedicata a un'applicazione specifica. Tuttavia, settori complessi necessitano di un approccio più olistico, in cui i DT non solo replicano lo stato delle singole entità, ma anche le loro interazioni dinamiche [13]. Per trovare una soluzione all'esigenza di rappresentare realtà complesse è stato introdotto il paradigma del Web of Digital Twin (WoDT) Il WoDT amplia il concetto di Digital Twin visto in precedenza permettendo la virtualizzazione di asset interconnessi, anche se essi appartengono a diversi settori e organizzazioni, in una prospettiva di sistema aperto. Questa idea richiede, oltre alle tecnologie necessarie, un modello e una struttura adeguati. Questi devono essere abbastanza generali da cogliere gli aspetti chiave, indipendentemente dai vari settori e dalle tecnologie specifiche utilizzate, ma anche abbastanza chiari da poter servire come guida per sviluppare architetture e tecnologie concrete [12].

2.1 Architettura del Web e il Semantic Web

Il concetto di Web of Digital Twin prende ispirazione dai principali principi concettuali e architettonici del Web, considerando il Web non solo come un mezzo di comunicazione, ma come un'architettura e una piattaforma na-

turale per il deployment dei digital twin [12]. In questo contesto, è quindi fondamentale esplorare le caratteristiche e le tecnologie del web.

Il World Wide Web, è stato concepito da Tim Berners-Lee nel 1989 mentre lavorava al CERN. L'obiettivo era quello di trovare una soluzione per gestire in modo ottimale le informazioni e le sfide legate alla complessità dei progetti, utilizzando sistemi di informazioni collegate, noti come "*Hypertext*". Secondo Berners-Lee, l'adozione di questi sistemi, non solo avrebbe portato giovamento al CERN, ma sarebbe potuto servire anche come modello per altre organizzazioni che si sarebbero potute trovare in situazioni simili in futuro [5]. Era quindi necessario l'introduzione di un sistema globale, che fino ad allora non era mai esistito a causa della mancanza uno schema di denominazione comune per i documenti, di protocolli di accesso di rete comuni e di formati di dati comuni per l'ipertesto. L'architettura del World Wide Web doveva essere in grado di affrontare un insieme distribuito e eterogeneo di computer che eseguono applicazioni diverse e utilizzano formati di dati differenti[4].

Nel 1991, il Web ha fatto il suo debutto pubblico, ma la sua crescita è avvenuta con l'introduzione di browser più intuitivi come Mosaic nel 1993. Questo ha reso il Web accessibile a un pubblico più vasto, trasformando il modo in cui le persone interagiscono con le informazioni. A partire dal 1994, la crescente popolarità del Web ha portato alla formazione del World Wide Web Consortium (W3C), un ente dedicato allo sviluppo di standard comuni per garantire l'interoperabilità e la qualità del Web. Da allora, il Web è evoluto notevolmente, integrando nuove tecnologie e servizi che hanno facilitato la comunicazione globale, il commercio elettronico e l'accesso a un'enorme quantità di contenuti [3].

Il Web si basa sullo stile architetturale **REST**, con l'obiettivo di fornire un accesso alle informazioni che sia interoperabile e scalabile. L'architettura REST si basa sui seguenti vincoli:

- **Identificazione delle risorse**:Ogni risorsa in un sistema REST è identificata in modo univoco tramite un URL (Uniform Resource Locator). Una risorsa può essere un oggetto, una collezione di dati, un'entità o anche un'operazione, come un'azione di ricerca. L'idea di base è che ogni risorsa abbia una rappresentazione, che può essere manipolata e trasferita tra client e server. Il client accede a una risorsa tramite il suo URL, e le interazioni tra client e server avvengono tramite la manipolazione di queste risorse identificabili.

- **Manipolazione delle risorse tramite rappresentazioni**: In REST, le risorse non sono direttamente manipolabili. Invece, le risorse sono rappresentate da documenti o oggetti che contengono lo stato della risorsa in

un formato leggibile. Le rappresentazioni più comuni includono formati come HTML, XML e JSON.

- **Messaggi auto-descrittivi**: I messaggi di una comunicazione REST sono auto-descrittivi, cioè contengono tutte le informazioni necessarie per essere compresi senza fare affidamento su uno stato precedente o su sessioni persistenti.

- **Ipermedia come motore dello stato applicativo** :è un principio fondamentale del REST secondo cui il client interagisce con il server esclusivamente attraverso link (ipermedia) che vengono forniti nelle risposte del server. Ogni risposta non solo fornisce dati, ma anche link ad altre risorse o azioni che possono essere eseguite successivamente, indicando il percorso che il client deve seguire per navigare l'applicazione. Questo significa che il client non ha bisogno di conoscere in anticipo tutte le possibili azioni o percorsi nel sistema, ma può "navigare" dinamicamente attraverso le risorse a partire dalle informazioni contenute nei messaggi di risposta. [7]

Nel 2001 Tim Berners-Lee introdurrà il concetto di **Web Semantico** in modo da permettere a computer e agenti intelligenti di interagire in modo più efficace con le informazioni disponibili.

Il Web Semantico è un'estensione dell'attuale web, dove le informazioni sono dotate di un significato ben definito, migliorando così la collaborazione tra computer e persone. Per consentire al Semantic Web di funzionare, i computer devono avere accesso a informazioni strutturate e insiemi di regole inferenziali che possono utilizzare per condurre ragionamenti automatici [16]. L'idea centrale del Web Semantico è fornire una semantica dei dati, che consente a un sistema di manipolarli in catene di passi deduttivi significativi, secondo regole solide. Include un livello di metadati che le macchine possono interpretare, permettendo così ai computer di cogliere il significato di una pagina web. Per realizzare ciò, è necessario disporre di un metodo per descrivere i concetti e le relazioni tra di essi [16]. Il Web Semantico è suddiviso in due principali rami per la modellazione della conoscenza e il ragionamento: OWL e RDF/RDFS [6].

RDF: RDF è un formato di dati con semantica che fornisce un modello standard per lo scambio di dati su Web nel contesto del Semantic Web [16], consentendo di esprimere relazioni tra risorse in modo comprensibile per le macchine [6]. RDF utilizza una struttura basata su "triple," composte da tre elementi che possono essere serializzate in diversi formati:

- Soggetto
- Predicato
- Oggetto

Ogni tripla rappresenta un'affermazione che il soggetto è collegato all'oggetto tramite il predicato. Gli elementi di una tripletta possono essere URI univoci, nodi vuoti (che non sono consentiti all'interno dei Knowledge Graph) o letterali (come numeri o stringhe) [6].

```
<http://dbpedia.org/resource/Paris>
    <http://purl.org/dc/terms/subject>
    <http://dbpedia.org/resource/Category:Capitals_in_Europe> .
```

Questa affermazione, espressa nel formato N-TRIPLES, indica che Parigi (soggetto) appartiene alla categoria delle capitali in Europa (oggetto), utilizzando "subject" come predicato per indicare la relazione [6]. Un formato molto utilizzato per serializzare i dati RDF è **Turtle**,una sintassi compatta , che rende la rappresentazione delle triple più leggibile e meno verbosa rispetto a N-Triples. Per facilitare la leggibilità, Turtle usa "qnames" che abbreviano le URI, definite all'inizio del file tramite il comando @prefix seguito dall'URI globale. Una tripla in Turtle segue l'ordine soggetto/predicato/oggetto, terminando con un punto. Quando diverse triple condividono lo stesso soggetto, Turtle permette di omettere il soggetto, separando le dichiarazioni con un punto e virgola. Se, invece, soggetto e predicato sono condivisi, si usa una virgola [1].

OWL: OWL (Web Ontology Language) è un linguaggio di modellazione sviluppato per il Semantic Web che consente di descrivere complesse relazioni tra classi, entità e proprietà, aggiungendo un alto livello di espressività logica rispetto agli standard meno avanzati come RDF e RDFS [1]. Grazie a OWL, i modellatori possono definire vincoli dettagliati e specificare in modo preciso come le classi e gli individui si relazionano tra loro. Ad esempio, si possono esprimere regole per vincolare le proprietà, come stabilire che una certa proprietà è unica o transitiva, oppure creare gerarchie complesse di classi e sottoclassi[1]. OWL è particolarmente utile per gestire dati distribuiti su larga scala e per costruire sistemi di inferenza automatica che permettono di derivare nuove conoscenze dai dati esistenti. In OWL, i concetti possono essere specificati tramite l'uso di URI (Uniform Resource Identifier), che rappresentano un insieme di termini standardizzati e riutilizzabili su internet. Grazie a questa struttura, è possibile accedere a descrizioni e a documentazione aggiuntiva, come per le URI delle proprietà come rdf:type o rdfs:subClassOf, che

forniscono informazioni su come questi termini vengono utilizzati nei modelli di OWL [1].

SPARQL: SPARQL è un linguaggio di interrogazione RDF standardizzato dal World Wide Web Consortium (W3C) [14], sviluppato per recuperare e manipolare i dati memorizzati nel formato RDF [16]. Progettato specificamente per lavorare con la struttura RDF, SPARQL consente di formulare query per recuperare informazioni dai dati rappresentati come triple RDF (soggetto-predicato-oggetto), che rappresentano le connessioni tra risorse [1]. La sua espressività è comparabile all'algebra relazionale, consentendo un'agevole integrazione con ontologie come OWL (Web Ontology Language) e rappresentando un mezzo versatile per formulare obiettivi, controllare stati preesistenti e dedurre transizioni di stato nel mondo digitale [14] La possibilità di strutturare le query SPARQL in forma di pattern consente di cercare risorse anche quando i nomi degli attributi o delle proprietà possono variare, rendendo flessibile il recupero delle informazioni [1].

2.1.1 Paradigma del Web of Things

Il concetto di WoT è apparso agli inizi degli anni 2000, ma ha cominciato a svilupparsi in modo più strutturato nel 2008, grazie ai contributi di Guinard e Trifa, i quali proposero un'architettura orientata alle risorse (resource-oriented architecture) come base per la comunicazione dei dispositivi [15]. Il Web of Things (WoT) rappresenta l'evoluzione dell'Internet of Things (IoT), portando il paradigma del Web all'interno di una rete di dispositivi connessi. Mentre l'IoT mira principalmente a connettere fisicamente oggetti e dispositivi tramite Internet, il WoT estende questo concetto, utilizzando il Web come piattaforma unificante che consente l'interazione, la gestione e l'integrazione di dispositivi intelligenti attraverso protocolli e standard Web. Grazie al WoT, dispositivi di produttori diversi e con tecnologie differenti possono operare in un ecosistema condiviso, migliorando così l'interoperabilità e riducendo le barriere che impedivano una comunicazione fluida tra dispositivi e piattaforme di diversa natura. Una delle caratteristiche principali del WoT è l'uso delle API RESTful e dei protocolli HTTP, che permettono agli oggetti intelligenti di essere rappresentati come risorse Web. Ogni dispositivo è trattato come una risorsa identificabile e accessibile tramite URL, e le operazioni standard del Web, come il GET e il POST, possono essere utilizzate per interrogare, aggiornare e interagire con questi dispositivi. Questo approccio semplifica lo sviluppo di applicazioni per il WoT, poiché gli sviluppatori possono utilizzare tecnologie Web consolidate come HTML, JavaScript e CSS per accedere ai dispositivi e visualizzare i dati [9].

Il WoT si basa su una struttura di architettura stratificata che include vari livelli, come quello di connettività, interazione e servizio, progettati per rispondere alle esigenze di interoperabilità e scalabilità. Questo approccio consente di integrare facilmente nuovi dispositivi e servizi e supporta un ampio numero di applicazioni [15].

L'uso del WoT porta molti vantaggi per la creazione di ambienti intelligenti, come le smart cities e le smart home, poiché consente a diversi dispositivi di collaborare in un sistema interconnesso. Inoltre, consente di applicare tecnologie semantiche per descrivere i dispositivi e i servizi offerti, facilitando la comprensione e l'interazione automatizzata tra le cose e le applicazioni [9].

2.2 Fondamenti del Web of Digital Twins

I DT sono una tecnologia che sta prendendo rapidamente piede in diversi campi. Negli ultimi anni grazie alla ricerca, questa tecnologia ha subito una forte innovazione che ne ha migliorato le prestazioni e ha aumentato le possibilità di utilizzo. I DT possono essere ora applicati a contesti più complessi e a dinamiche che possono includere asset appartenenti a domini applicativi molto diversi tra loro, realizzando un ecosistema distribuito formato da DT interconnessi tra loro che prende il nome di Web of Digital Twins. In questa visione il WoDT definisce uno strato software orientato ai servizi sopra al quale possono essere progettate applicazioni intelligenti. Grazie alle sue funzionalità è possibile accedere e interagire con gli asset fisici interconnessi [12].

2.2.1 Caratteristiche del WoDT

L'idea alla base del WoDT si basa sul principio "DT-as-a-Service", secondo il quale ogni asset fisico di un'organizzazione deve avere un corrispettivo digitale che ne rispecchia e aumenta le funzionalità e rende disponibili le sue funzioni per utenti esterni, interconnettendosi con gli altri asset in un ecosistema di digital twins [7] [12].

Gli aspetti principali di un Digital Twin all'interno di un contesto WoDT, riguardano principalmente la sua natura dinamica. Infatti, i DT associati a PA sono dinamici poiché possono esistere all'interno del WoDT sin dall'inizio oppure essere creati in un secondo momento in modo dinamico e poi eliminati quando non sono più necessari. Un aspetto interessante degli ecosistemi di Digital Twin è che le relazioni tra i vari asset interconnessi non sono statiche, esse infatti possono mutare nel tempo. In un WoDT,le relazioni devono essere rappresentate in modo esplicito a livello di DT. A tale scopo, si utilizzano semantiche definite tramite ontologie di dominio, in modo analogo a quanto accade nel Semantic Web [12].

Knowledge Graph Ogni DT all'interno di un WoDT è rappresentato da un **knowledge graph**, che contiene tutte le informazioni relative al dominio e ai dati dell'asset fisico.

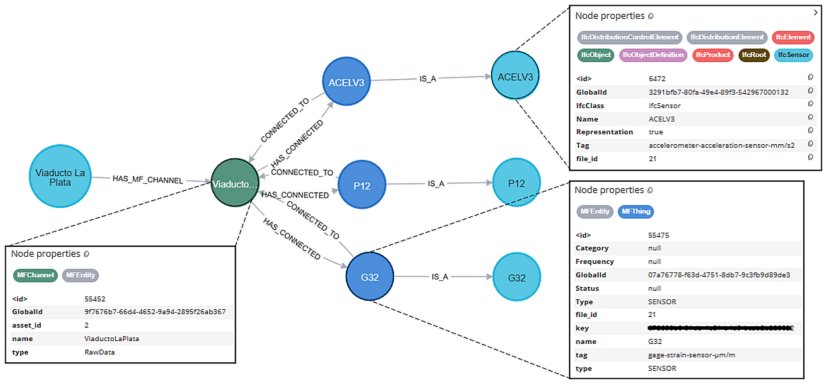

Figura 2.1: Esempio di Knowledge Graph[11]

Quando si verificano delle interazioni tra i Physical Assets di diverse organizzazioni o essi quando partecipano a diverse organizzazioni con ruoli differenti, i Knowledge Graph dei rispettivi DT vengono collegati sfruttando ontologie specifiche e i DT vengono messi in relazione tra loro. Questo richiede al WoDT di affrontare la sfida dell'interoperabilità, consentendo, di utilizzare ontologie diverse, anche tra settori differenti.

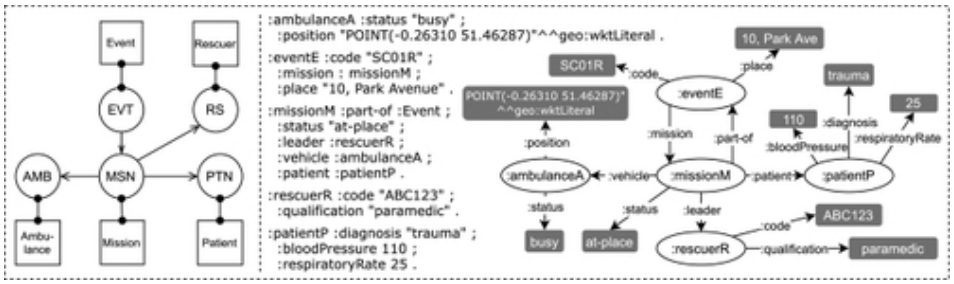

Figura 2.2: Kwoledge Graph costruito sulla base di triplette RDF [12]

È possibile utilizzare RDF come linguaggio di rappresentazione per knowledge Graph. Il grafo sarà quindi rappresentato da triple RDF che contengono informazioni dinamiche, come proprietà, relazioni ed eventi, oltre alle informazioni statiche del DT. Nelle triple relative alle proprietà il predicato rappresenta il nome della proprietà e l'oggetto rappresenta il valore della proprietà, che può essere un dato letterale o l'IRI di un'altra risorsa. Nelle triple relative alle relazioni, il predicato identifica il nome della relazione e l'oggetto è l'IRI del DT collegato, corrispondente al PA di destinazione che è in relazione con il PA di origine, riflesso attraverso il DT di collegamento [12]

Modello Astratto Il modello M rappresenta come il PA è strutturato e monitorato digitalmente. Le sue componenti principali sono:

- **Propietà:** Sono gli attributi osservabili del PA, rappresentati come variabili di dati etichettate. Questi attributi descrivono lo stato del PA in termini di caratteristiche specifiche e possono evolvere dinamicamente nel tempo.
- **Relazioni:** Descrivono i collegamenti tra il PA e altri PA all'interno del WoDT, stabilendo connessioni a livello di DT con altre entità digitali. Le relazioni possono essere osservabili e anch'esse mutevoli nel tempo, permettendo così la modellazione di network complessi tra entità diverse. A differenza delle proprietà, le relazioni estendono il DT oltre il PA locale, facilitando l'interazione e l'integrazione con altri DT.
- **Eventi:** Rappresentano gli eventi significativi che si verificano a livello del PA. Questi eventi permettono il monitoraggio delle dinamiche del PA e attivano cambiamenti di stato nel DT che ne riflettono l'evoluzione [12].

Un modello concreto definisce le attuali proprietà e relazioni usate da un DT per rappresentare il PA e gli eventi che lui può generare dinamicamente.

Stato dinamico Dato un modello M lo **stato dinamico** (STD) è descritto dalla tupla

$$SDT = \langle P, R, E, t \rangle$$

dove:

- ***P:*** l'insieme corrente delle proprietà con i relativi valori
- ***R:*** l'insieme delle relazioni attuali
- ***E:*** la sequenza di eventi generati finora
- ***t:*** un timestamp logico che rappresenta il tempo corrente modellato

Processo di Shadowing Come descritto nel capitolo precedente, il meccanismo di shadowing è una proprietà fondamentale dei DT che permette di mantenere sincronizzato il DT con il corrispettivo PA, basandosi sul modello M. Questo processo si svolge attraverso tre fasi principali:

- Ogni variazione rilevante nello stato S_{PA} del PA viene catturata come un evento ev_{PA}.

- L'evento ev_{PA} viene propagato al DT.
- In presenza di un nuovo evento ev_{PA}, lo stato S_{DT} del DT viene aggiornato tramite una funzione di shadowing $Shad_{PA \to DT}$ che dipende dal modello M: $S'_{DT} = Shad_{PA \to DT}(S_{DT}, ev_{PA})$.

Un DT deve essere in grado di supportare eventi che avvengono all'interno del mondo fisico e che impattano sul PA e aggiornare lo stato del DT di conseguenza dopo l'applicazione dell'azione.

Il WoDT è composto da una serie di DT autonomi, ognuno con il proprio modello M e stato, collegati tra loro attraverso le relazioni definite a livello di PA. Questa struttura distribuita rende il WoDT asincrono e decentralizzato [12]. Infatti, i DT all'interno di un WoDT possono operare con tempistiche indipendenti e aggiornarsi autonomamente.

Modello di Interazione Il modello di interazione si occupa della comunicazione e dell'interazione con i DT utilizzando delle primitive API fornite a livello applicativo. Queste primitive consentono di poter sfruttare al massimo le funzionalità del DT e di garantire che le informazioni relative agli stati degli asset fisici e gli eventi che riguardano l'ambiente del dominio applicativo siano osservabili in tempo reale senza però interferire con il processo di shadowing [12].

Per soddisfare questa esigenza, sono previste due modalità principali di interazione: interrogare attraverso delle query e l'osservazione dei DT. [12].

- Interrogazione: Questa modalità prevede richieste uniche di informazioni sullo stato corrente di un DT o di un gruppo di DT. Grazie al modello semantico implementato, tali interrogazioni possono essere gestite utilizzando linguaggi standard come SPARQL, ottimizzando così l'accesso alle informazioni specifiche richieste dall'utente [12].
- Monitoraggio: Il monitoraggio consiste nel richiedere costantemente lo stato corrente di un DT o di un grafo di DT per ricevere notifiche su tutti gli eventi osservabili (e quindi rilevanti) che si verificano nel PA corrispondente. Questo flusso di eventi può essere filtrato in base a criteri definiti durante il processo di sottoscrizione. L'adesione è dinamica, e tramite le API è possibile avviare o interrompere il monitoraggio in qualsiasi momento.

È possibile interagire con i DT aggiungendo ed eliminando i DT all'interno di un WoDT. Questa modalità di interazione riflette la complessità e la dinamicità si questi sistemi complessi di asset interconnessi, permettendo ad un

WoDT di evolvere nel tempo dinamicamente. Così facendo è possibile creare un ambiente che muta in base alle esigenze creando un sistema scalabile [12].

Esistono tre approcci principali per la creazione di nuovi DT:

1. **Creazione statica**: Un DT può essere creato in modo stabile e configurato manualmente dagli amministratori. Questa modalità viene spesso applicata per istanziare DT autonomi o per creare DT di livello radice in un WoDT complesso, che poi gestirà la creazione dinamica degli altri DT [12].

2. **Creazione dinamica per shadowing**: Un DT può nascere in modo dinamico come conseguenza del processo di shadowing di un PA o di un DT esistente. In questo caso, il nuovo DT è generato a partire dal DT che sta tracciando, ed è possibile definire collegamenti al "genitore" tramite relazioni, che fanno parte dell'insieme delle relazioni R nel modello astratto [12].

3. **Creazione dinamica a livello applicativo**: Infine, un DT può essere creato su richiesta di un'operazione specifica, avviata dal livello applicativo. Questo approccio può riguardare la creazione di un DT autonomo oppure di un DT legato a uno esistente tramite una relazione, generando quindi una struttura gerarchica o collegata in rete [12].

2.2.2 Aumento delle funzionalità di un DT

La rappresentazione di un oggetto fisico attraverso un corrispettivo digitale non consente solo di clonare virtualmente le sue caratteristiche e comportamenti ma permette anche di aumentarne le funzionalità. Rientrano in questo concetto anche le funzioni di previsione di eventi e simulazione, comunemente descritte nella letteratura sui DT poiché, pur non essendo strettamente correlate all'asset fisico, utilizzano il modello di riferimento M, lo stato S_{DT} e altri dati disponibili per fornire informazioni sulle possibili evoluzioni del comportamento dell'asset fisico [12].

Nel modello astratto, ciò viene rappresentato con uno stato aumentato, S_{AU}, che contiene ulteriori proprietà, relazioni ed eventi oltre a quelli derivati dal processo di shadowing del PA. L'augmentation si basa su una funzione astratta $Augm$, parte anch'essa del modello M, in modo analogo alle funzioni di shadowing $Shad$.

Per esempio, un evento ev_{PA} rilevato a livello dell'asset fisico può aggiornare sia lo stato S_{DT} attraverso il processo di shadowing, sia lo stato aumentato S_{AU} tramite la funzione di augmentation, rappresentata da

$$S'_{AU} = Augn_{PA \to DT}(S_{AU}, S_{DT}, ev_{PA}).$$

Inoltre, un evento azione a_{DT}, parte del comportamento aumentato, può causare l'aggiornamento dello stato aumentato S_{AU} e generare un evento a_{PA} da propagare all'asset fisico, secondo un processo simile a quello di shadowing [12].

2.2.3 Ciclo di vita e Architettura

Il modello WoDT richiede un ciclo di vita astratto per i Digital Twin e un'adeguata architettura software per essere implementato in modo efficace e operativo.

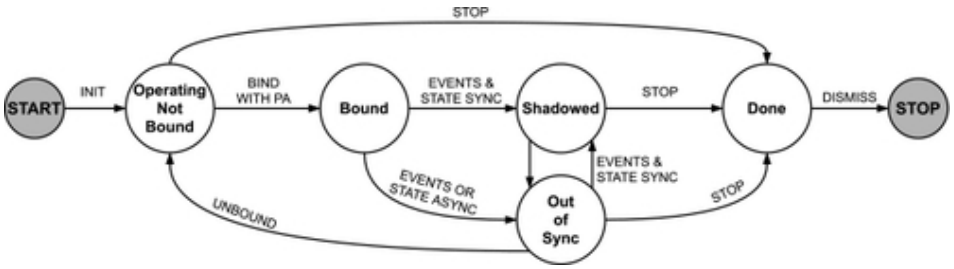

Figura 2.3: Ciclo di vita di un DT [12]

Come descritto nella figura 2.3, dopo l'avvio, il DT s trova in uno stato operativo: in questa fase, tutti i moduli interni sono attivi, ma il DT non è ancora associato al suo Asset Fisico (PA). Il collegamento tra DT e PA avviene attraverso una procedura di "binding" che li associa rispettando i requisiti specifici del dominio di applicazione. Una volta completata la procedura, il DT entra nello stato Bound e può gestire eventi bidirezionali, interagire con il PA e avviare il processo di shadowing, sincronizzando così gli stati e gli eventi in tempo reale per raggiungere lo stato Shadowed. Qualsiasi errore di sincronizzazione porta il DT in uno stato Out of Sync, dove perde la capacità di gestire eventi, di mantenere l'allineamento dello stato o di interagire con il mondo esterno fino a quando non viene ripristinata la sincronizzazione[12]. Il DT può tornare allo stato Shadowed solo dopo aver recuperato la sincronizzazione. Durante il ciclo di vita, il DT può essere fermato e trasferito allo stato Done: in questa fase è ancora accessibile per applicazioni esterne, mantiene la sua memoria e il registro eventi, ma non è più associato o sincronizzato con il PA. Al termine del ciclo di vita, il DT può infine essere dismesso entrando nello stato Stop. La complessità sia della piattaforma che dei DT è affrontata decomponendo il sistema in moduli gestibili e a eventi, facilitando così sviluppo, manutenzione e scalabilità. Questi moduli possono essere distribuiti su diverse

architetture, come Edge o Cloud, mantenendo sempre la comunicazione tra eventi e conoscenze distribuite[12].

Basandosi sul ciclo di vita precedentemente descritto, emergono alcuni requisiti architetturali per il WoDT:

- **Creazione e Associazione Dinamica di DT e PA:** Il sistema deve consentire la creazione dinamica dei DT e la loro associazione con gli Asset Fisici (PA), assegnando a ciascun DT e PA un identificatore unico.
- **Uniformità nella Cattura degli Eventi del PA:** Le modifiche di stato di un PA devono essere rilevate e rappresentate in una forma unificata. Anche se le fonti PA sono eterogenee, i DT associati devono poter aggiornare il loro stato in base a queste variazioni, seguendo i modelli predefiniti dei DT.
- **Tracciabilità dei Collegamenti Dinamici tra DT:** È necessario un knowledge graph etichettato multi-grafo per tenere traccia dei collegamenti dinamici tra DT. Il sistema deve quindi offrire strumenti per navigare e interrogare questo grafo attraverso delle query.
- **Sincronizzazione tra PA e DT:** Il processo di shadowing (sincronizzazione continua tra PA e DT) deve rispettare i vincoli di servizio stabiliti nei modelli dei DT, come i tempi di risposta e altri parametri di qualità.
- **Esecuzione del Modello Operativo dei DT:** È fondamentale che i modelli operativi dei DT siano sempre disponibili in modalità eseguibile per governare la cattura degli eventi, gli aggiornamenti di stato, la sincronizzazione, il collegamento e il funzionamento interno del DT.
- **Servizi Indipendenti dal DT:** Alcuni servizi, come quelli per la creazione e l'interrogazione dei DT, devono essere accessibili anche senza riferimenti a un DT specifico.
- **Accesso Costante a Stato e Contesto del DT:** Gli stati correnti e passati dei DT, i modelli utilizzati, il flusso di eventi rilevati e altri dati contestuali devono essere accessibili in ogni momento da entità esterne, indipendentemente dalla loro eterogeneità.
- **Interazione e Azioni sui DT:**È necessario garantire la possibilità di interazione costante con i DT e, di conseguenza, con i PA per attivare azioni o funzioni. Queste interazioni generano eventi che forniscono feedback sull'azione e possono generare aggiornamenti di stato sia nel PA che nel DT.

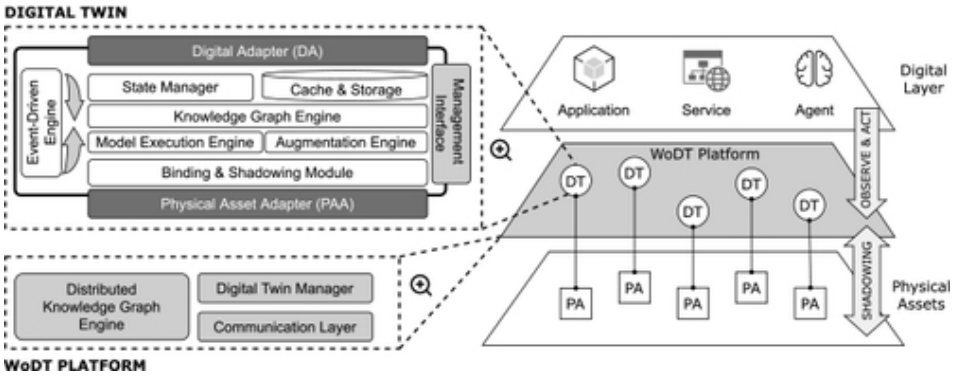

Figura 2.4: Architettura di un WoDT [12]

L'architettura base mostrata in figura 2.4 è concepita per soddisfare i requisiti chiave attraverso una suddivisione dei componenti in quattro categorie principali. Tra gli elementi interni al Digital Twin, il Model Execution Engine rappresenta il cuore operativo, eseguendo e gestendo il modello del sistema. Sul perimetro tra vari DT, il Management Interface agevola le connessioni e offre un monitoraggio centralizzato e strutturato, che facilita il collegamento tra DT e mantiene un controllo centralizzato e supervisionato sulle operazioni. I componenti che mediano tra i DT e i client esterni, come il Digital Adapter, semplificano l'accesso ai dati e alle funzionalità del DT, agevolando le interazioni con altri sistemi. Infine, tra le infrastrutture esterne al DT, il Digital Twin Manager (DTM) è il punto di riferimento per la gestione dell'intero sistema, garantendo la comunicazione tra eventi e un'ampia disponibilità di conoscenze distribuite, necessarie per mantenere una struttura efficiente [12].

- **Physical Asset Adapter (PAA)**:Questo componente si occupa di ricevere dati ed eventi dal PA associato, e di inviare eventi alla PA in risposta a richieste delle applicazioni tramite il Digital Adapter (DA) [12]. Sebbene le caratteristiche dei PA possano variare ampiamente, il PAA trasforma ogni evento in una rappresentazione uniforme, agevolando la compatibilità e il coordinamento tra diversi DT [12].

- **Binding e Shadowing Module (BSM)**: Questo modulo è centrale per la sincronizzazione, sia nel collegamento iniziale che nel mantenimento dell'allineamento continuo tra DT e PA. Il BSM collabora con l'Event-driven Engine per inviare aggiornamenti al Model Execution Engine (MEE), che regola quando e come aggiornare il Knowledge Graph (KG) e lo stato complessivo del DT [12].

- **Event-driven Engine (EE)**: Connette le varie parti del DT, facilita le interazioni tra i componenti, implementando un sistema basato su eventi.

Anche se il concetto base è quello di un bus di eventi, la sua effettiva implementazione può variare in base alle esigenze tecnologiche [12].

- **Model Execution Engine (MEE)**: Agisce come il "cervello" del DT, supervisionando i vari moduli secondo i modelli prestabiliti, decidendo quali eventi registrare e come influenzano lo stato e il comportamento del DT. Grazie al MEE, il progettista definisce l'operatività del DT, rendendo i modelli funzionali nella fase di esecuzione [12].

- **State Manager (SM)**:Questo modulo si occupa di monitorare e aggiornare lo stato del DT in conformità ai parametri definiti nel MEE, assicurando che gli aggiornamenti siano coerenti con le relazioni impostate con altri DT [12].

- **Knowledge Graph Engine (KGE)**: È responsabile della gestione del knowledge graph associato al DT, mantenendo i collegamenti con gli altri DT e rendendo le informazioni disponibili [12].

- **Cache e Storage (CS)**:Fornisce capacità di memorizzazione e caching, registrando aggiornamenti di stato, modifiche del KG e dati sugli eventi per un accesso rapido e una migliore efficienza del DT [12].

- **Management Interface (MI)**: Questa interfaccia gestisce le richieste di informazioni sul ciclo di vita del DT e permette la creazione o la connessione con altri DT all'interno del WoDT [12].

- **Digital Adapter (DA)**: Supporta le funzioni del PAA, rendendo gli eventi del DT accessibili a entità esterne attraverso traduzioni di eventi interni del DT in eventi che possono essere elaborati da entità esterne o l'osservazione delle proprietà del DT.

- **Augmentation Engine (AE)**:Questo componente aggiunge funzioni avanzate, permettendo la generazione di azioni predittive o di dati derivati, utilizzabili da applicazioni esterne per estendere le capacità del DT [12].

Questa architettura modulare e strutturata consente ai Digital Twins di comunicare in modo coordinato e fluido con gli asset fisici, mantenendo allineamenti continui e rendendo possibile la realizzazione di applicazioni scalabili e complesse[12].

La struttura include anche componenti esterni ai DT individuali per fornire supporto aggiuntivo e garantire un funzionamento efficace su scala. Uno di questi è il **Distributed Knowledge Graph Engine (DKGE)**, che permette la navigazione e la gestione della rete di DT, fungendo da grafo etichettato per

monitorare le connessioni tra i vari DT, facilitando l'accesso alle proprietà e ai flussi di dati.

Il **Digital Twin Manager (DTM)** coordina le fasi del ciclo di vita dei DT, dalla creazione all'eliminazione, e fornisce identificativi univoci per ogni DT[12]. Il **Communication Layer (CL)** agisce come interfaccia principale tra la piattaforma WoDT e le entità esterne, garantendo la possibilità di comunicazione con vari componenti [12].

2.2.4 Limitazioni delle Tecnologie Attuali e Soluzione

Dopo aver analizzato, nella sezione 1.5, gli strumenti attualmente disponibili sul mercato, emergono alcune considerazioni importanti. Al giorno d'oggi, sono ancora relativamente pochi gli strumenti che offrono una visione integrata e completa degli ecosistemi di Digital Twins, il che rappresenta un problema significativo, come già sottolineato in precedenza. La mancanza di strumenti che supportino una gestione e visualizzazione ad ecosistemi eterogenei limitano l'utilizzo dei Digital twin a situazioni più "semplici".

Tra gli strumenti analizzati, lo Azure Digital Twins si distingue come la soluzione più completa e avanzata. Questo strumento consente di visualizzare e interagire con il Knowledge Graph, permettendo una gestione efficiente dei dati provenienti dai Digital Twins. Tuttavia risulta essere un sistema chiuso, poiché non è possibile creare relazioni tra i DT di diverse istanze di Azure o con DT esterni alla piattaforma.

Ciò che manca a queste tecnologie è la possibilità di poter modellare ecosistemi di DT indipendentemente dalla tecnologia sottostante. Questo problema si risolve traendo ispirazione dagli standard del web, dall'architettura REST e dal concetto di Hypermedia.

2.3 WoDT Basata su Hypermedia

Sfruttando tecnologie ispirate dal Web come l'utilizzo di un'architettura REST e l'uso di Hypermedia è possibile risolvere il problema dell'eterogeneità tecnologica negli ecosistemi di Digital Twin.

Nei DT possiamo individuare due livelli di eterogeneità una a livello tecnologico e una a livello di modellazione. L'approccio basato sull'Hypermedia implementa una versione del WoDT sfruttando gli standard del web, ciò permette di ottenere un'interfaccia uniforme che nasconda l'eterogeneità dei DT consentendo la creazione di ecosistemi di DT aperti, navigabili e scopribili. Esponendo un'interfaccia uniforme, ogni DT all'interno di un ecosistema, permette ai consumatori di accedere ai dati ricavati dall'oggetto fisico.[7]

2.3.1 Piattaforma Web-Based WoDT

Una soluzione a questo problema è rappresentata dalla piattaforma Web-Based WoDT, concepita per offrire uno strumento navigabile e interattivo ispirato ai principi dell'architettura del web. Questa piattaforma adotta un approccio che sfrutta l'architettura REST, insieme a standard, protocolli e tecnologie web, per creare ecosistemi dinamici, aperti e longevi.

La piattaforma non solo realizza una versione implementabile del concetto di Web of Digital Twins, ma ne espande le potenzialità consentendo la creazione di ecosistemi composti da Digital Twin eterogene. Questo approccio facilita l'integrazione e la comunicazione tra Digital Twin di domini e organizzazioni diverse, mantenendo al contempo una visione coerente e condivisa dell'ecosistema complessivo [7]. In tal modo, si supera l'attuale frammentazione tipica delle piattaforme chiuse [7].

La piattaforma gestisce e unisce i dati dei DT, creando una vista contestualizzata della realtà e offrendo servizi applicativi sull'ecosistema. Per essere conforme alla visione HWoDT implementa un'interfaccia uniforme del HWoDT per esporre uno strato di compatibilità sulla sua infrastruttura tecnologica [7].

Capitolo 3

Uno Strumento per la Visualizzazione e Gestione di Ecosistemi di DT

In questo capitolo, ci concentreremo sull'analisi dei requisiti per la gestione e visualizzazione degli ecosistemi di Digital Twins, con un'attenzione particolare alla progettazione di una web-app dedicata alla visualizzazione del Knowledge Graph e esecuzione di query su un ecosistema di digital twin. Il tool si svilupperà sopra la piattaforma Web-Based WoDT, precedentemente descritta. L'obiettivo è sviluppare strumenti che migliorino l'interoperabilità e l'accessibilità, affrontando le necessità di una gestione più efficace dei dati provenienti dagli ecosistemi di DT, sfruttando i vantaggi offerti dalla piattaforma.

Inizieremo con l'identificazione dei requisiti funzionali e non funzionali per la nostra applicazione, tenendo conto delle esigenze degli utenti e delle sfide nel contesto della gestione dei Digital Twins. In questa fase, delineeremo le funzionalità chiave necessarie per l'interazione con il Knowledge Graph, come l'esecuzione di query avanzate e la visualizzazione intuitiva dei dati.

Successivamente, presenteremo un mockup dell'interfaccia utente, illustrando come gli utenti potranno navigare e interagire con il sistema. Questo mockup sarà una rappresentazione visiva del design, pensato per migliorare l'usabilità e l'esperienza dell'utente finale, facilitando l'accesso e l'analisi dei dati in modo semplice e diretto.

3.1 Visualizzazione e Gestione di Ecosistemi di DT

L'obiettivo di questa tesi i è sviluppare uno strumento che faciliti la gestione di ecosistemi di Digital Twin sfruttando una piattaforma Hypermedia-Based Web of Digital Twins esistente. Questo strumento, progettato per interagire con la piattaforma WoDT, intende offrire un'interfaccia uniforme per la gestione di DT che adottano modelli di dati e tecnologie diverse. L'idea di fondo è quella di ridurre la complessità tipica dell'integrazione e dell'interazione tra DT, rendendo l'ecosistema digitale più accessibile e navigabile, tanto per gli utenti finali quanto per i sistemi connessi.

Le motivazioni alla base della necessità di implementare questa applicazione derivano dalle sfide e dalle limitazioni riscontrate nella gestione dei Digital Twin in ambienti eterogenei e complessi. Con la crescente adozione del paradigma dei DT in settori come la produzione, la sanità e le smart city, è emersa l'esigenza di strumenti che facilitino la gestione e l'integrazione di questi ecosistemi digitali. Tuttavia, la varietà di modelli, tecnologie e protocolli utilizzati per sviluppare i DT crea ambienti frammentati, dove i diversi asset virtualizzati non sempre possono comunicare o operare in modo coordinato.

In particolare, molti degli strumenti attuali sono progettati per gestire singoli DT o gruppi di DT isolati, ma non supportano la creazione di ecosistemi interoperabili e dinamici. Questa mancanza di interoperabilità limita fortemente il potenziale dei DT come strumento per rappresentare e gestire digitalmente porzioni complesse della realtà, poiché ogni Digital Twin è spesso confinato a specifiche piattaforme o tecnologie proprietarie. Di conseguenza, diventa difficile visualizzare e comprendere le interazioni e le relazioni tra diversi DT all'interno di un ecosistema più ampio.

Il tool creato si propone di semplificare la gestione dei dati raccolti dai singoli DT interconnessi e anche offrire una visualizzazione di insieme dell'ecosistema evidenziandone le relazioni e gli aspetti dinamici tra i singoli asset. Per validare l'efficacia dello strumento sviluppato, la tesi prevede un caso di studio che simula un ecosistema di DT semplice, ma sufficientemente dinamico da mettere alla prova le capacità del tool di adattarsi e di supportare una gestione efficace. Questo caso di studio rappresenterà situazioni tipiche di un ecosistema digitale in continua evoluzione, osservando come il tool risponde a modifiche strutturali. In questo modo, si potrà verificare la flessibilità del tool nella gestione di ecosistemi DT complessi e la sua capacità di semplificare l'interazione tra le diverse entità.

3.2 Identificazione dei Requisiti

Sulla base dell'analisi delle tecnologie esistenti fatta nel capitolo precedente e delle funzionalità offerte dalle piattaforme attuali di Digital Twin, sono stati definiti i requisiti che guideranno lo sviluppo del tool.

La trattazione precedente ha evidenziato l'importanza di sviluppare strumenti idonei per la gestione di ecosistemi di Digital Twin, essenziali per supportare le complessità di un sistema interconnesso. Un aspetto cruciale per rendere la visualizzazione di questi ecosistemi facile e intuitiva è rappresentarli sotto forma di grafo, che mostri chiaramente le interazioni e le relazioni tra le varie componenti interconnesse. Questa rappresentazione visiva permette di comprendere immediatamente le connessioni e la struttura del sistema, rendendo più agevole l'analisi delle dipendenze e delle influenze reciproche tra i diversi Digital Twin.

È inoltre fondamentale che l'utente possa interagire attivamente con il grafo, accedendo in modo semplice a tutte le informazioni aggiornate provenienti dal mondo fisico. Questo include dati in tempo reale sullo stato di ogni Digital Twin e la possibilità di esplorare metriche, eventi e dettagli specifici relativi a ciascuna entità. Tale interattività non solo migliora l'esperienza dell'utente, ma consente anche un monitoraggio più efficace e tempestivo dell'ecosistema.

Da queste considerazioni derivano in modo naturale i requisiti necessari per l'implementazione del sistema, che includeranno sia aspetti funzionali, legati all'operatività e alle interazioni principali, sia requisiti non funzionali, che ne definiranno l'efficienza, la scalabilità e l'usabilità.

Nelle sezioni successive, verranno descritti nel dettaglio i requisiti funzionali e non funzionali, per garantire una comprensione completa degli obiettivi e delle caratteristiche essenziali del sistema.

3.2.1 Requisiti Funzionali

I requisiti funzionali sono elementi fondamentali per la progettazione e lo sviluppo di un sistema, poiché definiscono le funzionalità che il software deve offrire agli utenti. Questi requisiti stabiliscono le aspettative di comportamento del sistema e garantiscono che risponda adeguatamente alle esigenze degli utenti finali.

- **Visualizzazione del Knowledge Graph:** Il sistema deve supportare una visualizzazione intuitiva e completa del Knowledge Graph. Ogni nodo del grafo rappresenta un singolo Digital Twin, con la struttura del grafo che evidenzia le relazioni tra i vari DT in modo chiaro. L'idea iniziale è quella di mostrare una versione semplificata del KG dove saranno

visibili soltanto i Digital Twin in modo da evidenziarne gli aspetti dinamici e le interazioni con gli altri DT. Eventualmente si potrà aggiungere la possibilità di espandere un nodo del grafo in modo tale da mostrare quelli che sono gli aspetti statici sottoforma di Knowledge Graph del singolo DT.

- **Interazione con i Nodi del Grafo:**Gli utenti devono poter interagire con i nodi del grafo per esplorare i dati associati. Cliccando su un nodo, l'utente potrà osservare in tempo reale le Dashboard e le propietà relative al DT selezionato.

- **esecuzione di Query:** Il sistema deve supportare l'esecuzione di query avanzate sui dati del Knowledge Graph. e query permettono di estrarre informazioni specifiche su DT individuali e sulle loro interazioni. Il sistema deve supportare query basate su SPARQL, consentendo agli utenti di eseguire ricerche mirate e ottenere risposte in tempo reale.

3.2.2 Requisiti Non Funzionali

I requisiti non funzionali sono le specifiche che definiscono le caratteristiche generali del sistema, ma non si riferiscono direttamente alle funzioni che il sistema deve eseguire. Descrivono "come" il sistema deve comportarsi piuttosto che "cosa" deve fare.

- **Performance e Tempi di Risposta**: Il sistema deve essere in grado di fornire risposte rapide alle interazioni degli utenti. In particolare, la visualizzazione dei dati e l'esecuzione delle query devono avvenire in tempo reale o comunque in tempi di risposta che garantiscano un'esperienza utente fluida.

- **Usabilità**: Gli utenti devono essere in grado di navigare facilmente nel sistema senza necessitare di una formazione intensiva.

3.2.3 Modellazione delle Interazioni Utente-Sistema

In questa sezione, esploreremo le principali interazioni tra l'utente e la piattaforma, focalizzandoci su come l'attore principale interagisce con il sistema. Attraverso un diagramma UML dei casi d'uso, rappresenteremo le funzionalità chiave e i flussi di lavoro che descrivono come l'utente può accedere e utilizzare le risorse offerte dalla piattaforma. Questo diagramma fornisce una visione chiara delle operazioni che l'utente può eseguire, nonché delle relazioni tra le diverse azioni all'interno dell'ecosistema Web of Digital Twins.

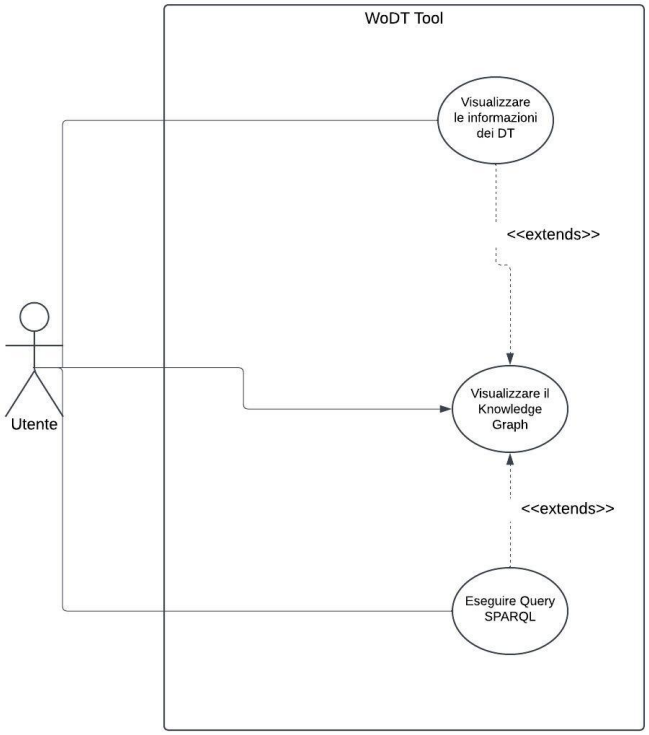

Figura 3.1: Diagramma dei casi d'uso

Come si può notare dalla Figura3.1, e in accordo con quanto precedentemente detto nei requisiti funzionali, l'utente può visualizzare il Knowledge Graph e interagire con esso, visualizzando i dati relativi ai nodi e eseguendo query. Nel diagramma, possiamo osservare una freccia di tipo "extend" tra le operazioni di query e visualizzazione delle informazioni. Questo è ovviamente dovuto al fatto che, per eseguire entrambe le operazioni, è necessario prima visualizzare il grafo.

Caso d'uso: Visualizzazione del Knowledge Graph

- Goal: Permettere all'utente di accedere a una rappresentazione visiva dell'ecosistema dei Digital Twins e di navigare tra i vari nodi.
- Attore Principale: Utente
- Postcondizioni: L'utente può visualizzare una mappa del Knowledge Graph, in cui ogni nodo rappresenta un Digital Twin.

Caso d'uso: Esecuzione di Query sul Knowledge Graph

- Goal: Consentire all'utente di eseguire query avanzate per estrarre informazioni specifiche sui Digital Twins e le loro interazioni.
- Precondizioni: L'utente ha visualizzato il Knowledge Graph e ha accesso ai permessi di interrogazione dei dati.
- Postcondizioni: L'utente può visualizzare una mappa del Knowledge Graph, in cui ogni nodo rappresenta un Digital Twin.
- Sequenza Principale degli Step:
  1. L'utente inserisce una query SPARQL sul Knowledge Graph.
  2. L'utente esegue la query.
  3. Il sistema elabora e visualizza i risultati, evidenziando i nodi e le relazioni richiesti.

Caso d'uso: Visualizzazione dei Dati del Nodo

- Goal: Consentire all'utente di esplorare i dati associati a un Digital Twin selezionato.
- Attore Principale: Utente
- Precondizioni: L'utente deve aver visualizzato il Knowledge Graph.
- Postcondizioni: I dettagli del nodo vengono visualizzati in un pannello laterale.
- Sequenza Principale degli Step:
  1. L'utente clicca su un nodo del Knowledge Graph.
  2. Il sistema recupera i dati associati al Digital Twin selezionato.
  3. Il sistema mostra i dati del nodo in tempo reale, compresi attributi come proprietà attuali, eventi, e possibili azioni.

3.3 Progettazione dell'Interfaccia Utente

In questa sezione, presentiamo i mockup realizzati per l'interfaccia utente del tool. Questi mockup sono stati progettati per illustrare le funzionalità e l'aspetto visivo dell'applicazione, facilitando una comprensione immediata di come gli utenti interagiranno con il sistema. I mockup mostrano le schermate

principali, le navigazioni e le interazioni previste, fornendo una base visiva per il successivo sviluppo e implementazione del tool. Attraverso questo processo, ci proponiamo di garantire un'esperienza utente intuitiva e accessibile.

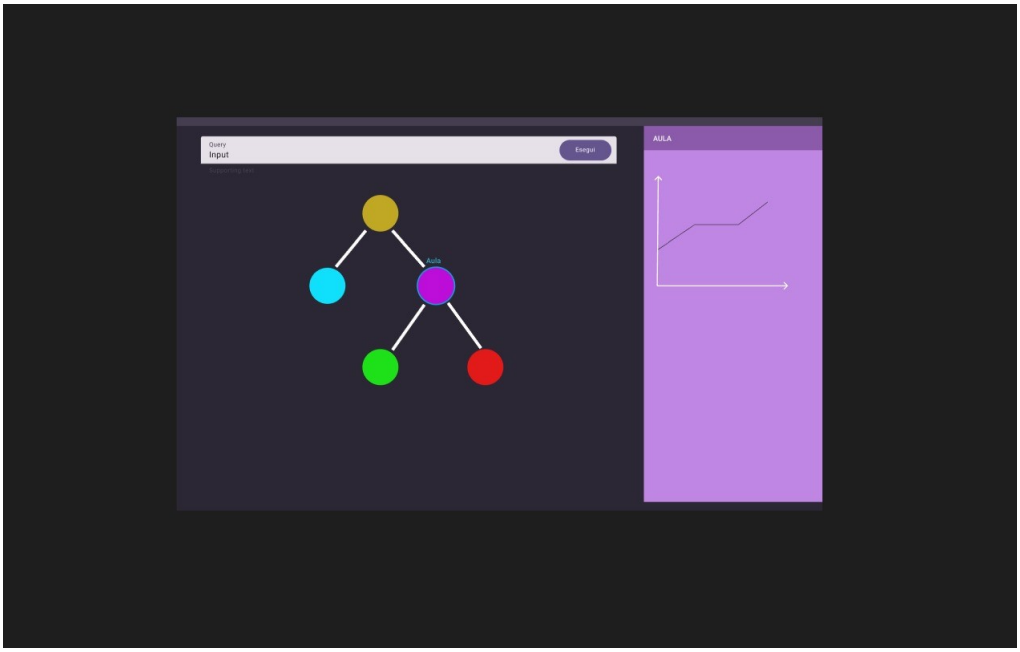

Figura 3.2: Mockup della Versione Desktop

Sono stati creati due mockup: uno per la modalità desktop (Figura 3.2) e l'altro per l'interfaccia in modalità mobile(Figura 3.3).

L'interfaccia utente è stata progettata con l'obiettivo di renderla il più semplice ed intuitiva possibile, per garantire un'esperienza fluida e accessibile anche agli utenti meno esperti. È stata posta particolare attenzione all'usabilità, cercando di ridurre al minimo la complessità e di offrire un'interazione chiara e diretta con le funzionalità principali. La piattaforma offre due modalità di visualizzazione: una versione desktop e una mobile, per garantire un'esperienza ottimale su schermi di diverse dimensioni e dispositivi. In questo modo, gli utenti possono accedere alla piattaforma in modo efficiente sia da computer desktop che da dispositivi mobili, mantenendo sempre una visualizzazione chiara e un'interazione semplice, indipendentemente dal tipo di dispositivo utilizzato. L'interfaccia permette la visualizzazione del grafo che rappresenta una struttura semplificata delle relazioni tra i vari Digital Twin interconnessi, fornendo così una panoramica chiara e immediata dell'ecosistema, dove ogni nodo del grafo rappresenta un singolo DT. Cliccando su un nodo l'utente potrà visualizzare le informazioni relative alle propità del DT. Nella parte superiore della schermata è presente un modulo dedicato all'inserimento

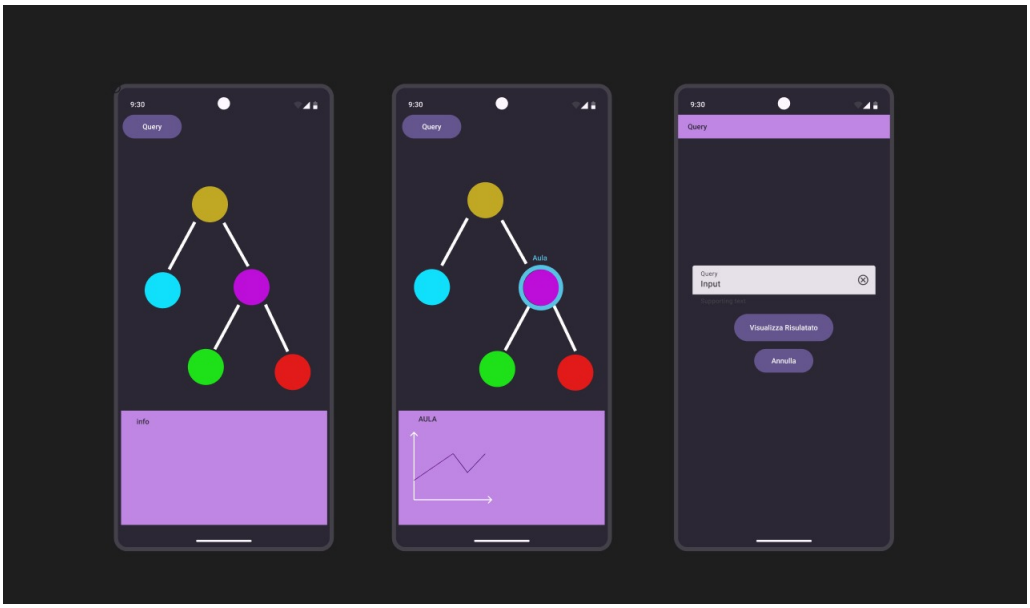

Figura 3.3: Mockup della Versione Mobile

di query SPARQL, uno strumento potente per interrogare e analizzare i dati all'interno dell'ecosistema dei DT. Inserendo una query SPARQL e avviando l'esecuzione, il sistema elabora la richiesta e ne visualizza immediatamente il risultato applicato sul grafo.

Al centro dello schermo viene visualizzato il grafo semplificato dei DT, che mostra i nodi principali e le relazioni tra di essi. Nella parte superiore, è presente un campo di input per l'inserimento delle query. Questa form permette agli utenti di eseguire query sui dati del grafo. Quando l'utente seleziona un nodo del grafo, un pannello laterale si attiva, mostrando le informazioni dettagliate relative al Digital Twin selezionato. Qui vengono visualizzate le proprietà del DT selezionato.

Capitolo 4

Sviluppo di un Prototipo

Nel seguente capitolo verrà analizzata in dettaglio l'implementazione del tool. Saranno descritte le tecnologie utilizzate, le scelte progettuali, e le fasi di sviluppo che hanno portato alla creazione del tool.

4.1 Tecnologie Utilizzate

In questa sezione, vengono descritte le principali tecnologie adottate per lo sviluppo del tool. Le principali tecnologie utilizzate nel progetto includono:

- Piattaforma Web-Based WoDT: per la gestione e l'integrazione dei Digital Twin.
- N3.js [1]: per il trattamento dei dati RDF in ambiente JavaScript.
- Vis Network [2]: per la visualizzazione interattiva del KG.
- White Label Digital Twin Framework (WLDT): per la creazione di Digital Twin personalizzabili [10].

Nei prossimi paragrafi, ciascuna di queste tecnologie sarà esplorata nel dettaglio, evidenziando il loro ruolo e il modo in cui contribuiscono al funzionamento del sistema WoDT.

Piattaforma Web-Based WoDT: Come già detto nel capitolo 2, la Piattaforma Web-Based WoDT è stata sviluppata per facilitare la gestione e l'integrazione di ecosistemi eterogenei di Digital Twin attraverso una struttura semantica e standardizzata. Aggrega i dati dai DT registrati ed espone un'inter-

[1] https://rdf.js.org/N3.js/
[2] https://visjs.github.io/vis-network/docs/network/

faccia verso i consumatori per permettere ad essi di interagire con l'ecosistema [7].

I DT possono registrarsi sulla piattaforma WoDT per entrare a far parte dell'ecosistema, con due modalità di registrazione. Una modalità è automatica tramite shadowing, in cui i DT possono auto-registrarsi al momento dell'attivazione; l'altra è manuale, gestita dall'amministratore della piattaforma. In caso di modifiche all'asset fisico, i DT notificheranno la piattaforma, la quale aggiornerà i dati del Digital Twin Descriptor (DTD) e di conseguenza il KG. Il DTD contiene una descrizione semantica statica dei metadati del DT a differenza del Digital Twin Knowledge Graph che invece contiene lo stato attuale dell'asset fisico e contiene triple RDF che comprendono valori delle proprietà, relazioni correnti e azioni disponibili. Se un Dt non dove essere più attivo o nel caso che l'asset fisico fosse rimosso, la piattaforma consentirà di eliminare il DT e tutti i dati associati al KG. La rimozione potrà essere richiesta dal DT stesso o dall'amministratore della piattaforma [7].

Per poter accedere ai dati del DTKG e eseguire query la piattaforma mette a disposizione delle REST-API

N3: N3.js è una libreria JavaScript progettata per manipolare e lavorare con dati RDF. Basata su RDF.js, N3.js fornisce un'interfaccia di programmazione che facilita il parsing, la creazione, l'elaborazione e la serializzazione di dati RDF. Questa libreria verrà utilizzata per facilitare la conversione dal formato RDF Turtle in formato JSON-LD per rendere più semplice la gestione dei dati in un ambiente Javascript.

Vis Network: Vis Network è una libreria JavaScript della suite vis.js utilizzata per creare e visualizzare reti complesse e grafi interattivi direttamente all'interno di applicazioni web. Questa libreria consente di rappresentare nodi e connessioni tra di essi, rendendo possibile la visualizzazione di dati in forma grafica.

Con Vis Network, è possibile:

- Personalizzare l'aspetto di nodi e bordi.
- Impostare parametri per layout automatici e dinamici
- Esplorare i grafi tramite funzionalità di zoom, drag-and-drop e clustering.

La libreria fornisce anche eventi interattivi per rilevare clic, selezioni e hover su elementi specifici della rete, facilitando la creazione di esperienze utente altamente interattive e dinamiche. Inoltre, Vis Network supporta ampie reti con migliaia di nodi, mantenendo un'ottima performance, grazie a un motore di rendering ottimizzato.

White Label Digital Twin Framework framework open-source progettato per facilitare la creazione e gestione di Digital Twin personalizzabili. Sviluppato per offrire flessibilità e interoperabilità. Il sistema è progettato per rendere più semplice la creazione e lo sviluppo dei Digital Twin. Infatti consente agli sviluppatori di creare facilmente nuove istanze di DT utilizzando moduli preesistenti o personalizzando il comportamento in base alle necessità specifiche. Pur mantenendo la semplicità, l'API è facilmente estendibile per permettere l'aggiunta di nuove funzionalità o moduli. Inoltre, WLDT è progettato per essere portabile consentendo la creazione di applicazioni DT indipendenti e modulari. Attraverso WLDT, gli sviluppatori possono creare Digital Twin che rispecchiano in tempo reale lo stato degli asset fisici e che si connettono facilmente con ecosistemi più ampi, garantendo un flusso dati continuo e una sincronizzazione ad alta fedeltà tra i DT e le loro controparti fisiche [10].

Le componenti principali dell'architettura sono:

- Digital Twin Engine: È il motore multi-thread che consente l'esecuzione simultanea di più DT. Orchestrando i vari moduli, gestisce l'esecuzione e il monitoraggio dei Digital Twin.
- Digital Twin: La struttura modulare del DT, che combina le funzionalità core con gli adattatori fisici e digitali. Gestisce lo stato del DT e definisce le proprietà, gli eventi e le azioni.
- Shadowing Function: Gestisce il processo di sincronizzazione tra il DT e il suo asset fisico, aggiornando lo stato del DT in base ai cambiamenti del PA.
- Physical Adapter: Implementa le funzionalità necessarie per gestire la comunicazione con l'asset fisico, descrivendo le sue proprietà, eventi, azioni e relazioni.
- Digital Adapter: Permette l'esposizione dello stato e delle funzionalità del DT tramite protocolli digitali esterni

Per creare un Digital Twin è necessario una funzione di Shadowing, un Digital Adapter e un Physical Adapter. Un DT potrebbe avere anche più Physical Adapter [10].

4.2 Implementazione

In questa sezione viene descritta l'implementazione del sistema per la visualizzazione e l'interazione con un Knowledge Graph contenente dati relativi

a Digital Twin. L'architettura del sistema è progettata per offrire un'esperienza dinamica e interattiva, consentendo agli utenti di esplorare i dati, eseguire query SPARQL e visualizzare i risultati direttamente all'interno di una pagina web. L'implementazione si concentra su tre principali funzionalità: la visualizzazione del Knowledge Graph, l'esecuzione di query SPARQL e l'esplorazione dei dati dei Digital Twin.

4.2.1 Visualizzazione del Knowledge Graph

Il processo descritto consente di visualizzare e aggiornare dinamicamente un knowledge graph all'interno di una pagina web, rappresentando nodi e relazioni tra di essi in modo interattivo. La visualizzazione grafica utilizza una libreria dedicata, Vis.js, per creare una rappresentazione chiara e navigabile dei dati. Il sistema è in grado di recuperare informazioni strutturate in formato JSON-LD, convertite da dati RDF in formato Turtle, permettendo così una facile integrazione di nuove informazioni.

In particolare, il flusso di lavoro gestisce le seguenti fasi:

- Inizializzazione del grafo: initializeGraph() configura una **network** di Vis.js, se il grafo non è ancora inizializzato, crea la rete e abilita l'interazione con i nodi.

- Recupero del knowledge graph: getKnowledgeGraph() effettua una richiesta HTTP per recuperare i dati RDF in formato Turtle. Dopo aver ricevuto i dati li converte in JSON richiamando la funzione parseTurtleToJSONLD(turtleData) e successivamente restituisce i dati nel formato corretto.

- Conversione dei dati da Turtle a JSON: Una volta ottenuti, i dati vengono convertiti in JSON-LD tramite parseTurtleToJSONLD(turtleData) che utilizza la libreria N3 per analizzare i dati in formato turtle restituendo un array di triplette (soggetto, predicato, oggetto) in formato JSON-LD. La funzione parseTurtleToJSONLD restituisce una Promise, che consente di gestire il processo asincrono di parsing dei dati. La Promise chiama resolve quando la conversione è completata e reject in caso di errori.

  - N3.Parser(): Crea un parser RDF che interpreta i dati Turtle.
  - N3.Store(): Crea un archivio temporaneo per memorizzare le triplette RDF.
  - let jsonld = []: Inizializza un array vuoto jsonld dove verranno memorizzati i risultati finali in formato JSON-LD.

Il parser esamina i dati Turtle passati come parametro (turtleData) e restituisce i risultati sotto forma di triplette RDF (quad) attraverso una funzione di callback. Una volta terminato il parsing, store.getQuads(null, null, null, null) recupera tutte le triple RDF memorizzate. Ogni tripletta viene trasformata in un oggetto JSON. Per ogni q (una tripletta), il subject, predicate, e object vengono convertiti in proprietà JSON: soggetto, predicato e oggetto.

```
    function parseTurtleToJSONLD(turtleData) {
    return new Promise((resolve, reject) => {
        const parser = new N3.Parser();
        const store = new N3.Store();
        let jsonld = [];

        parser.parse(turtleData, (error, quad) => {
            if (error) return reject(error);
            if (quad) store.addQuad(quad);
            else {
                jsonld = store.getQuads(null, null, null,
                    null).map(q => ({
                    subject: q.subject.value,
                    predicate: q.predicate.value,
                    object: q.object.value
                }));
                resolve(jsonld);
            }
        });
    });
}
```

Listato 4.1: Conversione dei dati da Turtle a JSON

- Creazione e aggiornamento del grafo: createGraph(data) utilizza i dati JSON-LD per creare nodi e archi nel grafo, aggiungendo nodi solo se sono tra i digital twins (nodi di interesse nel knowledge graph). La visualizzazione del grafo si aggiorna automaticamente ogni pochi secondi, consentendo di visualizzare in tempo reale eventuali cambiamenti e nuove connessioni.

4.2.2 Esecuzione delle Query

La visualizzazione delle query nel sistema avviene attraverso un'interazione semplice e diretta con un'interfaccia utente che consente di eseguire query

SPARQL per estrarre dati dal knowledge graph. Quando un utente invia una query, il sistema invia la richiesta al server e, una volta ricevuti i risultati, aggiorna dinamicamente la visualizzazione del grafo.

- Recupero della Query: L'utente inserisce una query SPARQL in un campo di testo dedicato e invia la richiesta. Questo avviene premendo un pulsante di invio. La query viene inviata alla piattaforma, dove viene elaborata per estrarre i dati richiesti.

- Gestione della Risposta: Una volta ricevuta la risposta dalla piattaforma, che contiene i risultati della query in formato JSON, il sistema esegue un'elaborazione dei dati.

- Aggiornamento del grafo: I nodi esistenti nel grafo vengono sostituiti con quelli nuovi derivanti dai risultati della query. La visualizzazione grafica viene aggiornata, e i nodi pertinenti vengono aggiunti alla rete in modo dinamico.

- Reset dei Risultati: Dopo l'esecuzione della query, viene fornito un pulsante di reset che consente di ripristinare la visualizzazione precedente del grafo. Quando il pulsante viene premuto, i nodi precedenti vengono ripristinati e la visualizzazione ritorna allo stato iniziale, consentendo un nuovo caricamento o l'esecuzione di una nuova query.

4.2.3 Esplorazione dei dati dei Digital Twin

Quando l'utente interagisce con il grafo, selezionando un nodo, il sistema recupera ulteriori dettagli su quel nodo specifico. Questo avviene tramite una richiesta ad un endpoint che restituisce dati. I dati recuperati vengono presentati all'utente sotto forma di una tabella che mostra le proprietà del nodo selezionato e i relativi valori. Ogni proprietà rappresenta un attributo del Digital Twin, mentre il valore associato a quella proprietà può essere un dato in tempo reale, come temperatura o pressione, o informazioni legate al comportamento dell'oggetto fisico che il nodo rappresenta.

```
async function getDigitalTwinData(digitalTwinUri) {
try {
    const response = await fetch(digitalTwinUri);
    if (!response.ok) {
        throw new Error(`Error fetching data for Digital Twin:
            ${response.status}`);
    }
```

```
        const data = await response.text();
        displayTwinData(data);
    } catch (error) {
        console.error('Error fetching digital twin data:', error);
    }
}
```

Listato 4.2: Funzione per la richiesta dei dati dei singoli nodi del Knowledge Graph

```
    function createTable(jsonldData) {
    const table = document.createElement('table');
    table.classList.add('table', 'table-striped');

    const thead = document.createElement('thead');
    thead.innerHTML = `<tr><th>Property</th><th>Object</th></tr>`;
    table.appendChild(thead);

    const tbody = document.createElement('tbody');
    jsonldData.forEach(triple => {
        const row = document.createElement('tr');
        const propertyCell = document.createElement('td');
        propertyCell.textContent = triple.predicate || 'N/A';
        const objectCell = document.createElement('td');
        objectCell.textContent = triple.object || 'N/A';
        row.appendChild(propertyCell);
        row.appendChild(objectCell);
        tbody.appendChild(row);
    });

    table.appendChild(tbody);
    return table;
}
```

Listato 4.3: Creazione della tabella per mostrare i dati

4.3 Caso di Studio

Per testare il software sviluppato, è stato creato un caso di studio che, sebbene semplice, offre una visione complessiva dell'ecosistema. Questo caso di studio è stato progettato con l'intento di evidenziare i vari aspetti dinamici e le interazioni tra i digital twin.

4.3.1 Descrizione del Caso di Studio

Il caso di studio progettato si basa sulla creazione di un ecosistema di digital twin, con l'obiettivo di simulare e monitorare le dinamiche interne a un ambiente fisico complesso. In particolare, è stato modellato il digital twin di un campus, al quale sono state integrate le rappresentazioni virtuali di due stanze situate all'interno della struttura. Per approfondire l'analisi delle dinamiche e degli aspetti interattivi tra i vari componenti dell'ecosistema, è stato aggiunto un ulteriore digital twin che rappresenta una persona all'interno del campus. Questo digital twin della persona viene costantemente monitorato, con particolare attenzione ai suoi spostamenti tra le diverse stanze del campus. Grazie a questa integrazione, è possibile tracciare in tempo reale i movimenti dell'individuo, così come le interazioni con l'ambiente circostante. Ciò permette di osservare l'evoluzione delle relazioni tra i digital twin in un contesto dinamico, mettendo in evidenza come i dati relativi agli spostamenti possano influire sulle simulazioni in corso, garantendo una visione precisa e interattiva dell'ecosistema.

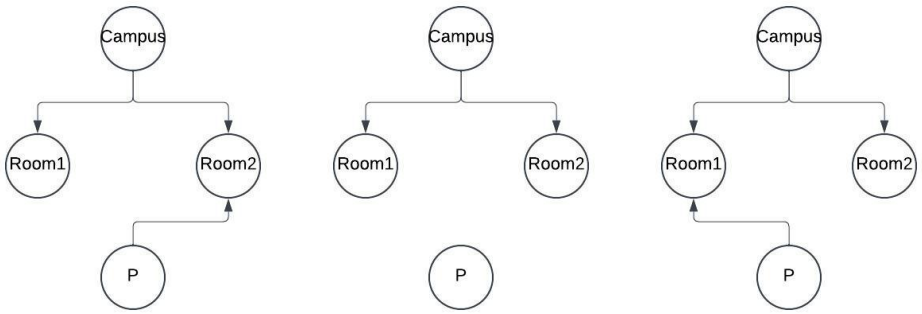

Figura 4.1: Rappresentazione delle relazioni tra i vari DT del caso d'uso

Come si può osservare negli schemi riportati sopra, esistono due tipi di relazioni tra gli oggetti rappresentati. Tra il campus e le stanze c'è una relazione statica: il campus, infatti, include le stanze al suo interno, come una struttura che le contiene permanentemente. Questa relazione è di natura fissa, poiché il campus è il "contenitore" delle stanze. Al contrario, la relazione tra le stanze e la persona è di tipo dinamico. Quando una persona entra in una stanza, si crea una connessione temporanea tra la persona e quella stanza. Tale relazione persiste finché la persona rimane all'interno della stanza. Nel momento in cui la persona esce dalla stanza, questa connessione viene eliminata, poiché non sussiste più la condizione di "presenza" nella stanza stessa. Inoltre, è stato modellato anche il caso in cui una persona non si trovi all'interno di nessuna delle

stanze. In questo scenario, non esiste alcuna relazione attiva tra la persona e le stanze, e la persona rimane fuori dal contesto delle stanze modellate.

4.3.2 Modellazione e Componenti dei Digital Twin

Nel caso di studi preso in considerazione vengono modellati tre tipi Digital Twin Campus, Room, Person. La distinzione delle due stanze verrà poi modellata in seguito. Ogni Digital Twin nel nostro sistema è strutturato attorno a tre componenti principali che ne definiscono il funzionamento e le modalità di interazione con il mondo fisico e digitale.

- Digital Adapter della piattaforma Web-based WODT
- Physical Adapter WLDT
- Ontologia

Tutti i Dt sono poi collegati ad una funzione di Shadowing comune che ha il compito di mantenere sincronizzato il mondo fisico e digitale.

Ontologia: Le classi che implementano l'interfaccia DTOntology hanno il compito di definire:

- Tipo del DT: con il metodo *getDigitalTwinType()* restituisce l'URI che identifica il tipo di Digital Twin.
- Mappatura delle proprietà: La mappa associa nomi di proprietà (ad esempio, "room-name") agli URI delle proprietà semantiche
- Gestione delle proprietà:
  - *obtainProperty()* Verifica se una determinata proprietà è definita nell'ontologia e ne restituisce un Property, se disponibile.
  - *obtainPropertyValueType()* Restituisce il tipo di valore atteso per una data proprietà (ad esempio, "string" per il nome della stanza).
  - *convertPropertyValue()* Converte un valore generico associato a una proprietà in un nodo utilizzabile nell'ontologia, permettendo così al sistema di riconoscere e gestire correttamente le istanze della proprietà.
- Gestione delle relazioni:
- Gestione delle azioni:

Shadowing Function: La classe *UseCaseShadowingFunction* è responsabile del mantenimento della sincronizzazione tra il Digital Twin e l'asset fisico, gestendo eventi, proprietà e relazioni. Alcune sue funzionalità sono:

- Gestione degli eventi di variazione delle proprietà: *onPhysicalAssetPropertyVariation()* rileva e gestisce le variazioni delle proprietà dell'asset fisico. Ogni volta che una proprietà fisica cambia, il metodo inizia una transazione sullo stato del DT, aggiorna la proprietà corrispondente e conclude la transazione

- Gestione delle relazioni: *onPhysicalAssetRelationshipEstablished()* gestisce l'evento di creazione di una relazione tra l'asset fisico e un altro oggetto. Il metodo crea un'istanza della relazione, avvia una transazione sullo stato del DT, aggiunge l'istanza della relazione e conclude la transazione. Il metodo *onPhysicalAssetRelationshipDeleted()* si occupa della rimozione di una relazione.

- Associazione e disassociazione del Digital Twin: *onDigitalTwinBound()* è uno dei metodi più rilevanti e si attiva quando il Digital Twin viene associato (bound) con il proprio asset fisico. Il metodo inizia una transazione sullo stato del DT, itera su tutte le propietà , azioni e relazioni dichiarate negli PhysicalAssetDescription degli adattatori fisici collegati, aggiornando lo stato del DT:

  - Proprietà: Crea proprietà digitali corrispondenti alle proprietà fisiche e imposta l'osservazione per aggiornamenti futuri.
  - Azioni: Abilita azioni digitali per i tipi di azioni che l'asset fisico supporta.
  - Relazioni: Crea relazioni tra il Digital Twin e altri elementi specificati, osservando eventuali aggiornamenti di relazione.

  infine conclude la transazione e inizia l'osservazione.

Physical Adapter: Le classi CampusPA, PersonPA, RoomPA estendono PhysicalAdapter e svolgono una funzione di collegamento tra l'entità fisica e il DT, assicurando che le informazioni sullo stato e le proprietà dell'oggetto fisico siano continuamente aggiornate e sincronizzate con la rappresentazione digitale..

Ogni adattatore fisico è responsabile di:

- Mappare le Proprietà Fisiche: Gli adattatori raccolgono e trasmettono le caratteristiche specifiche dell'entità fisica, come nome, posizione o stato.

- Definire e Gestire le Relazioni Fisiche: Gli adattatori modellano le relazioni tra l'entità fisica e altre entità.
- Rilevare Eventi e Azioni: Gli adattatori possono gestire e segnalare eventi che accadono nell'entità fisica e rispondere alle azioni richieste dal sistema DT.

Il metodo *onAdapterStart()* viene invocato all'avvio dell'adattatore e ha il compito di configurare l'entità, compresa la descrizione delle sue proprietà e delle sue relazioni. All'interno di questo metodo viene creato un PhysicalAssetDescription, ovvero un oggetto che descrive l'entità fisica del DT e aggiunge propietà e definisce relazioni. Il Physical Adapter invia una notifica (notifyPhysicalAdapterBound) che segnala l'avvio e l'associazione dell'adattatore fisico. Utilizzando publishPhysicalAssetRelationshipCreatedWldtEvent, l'adattatore pubblica eventi di creazione di relazione.

Collegamento con il Digital Adapter e creazione dei DT: Per creare e configurare un Digital Twin, bisogna seguire una serie di passaggi che coinvolgono l'integrazione di diversi componenti, come il Physical Adapter, il Digital Adapter e la funzione di sincronizzazione dei dati tra il mondo fisico e il digitale. Ogni Digital Twin è associato a un Digital Adapter, che è responsabile dell'interazione tra il Digital Twin e la piattaforma digitale. Questo adapter si configura con informazioni come:

- L'URL di connessione: l'indirizzo al quale il Digital Twin sarà esposto sulla rete.
- Ontologia.
- Il numero di porta: la porta sulla quale il Digital Twin ascolta le richieste o invia i dati.
- : Physical Adapter.

Il Digital Adapter deve essere configurato per comunicare con la piattaforma, il che implica l'uso di un URL della piattaforma a cui il Digital Twin è associato. Una volta configurati, i Digital Twin devono essere poi avviati.

4.3.3 Modellazione degli aspetti dinamici

Il Physical Adapter della persona (PersonPA), gestisce dinamicamente il movimento tra diverse stanze, aggiornando costantemente lo stato del Digital Twin associato. Questo avviene tramite l'aggiornamento di una relazione che

collega la persona alle stanze, riflettendo il cambiamento della sua posizione nel contesto fisico. Una relazione, denominata "person-in-room", collega la persona a una stanze. Ogni volta che la persona si sposta da una stanza all'altra, o quando lascia una stanza, questa relazione viene aggiornata per riflettere il cambiamento della sua posizione.

La funzione *updateRoomRelationship(uri)* gestisce l'aggiornamento della relazione tra una persona e una stanza. Se l'URI fornito (uri) rappresenta una stanza diversa da quella attuale, la funzione esegue i seguenti passaggi:

- Se la persona era già associata a una stanza precedente, la relazione con quella stanza viene eliminata (PhysicalAssetRelationshipInstanceDeletedWldtEvent)

- Viene creata relazione con la nuova stanza (PhysicalAssetRelationshipInstanceCreatedWldtEvent), aggiornando lo stato del currentRoom (campo che tiene traccia dell'uri della stanza in cui si trova la persona in quel monìmento) per riflettere il nuovo ambiente.

- Se l'URI passato rappresenta l'assenza da qualsiasi stanza, la relazione con la stanza attuale viene eliminata e il currentRoom viene aggiornato per indicare che la persona non si trova in nessuna stanza.

La simulazione del movimento della persona avviene tramite un processo asincrono, che viene avviato quando l'adattatore fisico entra in funzione. Il flusso di movimento segue questi passaggi: La persona viene inizialmente associata a una stanza, successivamente, viene dissociata dalla stanza (la relazione viene eliminata).

- La persona viene associata a una stanza.

- La relazione "person-in-room" viene aggiornata (associando o dissociando la persona dalla stanza).

- La persona lascia la stanza dopo un intervallo di tempo.

- La persona si sposta in una nuova stanza e ripete il processo.

Questo ciclo viene ripetuto per un numero predefinito di iterazioni, con intervalli temporali tra ciascun spostamento, emulando un comportamento di movimento tra stanze o di assenza da esse.

Il codice sottostante rappresenta il processo precedentemente descritto.

```
    private Runnable personMovementEmulation(){
       return ()-> {

           try {

               for(int i = 0; i< 10; i++){
                   Thread.sleep(5000);
                   updateRoomRelationship(room1Uri);
                   Thread.sleep(5000);
                   updateRoomRelationship(NOT_IN_ANY_ROOM);
                   Thread.sleep(5000);
                   updateRoomRelationship(room2Uri);
                   Thread.sleep(5000);
                   updateRoomRelationship(NOT_IN_ANY_ROOM);
               }

           } catch (InterruptedException e) {
               e.printStackTrace();
           }
       };
   }
```

Per garantire che il movimento della persona non blocchi altre operazioni, l'emulazione del movimento avviene in un thread separato. In questo modo, il comportamento dinamico della persona viene gestito in parallelo, senza interferire con il flusso principale dell'applicazione.

```
   new Thread(personMovementEmulation()).start();
```

4.4 Risultato finale

In questa sezione verranno presentati i risultati ottenuti dall'implementazione, con una descrizione dettagliata delle funzionalità disponibili. Saranno inclusi screenshot dell'interfaccia utente per fornire una panoramica visiva e facilitare la comprensione delle operazioni.

Dopo aver collegato i DT alla piattaforma, nella schermata principale, come mostrato in figura 4.2, sarà visibile il grafo principale, che si aggiorna per riflettere i cambiamenti dinamici nelle relazioni tra i vari Digital Twin. Ogni nodo del grafo rappresenta un Digital Twin e la sua posizione e connessione cambiano in base alle interazioni e agli aggiornamenti dell'ecosistema.

Figura 4.2: Schermata con visualizzazione del knowledge Graph

Cliccando su un nodo specifico, è possibile visualizzare una versione estesa del knowledge graph relativo a quel nodo, come mostrato in figura 4.3. I nodi rossi rappresentano i singoli componenti del knowledge graph del DT selezionato. Accanto al grafo, nella tabella, vengono visualizzate anche le proprietà dettagliate relative al Digital Twin selezionato

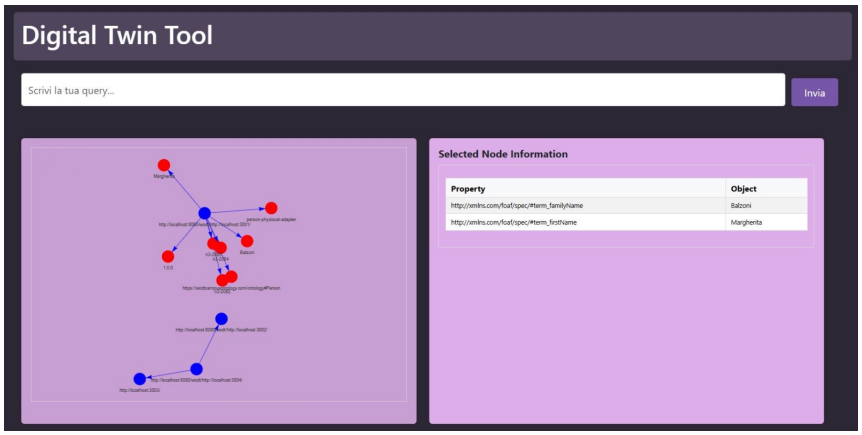

Figura 4.3: Schermata che mostra l'espansione del nodo del grafo e la tabella con le propietà

In alto nella schermata, è presente un campo dedicato all'inserimento di query. Se il risultato della query corrisponde a un Digital Twin specifico, verrà mostrato un grafo contenente esclusivamente il nodo corrispondente al risultato della ricerca. In ogni caso, verrà visualizzata una tabella che riassume i risultati della query, indicando tutte le informazioni rilevanti. Una volta eseguita la query, sarà visibile un pulsante di "reset" che consente di annullare

la ricerca e ripristinare la visualizzazione del grafo nella sua configurazione originale (Figura 4.4).

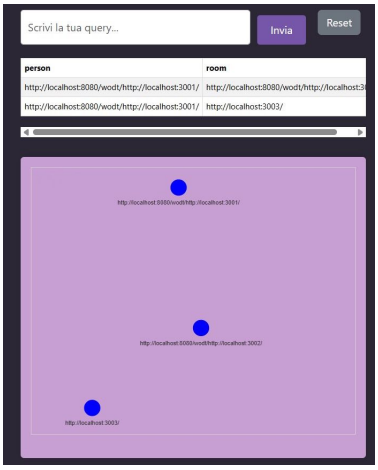

Figura 4.4: Schermata che mostra i risultati dell'esecuzione di una query

Conclusioni

Il lavoro svolto in questa tesi ha mirato a colmare un'importante lacuna nella gestione e visualizzazione degli ecosistemi di Digital Twin all'interno della visione Web of Digital Twins. Le soluzioni esistenti, sebbene efficaci per applicazioni verticali, spesso si scontrano con limiti significativi in termini di interoperabilità, apertura e navigabilità degli ecosistemi digitali.

Gli strumenti sviluppati dimostrano che è possibile rendere un ecosistema di DT più accessibile e navigabile, anche in presenza di tecnologie e modelli di dati differenti.Uno dei principali risultati ottenuti è stata la realizzazione di un'interfaccia web che consente di gestire ecosistemi eterogenei di DT, integrando funzionalità di visualizzazione interattiva e strumenti per eseguire query. Questi elementi non solo migliorano la comprensione e l'accessibilità dei dati, ma permettono anche una gestione più dinamica e trasparente delle relazioni tra i DT.

Il lavoro ha evidenziato che il paradigma WoDT, quando implementato attraverso un approccio basato su tecnologie Web, può rappresentare una soluzione concreta per abilitare applicazioni che operano in contesti e organizzazioni diverse. Ciò consente di trasformare un insieme di DT in un ecosistema interconnesso che offre un valore aggiunto significativo sia per le applicazioni che per gli utenti finali. La possibilità di navigare tra i dati dei DT e di osservare le relazioni e gli stati delle entità fisiche rappresentate, offre un'innovazione sostanziale nella gestione delle realtà digitali e ha offerto un esempio pratico di come gli ecosistemi di DT possano essere resi più accessibili e integrati.

Bibliografia

[1] Dean Allemang and James Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011.

[2] Mohsen Attaran and Bilge Gokhan Celik. Digital twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal*, 6:100165, 2023.

[3] Tim Berners-Lee. Www: Past, present, and future. *Computer*, 29(10):69–77, 1996.

[4] Tim Berners-Lee, Robert Cailliau, Jean-François Groff, and Bernd Pollermann. World-wide web: the information universe. *Internet Research*, 2(1):52–58, 1992.

[5] Timothy J Berners-Lee. Information management: A proposal. Technical report, 1989.

[6] Simona Colucci, Francesco M Donini, and Eugenio Di Sciascio. A review of reasoning characteristics of rdf-based semantic web systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1537, 2024.

[7] Andrea Giulianelli, Samuele Burattini, Andrei Ciortea, and Alessandro Ricci. Engineering interoperable ecosystems of digital twins: A web-based approach. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, pages 476–485, 2024.

[8] Roberto Minerva, Gyu Myoung Lee, and Noël Crespi. Digital twin in the iot context: A survey on technical features, scenarios, and architectural models. *Proceedings of the IEEE*, 108(10):1785–1824, 2020.

[9] Firoj Parwej, Nikhat Akhtar, and Yusuf Perwej. An empirical analysis of web of things (wot). *International Journal of Advanced Research in Computer Science*, 10(3), 2019.

[10] Marco Picone, Marco Mamei, and Franco Zambonelli. Wldt: A general purpose library to build iot digital twins. *SoftwareX*, 13:100661, 2021.

[11] Carlos Ramonell, Rolando Chacón, and Héctor Posada. Knowledge graph-based data integration system for digital twins of built assets. *Automation in Construction*, 156:105109, 2023.

[12] Alessandro Ricci, Angelo Croatti, Stefano Mariani, Sara Montagna, and Marco Picone. Web of digital twins. *ACM Trans. Internet Technol.*, 22(4), November 2022.

[13] Alessandro Ricci, Angelo Croatti, and Sara Montagna. Pervasive and connected digital twins—a vision for digital health. *IEEE Internet Computing*, 26(5):26–32, 2021.

[14] Marco Luca Sbodio, David Martin, and Claude Moulin. Discovering semantic web services using sparql and intelligent agents. *Journal of Web Semantics*, 8(4):310–328, 2010.

[15] Luca Sciullo, Lorenzo Gigli, Federico Montori, Angelo Trotta, and Marco Di Felice. A survey on the web of things. *IEEE access*, 10:47570–47596, 2022.

[16] Xiang Su, Ekaterina Gilman, and Xiaoli Liu. Towards semantic web of things: Reference architecture and gap analysis. In *2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pages 151–158. IEEE, 2023.

[17] Fei Tao, Bin Xiao, Qinglin Qi, Jiangfeng Cheng, and Ping Ji. Digital twin modeling. *Journal of Manufacturing Systems*, 64:372–389, 2022.