

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA
DIPARTIMENTO DI INFORMATICA - SCIENZA E
INGEGNERIA

Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

APPLICAZIONE DEL PARADIGMA DIGITAL
TWIN IN AMBITO SANITÀ DIGITALE: IL
SISTEMA DI GESTIONE DEI TRASPORTI
SECONDARI AUSL ROMAGNA NUC 116-117
COME CASO DI STUDIO

Tesi in: Pervasive Computing

Relatore:

Prof. ALESSANDRO RICCI

Co-Relatore:

Prof. ANGELO CROATTI

Presentata da:

GIADA GIBERTONI

ANNO ACCADEMICO 2020-2021

PAROLE CHIAVE

Digital Twin
Sanità digitale
Internet Of Things
NUE 116-117
AUSL Romagna

Ai miei genitori, che mi hanno dato la possibilità di svolgere questo percorso. Un ringraziamento anche a tutte le persone che mi sono state vicine in questi ultimi anni.

Indice

Introduzione	xi
1 Digital Twin	1
1.1 Origini e definizione del concetto	1
1.1.1 Key concepts	2
1.1.2 Life Cycle dei Digital Twin	2
1.2 Campi di utilizzo	4
1.2.1 Industry 4.0	4
1.2.2 Smart City e Smart Building	5
1.2.3 Military field	6
1.2.4 Digital Health	8
2 Digital Twin per Digital Health	9
2.1 Esempi di ambiti applicativi nella Digital Health	10
2.1.1 Digital Twin come tecnologia abilitante per la <i>precision medicine</i>	10
2.1.2 Digital Twin per la virtualizzazione dei processi ospedalieri	12
2.1.3 Digital Twin per la digitalizzazione di strutture ospedaliere	14
3 Un caso di studio: applicazione dei Digital Twin ai Trasporti secondari 116-117	17
3.1 Applicazione del paradigma Digital Twin al caso di studio Trasporti secondari 116-117	18
3.2 Challenge da affrontare durante l'analisi e lo sviluppo del caso di studio	19

4	Analisi	21
4.1	Contesto di riferimento	21
4.2	Situazione attuale del servizio	24
4.3	Proposta di riorganizzazione	25
4.4	Scenari e casi d'uso	27
4.4.1	Trasporto per dimissioni da ospedale	27
4.4.2	Trasporto per visita da casa a ospedale, paziente con ridotta mobilità	28
4.4.3	Trasporto per trattamenti ricorrenti	29
4.4.4	Trasporto per trasferimento da un ospedale ad un altro, non programmato	29
4.4.5	Giornata lavorativa di un operatore trasporto	30
4.5	Ubiquitous language	40
4.6	Requisiti del sistema	42
4.6.1	Caratteristiche del NUE 116-117	43
4.6.2	Requisiti	43
4.7	Context map	45
4.7.1	Bounded Context	45
4.7.2	Entità del dominio	48
5	Progettazione di una soluzione basata su Digital Twin	55
5.1	Integrazione Digital Twin nel modello 116-117	55
5.1.1	Digital Twin Prototype	56
5.1.2	Synchronization (twinning) tra digital twin e physical twin	66
5.1.3	Piattaforma per lo sviluppo dei digital twin	67
5.2	Interoperabilità	68
5.2.1	FHIR	69
5.2.2	SNOMED CT	71
5.2.3	Mapping tra entità del dominio e risorse FHIR	71
5.3	Sicurezza e privacy	80
5.4	Architettura finale proposta	81
6	Implementazione	83
6.1	Servizi Azure utilizzati	83
6.1.1	Creazione e gestione dei Digital Twin	85
6.1.2	Connessione tra Digital Twin e device IoT	85

6.1.3	Strumenti di analisi	87
6.2	Struttura delle API sviluppate	88
6.2.1	Trasporti 116-117 REST API	88
6.2.2	Interazione con Azure tramite Azure SDK	89
6.2.3	HAPI FHIR	93
6.3	Set di API sviluppate	93
6.3.1	Bounded Context Paziente	93
6.3.2	Bounded Context Veicolo	94
6.3.3	Bounded Context Trasporto	95
7	Testing	97
7.1	ADT Explorer	97
7.2	Unit testing	99
7.3	Testing Rest API tramite Postman	99
7.4	Invio dati IoT simulati all'ambulanza	99
8	Validazione e discussione	101
8.1	Client di esempio - Call Center Client	101
8.2	Altri ipotetici client	105
8.2.1	Vehicle client	105
8.2.2	User client	106
8.2.3	Sistema 118	107
8.3	Ulteriori query sul Digital Twin Graph	107
8.4	Scelta e costi della piattaforma	108
9	Conclusioni	109
9.1	Miglioramenti futuri	110

Introduzione

Negli ultimi anni sta emergendo sempre di più il concetto di Digital Twin. Nell'era dell'industria 4.0 molti settori stanno vivendo una trasformazione digitale, il paradigma Digital Twin può portare a molti vantaggi in termini di competitività sul mercato, produttività ed efficienza.

L'utilizzo della tecnologia si sta evolvendo moltissimo anche nel campo della sanità digitale, dove le modalità di cura tradizionali non sono più sufficienti per far fronte alla sempre più crescente richiesta in questo ambito. I digital twin possono, dunque, giocare un ruolo importante, come dimostrano molti progetti presenti in letteratura, che spaziano in diverse tipologie di applicazioni, come ad esempio la precision medicine per la cura del paziente, la virtualizzazione dei processi ospedalieri o la digitalizzazione di mezzi o strutture.

In questa tesi si è scelto di applicare il paradigma al caso di studio Trasporti 116-117, che al momento evidenzia una serie di problematiche, come la mancanza di interoperabilità con una conseguente difficoltà nella comunicazione, un progressivo aumento dei costi ed una scarsa tracciabilità delle risorse nel sistema. Il progetto mira, dunque, a far fronte a queste problematiche con un miglioramento della gestione dei trasporti ospedalieri secondari, ovvero, ciò di cui si occupa il numero unico europeo 116-117 adibito alle cure mediche non urgenti ed altri servizi territoriali a bassa priorità. Tra i vari servizi erogati tramite questo numero, ci si focalizzerà prevalentemente sui trasporti sanitari non urgenti.

Nel capitolo 1 si introdurrà il concetto di digital twin partendo dalle sue origini e dalla sua definizione, passando poi alla sua applicazione in ambiti quali industria 4.0, smart city, campo militare per finire con la digital health. Nel capitolo 2 si approfondisce l'utilizzo del paradigma nel campo dell'healthcare, analizzando alcuni casi rilevanti presenti in letteratura.

Nel capitolo 3 si introduce il caso di studio Trasporti 116-117, dando una

spiegazione generale del contesto di riferimento, analizzando i vantaggi che i digital twin possono portare ad esso e le challenge da affrontare durante il ciclo di sviluppo.

Nel capitolo 4 e 5 si passa alle fasi di analisi e progettazione, svolte usando un approccio Domain Driven. Per prima cosa si fa un'analisi del contesto di riferimento attuale e la relativa proposta di riorganizzazione fornita dall'esperto del dominio. Si ipotizzano, poi, alcuni scenari arrivando ad una lista di requisiti che il sistema deve avere, infine si identificano i principali bounded context all'interno del dominio.

Si passa poi alla progettazione, integrando i digital twin all'interno del modello definendo la loro struttura e la piattaforma che li ospiterà. Per finire si approfondisce il problema dell'interoperabilità sollevato nel capitolo 3.

Con il capitolo 6 si passa poi alla fase implementativa, questo capitolo ha lo scopo di dare una panoramica dello sviluppo svolto, il progetto rimane consultabile al repository <https://github.com/giadagibertoni/Trasporti116-117>. Nel capitolo 7 vengono spiegati i tipi di test svolti per assicurare la correttezza del lavoro svolto.

Per finire, nel capitolo 8 si valida il progetto tramite alcuni client d'esempio che possono usufruire delle API sviluppate, inoltre si discute della possibilità di ampliare il set di API e dei costi di una soluzione cloud.

Capitolo 1

Digital Twin

1.1 Origini e definizione del concetto

Il termine è stato coniato nel 2003 da Michael Grieves alla University of Michigan ma, al momento dell'introduzione del concetto, le rappresentazioni virtuali di prodotti fisici erano nuove e ancora immature e le informazioni che potevano essere raccolte dall'asset fisico erano molto limitate [12]. A causa, quindi, dei limiti tecnologici, come, ad esempio, bassa potenza di calcolo, poca o assente connettività dei dispositivi, scarsa capacità di archiviazione e gestione dei dati, i Digital Twin non hanno avuto applicazioni pratiche all'epoca.

Il concetto si è evoluto con l'avvento dell'Internet of Things (IoT). A dare rilevanza al concetto è stata la NASA in un articolo del 2012 intitolato *"The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles"* [11].

L'articolo spiega che le future generazioni di veicoli prodotti dalla NASA richiederanno particolari caratteristiche, questi veicoli dovranno essere in grado di resistere a condizioni estreme per periodi prolungati. Gli approcci a disposizione per la certificazione e la gestione delle flotte non sono in grado di soddisfare i requisiti richiesti e le condizioni ambientali sono impossibili da simulare all'interno di un laboratorio. In alcune missioni complesse, specialmente quelle dove non è possibile il supporto esterno, si richiede una gestione in tempo reale del veicolo. Per queste motivazioni la NASA si vede costretta a un cambio di paradigma che porta all'utilizzo dei Digital Twin. Di seguito è riportata la definizione data dalla Nasa nel 2012.

A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin.

Un digital twin è, dunque, una rappresentazione virtuale di un qualsiasi asset o processo fisico. Esso ha la capacità di collegare mondi fisici e virtuali in tempo reale, ciò può fornire una misurazione più realistica di scenari imprevisi e imprevedibili [21]. Essi sono utilizzati per simulazioni, per controllo, manutenzione e predizione, ovvero per identificare possibili guasti o malfunzionamenti prima che si verifichino.

Le prime applicazioni dei digital twin erano per la manutenzione di apparecchiature. Da quando il paradigma ha trovato la sua applicazione in altri settori, oltre che nella produzione, la definizione non è più limitata a entità non viventi o prodotti industriali, ora può essere utilizzata anche per descrivere processi o sistemi biologici complessi come esseri umani e alberi.

1.1.1 Key concepts

I digital twin possono essere caratterizzati da tre concetti chiave [22]. Il **Digital Twin Prototype (DTP)** è la rappresentazione del modello dell'asset fisico, esiste un unico DTP per ogni tipo di digital twin. La **Digital Twin Instance (DTI)**, invece, è l'istanza di ogni entità fisica, basata sul DTP, quindi più DTI possono utilizzare lo stesso DTP. Infine il **Digital Twin Aggregate (DTA)** è una collezione di DTI ed eventualmente altri DTA e dei meccanismi per interrogarli. Il DTA può fornire informazioni aggiuntive sull'aggregato, che non possono essere ottenute a livello di DTI. Esiste anche il concetto di **Twin2Twin (T2T)** dove diversi digital twin possono essere connessi ad altri creando un T2T ecosystem, il T2T non ha una natura gerarchica.

1.1.2 Life Cycle dei Digital Twin

Il Digital Twin esiste per l'intero ciclo di vita dell'asset fisico, dalla nascita fino a alla sua dismissione. Tuttavia non è detto che il ciclo di vita del asset fisico e del gemello digitale corrisponda, il ciclo di vita del digital twin, infatti, può anche iniziare prima della creazione dell'asset fisico. Uno dei

punti di forza dei digital twin è la possibilità di simulare il funzionamento della controparte fisica prima della sua produzione, questo permette di individuare colli di bottiglia all'interno del processo, di ridurre i costi e di analizzare i livelli di efficienza e sicurezza dell'asset fisico. La vita del Digital Twin può continuare anche dopo le dismissioni della sua controparte, può infatti aiutare nello smantellamento di questa e successivamente può essere utile nella progettazione e produzione della generazione successiva.

Al momento qualsiasi controparte virtuale di un sistema fisico è chiamata Digital Twin. Con questa ampia definizione si lascia spazio a molti dubbi e domande; per risolvere questa ambiguità M. Madni, C. Madni e Lucero [15], hanno definito quattro livelli di Digital Twin, ognuno con uno scopo specifico. Di seguito si riportano i quattro livelli e le relative caratteristiche:

Level	Model Sophistication	Physical Twin	Data Acquisition from Physical Twin	Machine Learning (Operator Preferences)	Machine Learning (System/Environment)
1 Pre-Digital Twin	virtual system model with emphasis on technology/technical-risk mitigation	does not exist	Not applicable	No	No
2 Digital Twin	virtual system model of the physical twin	exists	performance, health status, maintenance; batch updates	No	No
3 Adaptive Digital Twin	virtual system model of the physical twin with adaptive UI	exists	performance, health status, maintenance; real-time updates	Yes	No
4 Intelligent Digital Twin	virtual system model of the physical twin with adaptive UI and reinforcement learning	exists	performance, health status, maintenance, environment; both batch/real-time updates	Yes	Yes

Figura 1.1: Livelli Digital Twin

1. **Pre-Digital Twin:** Rappresenta il prototipo digitale. Viene utilizzato principalmente per confermare eventuali decisioni chiave da prendere durante la progettazione dell'asset fisico e per mitigare rischi tecnici. La controparte fisica in questo livello non esiste ancora.
2. **Digital Twin:** Rappresentazione digitale che incorpora dati dal gemello fisico, questa riceve aggiornamenti real-time dal sistema fisico.

Poiché l'interazione tra le due parti è bidirezionale, vi sono ampie opportunità per il gemello fisico di utilizzare le conoscenze acquisite dal gemello digitale per migliorare le proprie prestazioni.

3. **Adaptive Digital Twin:** Offre un'interfaccia grafica adattiva, sensibile alle preferenze e alle necessità dell'utente. La capacità chiave di questo livello è quella di apprendere le preferenze degli operatori in diversi contesti, ciò è possibile tramite l'utilizzo di reti neurali basate su learning supervisionato. A questo livello il Digital Twin può supportare la pianificazione e il processo decisionale real-time durante le varie operazioni, la manutenzione e l'assistenza.
4. **Intelligent Digital Twin:** Ha le capacità del livello 3 con, in aggiunta, l'utilizzo di learning non supervisionato per comprendere oggetti e modelli incontrati nell'ambiente in cui opera. Inoltre utilizza reinforcement learning per operare in un ambiente incerto e parzialmente osservabile. A questo livello il Digital Twin ha un alto grado di autonomia e può analizzare dati ad un maggior livello di dettaglio.

1.2 Campi di utilizzo

1.2.1 Industry 4.0

L'industria 4.0 è il futuro del campo manifatturiero, essa punta a migliorare l'efficienza e le prestazioni dei sistemi di produzione [6]. Questa implica la trasformazione dei vecchi sistemi in sistemi più flessibili e adattabili con la capacità di apprendere e prendere decisioni in maniera autonoma. L'industria 4.0 fa emergere molti concetti chiave come la decentralizzazione, l'interoperabilità, la virtualizzazione e la comunicazione real-time. Il concetto di virtualizzazione può essere definito come la capacità di creare una copia virtuale di un sistema fisico, questo concetto si è poi evoluto nel paradigma Digital Twin che acquisisce un'importanza cruciale nello sviluppo di soluzioni per l'industria 4.0 [19]. L'applicazione dei Digital Twin nel mondo manifatturiero influisce sulla progettazione, la realizzazione e la manutenzione dei prodotti. Il Digital Twin può aiutare a prendere decisioni in merito alla produzione, monitorare le prestazioni del prodotto, comandare e configurare le macchine per la produzione, individuare e gestire problemi

ed ottimizzare i processi di produzione. In questo ambito esso può anche essere utilizzato per la manutenzione predittiva, la formazione degli operatori, il supporto alle decisioni, l'analisi in tempo reale e la simulazione di un comportamento.

Uno dei punti cruciali nell'industria 4.0 è la Human Robot Interaction, al momento l'interazione ha ancora molte limitazioni, questo dipende spesso dalla mancanza di un giusto linguaggio di comunicazione tra uomo e macchina; ciò rende difficile al robot capire le intenzioni dell'essere umano e all'uomo di esprimersi. Questa barriera di comunicazione può portare problemi in termini di efficienza e sicurezza. L'utilizzo dei digital twin può potenzialmente essere un metodo efficace per mediare l'interazione tra uomo-robot. Negli ambienti classici l'uomo che si interfaccia con il robot è un osservatore poiché il flusso di dati è unidirezionale: l'uomo riceve informazioni dall'asset fisico o dal relativo digital twin. Per processi complessi, questa modalità di comunicazione unidirezionale non è sufficiente; può essere utile integrare anche l'uomo nel digital twin in modo che la propria operatività sia potenziata e che il proprio ruolo passi da osservatore del sistema a dominatore [23]. Un esempio dell'utilizzo dei digital twin nell'industria 4.0 si può vedere in figura 1.2, dove l'operatore umano, tramite il visore, vede l'ambiente virtuale del digital twin, tenendosi a distanza di sicurezza dall'asset fisico, ovvero il robot saldatore.

1.2.2 Smart City e Smart Building

Negli ultimi anni è in atto una rivoluzione dei dati che fornisce nuove informazioni sulle città, consentendo di comprendere le aspettative del cittadino in merito a smart building e smart city. La challenge è determinare qual'è il modo migliore per i cittadini di interagire con questi ambienti [8]. È necessario modellare e visualizzare dati complessi ed i Digital Twin sono in grado di affrontare queste sfide. Essi consentono lo scambio di dati e possono contenere modelli, simulazioni e algoritmi che descrivono dettagliatamente la controparte fisica, inoltre possono essere implementati nella realtà virtuale, il che consente un'ampia partecipazione dei cittadini, un esempio in figura 1.3.

Le relazioni tra esseri umani, infrastrutture e le tecnologie nelle città non possono essere considerati come semplici aggregati di entità indipendenti. L'avvento dei social media, ad esempio, ha creato un grande flusso di dati

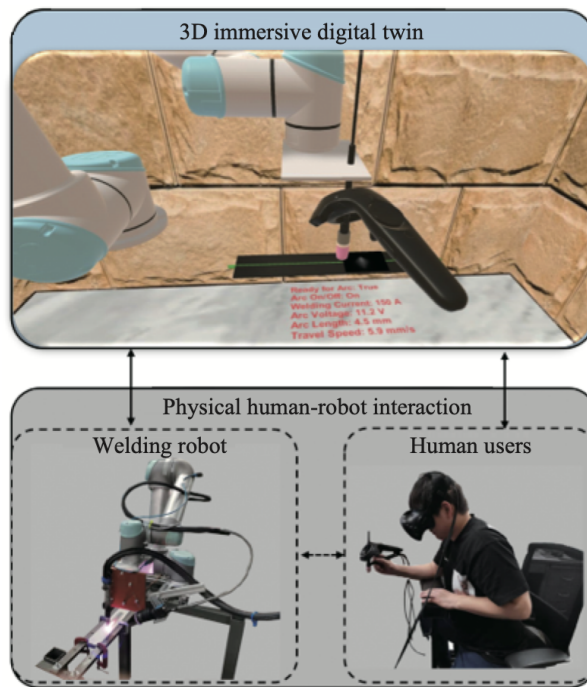


Figura 1.2: DT per l'industria 4.0 nel campo della saldatura [23]

socio-economici ad alta risoluzione spazio-temporale, questi dati sono però eterogenei e non strutturati, il che rende difficile la loro astrazione e aggregazione. La simulazione e la creazione di modelli digitali sembra essere una buona strada per facilitare questa sfida [17].

L'obiettivo è di raggiungere una somiglianza con il mondo reale ad un livello di dettaglio abbastanza accurato da permettere di affrontare problemi complessi [9].

1.2.3 Military field

In campo militare i Digital Twin sono molto rilevanti data la presenza di molti sistemi critici con bassa tolleranza ai guasti. I Digital Twin si distinguono in questo campo per i vantaggi che offrono in termini di produzione di soluzioni ad un giusto costo e l'aumento di efficienza e qualità. Nel campo dell'aviazione e dello spazio, i test, le previsioni future e lo studio del comportamento del veicolo sono processi molto complessi. Con l'ausilio dei Di-



Figura 1.3: Interazione tra DT e i cittadini [9]

digital Twin vari scenari possono essere realisticamente testati nell'ambiente digitale. Ad esempio, nel campo dell'aviazione e dell'industria aerospaziale, la produzione e il collaudo di nuovi modelli sono molto costosi; i Digital Twin, insieme al machine learning e a modelli predittivi, forniscono consigli per aiutare i produttori e i piloti a ottimizzare il consumo di carburante, per essere proattivi sulla manutenzione e per gestire i costi delle flotte. Ad esempio, grazie ai terabyte al secondo di dati real-time forniti dal motore e dai dati passati raccolti, è possibile prevedere futuri guasti.

Uno dei primi Digital Twin in campo militare è il programma AVM (Adaptive Vehicle Production) partito nel 2010. Il programma mira ad abbreviare il ciclo di ricerca e sviluppo che trasformerà l'industria manifatturiera militare.

L'utilizzo di Digital Twin nell'aviazione rimane, al momento, il campo più emergente in ambito militare. È stato annunciato che l'esercito americano vuole creare cloni digitali del caccia F-35, questi cloni saranno utilizzati per sostituire i test eseguiti sull'asset fisico. Questo permette di prevedere le prestazioni, l'aspettativa di vita e il tasso di guasti dell'asset fisico. L'utilizzo dei Digital Twin non aiuta solo in questioni come il rilevamento

precoce dei guasti ma può anche fornire un significativo risparmio sui costi di manutenzione, che al momento ammonta a 75 milioni di dollari annui su una flotta di 14 mila velivoli.

L'applicazione dei Digital Twin non si limita soltanto agli aerei, essi, infatti, vengono utilizzati anche, ad esempio, per i sistemi satellitari, in quanto hanno un serio impatto in ambito militare. Si prevede che questa tecnologia si diffonderà sempre di più per vari campi di studio militari [16].

1.2.4 Digital Health

I Digital Twin svolgono un ruolo importante nel campo dell'assistenza sanitaria, in particolare per la pianificazione del percorso medico, l'allocazione delle risorse, la previsione di attività mediche e molto altro. Nel capitolo 2 si approfondiscono l'utilizzo e i vantaggi dei Digital Twin per la Digital Health.

Capitolo 2

Digital Twin per Digital Health

La crescita e lo sviluppo che la tecnologia sta avendo nell'ambito dell'assistenza sanitaria non ha precedenti, ciò che prima era impossibile ora sta diventando possibile. In termini di IoT i dispositivi sono più economici e più semplici da implementare, per questo motivo aumenta la connettività all'interno del settore sanitario. L'aumento della connettività aumenta, di conseguenza, il potenziale dei Digital Twin. L'utilizzo di questo paradigma offre a ricercatori, medici, ospedali e operatori sanitari la possibilità di simulare determinati ambienti o situazioni in base alle loro esigenze, sia in real-time che per predizioni o utilizzi futuri [10].

Nonostante i progressi dei servizi medici intelligenti, persistono una serie di problemi, come ad esempio:

- Non ci sono interazioni real-time tra l'istituzione medica e il paziente.
- La fusione tra i sistemi medici fisici e i sistemi informativi non è ancora stata attuata.
- L'accuratezza di un servizio dall'allerta in caso di crisi di un paziente non è sufficientemente elevata e non esiste una supervisione intelligente.
- I sistemi o le piattaforme esistenti non forniscono servizi continui di gestione della salute in tutto il ciclo di vita del paziente.

Per affrontare questi problemi serve risolvere quello di interazione e convergenza tra lo spazio fisico e quello virtuale. Un modo efficace è utilizzare il concetto di Digital Twin [14].

2.1 Esempi di ambiti applicativi nella Digital Health

Nonostante il paradigma Digital Twin ha avuto rilevanza soltanto negli ultimi anni e specialmente nell'ambito industriale, ci sono già molti riscontri da parte del mondo della digital health.

Il paradigma, ad oggi, viene utilizzato per molteplici scopi che spaziano dall'organizzazione e monitoraggio delle strutture sanitarie, all'assistenza durante attività e processi medici fino al supporto decisionale in merito a diagnosi e terapie da sottoporre ai pazienti. Di seguito sono riportati alcuni esempi di ambiti applicativi corredati da casi di studio presenti in letteratura.

2.1.1 Digital Twin come tecnologia abilitante per la *precision medicine*

Negli ultimi anni la medicina si sta evolvendo dalla *evidence-based medicine*, ovvero utilizzo di ricerca, valutazione e uso sistematico dei risultati della ricerca come base per le decisioni cliniche, verso la *precision medicine*, ovvero un modello che propone la personalizzazione dell'assistenza sanitaria, con decisioni, trattamenti e prodotti medici specifici per un determinato individuo. Nella precision medicine un obiettivo importante è prendere di mira una malattia prima che si manifesti, utilizzando, ad esempio, la genetica e i sistemi intelligenti di monitoraggio sanitario. Questo porta ad una assistenza sanitaria costruita ad hoc sul paziente, ciò richiede una modalità di servizio sanitario continuo e più intelligente, che porta alla costruzione di nuovi sistemi e piattaforme per la gestione di dati personalizzati [10].

La precision medicine richiede non solo dati migliori e più dettagliati, ma anche una crescente capacità computazionale per analizzare, integrare e sfruttare questi dati per costruire il digital twin del paziente. Questo rappresenta la visione di uno strumento virtuale completo che integra i dati clinici di un paziente. Il gemello digitale seguirà la vita di ogni persona e

sfrutterà sia i dati provenienti dai sensori sia lo stile di vita che lo stesso può registrare.

Il Digital Twin è la via verso la “precision medicine”, esso creerà la transizione da sistemi sanitari basati sulla descrizione della malattia a sistemi sanitari focalizzati sulla previsione della risposta, in questo modo la scelta del trattamento migliore non sarà basata sullo stato attuale del paziente ma sul suo stato futuro.

Una soluzione è stata proposta dallo Swedish Digital Twin Consortium che mira a sviluppare una strategia per la medicina personalizzata [4]. L’idea si basa su i seguenti quattro punti, come riportato in figura 2.1:

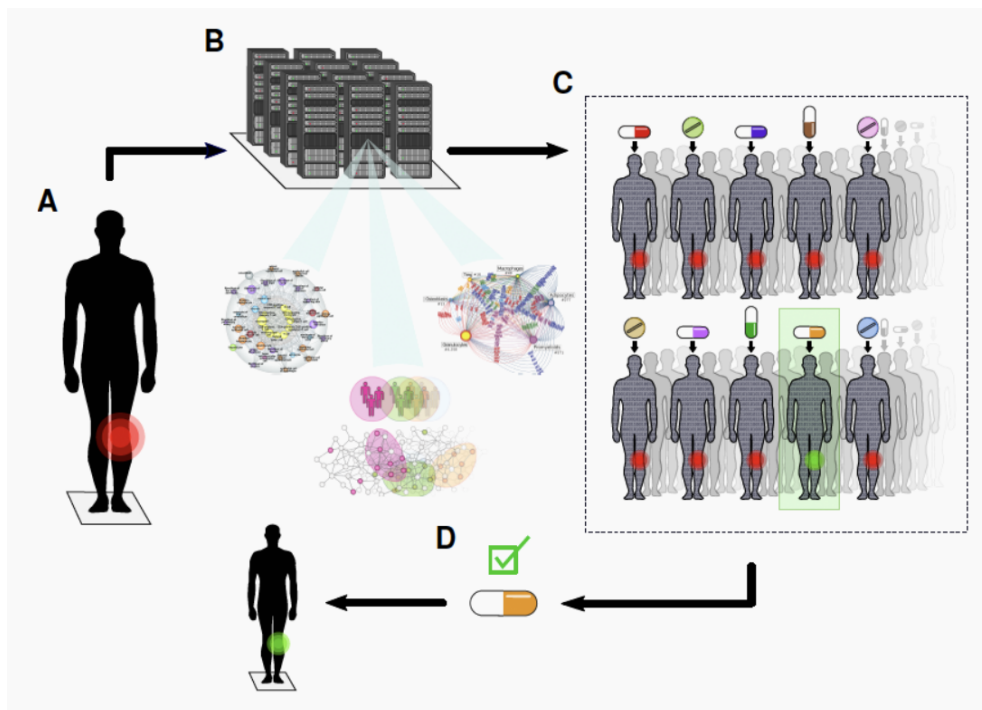


Figura 2.1: Step del modello SDTC

- A. Il paziente accusa un dolore.
- B. Si costruisce un gemello digitale del paziente in copie illimitate, sulla base di migliaia di variabili rilevanti per la malattia.

- C. Ogni istanza del Digital Twin viene trattata con uno o più farmaci, questo si traduce nella cura digitale del paziente.
- D. Il farmaco che ha un effetto migliore sul paziente virtuale viene selezionato per il trattamento del paziente fisico.

Sono da tenere in considerazione, però, alcuni problemi, come l'utilizzo di questi dati da parte delle organizzazioni sanitarie, dove la sicurezza e la riservatezza dei dati sensibili sono molto importanti. Un altro problema è la struttura dei dati, infatti i dati medici contenuti nelle cartelle cliniche dei pazienti e nei sistemi informativi sanitari sono molto eterogenei e non strutturati [7].

In generale l'implementazione di un Digital Twin di un paziente richiederà la risoluzione di molte sfide tecniche, mediche ed etiche; ma potrebbe portare ad un'assistenza sanitaria notevolmente migliorata e alla profonda comprensione di alcune malattie [5].

2.1.2 Digital Twin per la virtualizzazione dei processi ospedalieri

I digital twin applicati alla sanità digitale permettono di abilitare la virtualizzazione di complessi processi ospedalieri, come ad esempio monitorare degenze, assistere i medici durante un intervento, gestire terapie e molto altro.

Uno dei campi più complicati da gestire per l'assistenza sanitaria è, ad esempio, la gestione delle emergenze. Un esempio in letteratura che virtualizza questo processo è il progetto Trauma Tracker, di seguito illustrato.

Trauma Tracker [18] ha lo scopo di assistere i medici e gli operatori per tutto il ciclo di vita dell'emergenza. In questi scenari opera un team formato da persone con competenze eterogenee che fanno capo al leader del team. La prima ora di trattamento è molto spesso fondamentale per la salute del paziente, quindi il team deve essere il più veloce possibile nell'analizzare la situazione e fornire il giusto aiuto medico.

Trauma Tracker ha lo scopo di assistere il leader del team tenendo traccia di tutte le informazioni e documentazioni relative alla gestione del trauma. Durante un primo periodo di utilizzo di Trauma Tracker è emerso il bisogno di un monitoraggio continuo di tutto il processo di assistenza; si è scelto

quindi di integrare la soluzione con dei Digital Twin che modellano le due fasi principali del processo, di seguito riportate [20].

- **Pre-hospital phase (*The PreH Digital Twin*)**: consiste nel primo soccorso al paziente effettuato sul luogo dell'incidente. L'istanza del Digital Twin viene creata quando si riceve la richiesta di assistenza e contiene informazioni sulle seguenti parti:
 - il veicolo con il quale viene gestita l'emergenza;
 - EMT (emergency medical technician) e il soccorritore;
 - il luogo e la dinamica dell'infortunio;
 - il paziente.

Il Digital Twin raccoglie informazioni real-time tramite i dati forniti dall'unità centrale e dai vari sensori e dispositivi intelligenti disponibili. Il punto cruciale di questa fase è quando i soccorritori decidono il grado di gravità del trauma.

- **Operative phase (*The Trauma Digital Twin*)**: consiste nella fase in cui il paziente viene assistito all'interno del pronto soccorso. L'istanza di questo Digital Twin viene creata dal momento in cui il trauma è stato contrassegnato come grave. È importante che il gemello digitale venga creato prima dell'arrivo in pronto soccorso, in questo modo il team traumatologico è già informato sulle dinamiche dell'incidente e sulle condizioni del paziente prima del suo arrivo in ospedale. Quando il paziente arriva al pronto soccorso lo stato del Digital Twin cambia, esso conterrà le seguenti risorse:
 - il paziente;
 - il leader e il team di soccorso;
 - la shock-room, ovvero la sala del pronto soccorso dove viene gestito il trauma;
 - altri eventuali strumenti e attrezzature.

Il ciclo di vita del Digital Twin termina quando il paziente viene ricoverato.

Il Digital Twin può essere osservato ed essere accessibile dai diversi operatori che gestiscono o fruiscono delle informazioni al suo interno.

2.1.3 Digital Twin per la digitalizzazione di strutture ospedaliere

Un altro ambito dove i digital twin si possono dimostrare estremamente validi è quello della virtualizzazione di strutture ospedaliere. Un gemello digitale di una sala operatoria, di un reparto o addirittura di un intero ospedale può portare a una moltitudine di vantaggi. Questi digital twin possono essere usati per gli scopi più disparati, come ad esempio il monitoraggio, la manutenzione, l'amministrazione, la gestione di emergenze, l'individuazione di guasti e molto altro ancora.

Un esempio rilevante in letteratura è HospiT'Win che ha lo scopo di mitigare l'impatto di eventi imprevisti all'interno dell'ospedale, prevedendo gli scenari futuri che potrebbero accadere al verificarsi di un evento imprevisto. Di seguito si illustra brevemente il progetto. HospiT'Win è composto da tre livelli di funzionalità:

1. **Construction and Validation Level:** è un livello offline; la sua funzione è la costruzione di un meta-modello per ogni sensore dell'asset fisico e la verifica della sua corretta implementazione per avere lo stesso stato e lo stesso comportamento dei sensori fisici. Successivamente questo meta-modello verrà utilizzato nel livello 2.
2. **Tracking and Monitoring Level:** questo livello online è composto da sensori, devices IoT, RFID utilizzati per sincronizzare il mondo virtuale e quello reale. Questo livello contiene tre sotto livelli, di seguito riportati:
 - Data Preparing and Cleaning layer: ha il compito di pulire e preparare i dati prima di inviarli al modello virtuale, poichè i dati forniti dai sensori potrebbero non essere affidabili e devono essere controllati prima del loro utilizzo.
 - Security layer: ha il compito di garantire un trasferimento sicuro dei dati dall'asset fisico al Digital Twin.
 - Dynamic Database layer: è utilizzato per memorizzare i dati e le informazioni aggiornate. Il Digital Twin utilizza il database per tracciare e monitorare il flusso di dati dai sensori reali al modello digitale. Il database memorizza i feedback sui comportamenti del

modello digitale per confermare il corretto funzionamento, poiché esso deve rispettare i KPIs dell'ospedale.

Lo scopo di questo livello è fornire informazioni sullo stato dell'asset fisico in tempo reale (es. flusso dei pazienti, monitoraggio della sala operatoria, ecc...); inoltre deve rilevare situazioni impreviste, prima che si verifichi un problema deve chiamare il modello predittivo per vedere cosa succederà in futuro.

- 3. Vision of the Future Level:** questo livello è composto da modelli predittivi, simulazioni stocastiche e tecniche di intelligenza artificiale. È una rappresentazione digitale che applica le tecniche di simulazione sui dati che riceve dal livello due. In caso di eventi inaspettati predice il futuro prossimo per vedere l'impatto di questi eventi sull'ospedale reale. Inoltre può generare vari scenari per minimizzare l'impatto di questi eventi, utilizza l'intelligenza artificiale per determinare qual'è lo scenario migliore da adottare.

Capitolo 3

Un caso di studio: applicazione dei Digital Twin ai Trasporti secondari 116-117

Come esposto nei capitoli 1 e 2, i digital twin possono portare una moltitudine di vantaggi in diversi campi. In letteratura spiccano i risultati ottenuti dall'unione di digital twin e assistenza sanitaria, per questo motivo nel prosieguo di questa tesi ci si focalizzerà su un caso di studio nell'ambito sanità digitale.

Il caso di studio preso in considerazione è il nuovo numero unico europeo 116-117 per l'accesso alle cure mediche non urgenti e ad altri servizi sanitari territoriali a bassa priorità. L'istituzione di questo numero è frutto di una direttiva dell'unione Europea, a seguito di questa l'Italia ha disposto il numero unico non urgente 116-117. Questo ha lo scopo di rispondere a quesiti di non emergenza, alleggerendo il carico di lavoro del 112 e 118, lasciando questi numeri per le sole emergenze.

L'idea del caso di studio nasce da una collaborazione con l'AUSL Romagna, che, a seguito delle nuove normative europee e dell'accordo Stato Regioni del 24/11/2016 sulle "Linee di indirizzo sui criteri e le modalità di attivazione del numero europeo armonizzato a valenza sociale 116-117", ha trovato necessaria l'implementazione di una centrale operativa 116-117, con valenza sovra-provinciale, nel breve periodo.

Questa tesi ha dunque lo scopo di sperimentare l'applicazione dei digital twin a questo contesto e verificare i vantaggi/svantaggi che esso può trarre

dal paradigma.

3.1 Applicazione del paradigma Digital Twin al caso di studio Trasporti secondari 116-117

Con l'applicazione dei Digital Twin a questo caso di studio si mira ad un miglioramento della gestione dei trasporti ospedalieri secondari, che ad oggi presenta molte lacune. Si è scelto di applicare il paradigma dei Digital Twin al fine di riportare una rappresentazione digitale del mondo reale, in modo da facilitare l'analisi dello stato attuale del servizio fornito e una migliore comprensione di quali potrebbero essere i miglioramenti da apportare.

Si prevede che l'applicazione di questo paradigma porti a diversi vantaggi:

- Monitoraggio real-time del processo di trasporto: i Digital Twin permetteranno di monitorare in real-time tutti gli elementi del dominio. Sarà possibile conoscere la posizione esatta dell'ambulanza e dell'operatore in ogni momento, si potrà così individuare e gestire eventuali ritardi/anticipi.
- Monitoraggio delle richieste del servizio: Si potrà monitorare le richieste di servizio ricevute, questo permette di individuare le fasce orarie/periodi con traffico più intenso portando ad individuare colli di bottiglia all'interno del sistema.
- Ottimizzazione dell'allocazione delle ambulanze: analizzando i dati storici si possono individuare periodi dove le risorse come ambulanze e operatori non erano sufficienti per soddisfare la mole di richieste, oppure, al contrario, erano in eccesso. Con queste informazioni si può decidere di integrare l'organico e i mezzi in modo mirato evitando spreco di denaro e risorse.
- Ottimizzazione dei percorsi delle ambulanze: tramite i Digital Twin si possono assegnare dinamicamente le ambulanze ai trasporti da eseguire nel prossimo futuro, in questo modo si evitano viaggi a vuoto che, al momento, le ambulanze compiono.

- Previsioni future del servizio: i Digital Twin permettono di prevedere la fruizione futura del servizio, permettendo, ad esempio, di individuare fasce orarie o periodi in cui probabilmente saranno necessarie maggiori risorse per garantire un determinato livello di qualità del servizio.
- Produzione di report e studio dell'andamento del servizio tramite KPI
- Simulazioni di situazioni critiche che potrebbero accadere: potrebbe essere utile simulare eventi critici o possibili incidenti che possono accadere all'interno del dominio, in questo modo si possono simulare vari scenari in risposta al problema ed individuare il migliore. Questo aiuterebbe a determinare una serie di protocolli da attuare in caso di emergenza. Inoltre la simulazione potrebbe essere utile per simulare, ad esempio, un alto picco di richieste di servizio e vedere, quindi, come reagirebbe il sistema.

3.2 Challenge da affrontare durante l'analisi e lo sviluppo del caso di studio

- **Latenza di comunicazione tra asset fisico e digital twin**: questa challenge è importante nell'ambito Digital Twin ma dipende dall'applicazione. È lo scenario che determina il grado massimo di latenza tra asset fisico e digital twin. I costi di sviluppo del sistema aumentano significativamente man mano che questo requisito diventa rigoroso. Nel caso di studio preso in considerazione questo requisito non dovrebbe causare problemi in quanto nel sistema non è richiesto un hard real-time.
- **Connettività**: in questo caso di studio la connettività deve essere garantita e disponibile in tutte le centrali operative e su tutti i mezzi utilizzati per il servizio. Questo è un punto chiave in quanto il sistema, in assenza di connessione, perderebbe di utilità in quanto non assicurerebbe dati aggiornati.
- **Interoperabilità**: un punto chiave è la definizione di interfacce comuni per lo scambio di informazioni. In ambito healthcare i dati

mancano di interoperabilità, i dati medici devono essere trasformati in informazioni significative. Al momento la maggior parte dei dati medici sono nascosti in “silos” di dati isolati e incompatibili tra loro. Questi sono difficili da scambiare, elaborare e interpretare. Questa challenge verrà approfondita nella sezione 5.2 in quanto è un punto fondamentale per il dominio Trasporti 116-117.

- **Sicurezza e privacy:** in ambito sanitario la privacy è un argomento complesso in quanto al suo interno sono presenti molti dati sensibili che vanno trattati con una serie di cautele specifiche alla tipologia di dato. Il Gdpr impone, per i dati sanitari, una protezione rafforzata, in generale il trattamento di questi dati è permesso solo nel caso in cui il loro utilizzo sia indispensabile, ad esempio se vengono utilizzati per finalità di cura del paziente da parte di medici o operatori sanitari.

Capitolo 4

Analisi

Durante lo sviluppo del caso di studio Trasporti 116-117 si utilizza un approccio *Domain Driven Design (DDD)*, al fine di ottenere un'analisi e una progettazione chiara, comprensibile e facilmente estendibile. Un altro motivo per il quale si è scelto di utilizzare questo approccio è dato dalla complessità e dalla tecnicità del dominio studiato, il DDD, infatti, permette di comprenderlo gradualmente, aiutando a capire ed analizzare domini anche molto complessi.

In questo capitolo si inizia analizzando i documenti e le informazioni fornite dall'esperto di dominio, da questo si producono possibili scenari e casi d'uso. Si individuano, quindi, i requisiti richiesti.

Successivamente si individuano i bounded context principali, le relazioni tra questi, le entità al loro interno e le relative funzionalità.

4.1 Contesto di riferimento

Il numero 116-117 è il numero unico europeo che permette di accedere a cure mediche non urgenti ed ad altri servizi sanitari secondari, ovvero non di emergenza [2]. Questo servizio è fruibile gratuitamente da tutti i cittadini, gli operatori risponderanno in lingua italiana o inglese.

Il numero garantisce a tutti i cittadini di ricevere risposte a tutte le richieste di cure mediche non urgenti, questo servizio è disponibile 24 su 24 per 7 giorni alla settimana. È un numero a chiamata rapida e non necessita di prefisso.

La struttura organizzativa del servizio in Italia può variare da Regione a Regione. In questo caso di studio si fa riferimento al servizio offerto dall'AUSL Romagna.

Il numero unico europeo 116-117 fornisce una moltitudine di servizi, attraverso questo si gestiscono diversi settori, di seguito in breve riportati:

- **Servizio di continuità assistenziale:** è un servizio che permette di filtrare le chiamate ai medici per le visite domiciliari, alle USCA e alla guardia medica turistica. Inoltre permette di individuare e trasferire le richieste di soccorso sanitario urgente al 118.
- **Centrale trasporti secondari:** permette agli operatori di coordinare i trasporti secondari dell'AUSL Romagna, in modo da consentire una più efficiente gestione dei mezzi di trasporto nei vari ambiti.
- **Infermiere di territorio:** ha l'obiettivo di fornire informazioni agli utenti e svolgere funzioni utili al monitoraggio ed all'assistenza in telemedicina.
- **Telemedicina:** monitoraggio di pazienti a domicilio attraverso sistemi di trasmissione dati che consentono una valutazione dello stato di salute del paziente in modo da fornire, se necessario, risposte sanitarie urgenti.

Per questo caso di studio si terrà in considerazione solamente il settore relativo alla **Centrale trasporti secondari**.

Per una prima esplorazione del caso di studio si è tenuto un incontro con l'esperto del dominio, di seguito sono riportati i punti principali di questo. All'inizio della call il Dottor Menarini spiega com'è nata la gestione delle emergenze/urgenze sanitarie, i trasporti secondari e la guardia medica. Inizialmente nacquero come un unico sistema, ovvero i trasporti secondari erano gestiti dallo stesso 118. Successivamente furono gestiti in maniera distaccata in quanto ritenuti servizi meno importanti ed ora hanno vita propria e sono divise per ambito con varie inefficienze. Con la comparsa del Covid-19 ci si rende conto che manca un'interfaccia tra territorio ed ospedale; il 118 si è dovuto prendere carico anche di situazioni che di fatto non erano di emergenza, ad esempio per informazioni o paura dei cittadini.

Oltre al 118, andrebbero sviluppati molti altri settori come trasporti, telemedicina e teleassistenza che ancora sono mal gestiti o assenti. Il vantaggio

derivante dallo sviluppo di questi settori è quello di avere un collegamento diretto tra cittadino e assistenza sanitaria. Attraverso questa interfaccia si potrebbero dare risposte migliori e indirizzare al meglio il cittadino e, in alcuni casi, intercettare prima i bisogni di quest'ultimo.

Considerando la situazione sanitaria attuale uno dei settori più in sofferenza è proprio quello che dovrebbe essere gestito dal 116-117 poiché non è stato progettato in precedenza.

Il 112 dovrebbe essere utilizzato per le emergenze e il 116-117 per le non emergenze. Idealmente, nel nuovo sistema, nel momento in cui le chiamate non sono indirizzate al giusto centralino, verranno reindirizzate a quello di competenza, poiché le centrali saranno distaccate fisicamente ma interdipendenti tra loro. Questo è un elemento da considerare in fase progettuale.

Già dal 2013 nel decreto rilancio era presente un comma che faceva riferimento alle centrali regionali con il 116-117, ma soltanto dopo 8 anni e l'emergenza Covid-19 se ne è capita davvero l'importanza.

Per quanto riguarda la dotazione dei mezzi di trasporto, nel modello idealizzato del passato erano previsti due sottosistemi, ovvero il trasporto non-emergenziale con ambulanza e il trasporto emergenziale con ambulanza. Questi sottosistemi hanno sempre vissuto in maniera separata ma si è resa evidente l'esigenza che l'uno si debba interfacciare con l'altro, per questo servirebbero dei mezzi equipotenti (soprattutto per quanto riguarda la trasmissione dati), tenuti comunque separati a livello tecnico ma intercambiabili e interfacciabili tra loro. Un esempio di situazione in cui sorge questa necessità, è il caso in cui si esauriscano i mezzi per le emergenze e debbano essere utilizzati, al loro posto, i mezzi adibiti ai trasporti secondari, che però, al momento, non sono equipaggiati a dovere.

Attualmente i mezzi del 118 sono equipaggiati con tablet in connessione con la centrale operativa. Per quello che riguarda i trasporti secondari, non c'è ancora una tracciabilità dei mezzi, ci sono soltanto sistemi di radiocomunicazione e le chiamate viaggiano su telefono. Si può presupporre comunque, la possibilità di avere un tablet associato al mezzo secondario e che i professionisti che usano questi mezzi siano dotati di un proprio tablet/smartphone associato al ruolo e non al mezzo.

Durante la call è stato richiesto all'esperto del dominio di fornire alcune user story per comprendere meglio il dominio. Sono stati identificati tre assi principali: chiamata di emergenza standard, trasporto standard e chiamata

che non rientra in nessuno dei due casi precedenti. Alcuni scenari specifici di casi in cui può essere utile rilevare certe tipologie di dati piuttosto che altre possono essere: trauma, infarto e stroke. In questi scenari si entra in ospedale con un'emergenza ma alla dimissione, potrebbe essere necessario un trasporto secondario per il rientro all'abitazione.

Uno scenario emerso è: una persona entra a Cesena con un trauma cranico. Viene spostata a Ravenna per finire la riabilitazione poi, viene dimessa. Però, prima di concludere il proprio percorso, ha bisogno di essere trasportato altre volte in ospedale per una serie di terapie. Se si segue qualcuno in telemedicina/teleassistenza, può essere che questa persona entri, successivamente, nel sistema per necessità di trasporto o di emergenza e quindi questi aspetti sono strettamente collegati e la tracciabilità deve essere mantenuta in ognuno. Da questo incontro sono emersi due aspetti molto rilevanti: la necessità di avere una tracciabilità sia dal lato dei mezzi di trasporto sia dal lato del paziente e l'importanza dell'interoperabilità e dell'interazione tra i diversi servizi.

4.2 Situazione attuale del servizio

All'interno di AUSL Romagna vi sono tre centrali operative per i trasporti secondari, di cui soltanto una è operativa h24. Queste centrali non sono collegate tra loro o con la centrale del 118, se non per telefono; questo causa una forte disomogeneità che comporta inefficienze rilevanti come ad esempio gestione dei mezzi per ambito e senza fare considerazioni di tipo geografico. Inoltre, a causa della non continuità del servizio, può capitare che le richieste effettuate dopo l'orario di chiusura del sistema di prenotazione, vengano passate alla centrale operativa 118 utilizzando ambulanze destinate all'emergenza territoriale.

È stato calcolato che i tempi di impiego delle ambulanze destinate alle emergenze per codici bianchi o verdi siano all'incirca 6'000 ore l'anno (circa 16 ore al giorno, l'1,60% delle ore/mezzo destinate all'emergenza territoriale). Nel modello attuale la centrale operativa 118 gestisce le emergenze ed i trasporti interospedalieri urgenti, ma anche i trasporti sanitari non urgenti nelle ore notturne. La situazione ha attualmente una serie di punti critici:

- assenza di un software unico tra le varie centrali con conseguenti difficoltà organizzative,

- assenza di un sistema di telecomunicazione unico con conseguente difficoltà di comunicazione tra le centrali ed i mezzi tra ambiti diversi,
- aumento dei costi progressivo senza evidenti benefici per il sistema,
- assenza di tracciabilità degli spostamenti con conseguente difficoltà nell'organizzare e tracciare i tempi degli stessi.

Inoltre serve considerare che un paziente può entrare nel sistema trasporti in diversi modi:

- con trasporto di emergenza per un primo soccorso, richiedendo successivamente trasferimenti per trattamenti o consulenze non emergenziali,
- con trasporto di emergenza e successive dimissione dal pronto soccorso,
- con trasporto non urgente dal domicilio al pronto soccorso e successivamente in ospedale di ricovero diverso dalla destinazione iniziale.

In questi vari scenari non si ha a disposizione una tracciabilità di tutti gli spostamenti del paziente, in quanto al momento i vari sistemi sono disomogenei e con scarsi livelli di interoperabilità.

4.3 Proposta di riorganizzazione

Nell'ambito dei trasporti si identificano due sistemi (figura 4.1) distinti, ma connessi tra loro:

- Sistema di emergenza sanitaria territoriale gestita dalla centrale operativa del 118: questa gestisce solamente i servizi di emergenza preospedalieri, l'intervento si conclude, quando necessario, con un trasporto verso l'ospedale più idoneo.
- Sistema trasporti sanitari gestito dalla centrale operativa unificata trasporti: gestisce solamente i trasporti secondari, ovvero quelli non emergenziali. Questo sistema può mantenere fisicamente sedi distinte ma necessita di un unico coordinamento operativo.

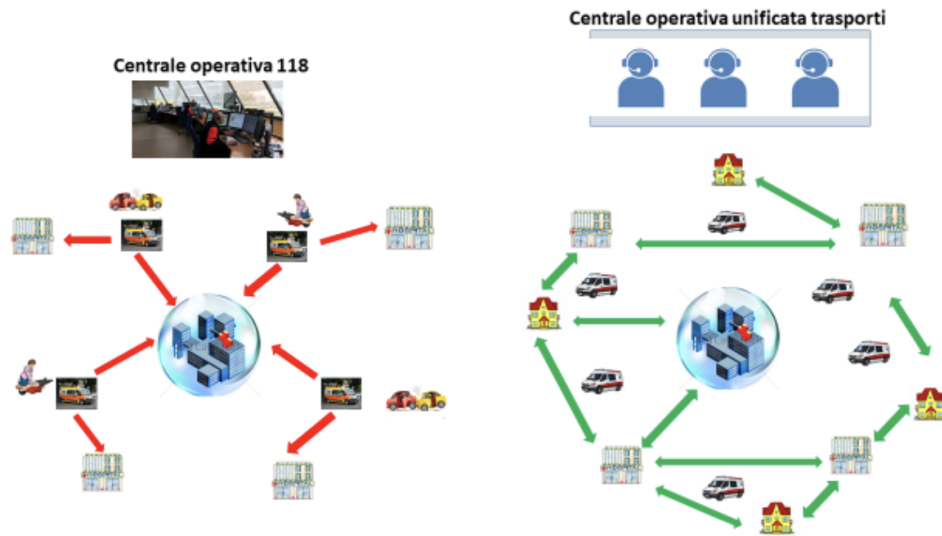


Figura 4.1: Condizione ideale delle centrali operative

Nella realtà può accadere che il primo ospedale di destinazione non sia quello idoneo e che si rilevi la necessità di trasferimento del paziente in un altro ospedale. Poiché si tratta di un trasferimento a carico dei trasporti secondari ma non programmato, il trasporto potrebbe non essere disponibile e potrebbe essere necessario rivolgersi al 118. La sinergia tra i due sistemi è un elemento di forza per rispondere meglio alle esigenze: è necessaria un'organizzazione che tenga conto di entrambi i sistemi, per evitare inefficienze, sovrapposizioni e ritardi.

Dallo schema riportato in figura 4.2 emerge un elemento fondamentale: la vicinanza tra le due centrali operative, del 118 e del 116-117. È importante considerare che in caso di emergenza, come è successo ad esempio per la pandemia COVID-19, deve essere possibile rimodulare rapidamente il sistema per fornire la risposta adeguata alla situazione.

Il modello proposto prevede inizialmente il mantenimento delle tre centrali separate ma, unificate a livello organizzativo. In un futuro si vorrebbe creare un'unica centrale operativa per i trasporti secondari come riportato in figura 4.1.

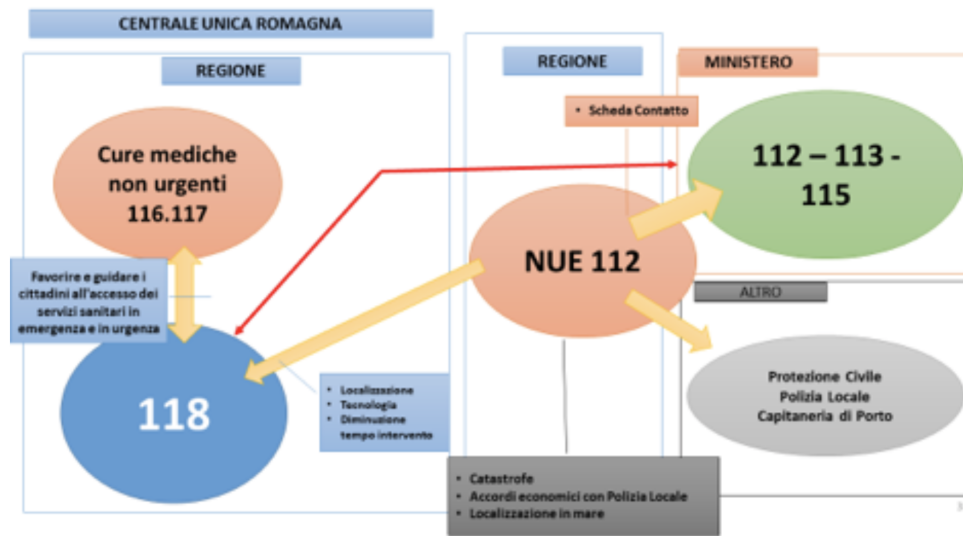


Figura 4.2: Proposta nuovo modello di comunicazione tra i vari settori (112, 118, 116-117)

4.4 Scenari e casi d'uso

In questa sezione si riportano alcuni ipotetici scenari che possono verificarsi all'interno del dominio. Si mostra inoltre il diagramma dei casi d'uso in figura 4.3. Successivamente si analizzano i casi d'uso nel dettaglio tramite domain storytelling, in figura 4.4, 4.5, 4.6 e 4.7, e diagrammi delle attività, in figura 4.9, 4.10 e 4.11.

4.4.1 Trasporto per dimissioni da ospedale

Un paziente ricoverato al Bufalini di Cesena al reparto di ortopedia, al termine della propria degenza, deve essere trasferito a casa mediante trasporto in ambulanza poiché ha una mobilità limitata. Il giorno della dimissione, il paziente deve essere trasferito dall'ospedale Bufalini di Cesena al proprio domicilio a Gambettola. L'infermiere del reparto contatta il 116-117 per organizzare il trasporto. Il risponditore di cortesia risponde all'utente e lo mette in attesa del primo operatore disponibile. Al termine dell'attesa l'utente viene messo in contatto con l'operatore (Storytelling in figura 4.4).

Durante la telefonata, l'operatore del centralino chiede la partenza e la destinazione del paziente, l'orario delle dimissioni, le condizioni di mobilità, lo stato di salute e i dati anagrafici del paziente. L'infermiere fornisce i dati anagrafici del paziente, inoltre, informa che il paziente dovrà essere trasferito dal Bufalini di Cesena a Gambettola, la dimissione avverrà all'incirca verso le 14:00 e che il paziente può stare solamente in posizione supina, per questo motivo sarà necessario l'accompagnamento fino all'interno dell'abitazione (Storytelling in figura 4.5).

A questo punto l'operatore registra la prenotazione dell'ambulanza, inserendo all'interno del sistema i dati anagrafici, lo stato di salute e l'autonomia del paziente, inoltre inserisce l'itinerario relativo al trasporto, la data e ora ed infine controlla i veicoli che possono effettuare il trasporto ed associa l'ambulanza alla prenotazione (Storytelling in figura 4.6).

Il giorno dell'appuntamento l'operatore che guida l'ambulanza, dopo aver concluso il precedente trasporto, controlla il nuovo trasporto da effettuare sul tablet associato all'ambulanza. Alle 14:00 parte per raggiungere l'ospedale. Alle 14:15, l'operatore arriva in reparto per prelevare il paziente e lo trasporta fino al domicilio a Gambettola. Una volta che il paziente si trova a destinazione, il processo del trasporto si ritiene concluso e l'ambulanza può eseguire altri trasporti (Storytelling in figura 4.7).

4.4.2 Trasporto per visita da casa a ospedale, paziente con ridotta mobilità

Un paziente rimasto paralizzato a seguito di un ictus ha in programma una visita per il mese successivo e intende prenotare l'ambulanza che lo trasporti in ospedale. Il tutore del paziente effettua una chiamata al 116-117 per prenotare il trasporto. Il centralino chiede i dati anagrafici, gli indirizzi di partenza e destinazione, la data e l'orario in cui si dovrà svolgere il trasporto, lo stato di salute e le condizioni di mobilità del paziente. Il tutore fornisce i dati richiesti ovvero: il trasporto sarà con partenza da Viale della Libertà, 15 a Forlì e destinazione Ospedale Santa Maria delle Croci Ravenna. La visita è prevista per le ore 10:00. Le due parti si accordano sul fatto che sarà necessario anche il viaggio di ritorno dall'ospedale al domicilio, che verrà prenotato per le 11:30. A questo punto, l'operatore del centralino controlla la disponibilità e conferma le due prenotazioni (andata e ritorno) e le inserisce nel sistema assegnando un'ambulanza.

Il giorno del trasporto, l'ambulanza, appartenente all'ente ravennate, si trova già a Forlì per un precedente trasporto. Quindi, alle 9:00 si porta verso l'indirizzo di partenza fornito dal paziente. Alle ore 9:40, l'ambulanza giunge all'indirizzo di destinazione, presso l'ospedale di Ravenna e trasferisce il paziente in ospedale. Il processo del trasporto di andata si ritiene concluso e l'ambulanza può eseguire altri trasporti. Alle 11:20 l'operatore di un'altra ambulanza, che si trovava già a Ravenna per il trasporto di un altro paziente, recupera il paziente e lo trasporta al suo domicilio. Il processo del trasporto di ritorno si ritiene concluso e l'ambulanza può eseguire altri trasporti.

4.4.3 Trasporto per trattamenti ricorrenti

Un paziente, a seguito di un grave incidente, ha necessità di seguire un trattamento di ossigenoterapia iperbarica che consiste in 15 sedute di durata di sessanta minuti ciascuna, da ripetere una volta a settimana. Il paziente, al momento, si trova ricoverato all'ospedale Bufalini di Cesena e dovrà effettuare il trattamento al centro iperbarico di Ravenna. L'infermiera del reparto contatta il centralino del 116-117 e richiede il trasporto di andata e di ritorno del paziente, ogni giovedì alle 10:00, per tutta la durata del ricovero. Il centralino controlla la disponibilità, conferma le varie prenotazioni e le inserisce nel sistema assegnando per ognuna l'ambulanza disponibile in tal data.

Il giorno del trasporto, l'ambulanza preleva il paziente alle 9:00, dall'ospedale Bufalini di Cesena e alle 9:40 arriva al centro iperbarico di Ravenna. Il processo del trasporto di andata si ritiene concluso e l'ambulanza può eseguire altri trasporti. L'operatore che si è occupato del trasporto, poiché non ha altri trasporti da effettuare e non ci sono colleghi in zona per il ritorno, aspetta il termine della terapia (intorno alle 11:00) e riporta il paziente all'ospedale Bufalini con arrivo alle 11:45. Una volta che il paziente è nuovamente in reparto il processo del trasporto di ritorno si ritiene concluso e l'ambulanza può eseguire altri trasporti.

4.4.4 Trasporto per trasferimento da un ospedale ad un altro, non programmato

La notte del 2 febbraio un paziente ha un grave incidente e viene trasportato dall'ambulanza del 118 al pronto soccorso di Ravenna. Dopo i primi

soccorsi viene ricoverato nel reparto di traumatologia dell'ospedale Santa Maria delle Croci di Ravenna. Il 5 febbraio i medici decidono di trasferire il paziente al Bufalini di Cesena, il trasporto deve essere effettuato in giornata. L'infermiera del reparto contatta il 116-117 per chiedere la disponibilità di un trasporto immediato. Il centralino controlla la disponibilità delle ambulanze e conferma la possibilità di effettuare il trasporto, dopodichè lo inserisce nel sistema assegnandogli un ambulanza.

Alle 12:30 del 5 febbraio l'operatore dell'ambulanza vede il nuovo trasporto da effettuare sul tablet e si dirige verso l'ospedale di Ravenna. A quel punto preleva il paziente dal reparto e lo trasporta al Bufalini di Cesena. Una volta portato il paziente in reparto il processo di trasporto si considera concluso.

4.4.5 Giornata lavorativa di un operatore trasporto

L'operatore, settimanalmente, controlla i turni della settimana successiva e la relativa ambulanza che dovrà utilizzare.

I turni sono assegnati dalla centrale operativa, essa ha il compito di decidere i giorni lavorativi dell'operatore e di assegnargli un'ambulanza.

L'operatore trasporto, 15 minuti dall'inizio del turno lavorativo, si reca nella sua sede di Cesena per prendere l'ambulanza a lui assegnata ed iniziare i trasporti della giornata. Per prima cosa controlla lo stato dell'ambulanza e si accerta che essa sia pronta per essere utilizzata. Successivamente consulta il calendario attività dal tablet dell'ambulanza e individua il primo trasporto da effettuare. Dopodichè si dirige verso l'indirizzo di partenza, preleva il paziente e lo porta a destinazione. Una volta giunti a destinazione scarica il paziente e conclude il primo trasporto. L'operatore consulta nuovamente il tablet e individua il prossimo trasporto da effettuare.

L'operatore, dopo aver effettuato i nove trasporti previsti nella giornata, si reca a rifornire il mezzo e lo riporta alla propria sede. Una volta depositato il mezzo la giornata lavorativa dell'operatore si ritiene conclusa (Storytelling in figura 4.8).

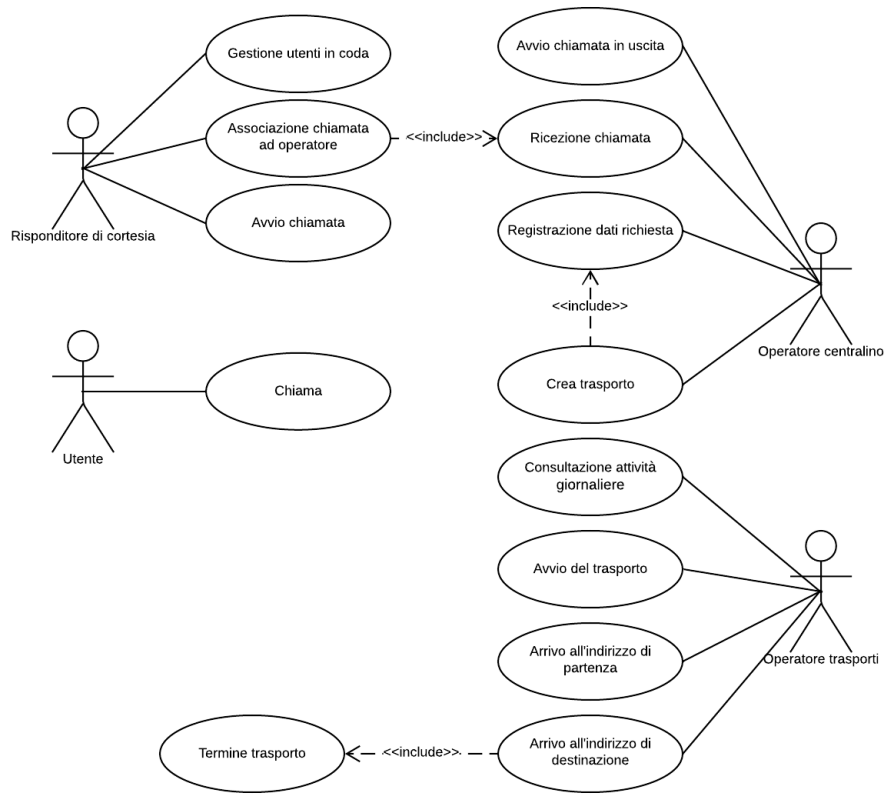


Figura 4.3: Diagramma dei casi d'uso

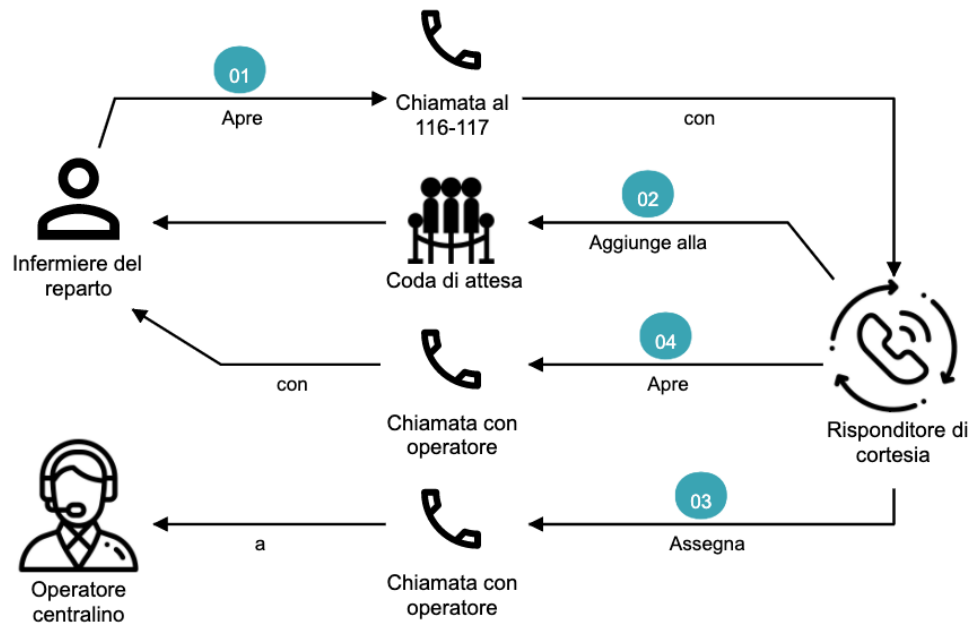


Figura 4.4: Domain Storytelling rappresentante la gestione della chiamata da parte del risponditore di cortesia

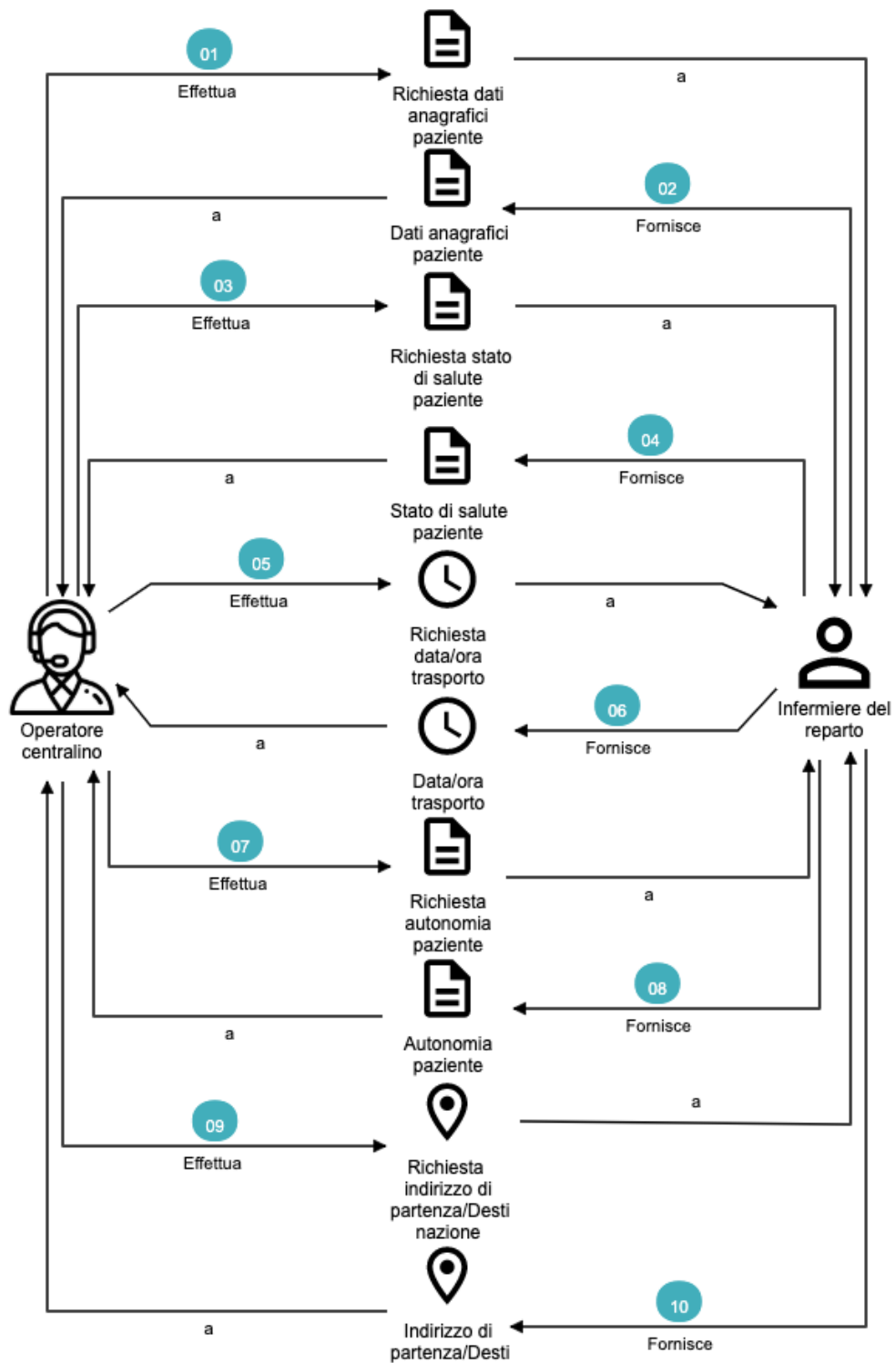


Figura 4.5: Domain Storytelling rappresentante la chiamata tra operatore e utente

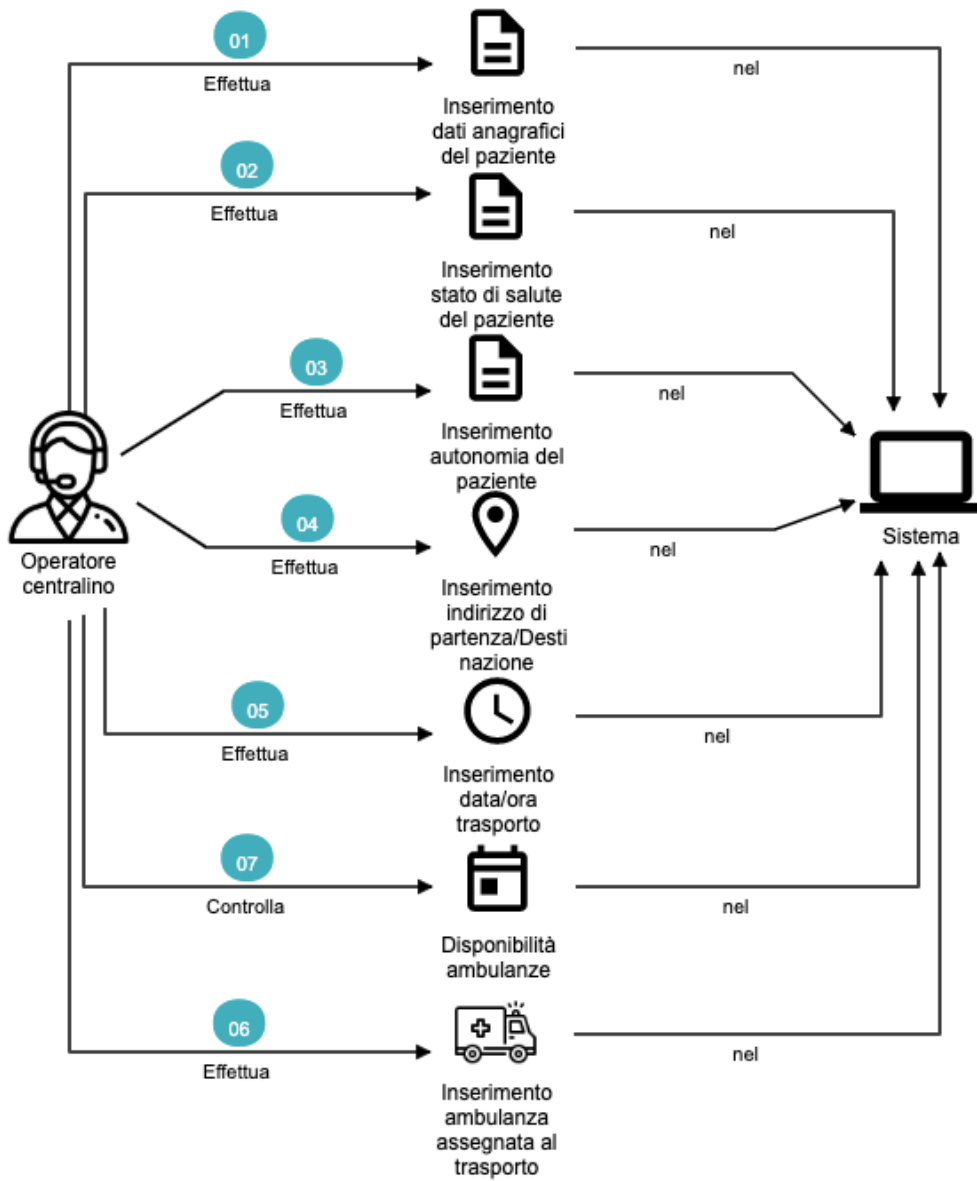


Figura 4.6: Domain Storytelling rappresentante l'inserimento del trasporto nel sistema

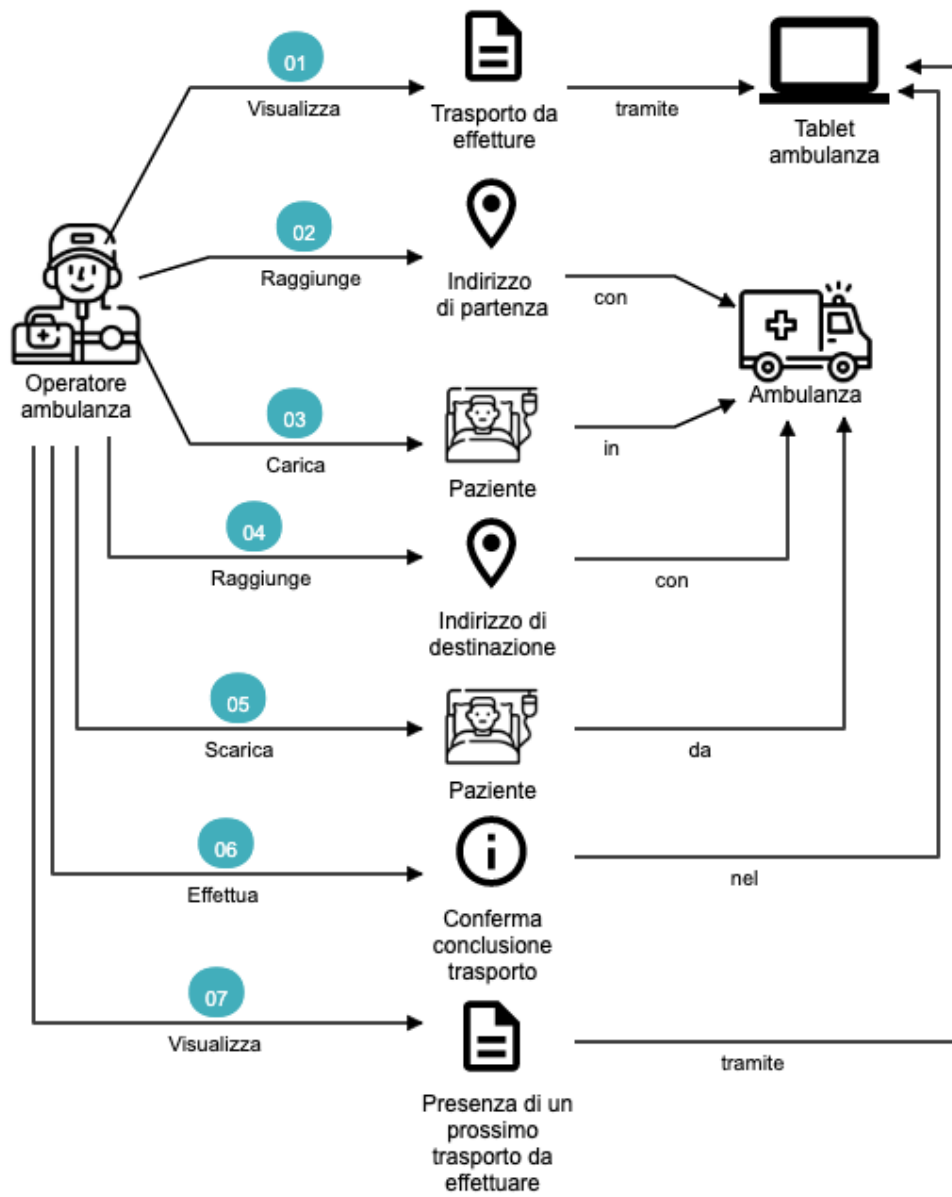


Figura 4.7: Domain Storytelling rappresentante l'esecuzione del trasporto

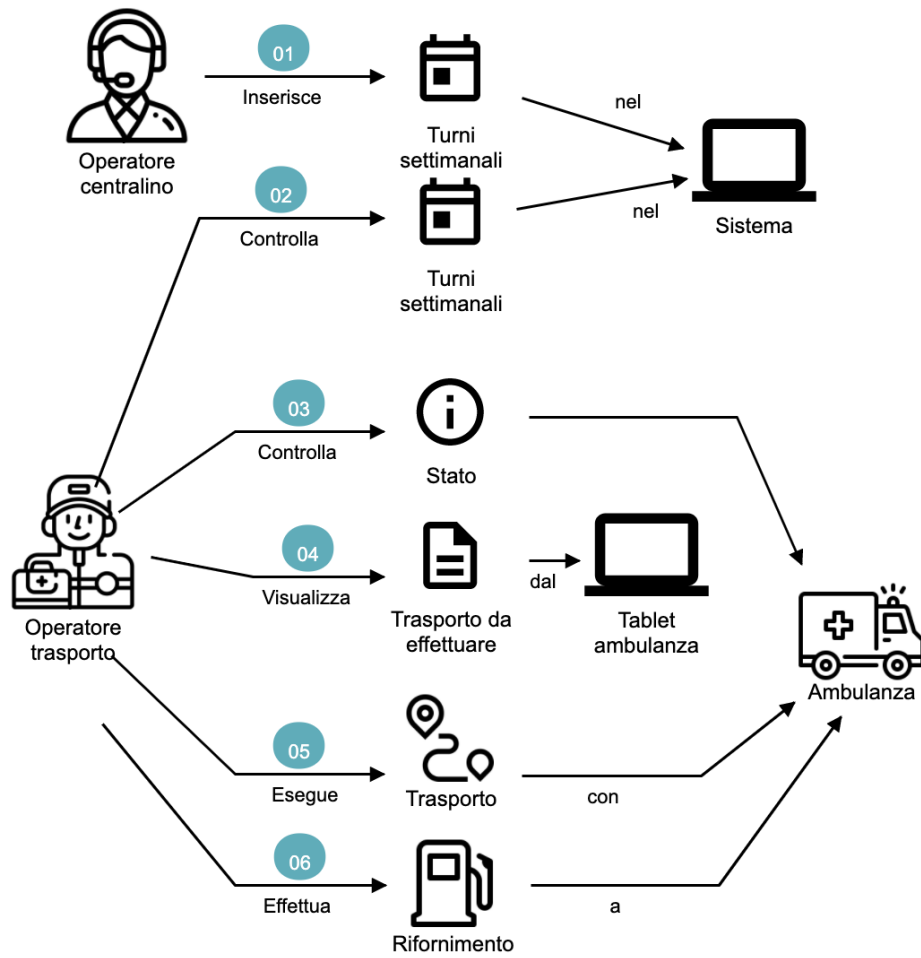


Figura 4.8: Domain Storytelling rappresentante la giornata lavorativa di un operatore trasporto

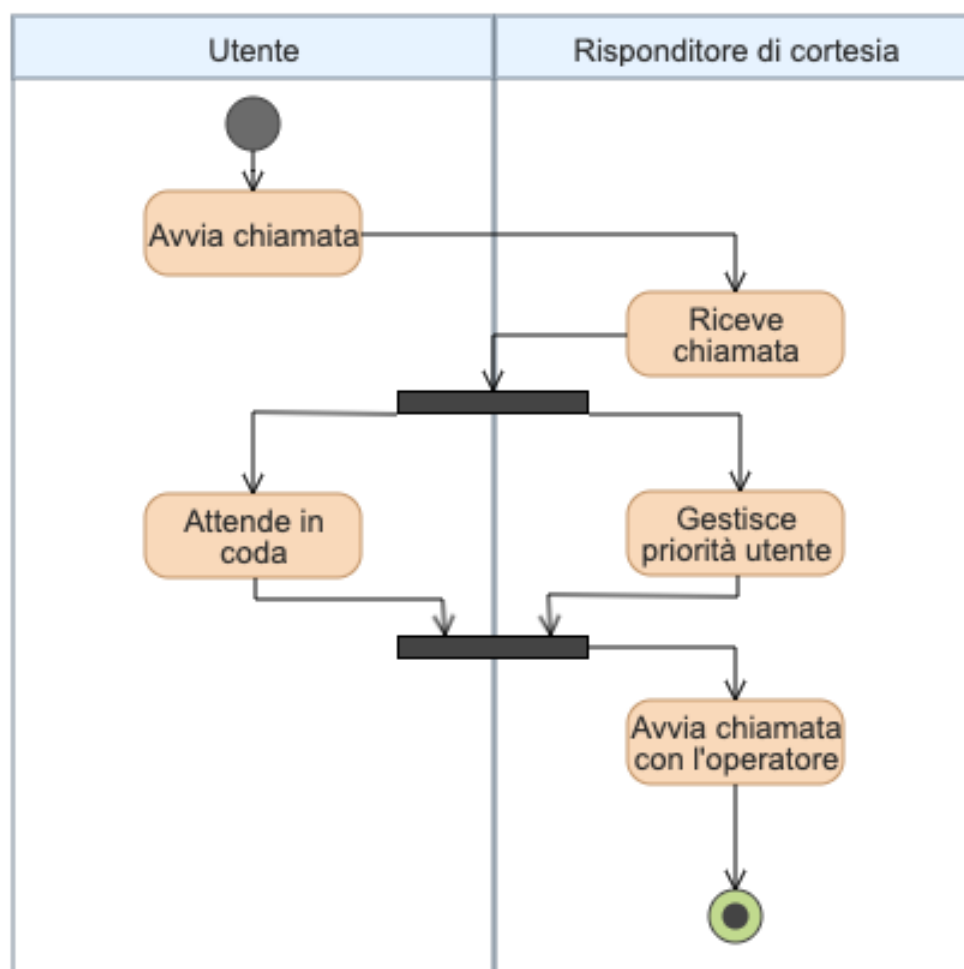


Figura 4.9: Diagramma delle attività rappresentante l'attesa in coda dell'utente

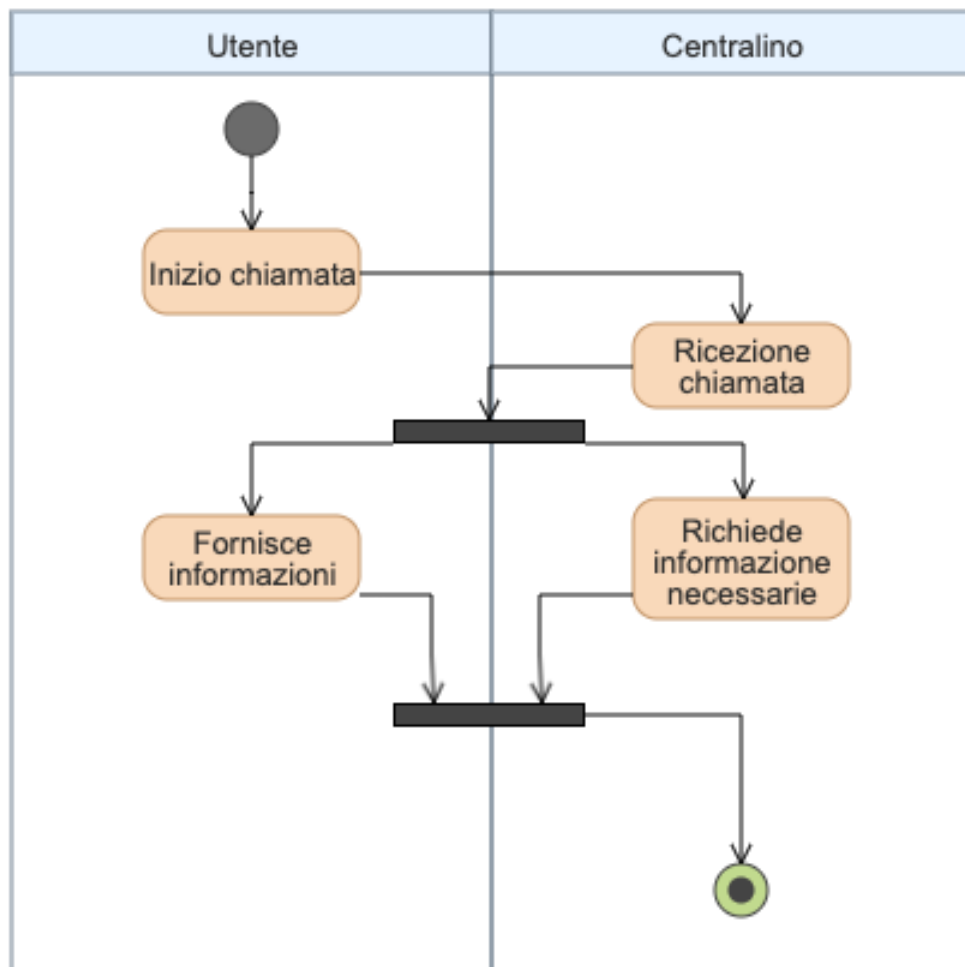


Figura 4.10: Diagramma delle attività rappresentante la chiamata tra operatore e utente

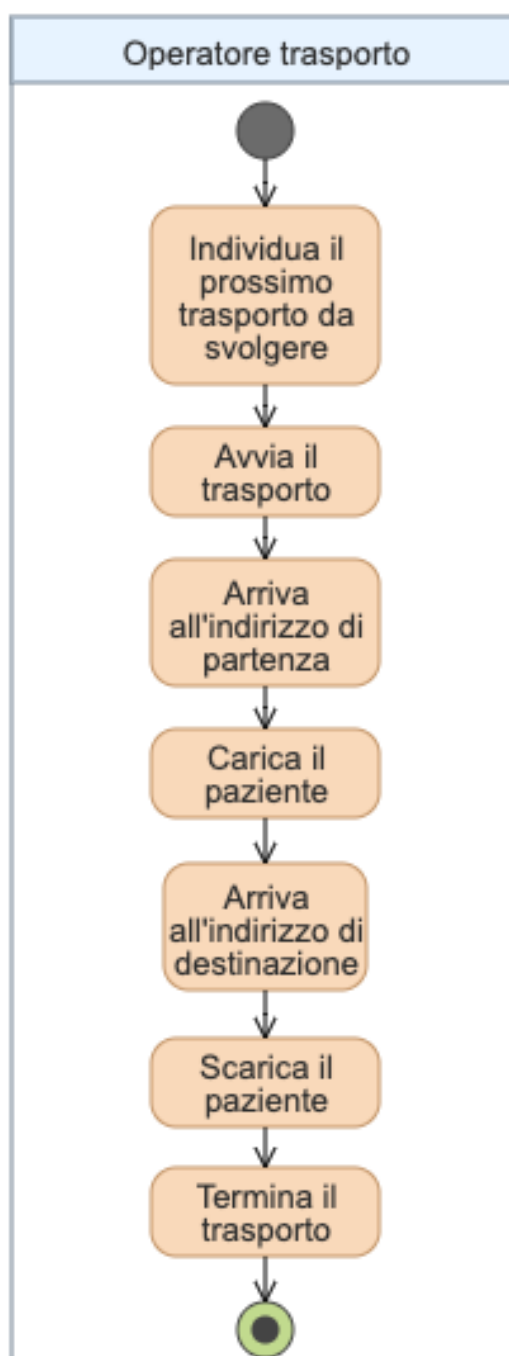


Figura 4.11: Diagramma delle attività rappresentante l'esecuzione del trasporto

4.5 Ubiquitous language

Il Domain Driven Design consiglia l'utilizzo dell'ubiquitous language, un linguaggio comune tra sviluppatori, stakeholder, esperti del dominio e utenti del dominio. Questo è molto utile per evitare ambiguità quando si comunica con gli esperti di dominio.

Nella tabella seguente è riportato l'ubiquitous language relativo al dominio trasporti 116-117:

ADI (Assistenza Domiciliare Integrata)	Formula assistenziale dedicata agli anziani e a tutte le persone che non sono autosufficienti. Il servizio viene erogato gratuitamente al domicilio del paziente.
Attività giornaliera	Evento inserito all'interno del calendario al momento della prenotazione da parte dell'operatore del centralino e consultato dagli operatori addetti al trasporto.
Calendario trasporti programmati	Insieme delle attività giornaliere.
Centrale operativa	Parte amministrativa del sistema che si occupa di gestire le chiamate degli utenti e memorizzare i trasporti, i veicoli e i pazienti. È formata dagli operatori centralino.
Operatore	Si tratta delle figure addette a fornire il servizio. Si distinguono in due tipologie: <ul style="list-style-type: none"> • Operatore del centralino: si occupa di rispondere alle chiamate degli utenti che vogliono richiedere un servizio di trasporto. Inoltre hanno il compito di memorizzare i trasporti all'interno del calendario dei trasporti programmati. Compongono la centrale operativa. • Operatore trasporto: è l'operatore che effettua il trasporto nel giorno prestabilito.

Patologie tempo-dipendenti	Pazienti che da un pronto soccorso o da un'area ospedaliera hanno una condizione di emergenza che richiede la centralizzazione in un hub.
Paziente	Entità passiva che usufruisce del servizio di trasporto.
Risponditore di cortesia	Risponditore automatico che mette in attesa l'utente che effettua la chiamata al 116-117 fino a che non sarà disponibile un operatore centralino.
Sistema	Sistema informativo contenente tutti i dati e le informazioni relativi al dominio 116-117
Tablet	Tablet presente all'interno di ogni veicolo che consente di visualizzare dati (come ad esempio i trasporti da effettuare). è equipaggiato con un sistema GPS che permette di tracciare l'ambulanza.
Trasporti Inter-ospedalieri	<p>Trasporti generati da esigenze di consulenze e trasferimenti di pazienti all'interno della rete ospedaliera aziendale (comprese gli ospedali privati accreditati). Si possono distinguere in:</p> <ul style="list-style-type: none"> • Trasporti in emergenza per patologie tempo-dipendenti (di fatto trasporti primari a partenza ospedaliera) o pazienti critici e complessi che richiedono letti intensivi o interventi chirurgici in emergenza. • Trasporti in urgenza, con necessità di tipo clinico ma non in emergenza e senza accompagnamento medico. • Trasporti non urgenti, in genere programmati per consulenze e trasferimenti.
Trasporti primari	Si definiscono trasporti primari quelli generati da emergenze/urgenze extraospedaliere gestite dalla centrale 118 con necessità di ospedalizzazione del paziente nell'ospedale idoneo.

Trasporti secondari, Trasporti non urgenti	Trasporti sanitari da e verso domicilio o struttura residenziale e strutture sanitarie. Si tratta dei trasporti che non hanno carattere di urgenza ma che rientrano in percorsi di cura di persone che necessitano di trasporto sanitario o di accompagnamento da e verso il domicilio (trasporti di pazienti ADI e dializzati, nonché prime visite postoperatorie).
Trasporto	Rappresenta tutto il processo di trasferimento del paziente. Inizia dal momento in cui l'operatore centralino salva la prenotazione e termina nel momento in cui l'operatore porta il paziente a destinazione. Si suddivide in 3 fasi: <ul style="list-style-type: none"> • Trasporto prenotato • Trasporto in corso • Trasporto completato
Utente	Entità attiva che richiede il servizio e comprende sia l'infermiere del reparto che richiede il trasferimento di un paziente, sia il tutore di un paziente, sia il paziente stesso che richiede autonomamente il servizio.
Veicolo	Rappresenta qualsiasi mezzo di trasporto utilizzato per soddisfare i trasporti 116-117, come ad esempio l'ambulanza, l'eliambulanza e l'automedica.

4.6 Requisiti del sistema

Durante l'analisi sono stati individuati, unitamente all'esperto del dominio, le seguenti caratteristiche e requisiti:

4.6.1 Caratteristiche del NUE 116-117

- Il servizio 116-117 è uno strumento di comunicazione che si rivolge a tutti i cittadini, italiani e stranieri, senza obbligo di registrazione preventiva.
- Il numero è unico in Italia ed in Europa.
- È un numero a chiamata rapida e non necessita di prefisso.
- Il numero è disponibile H24 per 7 giorni alla settimana.
- Fornisce assistenza e/o informazioni.
- Il servizio non è limitato nel tempo.
- Non è richiesto all'utente alcun pagamento per la chiamata.
- Le chiamate possono essere effettuate da qualunque apparecchio.

4.6.2 Requisiti

- **È auspicabile l'integrazione di soluzioni tecnologiche tra 118, 112 NUE e NE 116-117:** mantenendo comunque distinto l'accesso degli utenti alle due numerazioni, al fine di garantire affidabilità e sicurezza di funzionamento al sistema, presa in carico di chiamate provenienti da cittadini diversamente abili, fruibilità dei dati relativi ai pazienti da parte dei sanitari, utilizzo di sistemi di mediazione multilingue, registrazione informatizzata delle attività di localizzazione dei mezzi e di teletrasmissione.
- **Corretto utilizzo dei sistemi di gestione automatica:** nel centro di risposta 116-117 non sono previsti sistemi di risposta automatica che chiudono la chiamata con un messaggio informativo né segreterie telefoniche, così come sistemi di istradamento delle chiamate entranti, possono essere previsti risponditori di cortesia o sistemi di messaggistica che informano il chiamante sui tempi previsti di attesa.
- **Registrazione informatizzata dell'attività:** registrazione vocale delle chiamate in entrata ed in uscita e loro archiviazione, con possibilità di riascolto delle registrazioni archiviate secondo le disposizioni

in tema di sicurezza vigenti, anche al fine dell'elaborazione dei dati e dell'utilizzo di specifici indicatori che permettano di monitorare l'attività di servizio.

- **Tempi di attesa:** è stato stabilito a livello nazionale che il tempo di attesa della prima risposta da parte dell'operatore, a regime del servizio, non dovrà essere superiore ai 2 minuti nel 75% dei casi.
- **Governo delle centrali funzionalmente connesse nell'attività ordinaria ed emergenziale:** ovvero, la centrale 118 per la risposta all'emergenza sanitaria e la centrale dei trasporti secondari.
- **Governo unico delle centrali 118 e 116-117:** che assicura un coordinamento omogeneo basato sull'esperienza maturata con la centrale unica Romagna 118 e che massimizza l'efficienza della risposta all'utente per situazioni di urgenza.
- Solida interfaccia tra la richiesta anche con **servizio di interpretariato** in caso di persone che non parlano italiano, e la struttura in grado di fornire la risposta, anche indiretta, alle esigenze dell'utente. Le principali lingue disponibili sono italiano e inglese.
- **Infrastruttura tecnologica comune:** che consente un'uniforme raccolta dati, base per l'analisi dei servizi erogati che permette di avere risposte pronte a specifici quesiti come di predisporre report periodici sull'attività svolta e la qualità degli interventi.
- **Modellazione dell'ambulanza:** Si assume che tutte le ambulanze, sia quelle del reparto 116-117 sia quelle del reparto 118, siano equipaggiate allo stesso modo, con un dispositivo per il tracciamento e un tablet per il controllo e la comunicazione con la centrale operativa. Così che, in caso di necessità, siano intercambiabili con quelle disponibili per il servizio 118.
- **Tracciamento dei percorsi al fine di analisi.**
- **Storico degli eventi passati:** si vuole tenere traccia dei trasporti passati già conclusi, al fine di future analisi ed eventuali necessità di controllo.

- **Registrazione di eventi di trasporto futuri:** si vogliono memorizzare i trasporti prenotati per scopi organizzativi dell'occupazione dei mezzi e degli operatori.
- **Modellazione del paziente:** al fine di mantenere traccia di tutti i servizi di cui ha usufruito in modo da poter effettuare opportune verifiche in caso di necessità.
- **Possibilità di fruizione e scambio dei dati da parte di sistemi esterni:** ciò implica un elevato livello di interoperabilità.

4.7 Context map

All'interno del dominio Trasporti 116-117 sono stati individuati i seguenti tre sub-domain:

- **Trasporti:** identificato come core domain in quanto è la parte centrale del dominio ed è il sub-domain più complesso.
- **Paziente:** identificato come supporting-domain in quanto è fondamentale per il sub-domain trasporti ma ha una complessità molto bassa.
- **Centralino:** identificato come generic-domain in quanto la struttura è quella di un comune centralino. Come si evince dalle informazioni fornite dall'esperto di dominio questo sub-domain all'inizio verrà mantenuto su tre centrali separate, unificate solo a livello organizzativo. Successivamente l'obiettivo sarà quello di creare un'unica centrale operativa.

4.7.1 Bounded Context

Durante l'analisi del dominio sono stati individuati 4 bounded context di seguito illustrati.

- **Centralino:** bounded context che riguarda la ricezione della chiamata dell'utente da parte del centralino e la gestione della stessa, con l'eventuale trasferimento della chiamata ad un diverso operatore nel

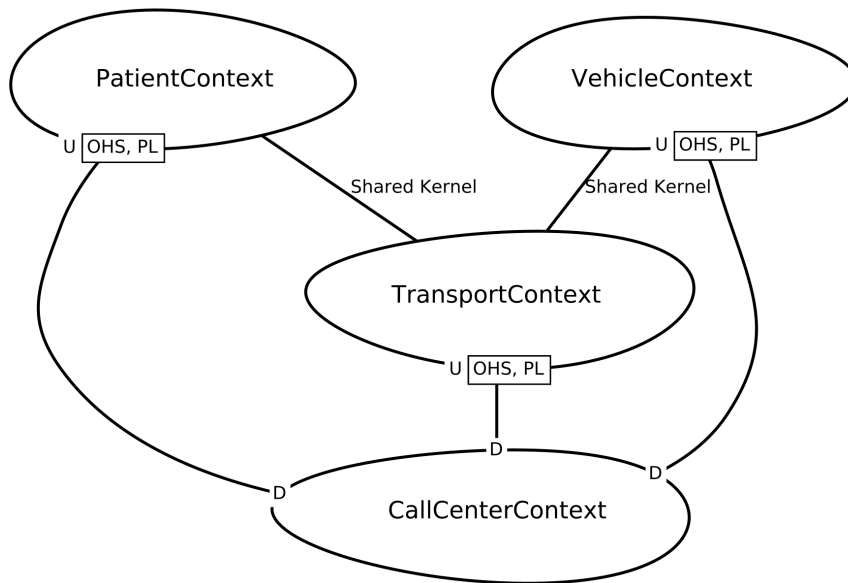


Figura 4.12: Context map

caso in cui si tratti di un utente che parla una lingua straniera e l'eventuale attesa in coda nel caso in cui tutti gli operatori del centralino siano occupati. Il centralino si occupa, inoltre, della memorizzazione di nuovi trasporti, veicoli e pazienti.

- **Paziente:** bounded context in cui sono state racchiuse tutte le informazioni relative al paziente da trasportare, compreso il suo stato di salute e il grado di autonomia, utile da conoscere per organizzare il trasporto.
- **Trasporto:** bounded context centrale del sistema che riguarda tutte le fasi del trasporto. Il ciclo di vita del trasporto inizia con la prenotazione dello stesso e termina con l'arrivo a destinazione del paziente, la destinazione non è intesa solo come indirizzo di destinazione ma anche come reparto o interno dell'abitazione, in caso il paziente ne avesse l'esigenza.
- **Veicolo:** bounded context contenente tutte le informazioni riguardanti i veicoli utilizzati per il trasporto dei pazienti e l'operatore che li guida. In uno stadio iniziale del dominio l'unico veicolo presente

all'interno del sistema 116-117 è l'ambulanza, ma si lascia aperta la possibilità di aggiungere altre tipologie di veicolo in caso di bisogno futuro.

Tra i 4 bounded context sono state individuate le seguenti relazioni, come si evince dalla figura 4.12:

- **Trasporto ↔ Veicolo, Trasporto ↔ Paziente:** per la relazione tra questi due bounded context si è scelta inizialmente una relazione di partnership, in quanto il loro sviluppo deve essere cooperativo. Successivamente si è scelto di rifattorizzare in una relazione shared kernel in quanto i tre bounded context condividono una piccola parte di funzionalità.
- **Centralino ← Trasporto, Centralino ← Paziente, Centralino ← Veicolo:** in questo caso si è scelto di utilizzare un open host service abbinato ad un published language.

Il published language rappresenta un linguaggio comune condiviso tra i due bounded context, all'interno del dominio 116-117 è rappresentato da FHIR, come si approfondirà nel capitolo 5.

Si è poi scelto di utilizzare un open host service in quanto i bounded context Vehicle, Transport e Patient dovranno fornire ai propri client un modo per consultare ed, in alcuni casi, modificare le informazioni al loro interno. L'open host service, infatti, permette di fornire un protocollo ben definito che abilita l'accesso ai tre bounded context. Questo protocollo è aperto, in modo che chiunque voglia avvalersi dei servizi messi a disposizione dai bounded context abbia la possibilità di farlo. In questo stadio di evoluzione del sistema l'unico client è il bounded context centralino, ma in un ampliamento futuro del dominio queste funzionalità potrebbero servire anche ad altri. Un esempio potrebbe essere la necessità del dominio relativo al 118 di accedere a informazioni relative a pazienti all'interno del sistema 116-117, oppure l'utilizzo, sempre da parte del dominio 118, di ambulanze appartenenti al dominio 116-117.

4.7.2 Entità del dominio

In questa sottosezione si andranno ad analizzare le varie entità individuate all'interno dei quattro bounded context riportati nella sottosezione 4.7.1.

Bounded Context Veicolo

- **Ambulanza:** rappresenta il mezzo fisico che trasporta il paziente ed è l'entità principale di questo bounded context. A bordo del veicolo è presente un tablet connesso in rete consultabile dall'operatore. Da questo l'operatore deve poter visualizzare i trasporti previsti per la giornata compresi di orario, partenza e destinazione. Inoltre l'ambulanza è equipaggiata con un GPS. In un futuro, se dovesse essere necessario integrare il modello con altri tipi di veicoli, sarà possibile farlo.
- **GPS:** deve permettere il tracciamento real-time del veicolo, ciò è necessario per tenere traccia di ogni trasporto svolto durante la giornata, inoltre è utile per individuare eventuali inefficienze durante i percorsi.
- **Operatore trasporto:** persona fisica alla guida il veicolo che esegue il trasporto. L'operatore viene assegnato al veicolo che deve condurre e non ai trasporti che deve svolgere durante la giornata.

Bounded Context Trasporto

- **Trasporto:** è l'unica entità all'interno di questo bounded context. È l'asset centrale del dominio Trasporti 116-117. Rappresenta tutto il processo di trasporto, il suo ciclo di vita inizia al momento della prenotazione di un trasporto e si conclude con l'arrivo a destinazione del paziente. Associata a questa entità si ha un enum fase che descrive, appunto, le fasi del suo ciclo di vita, di seguito illustrate:
 - Trasporto prenotato: L'entità nasce in questa fase, rappresenta quindi la prenotazione di un trasporto. Può capitare che durante questa fase l'entità trasporto non sia ancora associata ad un determinato veicolo.

- Trasporto in corso: Si passa in questa fase quando un operatore prende in carico un trasporto, ovvero quando parte per arrivare al luogo di partenza.
- Trasporto completato: Si passa in questa fase quando il paziente arriva a destinazione.

Bounded Context Paziente

- **Paziente**: persona fisica che usufruisce del servizio del trasporto. Ha tre value object che descrivono i dati anagrafici, l'autonomia del paziente e il suo stato di salute. Queste ultime due informazioni possono essere utili all'operatore per attrezzarsi adeguatamente in caso di pazienti in condizioni difficili. Fondamentalmente è un'entità passiva all'interno del sistema e non ha compiti specifici da svolgere, se non la memorizzazione dei dati relativi al paziente.

Bounded Context Centralino

- **Chiamata**: entità principale del bounded context. Rappresenta la chiamata tra l'operatore del centralino e l'utente che richiede il servizio. Questa entità ha come funzione principale mantenere le informazioni relative alla chiamata e la sua registrazione per eventuali problemi o indagini.
- **Operatore centralino**: persona fisica incaricata alla comunicazione con l'utente. Si deve occupare di richiedere tutti i dati necessari alla prenotazione del trasporto, come l'orario, il luogo di partenza e il luogo di destinazione, i dati e le condizioni del paziente.
- **Risponditore di cortesia**: rappresenta il sistema di risposta automatica, ha il compito di mettere in coda gli utenti in attesa di un operatore. Inoltre si occupa di aprire le chiamate tra operatore e utente in base al turno ed alla lingua parlata.
- **Utenti in coda**: rappresenta l'effettiva coda di utenti. Questa memorizza la lingua parlata dall'utente, in modo da indirizzare il risponditore di cortesia alla scelta dell'operatore adatto ad esso. L'entità è molto importante anche per l'analisi del sistema in quanto agevola nell'analisi di qualità e performance del servizio fornito.

```

1 ContextMap Trasporti116117 {
2   type = SYSTEM_LANDSCAPE
3   state = TO_BE
4
5   contains CallCenterContext, PatientContext, TransportContext, VehicleContext
6
7   VehicleContext [SK] <-> [SK] TransportContext
8
9   PatientContext [SK] <-> [SK] TransportContext
10
11  CallCenterContext [D] <- [U, OHS, PL] TransportContext {
12    implementationTechnology = "RESTful HTTP"
13  }
14
15  CallCenterContext [D] <- [U, OHS, PL] PatientContext {
16    implementationTechnology = "RESTful HTTP"
17  }
18
19  CallCenterContext [D] <- [U, OHS, PL] VehicleContext {
20    implementationTechnology = "RESTful HTTP"
21  }
22}
23
24 BoundedContext CallCenterContext implements CallCenter{
25   type = SYSTEM
26   domainVisionStatement = "Handles calls. It also deals with the storage of new transports,
27     vehicles and patients."
27   knowledgeLevel = META
28}
29
30 BoundedContext PatientContext implements Patient{
31   type = FEATURE
32   domainVisionStatement = "Represents the patient to be transported"
33   knowledgeLevel = CONCRETE
34
35   Aggregate Patient {
36     ValueObject PatientResidence{
37       -PatientAddress address
38       -PatientCity city
39       -PatientDistrict district
40       -PatientPostalCode postalCode
41     }
42
43     ValueObject PatientPersonalData{
44       -PatientFiscalCode fiscalCode
45       String name
46       String surname
47       Date dateOfBirth
48       -PatientResidence residence
49     }
50
51     ValueObject Condition{
52       String system
53       Integer code
54     }

```

```
55
56 ValueObject PatientFiscalCode {
57     String fiscalCode
58 }
59
60 ValueObject PatientAddress {
61     String address
62 }
63
64 ValueObject PatientCity {
65     String city
66 }
67
68 ValueObject PatientDistrict {
69     String district
70 }
71
72 ValueObject PatientPostalCode {
73     String postalCode
74 }
75
76 Entity Patient {
77     aggregateRoot
78     -PatientPersonalData personalData
79     -Condition patientCondition
80     Repository PatientDTRepository
81 }
82 }
83}
84
85 BoundedContext TransportContext implements Transport{
86     type = FEATURE
87     domainVisionStatement = "Represents all stages of transport. The life cycle of the transport
88         begins with the booking of the same and ends with the arrival at the destination."
89     knowledgeLevel = CONCRETE
90 }
91 Aggregate Transport{
92     enum Phase {
93         aggregateLifecycle
94         Scheduled, InProgress, Completed, Cancelled
95     }
96     ValueObject TransportRoute{
97         -Location departure
98         -Location destination
99     }
100
101     ValueObject Location {
102         -TransportAddress address
103         -TransportCity city
104         -TransportDistrict district
105         -TransportPostalCode postalCode
106     }
107
108     ValueObject TransportAddress {
109         String address
```

```
110     }
111
112     ValueObject TransportCity {
113         String city
114     }
115
116     ValueObject TransportDistrict {
117         String district
118     }
119
120     ValueObject TransportPostalCode {
121         String postalCode
122     }
123
124     Entity Transport {
125         aggregateRoot
126         String idTransport
127         -TransportRoute route
128         -Phase phase
129         Date startDateTime
130         Date endDateTime
131         Repository TransportDTRepository
132     }
133 }
134}
135
136BoundedContext VehicleContext implements Transport{
137     type = FEATURE
138     domainVisionStatement = "Represents the vehicles and the operator who drives them"
139     knowledgeLevel = CONCRETE
140
141     Aggregate Ambulance {
142         ValueObject Coordinates {
143             Double longitude
144             Double latitude
145         }
146
147         enum AmbulanceState {
148             aggregateLifecycle
149             Busy, Free, UnderMaintenance, Disused
150         }
151
152         Entity Ambulance {
153             aggregateRoot
154             String idAmbulance
155             -Coordinates coordinates
156             -AmbulanceState ambulanceState
157             Repository AmbulanceDTRepository
158         }
159     }
160
161     Aggregate TransportOperator {
162         ValueObject TransportOperatorResidence{
163             -OperatorAddress address
164             -OperatorCity city
165             -OperatorDistrict district
```

```
166     -OperatorPostalCode postalCode
167   }
168
169   ValueObject OperatorAddress {
170     String address
171   }
172
173   ValueObject OperatorCity {
174     String city
175   }
176
177   ValueObject OperatorDistrict {
178     String district
179   }
180
181   ValueObject OperatorPostalCode {
182     String postalCode
183   }
184
185   ValueObject TransportOperatorPersonalData{
186     String name
187     String surname
188     Date dateOfBirth
189     -TransportOperatorResidence residence
190   }
191
192   ValueObject IdTransportOperator {
193     String idTransportOperator
194   }
195
196   Entity TransportOperator{
197     aggregateRoot
198     String idTransportOperator
199     -TransportOperatorPersonalData personalData
200     Repository TransportOperatorDTRepository
201   }
202
203 }
204}
205
206Domain Trasporti116117Domain{
207  Subdomain CallCenter{
208    type GENERIC_SUBDOMAIN
209  }
210
211  Subdomain Patient{
212    type SUPPORTING_DOMAIN
213  }
214
215  Subdomain Transport{
216    type CORE_DOMAIN
217  }
218}
```

Capitolo 5

Progettazione di una soluzione basata su Digital Twin

In questo capitolo si approfondiscono gli aspetti tecnici relativi alla progettazione, derivanti dall'analisi del dominio svolta nel capitolo 4. In particolare, in questo capitolo, si individueranno e progetteranno le tipologie di digital twin presenti all'interno del dominio 116-117. Successivamente ci si occuperà del problema dell'interoperabilità sollevato nel capitolo 3 tramite l'integrazione dello standard FHIR nel dominio.

5.1 Integrazione Digital Twin nel modello 116-117

Per prima cosa sono state individuate le entità all'interno del dominio per le quali si è ritenuta opportuna una controparte digitale:

- Paziente
- Trasporto
- Ambulanza
- Operatore ambulanza

Questi saranno i quattro tipi di digital twin presenti all'interno del modello. All'interno della progettazione Domain Driven Design si è ritenuto opportuno rappresentarli come repositories. Questi permettono di nascondere la

complessità della tecnologia utilizzata per la costruzione dei digital twin, inoltre permettono di mantenere una chiara separazione logica tra l'asset fisico e la controparte digitale definendo un netto confine tra il modello reale e il modello virtuale.

La comunicazione tra digital twin e asset fisico, grazie all'utilizzo dei repositories, può avvenire soltanto attraverso la root dell'aggregato evitando che il mondo virtuale si infiltri eccessivamente nel dominio indebolendo la sua integrità.

5.1.1 Digital Twin Prototype

Per rappresentare il DTP è scelto di utilizzare il linguaggio DTDL (Digital Twin Definition Language). Questo è un linguaggio open source sviluppato da Microsoft, al fine di costruire modelli capaci di descrivere il digital twin e le sue relazioni; è un linguaggio flessibile e robusto che consente di mappare nel mondo digitale le complesse relazioni che caratterizzano il corrispettivo fisico.

DTDL utilizza una variazione di JSON, ovvero JSON-LD. Il DTDL utilizza sei tipi di metamodelli per definire il comportamento del digital twin, inoltre permette di definire relazioni tra le entità e supporta l'ereditarietà. Il DTDL non memorizza i dati storici dei Digital Twin, ma tiene in memoria solo lo stato corrente; per mantenere i dati storici bisogna avvelersi di altre piattaforme, come ad esempio un database o Time Series Insights di Azure [22].

Di seguito sono riportati i quattro modelli DTDL dei digital twin e il relativo diagramma in figura 5.1.

Bounded Context Veicolo

Digital twin Ambulanza Il digital twin ambulanza è rappresentato dalla proprietà *state* che rappresenta lo stato che questa può assumere, la proprietà *coordinates* che rappresenta le coordinate fornite in real-time dal GPS a bordo del veicolo ed infine la relazione *driveBy* che rappresenta la relazione che intercorre tra ambulanza ed operatore, questa relazione ha come proprietà una data, ciò per tenere traccia dei vari operatori che giornalmente sono assegnati a ciascun veicolo.

```
1 {  
2   "@id": "dtmi:num116117:ambulance;1",
```

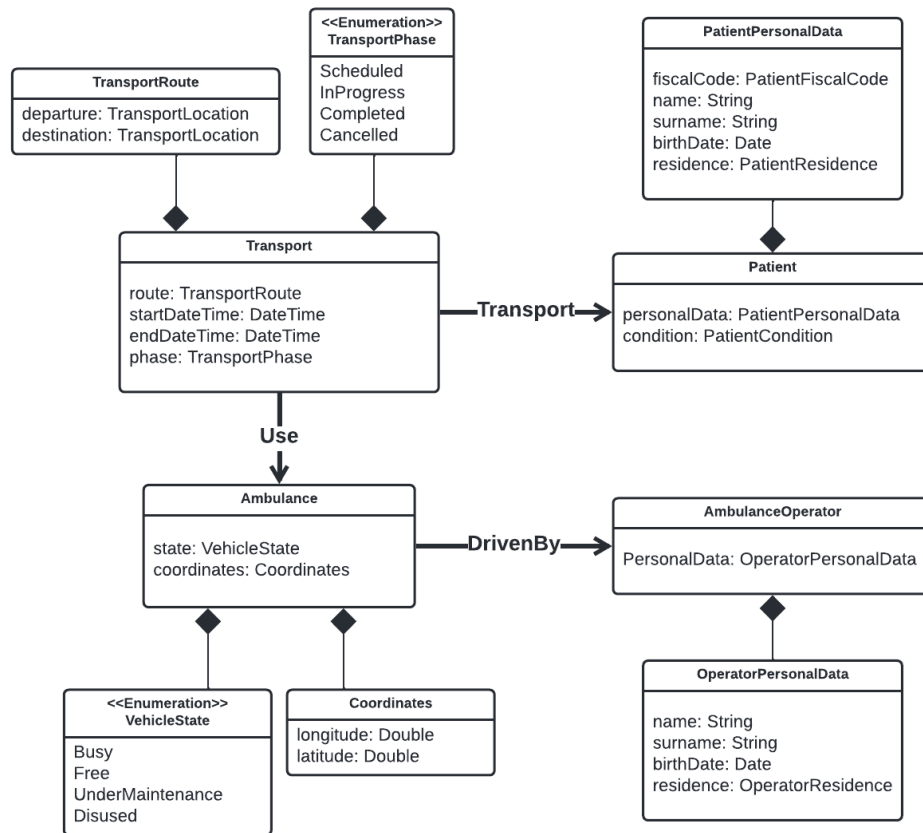



Figura 5.1: UML Diagram rappresentate il modello dei digital twin

```

3  "@type": "Interface",
4  "@context": [
5    "dtmi:dtdl:context;2"
6  ],
7  "displayName": "Ambulance",
8  "contents": [
9    {
10     "@type": "Property",
11     "name": "state",
12     "schema": "dtmi:num116117:veichleState;1"
13   },
14   {
15     "@type": "Property",
16     "name": "coordinates",
17     "schema": "dtmi:num116117:coordinates;1"
18   }
19 ]

```

```

19     {
20       "@type": "Relationship",
21       "name": "drivenBy",
22       "target": "dtmi:num116117:ambulanceOperator;1",
23       "properties": [
24         {
25           "@type": "Property",
26           "name": "workDate",
27           "schema": "date"
28         }
29       ]
30     }],
31   "schemas": [
32     {
33       "@id": "dtmi:num116117:veichleState;1",
34       "@type": "Enum",
35       "valueSchema": "string",
36       "enumValues": [
37         {
38           "name": "busy",
39           "enumValue": "busy"
40         },
41         {
42           "name": "free",
43           "enumValue": "free"
44         },
45         {
46           "name": "underMaintenance",
47           "enumValue": "underMaintenance"
48         },
49         {
50           "name": "disused",
51           "enumValue": "disused"
52         }
53       ]
54     },
55     {
56       "@id": "dtmi:num116117:coordinates;1",
57       "@type": "Object",
58       "fields": [
59         {
60           "name": "longitude",
61           "schema": "double"
62         },
63         {
64           "name": "latitude",
65           "schema": "double"
66         }
67       ]
68     }
69 ]

```

Digital twin Operatore ambulanza Il digital twin dell'operatore ambulanza è rappresentato solamente dalla proprietà *personalData* che contiene tutti i suoi dati personali.

```
1{
2  "@id": "dtmi:num116117:ambulanceOperator;1",
3  "@type": "Interface",
4  "@context": [
5    "dtmi:dtdl:context;2"
6  ],
7  "displayName": "AmbulanceOperator",
8  "contents": [
9    {
10     "@type": "Property",
11     "name": "personalData",
12     "schema": "dtmi:num116117:operatorPersonalData;1"
13   }
14 ],
15 "schemas": [
16   {
17     "@id": "dtmi:num116117:operatorPersonalData;1",
18     "@type": "Object",
19     "fields": [
20       {
21         "name": "name",
22         "schema": "string"
23       },
24       {
25         "name": "surname",
26         "schema": "string"
27       },
28       {
29         "name": "birthDate",
30         "schema": "date"
31       },
32       {
33         "name": "residence",
34         "schema": "dtmi:num116117:operatorResidence;1"
35       }
36     ]
37   },
38   {
39     "@id": "dtmi:num116117:operatorResidence;1",
40     "@type": "Object",
41     "fields": [
42       {
43         "name": "address",
44         "schema": "dtmi:num116117:operatorAddress;1"
45       },
46       {
47         "name": "city",
48         "schema": "dtmi:num116117:operatorCity;1"
49       },
50       {
51         "name": "district",
52         "schema": "dtmi:num116117:operatorDistrict;1"
53       },
54       {
55         "name": "postalCode",
```

```
56         "schema": "dtmi:num116117:operatorPostalCode;1"
57     }
58 ]
59 },
60 {
61     "@id": "dtmi:num116117:operatorAddress;1",
62     "@type": "Object",
63     "fields": [
64         {
65             "name": "address",
66             "schema": "string"
67         }
68     ]
69 },
70 {
71     "@id": "dtmi:num116117:operatorCity;1",
72     "@type": "Object",
73     "fields": [
74         {
75             "name": "city",
76             "schema": "string"
77         }
78     ]
79 },
80 {
81     "@id": "dtmi:num116117:operatorDistrict;1",
82     "@type": "Object",
83     "fields": [
84         {
85             "name": "district",
86             "schema": "string"
87         }
88     ]
89 },
90 {
91     "@id": "dtmi:num116117:operatorPostalCode;1",
92     "@type": "Object",
93     "fields": [
94         {
95             "name": "postalCode",
96             "schema": "integer"
97         }
98     ]
99 }
100 ]
101 }
```

Bounded Context Trasporto

Digital twin Trasporto Il digital twin trasporto contiene la proprietà *route* che descrive l'itinerario da compiere durante il processo di trasporto, le proprietà *startDateTime* e *endDateTime* che tengono traccia di data e

ora di inizio e fine del trasporto, la proprietà *phase* che descrive la fase in cui il processo di trasporto si trova, la relazione *use* che lega il trasporto all'ambulanza che lo eseguirà ed, infine, la relazione *transport* che specifica quale paziente usufruisce del servizio.

```

1{
2  "@id": "dtmi:num116117:transport;1",
3  "@type": "Interface",
4  "@context": [
5    "dtmi:dtdl:context;2"
6  ],
7  "displayName": "Transport",
8  "contents": [
9    {
10   "@type": "Property",
11   "name": "route",
12   "schema": "dtmi:num116117:transportRoute;1"
13  },
14  {
15   "@type": "Property",
16   "name": "startDateTime",
17   "schema": "dateTime"
18  },
19  {
20   "@type": "Property",
21   "name": "endDateTime",
22   "schema": "dateTime"
23  },
24  {
25   "@type": "Property",
26   "name": "phase",
27   "schema": "dtmi:num116117:transportPhase;1"
28  },
29  {
30   "@type": "Relationship",
31   "name": "use",
32   "target": "dtmi:num116117:ambulance;1"
33  },
34  {
35   "@type": "Relationship",
36   "name": "transport",
37   "target": "dtmi:num116117:patient;1"
38  }
39 ],
40 "schemas": [
41  {
42   "@id": "dtmi:num116117:transportPhase;1",
43   "@type": "Enum",
44   "valueSchema": "string",
45   "enumValues": [
46     {
47       "name": "scheduled",
48       "displayName": "Scheduled",
49       "enumValue": "scheduled"
50     }
51   ]
52  }
53 ]

```

```
51     {
52         "name": "InProgress",
53         "displayName": "InProgress",
54         "enumValue": "InProgress"
55     },
56     {
57         "name": "completed",
58         "displayName": "Completed",
59         "enumValue": "completed"
60     },
61     {
62         "name": "cancelled",
63         "displayName": "Cancelled",
64         "enumValue": "cancelled"
65     }
66 ]
67 },
68 {
69     "@id": "dtmi:num116117:transportRoute;1",
70     "@type": "Object",
71     "fields": [
72         {
73             "name": "departure",
74             "schema": "dtmi:num116117:transportLocation;1"
75         },
76         {
77             "name": "destination",
78             "schema": "dtmi:num116117:transportLocation;1"
79         }
80     ]
81 },
82 {
83     "@id": "dtmi:num116117:transportLocation;1",
84     "@type": "Object",
85     "fields": [
86         {
87             "name": "address",
88             "schema": "dtmi:num116117:transportAddress;1"
89         },
90         {
91             "name": "city",
92             "schema": "dtmi:num116117:transportCity;1"
93         },
94         {
95             "name": "district",
96             "schema": "dtmi:num116117:transportDistrict;1"
97         },
98         {
99             "name": "postalCode",
100            "schema": "dtmi:num116117:transportPostalCode;1"
101        }
102    ]
103 },
104 {
105     "@id": "dtmi:num116117:transportAddress;1",
106     "@type": "Object",
```

```

107     "fields": [
108       {
109         "name": "address",
110         "schema": "string"
111       }
112     ],
113   },
114   {
115     "@id": "dtmi:num116117:transportCity;1",
116     "@type": "Object",
117     "fields": [
118       {
119         "name": "city",
120         "schema": "string"
121       }
122     ]
123   },
124   {
125     "@id": "dtmi:num116117:transportPostalCode;1",
126     "@type": "Object",
127     "fields": [
128       {
129         "name": "postalCode",
130         "schema": "integer"
131       }
132     ]
133   },
134   {
135     "@id": "dtmi:num116117:transportDistrict;1",
136     "@type": "Object",
137     "fields": [
138       {
139         "name": "district",
140         "schema": "string"
141       }
142     ]
143   }
144 ]
145 }

```

Bounded Context Paziente

Digital twin Paziente Il digital twin paziente è rappresentato dalla proprietà *personalData* che contiene tutti i dati personali relativi ad esso e dalla proprietà *condition* che permette di tenere traccia delle condizioni fisiche del paziente.

```

1{
2  "@id": "dtmi:num116117:patient;1",
3  "@type": "Interface",
4  "@context": [
5    "dtmi:dtdl:context;2"

```

```
6 ],
7 "displayName": "Patient",
8 "contents": [
9   {
10    "@type": "Property",
11    "name": "condition",
12    "schema": "dtmi:num116117:patientCondition;1"
13  },
14  {
15    "@type": "Property",
16    "name": "personalData",
17    "schema": "dtmi:num116117:patientPersonalData;1"
18  }
19 ],
20 "schemas": [
21   {
22    "@id": "dtmi:num116117:patientPersonalData;1",
23    "@type": "Object",
24    "fields": [
25     {
26      "name": "fiscalCode",
27      "schema": "dtmi:num116117:patientFiscalCode;1"
28     },
29     {
30      "name": "name",
31      "schema": "string"
32     },
33     {
34      "name": "surname",
35      "schema": "string"
36     },
37     {
38      "name": "birthDate",
39      "schema": "date"
40     },
41     {
42      "name": "residence",
43      "schema": "dtmi:num116117:patientResidence;1"
44     }
45    ]
46   },
47   {
48    "@id": "dtmi:num116117:patientCondition;1",
49    "@type": "Object",
50    "fields": [
51     {
52      "name": "system",
53      "schema": "string"
54     },
55     {
56      "name": "code",
57      "schema": "integer"
58     }
59    ]
60   },
61   {
```



```
62     "@id": "dtmi:num116117:patientResidence;1",
63     "@type": "Object",
64     "fields": [
65       {
66         "name": "address",
67         "schema": "dtmi:num116117:patientAddress;1"
68       },
69       {
70         "name": "city",
71         "schema": "dtmi:num116117:patientCity;1"
72       },
73       {
74         "name": "district",
75         "schema": "dtmi:num116117:patientDistrict;1"
76       },
77       {
78         "name": "postalCode",
79         "schema": "dtmi:num116117:patientPostalCode;1"
80       }
81     ]
82   },
83   {
84     "@id": "dtmi:num116117:patientFiscalCode;1",
85     "@type": "Object",
86     "fields": [
87       {
88         "name": "fiscalCode",
89         "schema": "string"
90       }
91     ]
92   },
93   {
94     "@id": "dtmi:num116117:patientAddress;1",
95     "@type": "Object",
96     "fields": [
97       {
98         "name": "address",
99         "schema": "string"
100      }
101    ]
102  },
103  {
104    "@id": "dtmi:num116117:patientCity;1",
105    "@type": "Object",
106    "fields": [
107      {
108        "name": "city",
109        "schema": "string"
110      }
111    ]
112  },
113  {
114    "@id": "dtmi:num116117:patientDistrict;1",
115    "@type": "Object",
116    "fields": [
117      {
```

```
118         "name": "district",
119         "schema": "string"
120     }
121 ]
122 },
123 {
124     "@id": "dtmi:num116117:patientPostalCode;1",
125     "@type": "Object",
126     "fields": [
127     {
128         "name": "postalCode",
129         "schema": "integer"
130     }
131 ]
132 }
133 ]
134 }
```

5.1.2 Synchronization (twinning) tra digital twin e physical twin

Digital Twin Ambulanza

I dati relativi al gps, ovvero longitudine e latitudine, saranno aggiornati in real-time automaticamente, in quanto il sensore gps comunicherà direttamente i suoi aggiornamenti al digital twin. I restanti dati all'interno del digital twin saranno aggiornati manualmente dagli operatori.

Digital Twin Operatore Ambulanza

I dati relativi a questo digital twin saranno aggiornati, manualmente, dall'amministrazione del sistema Trasporti 116-117.

Digital Twin Trasporto

La parte dei dati relativi alla prenotazione, come ad esempio orario, partenza, destinazione e paziente saranno aggiornati, manualmente, dall'amministrazione del sistema Trasporti 116-117. Altri dati come lo stato del trasporto e la fine del trasporto saranno aggiornati, in corso d'opera, dall'operatore ambulanza. Per la fase di fine trasporto è stata valutata la possibilità di aggiornarla, automaticamente, nel momento in cui l'ambulanza raggiunge il punto di destinazione. Si è poi scelta l'opzione manuale, dove l'operatore segna l'effettiva conclusione del trasporto; questo perchè il

trasporto viene considerato effettivamente concluso con l'arrivo del paziente a destinazione, che potrebbe essere anche un reparto interno all'ospedale o l'interno di un'abitazione.

Digital Twin Paziente

I dati relativi a questo digital twin saranno aggiornati, manualmente, dall'amministrazione del sistema Trasporti 116-117.

5.1.3 Piattaforma per lo sviluppo dei digital twin

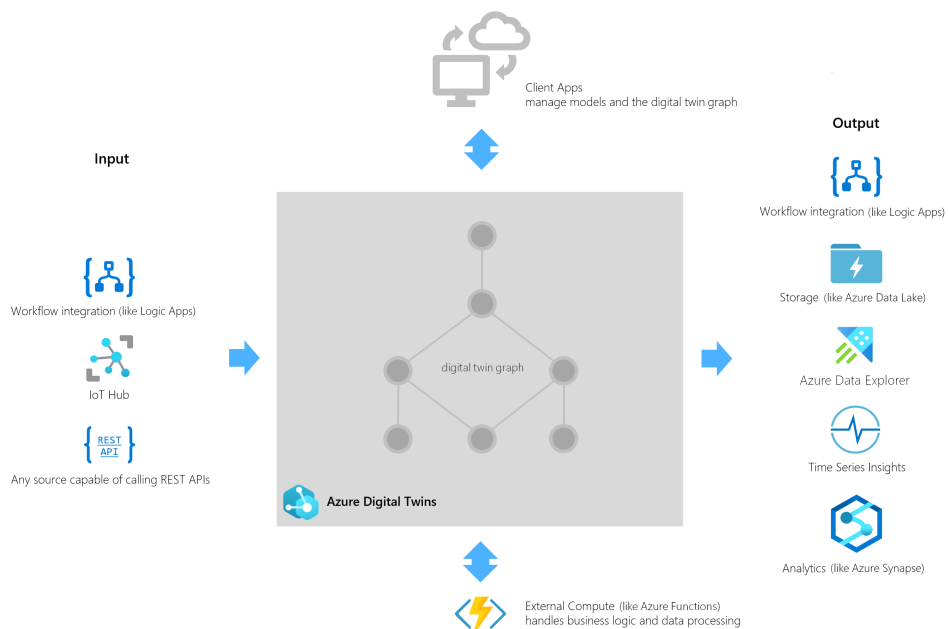


Figura 5.2: Esempio di configurazione di Azure Digital Twin e altri servizi messi a disposizione da Azure (immagine proveniente dalla documentazione Azure)

Per lo sviluppo e la gestione dei digital twin si è scelto di utilizzare **Azure Digital Twin (ADT)**. Azure Digital Twin è un PaaS (Platform as a service) che consente di modellare qualsiasi tipo di ambiente e implementare digital twin in modo scalabile e sicuro. ADT fa parte dei servizi offerti

da Azure IoT, una piattaforma che offre una moltitudine di servizi che possono essere combinati in diverse configurazioni, in base ai requisiti del sistema che si vuole realizzare. Per esempio, per avere un aggiornamento real-time dell'ambiente di esecuzione dei digital twin e il mondo fisico, è possibile utilizzare il servizio IoT Hub, messo a disposizione da Azure IoT, per connettere il gemello direttamente al dispositivo IoT fisico. È anche possibile gestire i digital twin da altre origini di dati, utilizzando le REST API. Azure Digital Twin, infatti, mette a disposizione un ampio set di API, che si dividono in due categorie:

- **Control Plane API:** utilizzate per gestire le risorse in una data sottoscrizione Azure.
- **Data Plane API:** utilizzate per gestire gli elementi all'interno di una digital twin instance. Includono operazioni come la creazione di route, il caricamento di modelli, la creazione di relazioni e la gestione dei gemelli.

I dati presenti all'interno dei digital twin possono essere instradati ad altri servizi di Azure per l'analisi o l'archiviazione, utilizzando elementi come Event Hubs, Event Grid o Service Bus per guidare il flusso di dati.

Azure Digital Twin può, quindi, essere utilizzato in combinazione con molti altri servizi messi a disposizione di Azure, come mostrato nell'esempio in figura 5.2.

5.2 Interoperabilità

Un requisito fondamentale del sistema Trasporti 116-117 è l'interoperabilità. Essa può essere definita come “la capacità di due o più sistemi di scambiare ed utilizzare informazioni”. Ci sono diversi livelli di interoperabilità [13]:

- **Interoperabilità tecnica:** Capacità di scambio di dati tra i sistemi, richiede canali di comunicazione e protocolli per la trasmissione dei dati. Con le reti e i protocolli odierni raggiungere questo livello è relativamente semplice.
- **Interoperabilità sintattica:** Specifica il formato e la struttura dei dati. Per quanto riguarda i dati sanitari questa interoperabilità è

supportata da organizzazioni come *HL7(Health Level Seven International)* o *IHE (Integrating the Healthcare Enterprise)*, che specificano gli standard IT sanitari ed il loro utilizzo. Uno standard emergente è *FHIR (Fast Healthcare Interoperability Resources)*, che specifica una serie di concetti sanitari comuni a cui è possibile accedere tramite le più comuni tecnologie web. Un'altra iniziativa è *openEHR* che consente a medici e esperti IT di definire contenuti clinici.

- **Interoperabilità semantica:** *FHIR* e *openEHR* definiscono anche la semantica di base dei dati sanitari, ma l'interoperabilità semantica è in realtà il dominio delle terminologie, nomenclature e ontologie mediche. Assicura che il significato dei concetti medici possa essere condiviso tra i sistemi, fornendo una lingua digitale comune. La terminologia *SNOMED CT*, con oltre 350'000 concetti medici, è adatta come linguaggio generico per far aumentare l'interoperabilità semantica in medicina. Questa può essere poi integrata con terminologie più specifiche del dominio, alcuni esempi sono *LOINC(Logical Observation Identifiers Names and Codes)*, *IDMP(Identification of Medicinal Products)*, *HGNC(HUGO Gene Nomenclature Committee)* e *HPO(Human Phenotype Ontology)*.
- **Interoperabilità organizzativa:** a questo livello l'interoperabilità coinvolge organizzazioni e legislazioni. Poiché gli stakeholder coinvolti nell'assistenza sanitaria hanno interessi diversi, che non sempre mirano all'interoperabilità, bisognerebbe che le organizzazioni politiche fornissero incentivi per aumentarla e, se necessario, imporla con regolamenti legali.

Relativamente al caso di studio Trasporti 116-117 l'interoperabilità tecnica si otterrà principalmente tramite lo sviluppo di un'architettura *REST (Representational state transfer)* in quanto questa è altamente scalabile e indipendente dal linguaggio o dalla piattaforma utilizzata. Per quanto riguarda l'interoperabilità sintattica e semantica si utilizzerà principalmente *FHIR* e *SNOMED CT*, di seguito riportati nel dettaglio.

5.2.1 FHIR

Data la sempre più crescente digitalizzazione dell'ambiente sanitario si è visto necessario trovare un modo per strutturare e standardizzare i dati al suo

interno, nasce quindi FHIR [1], uno standard per lo scambio di informazioni sanitarie. È stato sviluppato da HL7, un'organizzazione che ha come scopo lo sviluppo di standard per scambio, integrazione, condivisione e recupero di informazioni sanitarie elettroniche; infatti il numero sette all'interno dell'acronimo fa riferimento al settimo livello dello stack ISO/OSI, ovvero il livello di applicazione.

FHIR si basa su REST, alla base di questo ci sono dunque le risorse. Tutto ciò che si può scambiare è definito come risorsa, queste infatti definiscono il contenuto e la struttura delle informazioni. Lo scopo di FHIR è costruire un set base di risorse che soddisfino la maggior parte dei casi d'uso comuni, inoltre esso fornisce un'estensione integrata per coprire i contenuti non rappresentati nelle sue risorse.

Le risorse sono utilizzate per lo scambio e l'archiviazione di dati sanitari, sia clinici che amministrativi. La specifica suddivide le risorse in vari moduli, inoltre definisce anche vari modi per lo scambio delle risorse. Una risorsa ha le seguenti caratteristiche:

- ha un'identità con la quale può essere indirizzata (URL),
- si identifica come uno dei tipi di risorsa definiti dalla specifica,
- contiene un insieme di dati strutturati come descritto dalla definizione del tipo di risorsa,
- ha una versione che cambia ogni qualvolta cambia il contenuto della stessa.

FHIR si allinea con i seguenti principi architetturali:

- **Riutilizzo e Componibilità:** le risorse sono progettate per soddisfare solamente i requisiti generali o comuni a molti casi d'uso, questo per evitare l'aumento esponenziale delle risorse che potrebbero diventare ridondanti. Esistono estensioni per adattare risorse comuni a requisiti specifici dei casi d'uso. Inoltre le risorse FHIR sono altamente componibili in quanto è possibile far riferimento ad altre risorse, questo promuove ulteriormente il riutilizzo e consente di costruire strutture complesse da più risorse atomiche.
- **Scalabilità:** con l'adattamento delle API FHIR allo stile architetturale REST si garantisce che tutte le transazioni siano stateless, questo riduce l'utilizzo della memoria e abilita la scalabilità orizzontale.

- **Prestazioni:** le risorse FHIR sono leggere ed adatte allo scambio tramite la rete. Sono disponibili vari formati ottimizzati che permettono di migliorare le prestazioni anche con transazioni complesse.
- **Usabilità:** le risorse FHIR possono essere comprese sia da esperti del settore sia da persone esterne. Poiché la sintassi XML o JSON non può essere compresa da tutti è possibile visualizzare le risorse, tramite browser, in un linguaggio comune a tutti.
- **Fedeltà dei dati:** FHIR è fortemente tipizzato e ha meccanismi integrati per il collegamento dei dati e la convalida della terminologia tecnica. Inoltre, è possibile convalidare la sintattica dei documenti XML e JSON. Questo assicura un'elevata affidabilità dei dati, promuovendo l'utilizzo di FHIR per ottenere interoperabilità semantica.
- **Implementabilità:** FHIR, essendo uno standard, punta ad un'elevata adozione da parte di diverse comunità di sviluppatori, è quindi facilmente comprensibile e implementabile poiché utilizza tecnologie comuni di markup e scambio di dati.

5.2.2 SNOMED CT

SNOMED CT [3] è una collezione digitale organizzata di termini medici che fornisce codici, termini, sinonimi e definizioni in ambito sanità digitale. Al momento è la più completa e multilingue al mondo. Consente una rappresentazione coerente dei dati sanitari in formato elettronico, permettendo ai medici e agli operatori sanitari di registrare i dati con alta precisione. Questo strumento permette la condivisione di dati sia per la continuità assistenziale del paziente che nell'ambito di ricerca. SNOMED CT è in continua evoluzione, al 31 luglio 2021 includeva 350'936 concetti attivi. SNOMED CT in generale si limita a fornire terminologie, per questo caso di studio sarà utilizzato per descrivere ed adattare al dominio le risorse FHIR.

5.2.3 Mapping tra entità del dominio e risorse FHIR

Come introdotto nel capitolo 4, come linguaggio standard per la comunicazione tra i bounded context e per agevolare la comunicazione con nuovi contesti, dominio o sistemi futuri, si utilizza lo standard FHIR. Di seguito è

riportato, dunque, il mapping tra le entità principali individuate nell'analisi del dominio e le risorse FHIR.

Il mapping è relativo solamente al core-domain e al supporting-domain, ovvero i bounded context veicolo, trasporto e paziente. Il bounded context centralino non rientra nel mapping in quanto è classificato come generic-domain.

Bounded Context Veicolo

- **Ambulanza e GPS:** da convenzione FHIR considera l'ambulanza come una risorsa di tipo Location. Siccome nel dominio ambulanza e GPS sono due entità ben distinte si è scelto di utilizzare Location per la rappresentazione del GPS e Device per quella dell'ambulanza. Per il mapping di queste due entità si utilizzano, quindi, le seguenti due risorse:

Device: risorsa che tiene traccia di un dispositivo e della sua posizione. Per rappresentare un'autoambulanza si utilizza il codice SNOMED 465341007. Per rappresentare gli stati del digital twin rappresentante l'ambulanza si utilizzano gli stati messi a disposizione da FHIR all'interno della risorsa 'Device', nello specifico sono mappati come riportato di seguito:

- busy: active
- free: inactive
- under maintenance: entered in error
- disused: unknown

Se in un futuro bisognerà aggiungere altri tipi di veicoli medici SNOMED new fornisce la definizione.

Location: Dettagli e informazioni sulla posizione per un luogo fisico. Per rappresentare il GPS si utilizza il codice SNOMED 897293009. Questa risorsa sarà contenuta direttamente all'interno della risorsa principale rappresentante l'ambulanza.

```
1 {
2   "resourceType": "Device",
3   "contained":
4   [
5     {
```



```

6     "resourceType": "Location",
7     "id": "1",
8     "physicalType": {
9       "coding": [
10        {
11          "system": "http://snomed.info/sct", "code": "897293009"
12        }
13      ],
14     "position":
15     {
16       "longitude": 1.11,
17       "latitude": 1.11
18     }
19   }
20 ],
21 "identifier":
22 [
23   {
24     "value": "Ambulance1111"
25   }
26 ],
27 "status": "inactive",
28 "type":
29 {
30   "coding":
31   [
32     {
33       "system": "http://snomed.info/sct",
34       "code": "465341007"
35     }
36   ]
37 }
38 }

```

- **Operatore trasporto:** per rappresentare questa entità si utilizzano due risorse, di seguito riportate.

Practitioner: questa risorsa rappresenta un professionista, questi sono impegnati nel processo sanitario e nei servizi correlati all'assistenza sanitaria come parte delle loro responsabilità formali. La risorsa non deve essere utilizzata per le persone coinvolte senza una responsabilità formale come le persone che si prendono cura di amici, parenti o vicini.

PractitionerRole: specifica la posizione e i tipi di servizi che i Practitioner sono in grado di fornire a un'organizzazione. Nel caso dell'operatore addetto ai trasporti si è scelto il ruolo ambulanceman con codice SNOMED 159738005. Questa risorsa sarà contenuta direttamente all'interno della risorsa principale rappresentante l'operatore.

```
1{
2  "resourceType": "Practitioner",
3  "contained": [
4    {
5      "resourceType": "PractitionerRole",
6      "id": "1",
7      "specialty": [
8        {
9          "coding": [
10           {
11             "system": "http://snomed.info/sct",
12             "code": "159738005"
13           }
14         ]
15       }
16     ]
17   }
18 ],
19 "identifier": [
20   {
21     "value": "MarioRossi3"
22   }
23 ],
24 "name": [
25   {
26     "family": "Rossi",
27     "given": [
28       "Mario"
29     ]
30   }
31 ],
32 "address": [
33   {
34     "line": [
35       "Corso Cavour, 1"
36     ],
37     "city": "Cesena",
38     "district": "FC",
39     "postalCode": "47521"
40   }
41 ],
42 "birthDate": "1990-07-05"
43 }
```

Bounded Context Trasporto

- **Trasporto prenotato:** per la rappresentazione di questo stato del trasporto si utilizza la risorsa *Appointment*.

Appointment: risorsa che fa riferimento a una prenotazione di un evento sanitario tra paziente, medico, persona correlata e/o dispositivo per una data/ora specifica. Questa definizione prende i concetti

di appuntamenti in un contesto clinico e li estende anche per essere rilevanti nello spazio sanitario e per facilitare l'esposizione ad altri tipi di appuntamenti utilizzati al di fuori dell'assistenza sanitaria. Per la rappresentazione dell'itinerario del trasporto si utilizzano due risorse **Location**.

In futuro, se sarà necessario entrare nel dettaglio del flusso della prenotazione, questa risorsa offre la possibilità di:

- pianificare i dettagli della richiesta, con la risorsa *Schedule*;
- verificare la disponibilità;
- richiedere un appuntamento, tramite *Appointment.status = "proposed"*;
- rispondere alla richiesta di appuntamento tramite *AppointmentResponse*;
- verificare lo stato della richiesta, se è andata a buon fine si setta *Appointment.status = "booked"*.

```

1{
2  "resourceType": "Appointment",
3  "contained": [
4    {
5      "resourceType": "Location",
6      "id": "1",
7      "name": "departure",
8      "address": {
9        "line": [
10         "IV Settembre"
11        ],
12        "city": "Cesena",
13        "district": "FC",
14        "postalCode": "47521"
15      }
16    },
17    {
18      "resourceType": "Location",
19      "id": "2",
20      "name": "destination",
21      "address": {
22        "line": [
23         "Via Vicoli"
24        ],
25        "city": "Ravenna",
26        "district": "RA",
27        "postalCode": "48124"
28      }
29    }

```

```

30 ],
31 "identifier": [
32   {
33     "value": "Transport1111-appointment"
34   }
35 ],
36 "serviceCategory": [
37   {
38     "coding": [
39       {
40         "system": "http://terminology.hl7.org/CodeSystem/service-
41           category",
42         "code": "33",
43         "display": "Transport"
44       }
45     ]
46   }
47 ],
48 "serviceType": [
49   {
50     "coding": [
51       {
52         "system": "http://terminology.hl7.org/CodeSystem/service-type",
53         "code": "230",
54         "display": "Patient Transport"
55       }
56     ]
57   }
58 ],
59 "start": "2022-02-10T15:00:00.000+01:00",
60 "participant": [
61   {
62     "actor": {
63       "reference": "Patient/BNCFNC67H01C573J"
64     },
65     "status": "accepted"
66   },
67   {
68     "actor": {
69       "reference": "Device/Ambulance1111"
70     },
71     "status": "accepted"
72   }
73 ]

```

- **Trasporto in corso:** Quando inizia il trasporto vero e proprio si passa allo stato “Trasporto in corso” si crea la risorsa *Encounter*, questa conterrà la risorsa *Procedure* che descrive meglio il processo di trasporto.

Per rappresentare l’entità in questo stato si utilizzano quindi le seguenti due risorse:

Encounter: Comprende incontri ambulatoriali, di emergenza, domiciliari, ospedalieri e virtuali. Un incontro comprende il ciclo di vita dal pre-ricovero, l'incontro effettivo (per gli incontri senza ricovero) e il ricovero, la degenza e la dimissione (per i ricoveri). Durante l'incontro il paziente può spostarsi da un professionista all'altro e da un luogo all'altro.

Procedure: Questa risorsa viene utilizzata per registrare i dettagli delle procedure eseguite su o per un paziente. Una procedura è un'attività che viene eseguita su, con o per un paziente come parte della fornitura di cure. Gli esempi includono procedure chirurgiche, procedure diagnostiche, procedure endoscopiche, biopsie, consulenze, fisioterapia, servizi di supporto personale, servizi di assistenza diurna per adulti, trasporto non di emergenza, ecc. Le procedure possono essere eseguite da un operatore sanitario, un fornitore di servizi, un amico o un parente o in alcuni casi dal paziente stesso. La risorsa è descritta dal codice SNOMED 715537001 che rappresenta la procedura di trasporto con ambulanza.

```

1{
2  "resourceType": "Encounter",
3  "contained": [
4    {
5      "resourceType": "Procedure",
6      "id": "1",
7      "identifier": [
8        {
9          "value": "Transport1111-procedure"
10       }
11     ],
12     "status": "in-progress",
13     "code": {
14       "coding": [
15         {
16           "system": "http://snomed.info/sct",
17           "code": "715537001",
18           "display": "Transportation by ambulance (procedure)"
19         }
20       ]
21     },
22     "subject": {
23       "reference": "Patient/BNCFNC67H01C573J"
24     },
25     "encounter": {
26       "reference": "Transport1111-encounter"
27     },
28     "performer": [
29       {
30         "actor": {

```

```
31         "reference": "Practitioner/MarioRossi3"
32     }
33 },
34 {
35     "actor": {
36         "reference": "Device/Ambulance1111"
37     }
38 }
39 ]
40 }
41 ],
42 "identifier": [
43     {
44         "value": "Transport1111-encounter"
45     }
46 ],
47 "status": "in-progress",
48 "class": {
49     "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
50     "code": "AMB"
51 },
52 "serviceType": {
53     "coding": [
54         {
55             "system": "http://terminology.hl7.org/CodeSystem/service-type",
56             "code": "230",
57             "display": "Patient Transport"
58         }
59     ]
60 },
61 "subject": {
62     "reference": "Patient/BNCFNC67H01C573J"
63 },
64 "participant": [
65     {
66         "type": [
67             {
68                 "coding": [
69                     {
70                         "system": "http://terminology.hl7.org/CodeSystem/v3-
71                             ParticipationType",
72                         "code": "ESC"
73                     }
74                 ]
75             }
76         ],
77         "individual": {
78             "reference": "Practitioner/MarioRossi3"
79         }
80     }
81     "type": [
82         {
83             "coding": [
84                 {
85                     "system": "http://terminology.hl7.org/CodeSystem/v3-
```

```

86         ParticipationType",
87         "code": "DIR"
88     }
89 ]
90 ],
91 "individual": {
92     "reference": "Device/Ambulance1111"
93 }
94 }
95 ],
96 "appointment": [
97     {
98         "reference": "Transport1111-appointment"
99     }
100 ],
101 "period": {
102     "start": "2022-02-10T15:00:00+01:00"
103 }
104 }

```

- **Trasporto completato:** Se il processo di trasporto è andato a buon fine *Encounter* e *Procedure* entreranno in uno stato “*completed*”.

Bounded Context Paziente

- **Paziente:** per la modellazione di questa entità in FHIR si utilizza la risorsa *patient*, questa contiene già le informazioni anagrafiche della persona. Per rappresentare lo stato di salute e l’autonomia si è scelto di utilizzare la risorsa *condition*. Questa è utilizzata per registrare informazioni dettagliate su una condizione, problema, diagnosi o altro evento, situazione, problema o concetto clinico.

Per collegare le condizioni al paziente si utilizza il *contained* direttamente all’interno della risorsa *patient* in quanto la risorsa *condition* non ha senso di esistere senza il relativo paziente.

```

1{
2  "resourceType": "Patient",
3  "contained": [
4    {
5      "resourceType": "Condition",
6      "id": "1",
7      "code": {
8        "coding": [
9          {
10           "system": "http://snomed.info/sct",
11           "code": "60245001"
12         }
13       ]
14     }
15   ]
16 }

```

```
13     ]
14   }
15 }
16 ],
17 "identifier": [
18   {
19     "value": "BNCFNC67H01C573J"
20   }
21 ],
22 "name": [
23   {
24     "use": "official",
25     "family": "Bianchi",
26     "given": [
27       "Francesco"
28     ]
29   }
30 ],
31 "birthDate": "1967-06-01",
32 "address": [
33   {
34     "line": [
35       "Via Ferrari"
36     ],
37     "city": "Forli",
38     "district": "FC",
39     "postalCode": "47722"
40   }
41 ]
42 }
```

5.3 Sicurezza e privacy

Come anticipato nel capitolo 3.2 l'argomento privacy nell'ambito sanitario è un tema complesso.

Per rispettare le normative, nel caso del progetto Trasporti 116-117, i dati sanitari che si possono ricavare tramite i digital twin dovrebbero essere gestiti differenziando i ruoli degli utenti che li richiedono. Implementare questo livello di sicurezza direttamente in Azure non è possibile, poiché gli accessi sono gestiti a livello di istanza che quindi contiene i dati relativi a tutti i digital twin presenti, senza possibilità di differenziazione delle singole tipologie di digital twin.

Per la messa in opera del sistema serve quindi progettare un ulteriore livello di sicurezza che permetta il rispetto delle normative per il trattamento dei dati sanitari. L'argomento non verrà approfondito ulteriormente in quanto non è l'obiettivo primario di questa tesi. Si vuole però chiarire che

per l'effettiva utilizzazione del sistema servirà garantire il rispetto rigoroso di queste normative.

5.4 Architettura finale proposta

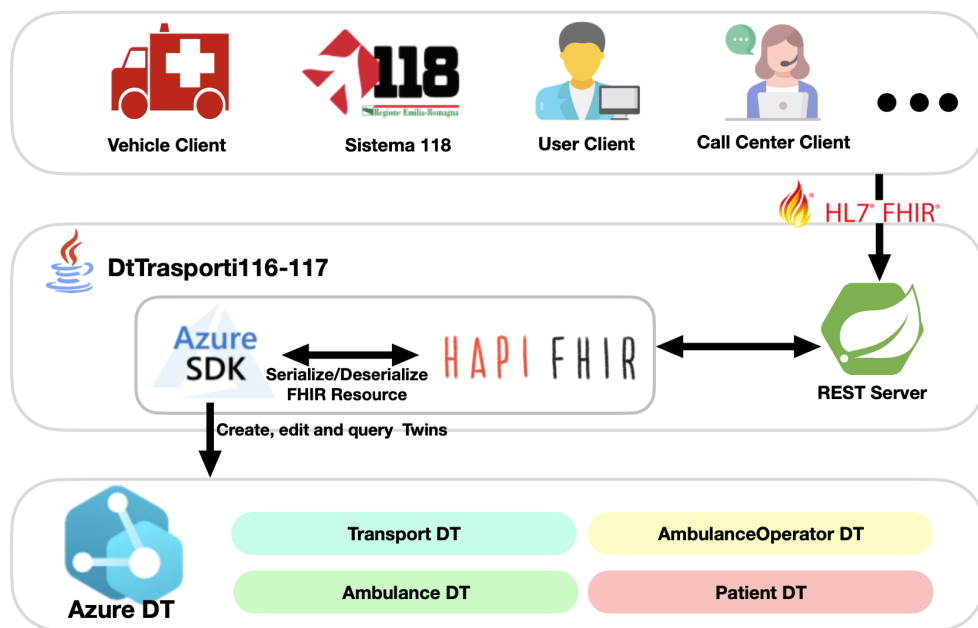


Figura 5.3: Schema architetturale

In figura 5.3 è rappresentato lo schema architetturale ottenuto al termine della fase di progettazione del sistema. L'architettura è suddivisa in tre macro livelli.

Il primo livello fa riferimento all'infrastruttura costruita all'interno di Azure, che ospita quindi l'ecosistema di digital twin.

Nel secondo livello si trova il REST Server che permette ai clienti di interagire con i digital twin, mantenendo una separazione tra le due parti, questo REST Server va ad implementare la relazione di Open Host Service discussa in fase di analisi del sistema. All'interno di questo blocco si utilizzerà l'Azure SDK per creare, modificare e interrogare i twins, inoltre ci si avvale di HAPI FHIR per la serializzazione/deserializzazione delle risorse

FHIR, FHIR infatti rappresenta il published language indicato nella fase di analisi ed approfondito all'interno di questo capitolo.

Il terzo livello fa riferimento a tutti gli ipotetici client che possono servirsi delle REST API sviluppate.

Nei capitoli seguenti si approfondiranno e implementeranno i tre livelli riportati ora in sintesi.

Capitolo 6

Implementazione

In questo capitolo si illustra l'implementazione del caso di studio Trasporti 116-117. Nello specifico nel capitolo si spiegano i servizi utilizzati messi a disposizione dalla piattaforma Azure, successivamente si spiega nel dettaglio la struttura delle API che permettono la comunicazione tra i digital twin rappresentanti il sistema ed i client che vogliono avvalersi delle loro funzionalità.

6.1 Servizi Azure utilizzati

Come detto nel capitolo 5.1.3, per ospitare i digital twin si è scelto di utilizzare la piattaforma Azure. Azure permette di implementare i digital twin in modo sicuro e scalabile e di connettere facilmente risorse come device IoT. Mette anche a disposizione servizi di analisi e AI per analizzare il passato e prevedere il futuro.

In figura 6.1 sono riportati i principali servizi di Azure utilizzati per il dominio Trasporti 116-117. Di seguito si approfondiranno gli aspetti principali.

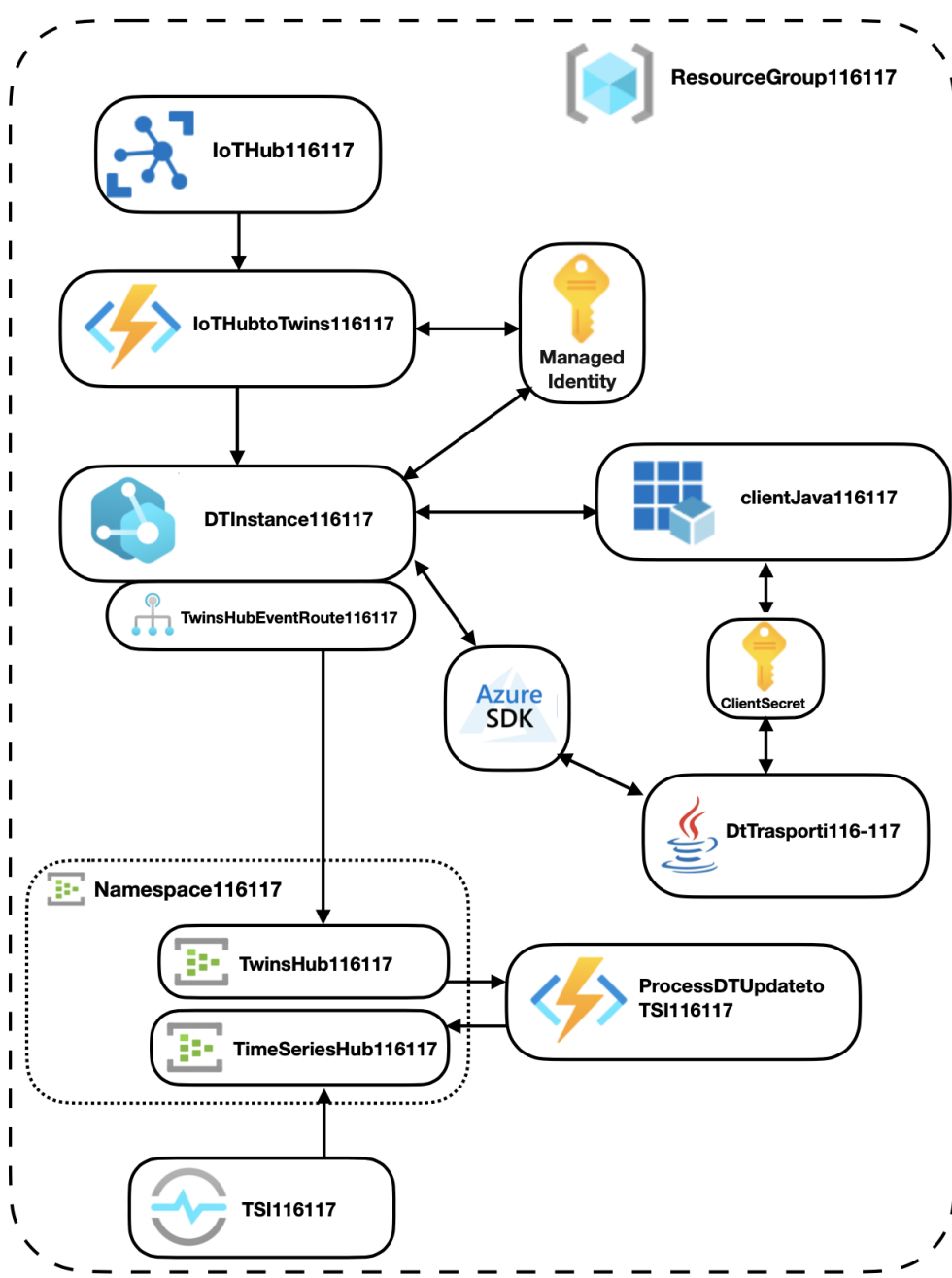


Figura 6.1: Schema dei servizi utilizzati in Azure

6.1.1 Creazione e gestione dei Digital Twin

Per potere ospitare i digital twin in Azure è necessario creare una **digital twin instance**, in tal caso denominata DTInstance116117. Durante la creazione della risorsa è necessario fornire un **resource group** di appartenenza, inoltre serve concedere le giuste autorizzazioni tramite **Azure Identity Management (IAM)**, queste serviranno per permettere di poter accedere e modificare i digital twin all'interno dell'istanza.

Nel caso di Trasporti 116-117 serve accedere ai digital twin anche tramite l'Azure SDK. Per fare questo si è creata un'applicazione all'interno di **Azure Active Directory (ADT)**, connessa tramite clientSecret all'API **DTTrasporti116-117**. Per permettere a questa applicazione di accedere ai gemelli digitali sono state settate le autorizzazioni necessarie nella DTInstance116117 tramite IAM, ovvero è stato assegnato il ruolo **Azure Digital Twins Data Owner** all'applicazione.

Azure mette a disposizione l'**Azure Digital Twin Explorer**, uno strumento che consente di visualizzare e interagire con i digital twin all'interno dell'istanza.

6.1.2 Connessione tra Digital Twin e device IoT

Azure offre la possibilità di connettere direttamente dispositivi IoT a digital twin, in questo modo ad ogni aggiornamento di stato del dispositivo verranno inviate telemetrie al digital twin, abilitando un aggiornamento real-time. Nel caso Trasporti 116-117 è stato fatto ciò per l'aggiornamento automatico della posizione dell'ambulanza.

Per prima cosa è necessario avere, come si evince dallo schema 6.2, oltre alla digital twin instance, un **IoT Hub**, che consente di acquisire elevati volumi di dati di telemetria dai device IoT per poi permetterne l'archiviazione e l'elaborazione.

Per la comunicazione tra IoT Hub e DTInstance si utilizza una **Azure Function**. Le Azure Function sono soluzioni serverless, quindi eseguite automaticamente da Azure ogni qual volta serve rispondere ad un evento definito.

Per il caso specifico si è quindi creata una funzione che esegue le seguenti operazioni in risposta ad un evento generato da un Event Grid:

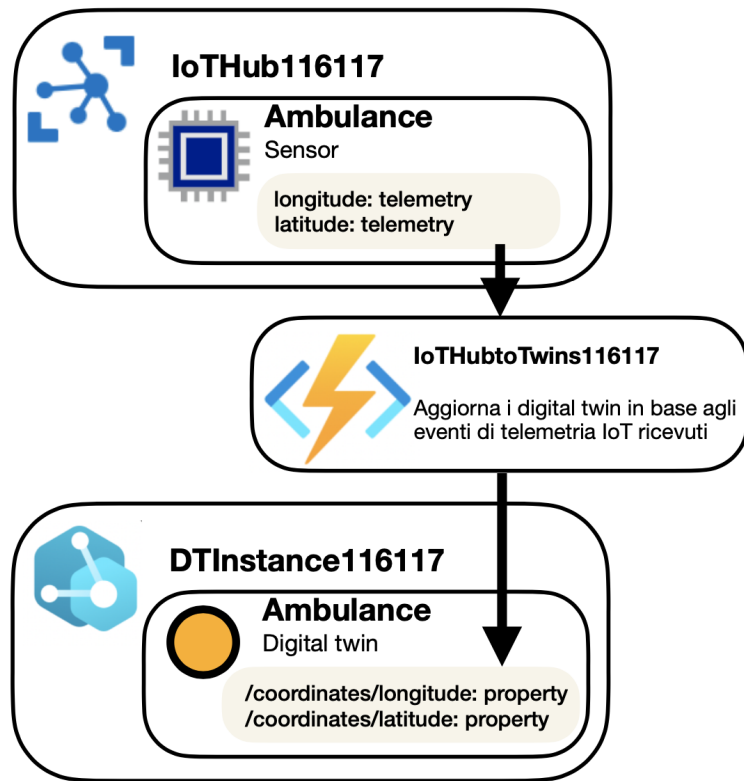


Figura 6.2: Comunicazione tra IoT Hub e digital twin instance

- crea un client che permette di accedere ai digital twin, eseguendo l'autenticazione alla DTInstance tramite managed identity;
- alla ricezione di un messaggio, che rappresenta l'evento, si deserializza e si estraggono le informazioni necessarie, in questo caso l'id del device (equivalente all'id del digital twin corrispondente) e le nuove coordinate.
- Infine si aggiorna il digital twin con i nuovi dati.

La funzione viene quindi pubblicata in una **Function App** in Azure, anch'essa deve avere le autorizzazioni necessarie per accedere alla DTInstance, come riportato nella sezione precedente 6.1.1.

Infine è necessario configurare la funzione creata come destinazione degli eventi generati dal device nell'IoT Hub. Per fare ciò bisogna, all'interno

dell'IoT Hub, creare un **event subscription** per la funzione, questa richiederà di specificare lo schema degli eventi, in tal caso Event Grid Schema, l'endpoint che è la funzione appena creata ed il tipo di eventi da filtrare, in questo caso solo gli eventi di tipo **Device Telemetry**.

6.1.3 Strumenti di analisi

Per l'analisi si propone l'utilizzo di **Time Series Insight (TSI)**. Grazie ad esso è possibile raccogliere, elaborare, archiviare ed analizzare i dati provenienti dai digital twin. Tramite TSI è possibile identificare anomalie nascoste in grandi moli di dati, che difficilmente sarebbero emerse.

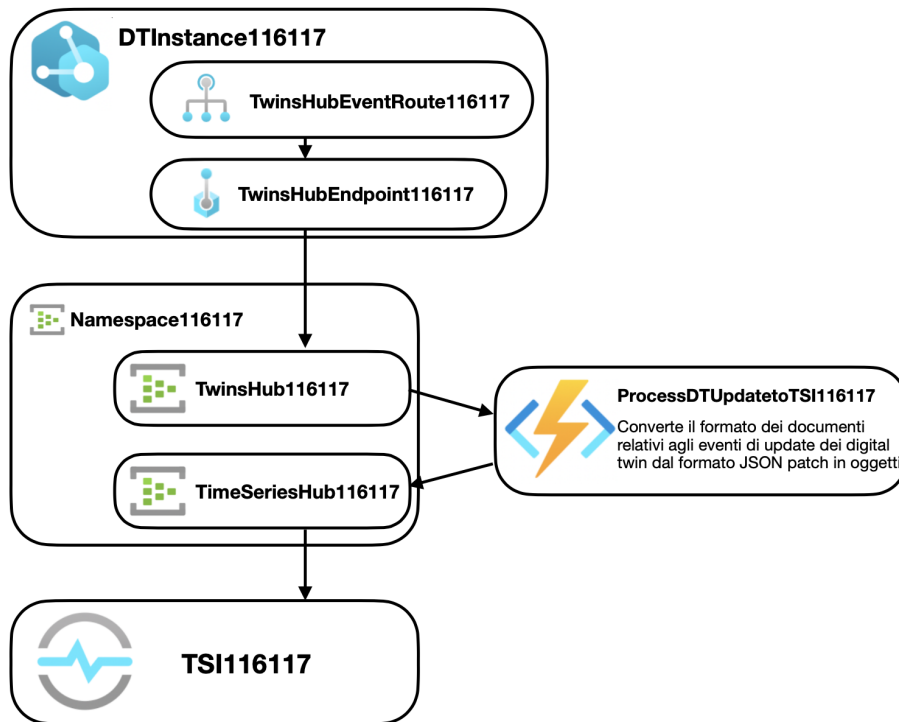


Figura 6.3: Comunicazione tra TSI e digital twin instance

Di seguito si espone brevemente il collegamento tra l'ambiente TSI e la digital twin instance, lo schema di collegamento si può vedere in figura 6.3. Per prima cosa si crea un **Namespace**, esso conterrà i due **event**

hub necessari, uno lato TSI, denominato timeserieshub116117 e uno lato DTInstance, denominato twinshub116117.

Timeserieshub116117 è utilizzato per trasmettere gli eventi a TSI mentre il secondo ha il compito di ricevere gli eventi di cambiamento di ogni digital twin. Per instradare questa tipologia di eventi verso l'event hub si creerà un **event route** all'interno della DTInstance.

Successivamente serve creare una function app che ha il compito di convertire il formato dei documenti, relativi agli eventi di update dei digital twin, dal formato JSON patch in oggetti JSON. Questa function app sarà di tipo event hub trigger. Anche in questo caso sarà necessario assegnare i ruoli di accesso in IAM e configurare le impostazioni per la connessione ai due event hub utilizzati.

Infine si crea l'istanza di Time Series Insight, specificando l'event source, in tal caso l'event hub timeserieshub116117 creato in precedenza. Per visualizzare i dati in TSI Azure mette a disposizione un ambiente chiamato TSIEplorer che consente di analizzare, archiviare e visualizzare tutti gli eventi in arrivo dall'event hub.

6.2 Struttura delle API sviluppate

Per abilitare la comunicazione tra i digital twin, ospitati dalla piattaforma Azure, ed i client che vogliono usufruire dei loro servizi è stata costruita un API in Java. Per favorire l'interoperabilità, come discusso nel capitolo 5.2, le API restituiranno in output esclusivamente risorse in formato FHIR o, eventualmente, gli id relativi.

6.2.1 Trasporti 116-117 REST API

Per l'interazione tra i digital twin e i client che necessitano di usufruire dei loro servizi sono state sviluppate delle REST API. Queste API implementano la relazione di Open Host Service che intercorre tra i tre bounded context Vehicle, Patient e Transport ed il bounded context CallCenter, per le motivazioni riportate nel capitolo 4.7.1.

Per lo sviluppo di queste API si è scelto di utilizzare il framework Spring Boot. Lo snippet di codice seguente riporta l'implementazione del controller che gestisce le richieste relative al paziente, nello specifico una post request

per la creazione di un digital twin paziente ed una get request per richiedere la lista dei digital twin paziente presenti nel sistema.

```
@RestController
public class PatientController {
    @PostMapping("/API/Patient/createPatient")
    @ResponseBody
    public String createPatient(@RequestBody String resource) {
        return PatientDigitalTwin.createPatient(resource);
    }

    @GetMapping("/API/Patient/getPatients")
    @ResponseBody
    public JSONArray getPatients() {
        JSONArray patients = new JSONArray();
        JSONParser parser = new JSONParser();
        PatientDigitalTwin.getPatients().forEach(p -> {
            try {
                patients.add(parser.parse(p));
            } catch (ParseException e) {
                e.printStackTrace();
            }
        });
        return patients;
    }
}
```

6.2.2 Interazione con Azure tramite Azure SDK

Come accennato nel capitolo 5.1.3, Azure mette a disposizione molte API per linguaggi quali C#, Java, JavaScript, Python e Go. Dunque, per creare, eliminare, modificare ed eseguire query sui digital twin è stato utilizzato il Java SDK a disposizione.

Creazione del client e autenticazione al portale Azure

Per poter accedere alla digital twin instance, all'interno del portale Azure, è necessario autenticarsi. Azure Identity mette a disposizione vari modi per farlo. In tal caso è stata registrata un'applicazione in Azure Active Directory (AAD), il servizio di gestione e protezioni delle identità di Azure. All'interno del package `azureservice`, è presente la classe `Client` incaricata della gestione del `DigitalTwinsClient` che permette di usufruire dei servizi messi a disposizione dalle API Azure citate sopra. Nello snippet di codice riportato sotto è riportata la creazione del client. Durante la creazione del client, per identificare l'applicazione registrata in AAD, si è

utilizzato il `ClientSecretCredentialBuilder()` che richiede il client secret dell'app, il suo id ed il tenant della quale fa parte. Oltre alle credenziali bisogna specificare l'endpoint che in questo caso fa riferimento alla digital twin instance.

```
private static void createClient() {
    dtClient = new DigitalTwinsClientBuilder()
        .credential(
            new ClientSecretCredentialBuilder()
                .tenantId(Constants.TENANT_ID)
                .clientId(Constants.CLIENT_ID)
                .clientSecret(Constants.CLIENT_SECRET)
                .build()
        )
        .endpoint(Constants.ENDPOINT)
        .buildClient();
}
```

Creazione di un Digital Twin

Per la creazione di un digital twin si parte dalla risorsa FHIR relativa a quel digital twin, si costruisce quindi il modello del digital twin mediante alcune classi di supporto che lo rappresentano e ne semplificano la serializzazione.

Dopodichè si utilizzano le classi messe a disposizione dall'Azure SDK per creare il digital twin. Nel seguente snippet di codice si crea un digital twin di tipo `BasicDigitalTwin` che richiede in input un id univoco. Tramite `setMetadata(BasicDigitalTwinMetadata metadata)` si specifica il modello DTDL che il digital twin dovrà avere e, tramite `addToContents(String key, Object value)`, si aggiungono le eventuali proprietà del gemello. Infine, attraverso `createOrReplaceDigitalTwin(String digitalTwinId, T digitalTwin, Class<T> clazz)`, si crea il vero e proprio digital twin all'interno della piattaforma Azure. Nel caso in cui venisse inserito un digital twin con un id già esistente questo verrebbe sostituito dal nuovo.

```
BasicDigitalTwin ambulanceDT = new BasicDigitalTwin(id)
    .setMetadata(
        new BasicDigitalTwinMetadata()
            .setModelId(Constants.AMBULANCE_MODEL_ID)
    )
    .addToContents("state", state.getValue())
    .addToContents("coordinates", coordinates);

BasicDigitalTwin basicTwinResponse = Client.getClient()
    .createOrReplaceDigitalTwin
        (id, ambulanceDT, BasicDigitalTwin.class);
```

Creazione di una relazione tra due Digital Twin

Una relazione tra due digital twin si crea con la stessa logica riportata nel punto precedente. La relazione deve necessariamente essere definita all'interno del modello DTDL.

Per creare la relazione si utilizza `createOrReplaceRelationship` (`String digitalTwinId`, `String relationshipId`, `T relationship`, `Class<T> clazz`) che richiede in input l'id del digital twin a cui si fa riferimento e l'id della relazione, vuole poi il modello della relazione che comprende l'id del digital twin source e target e l'id e il nome della relazione che si sta creando. È anche possibile aggiungere proprietà alle relazioni.

```
public static void addOperatorWorkDay(
    String ambulanceId, String operatorId, LocalDate date){
    String relationshipId =
        ambulanceId + "DriveBy" + operatorId;
    Client.getClient().createOrReplaceRelationship(
        ambulanceId,
        relationshipId,
        new BasicRelationship(
            relationshipId,
            ambulanceId,
            operatorId,
            Constants.AMBULANCE_OPERATOR_REL
        )
        .addProperty("workDate", date),
        BasicRelationship.class
    );
}
```

Registrazione di un device nel IoT Hub

Durante la creazione di un digital twin che riceve telemetrie in real-time dall'IoT Hub è necessario creare anche il relativo device. Nello stato attuale del sistema Trasporti 116-117 l'unico caso in cui è necessario compiere questa azione è durante la creazione del digital twin ambulanza, dove serve registrare il device che invia le telemetrie relative al GPS dell'ambulanza.

Per prima cosa si crea il `RegistryManager`, che richiede la stringa di connessione dell'IoT Hub; esso permette di gestire i device all'interno dell'Hub. Successivamente si controlla che il device non esista già all'interno dell'Hub ed infine si registra il device con il metodo `registryManager.addDevice` (`Device device`) e si chiude la connessione.

```
public static void addAmbulanceGPSDevice(String ambulanceId)
    throws Exception {
```

```

RegistryManager registryManager = RegistryManager
    .createFromConnectionString(
        Constants.IOT_HUB_CONNECTION_STRING);
Device device =
    Device.createFromId(ambulanceId, null, null);

if(registryManager.getDevices(Constants.MAX_DEVICES)
    .stream().noneMatch(d -> d.getDeviceId()
        .equals(ambulanceId)){
    try {
        registryManager.addDevice(device);
    } catch (IotHubException | IOException e) {
        e.printStackTrace();
    }
}
registryManager.close();
}

```

Eseguire query sul Twin Graph

In Azure è possibile eseguire query sul twin graph, grafo costituito da digital twin e dalle relazioni che li lega. Queste query sono scritte in un linguaggio SQL-like chiamato Azure Digital Twin query language. Si può utilizzare questo linguaggio per recuperare proprietà, modelli e relazioni all'interno del twin graph. Una volta scritta la query, basta eseguirla tramite il metodo `query(String query, Class<T> clazz)` fornito dal client. Questo metodo vuole in input la stringa rappresentante la query ed il modello della classe in cui si vuole deserializzare la risposta alla query. All'interno dell'API, come accennato sopra, sono forniti dei modelli che rappresentano i digital twin e ne facilitano la serializzazione e la deserializzazione. Infine si mappa la risposta della query in risorse FHIR tramite, questo caso, il metodo `FHIRAmbulanceResource.createFHIRResource(AmbulanceDtModel dt)`, per i motivi sopracitati.

```

public static List<String> getAmbulances() {
    List<String> ambulances = new ArrayList<>();
    String query = "SELECT * FROM DIGITALTWINS " +
        "WHERE IS_OF_MODEL('" +
        Constants.AMBULANCE_MODEL_ID + "')";
    PagedIterable<AmbulanceDtModel> response =
        Client.getClient()
            .query(query, AmbulanceDtModel.class);
    response.forEach(dt -> ambulances.add(
        FHIRAmbulanceResource.createFHIRResource(dt)));
    return ambulances;
}

```

Eliminazione di un Digital Twin

Per quanto riguarda l'eliminazione dei digital twin, non sono state previste API che la permettessero. È stata fatta questa scelta in quanto il ciclo di vita dei digital twin va oltre alla fine del ciclo di vita del relativo asset fisico, questo per favorire l'analisi del sistema. In caso di errori di inserimento sarà comunque possibile eliminare i digital twin utilizzando altri canali.

6.2.3 HAPI FHIR

Per facilitare la serializzazione e la deserializzazione delle risorse FHIR si utilizza **HAPI FHIR**.

HAPI FHIR è un'implementazione della specifica FHIR per Java, la libreria definisce classi per ogni tipo di risorsa e tipo di dato definito dalla specifica. HAPI FHIR richiede la creazione di un context nel quale serve specificare la versione FHIR utilizzata, in tal caso è stata utilizzata la versione R4 in quanto è la versione corrente proposta da HL7. L'API fornisce una moltitudine di servizi, in particolare è stato utilizzato il parser che permette di creare facilmente JSON in formato FHIR ed inoltre mette a disposizione strumenti per verificarne la correttezza.

All'interno del package `fhirSerializer` sono presenti una serie di metodi che permettono di convertire i modelli rappresentanti i digital twin in risorse FHIR.

6.3 Set di API sviluppate

Il set di API implementato è limitato in quanto è stato creato a scopo prototipale. Questo può essere facilmente esteso in caso di necessità.

Di seguito è riportato il set di API sviluppate suddiviso nei tre bounded context principali:

6.3.1 Bounded Context Paziente

- `/API/Patient/createPatient`: crea il digital twin del paziente partendo dalla relativa risorsa FHIR.
- `/API/Patient/getPatients`: restituisce tutti i pazienti all'interno del sistema codificati in risorse FHIR.

- `/API/Patient/getPatient`: prende in input l'id del paziente e restituisce la risorsa FHIR relativa ad esso.

6.3.2 Bounded Context Veicolo

- `/API/Vehicle/createAmbulance`: crea il digital twin dell'ambulanza partendo dalla relativa risorsa FHIR.
- `/API/Vehicle/createOperator`: crea il digital twin dell'operatore partendo dalla relativa risorsa FHIR.
- `/API/Vehicle/addOperatorWorkDay`: prende in input l'id dell'ambulanza e dell'operatore e una data. Assegna l'operatore all'ambulanza per l'intera giornata lavorativa nella data specificata. Alla relazione ambulanza/operatore viene assegnata una proprietà contenente il giorno lavorativo, in questo modo si tiene traccia di tutti gli operatori che hanno guidato l'ambulanza in determinate date, questo può servire a scopo di analisi o in caso di problemi.
- `/API/Vehicle/setAmbulanceUnderMaintenance`: imposta lo stato dell'ambulanza "in manutenzione".
- `/API/Vehicle/setAmbulanceFree`: imposta lo stato dell'ambulanza in "libero".
- `/API/Vehicle/setAmbulanceDisused`: imposta lo stato dell'ambulanza in "disuso", ovvero quando l'ambulanza viene dismessa.
- `/API/Vehicle/setAmbulanceBusy`: imposta lo stato dell'ambulanza in "occupata", ovvero sta eseguendo un trasporto.
- `/API/Vehicle/getAmbulances`: restituisce tutte le ambulanze all'interno del sistema codificate in risorse FHIR.
- `/API/Vehicle/getFreeAmbulance`: prende in input l'ora di inizio e l'ora di fine del presunto trasporto. Controlla se all'interno del sistema ci sono ambulanze libere nello slot di tempo indicato e disponibili a compiere il trasporto (quindi non in manutenzione o in disuso). In caso affermativo restituisce un'ambulanza che può compiere il trasporto.

- `/API/Vehicle/getOperators`: restituisce tutti gli operatori all'interno del sistema codificati in risorse FHIR.
- `/API/Vehicle/getAmbulance`: prende in input l'id dell'ambulanza e restituisce la risorsa FHIR relativa ad essa.
- `/API/Vehicle/getOperator`: prende in input l'id dell'operatore e restituisce la risorsa FHIR relativa ad esso.

6.3.3 Bounded Context Trasporto

- `/API/Transport/createTransport`: crea il digital twin del processo di trasporto partendo dalla relativa risorsa FHIR. Il trasporto viene creato in fase "prenotato".
- `/API/Transport/setTransportInProgress`: imposta il trasporto nella fase "in esecuzione".
- `/API/Transport/setTransportCompleted`: imposta il trasporto nella fase "completato". Quando si passa in questo stato si aggiorna in automatico anche l'orario effettivo di fine trasporto.
- `/API/Transport/setTransportDeleted`: imposta il trasporto nella fase "cancellato".
- `/API/Transport/getScheduledTransports`: restituisce tutti i trasporti prenotati all'interno del sistema codificati in risorse FHIR.
- `/API/Transport/getInProgressTransports`: restituisce tutti i trasporti in corso all'interno del sistema codificati in risorse FHIR.
- `/API/Transport/getCompletedTransports`: restituisce tutti i trasporti completati all'interno del sistema codificati in risorse FHIR.
- `/API/Transport/getCancelledTransports`: restituisce tutti i trasporti cancellati all'interno del sistema codificati in risorse FHIR.

Capitolo 7

Testing

7.1 ADT Explorer

Per testare l'infrastruttura si è seguita la sequenza consigliata in [22] e riportata in figura 7.1.

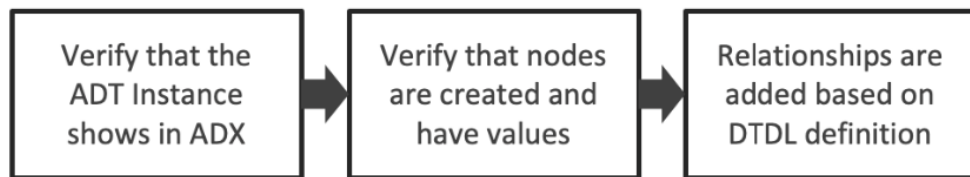


Figura 7.1: Test funzionale proposto in [22]

Nel primo step serve accedere a **ADT Explorer**, se si riesce ad accedere significa che la configurazione dell'ambiente è andata a buon fine e i vari permessi sono stati assegnati correttamente.

Nel secondo step si controlla che i nodi creati contengano i giusti valori, se questi valori non sono corretti significa che c'è qualche problema all'interno dei modelli DTDL.

Nel terzo step si controlla se le relazioni tra i digital twin sono corrette. Il completamento dei tre step conferma la correttezza infrastrutturale dell'ambiente.

Tramite ADT Explorer sono stati testati anche alcuni servizi forniti dalle API. Esso infatti permette di visualizzare i modelli e le istanze dei

digital twin e di creare, aggiornare o eliminare digital twin. Inoltre mette a disposizione un ambiente di esecuzione per le query.

ADT Explorer è stato utilizzato per provare la correttezza del digital twin graph dopo avere eseguito gli scenari tramite le API. In figura 7.2 si vede il digital twin graph dopo la prenotazione di un trasporto, in figura 7.3, invece, si vede il digital twin graph durante un trasporto in corso.

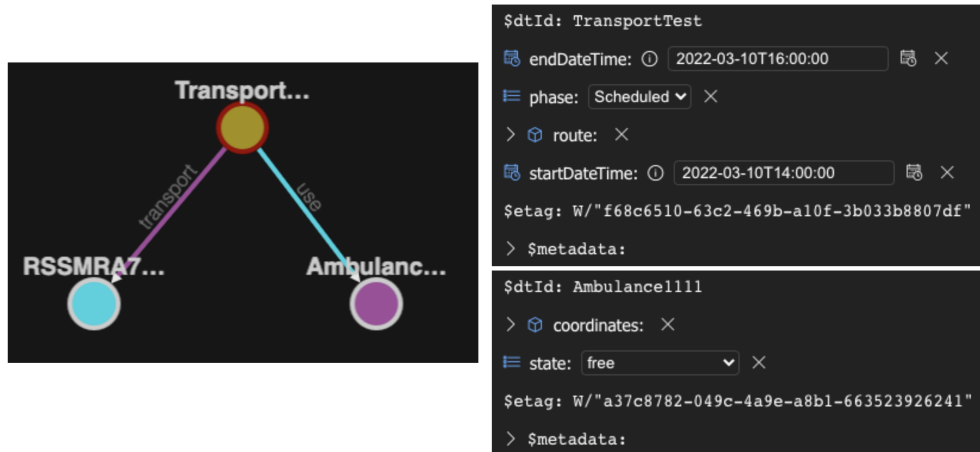


Figura 7.2: Digital twin graph con un trasporto prenotato

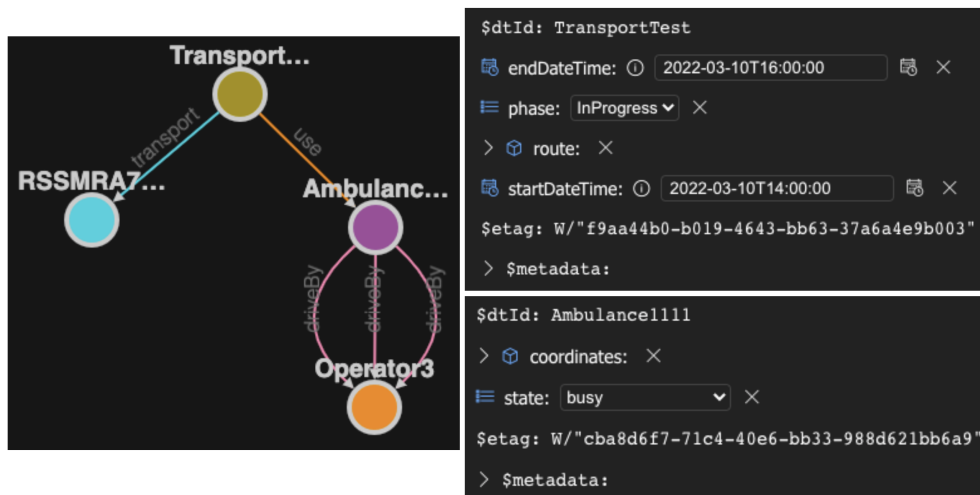


Figura 7.3: Digital twin graph durante un trasporto in corso

7.2 Unit testing

Parte dei servizi forniti dalle API sono stati testati mediante JUnit, un framework di testing per il linguaggio Java. Tramite la continuous integration con le GitHub Actions, ad ogni aggiornamento delle API, si eseguono automaticamente i test sopracitati in ambienti Linux, MacOS e Windows.

7.3 Testing Rest API tramite Postman

Per testare il set di Rest API sviluppate è stato utilizzato Postman, un tool che permette di eseguire facilmente richieste HTTP ad un qualsiasi server. Questo ha consentito un test delle API potente e veloce.

7.4 Invio dati IoT simulati all'ambulanza

Per testare la function app che instrada i dati da un dispositivo IoT, ovvero il gps all'interno dell'ambulanza, verso il relativo digital twin serviva simulare un flusso di telemetrie verso l'IoT hub. Per generare il flusso di telemetrie è stato customizzato il `DeviceSimulator`, un semplice programma in C#, fornito da Azure. Al suo interno si generano, ogni secondo, nuovi valori di longitudine e latitudine e li si inviano ad un IoT device specificato all'interno del programma. L'invio di queste telemetrie genereranno eventi all'interno dell'EventHub che attiveranno la function app citata sopra.

Capitolo 8

Validazione e discussione

Per la validazione del sistema sviluppato è stato implementato un client di esempio. In questo capitolo si illustreranno le funzionalità di questo client, inoltre si riporteranno altri client ipotetici che possono usufruire dei servizi messi a disposizione dall'API.

8.1 Client di esempio - Call Center Client

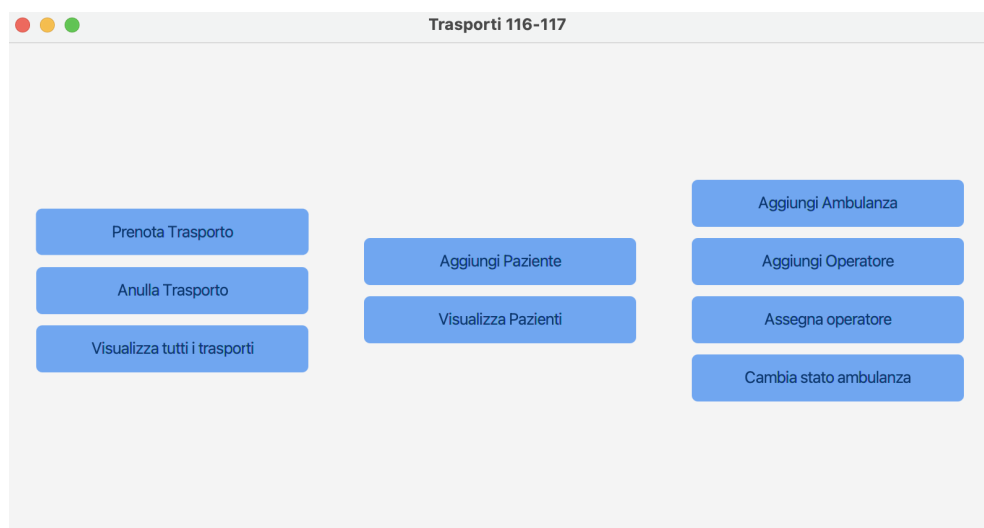


Figura 8.1: Call Center client

Call Center Client rappresenta un ipotetico client utilizzato all'interno della centrale operativa dei trasporti 116-117. Il client, tra le varie funzionalità, permette di eseguire le relazioni tra operatore centralino e il sistema individuate nello storytelling in figura 4.6 e 4.8.

Tramite l'applicativo, come si evince in figura 8.1, l'operatore può svolgere le seguenti operazioni:

- **prenotare un trasporto:** come si può vedere in figura 8.2, l'operatore deve inserire la partenza e la destinazione, la data del trasporto con l'orario di inizio e l'orario di fine ipotizzato ed il paziente che usufruisce del trasporto. Tramite il pulsante **Controlla ambulanza disponibile** l'operatore può verificare la disponibilità di ambulanze per la fascia oraria indicata;
- **annullare un trasporto:** l'operatore può cancellare una prenotazione di un trasporto;
- **visualizzare tutti i trasporti;**
- **aggiungere un nuovo paziente:** come si vede in figura 8.3, l'operatore deve inserire i dati anagrafici del paziente e le condizioni del paziente che includono sia stato di salute che autonomia, in particolare deve inserire il codice FHIR che rappresenta la sue condizioni;
- **visualizzare tutti i pazienti;**
- **aggiungere una nuova ambulanza:** permette di registrare una nuova ambulanza inserendo il suo id;
- **aggiungere un nuovo operatore:** come per il paziente, l'operatore centralino può inserire un nuovo operatore ambulanza specificando i suoi dati anagrafici;
- **assegnare i giorni lavorativi dell'operatore:** l'operatore, tramite la scheda in figura 8.4, può assegnare i giorni lavorativi all'operatore, per farlo deve assegnare l'operatore ad un ambulanza per una determinata data;
- **cambiare lo stato dell'ambulanza:** l'operatore, tramite la scheda in figura 8.5, può anche cambiare lo stato dell'ambulanza qualora questa venga dismessa oppure inizi o termini una manutenzione.

Prenotazione trasporto

Partenza

Via

Numero

Città

Provincia

Cap

Arrivo

Via

Numero

Città

Provincia

Cap

Data trasporto

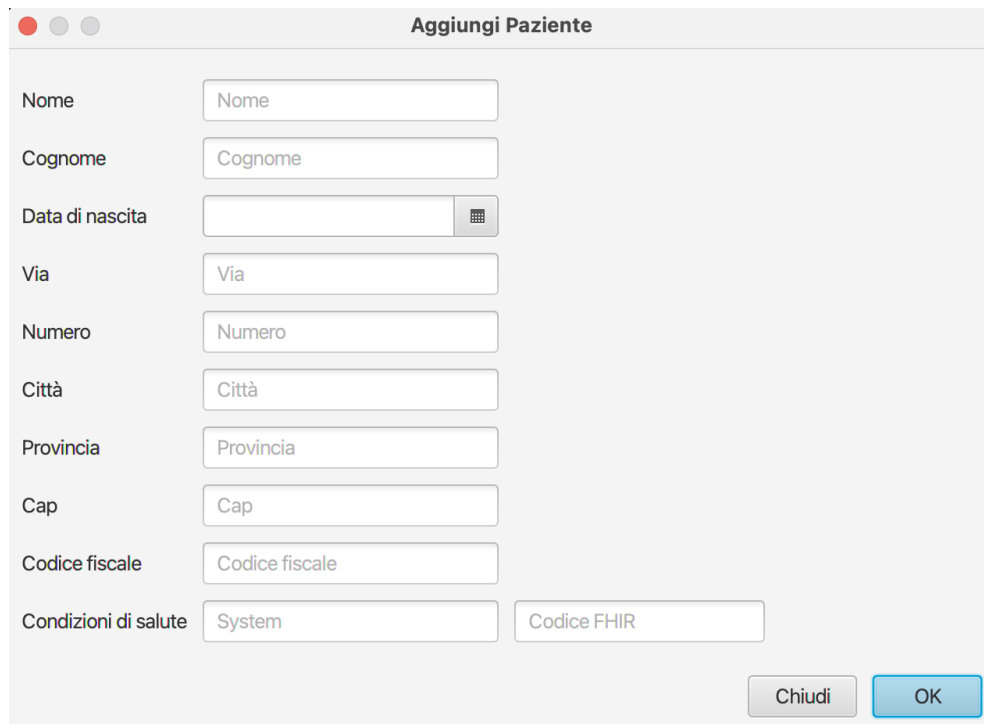
Ora inizio trasporto

Ora fine trasporto prevista

Ambulanza

Paziente

Figura 8.2: Scheda di prenotazione trasporto

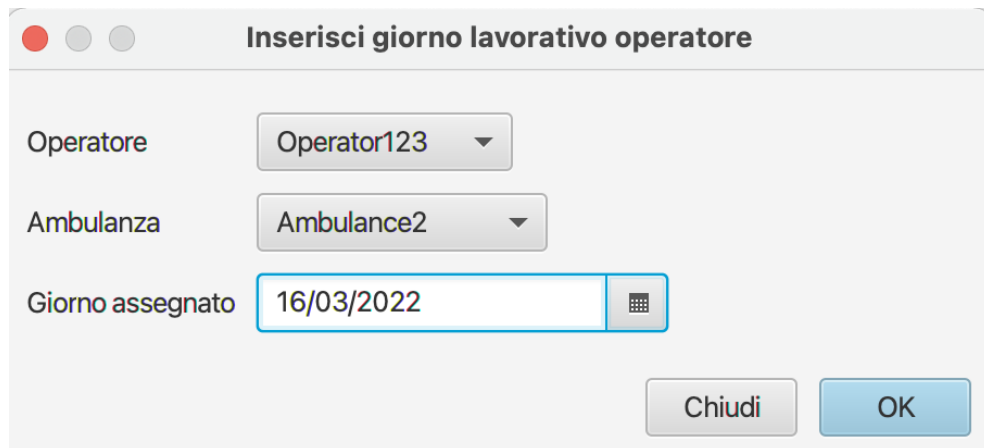


The screenshot shows a window titled "Aggiungi Paziente" with a standard macOS-style title bar (red, yellow, and green buttons). The form contains the following fields:

- Nome: Text input field with placeholder "Nome".
- Cognome: Text input field with placeholder "Cognome".
- Data di nascita: Text input field with a calendar icon on the right.
- Via: Text input field with placeholder "Via".
- Numero: Text input field with placeholder "Numero".
- Città: Text input field with placeholder "Città".
- Provincia: Text input field with placeholder "Provincia".
- Cap: Text input field with placeholder "Cap".
- Codice fiscale: Text input field with placeholder "Codice fiscale".
- Condizioni di salute: A dropdown menu showing "System" and a text input field for "Codice FHIR".

At the bottom right, there are two buttons: "Chiudi" (grey) and "OK" (blue).

Figura 8.3: Scheda di aggiunta nuovo paziente



The screenshot shows a window titled "Inserisci giorno lavorativo operatore" with a standard macOS-style title bar. The form contains the following fields:

- Operatore: A dropdown menu with "Operator123" selected.
- Ambulanza: A dropdown menu with "Ambulance2" selected.
- Giorno assegnato: A text input field containing "16/03/2022" and a calendar icon on the right.

At the bottom right, there are two buttons: "Chiudi" (grey) and "OK" (blue).

Figura 8.4: Scheda di aggiunta giorno lavorativo operatore

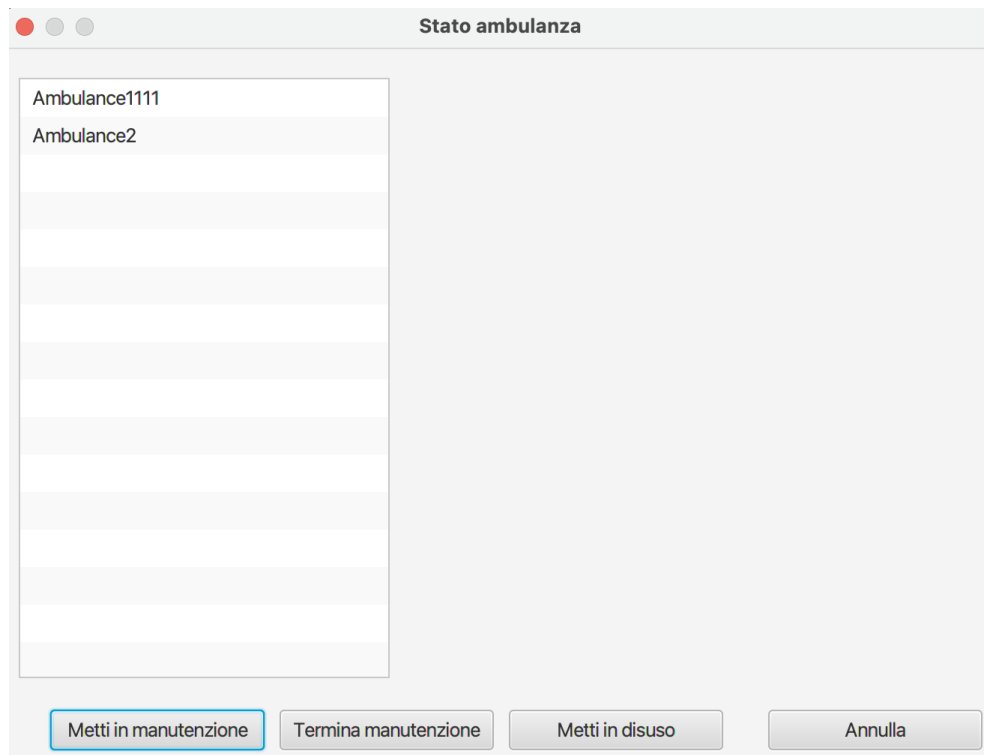


Figura 8.5: Scheda per il cambio stato dell'ambulanza

8.2 Altri ipotetici client

Di seguito si illustrano altri client, non implementati, che potrebbero usufruire delle API sviluppate.

8.2.1 Vehicle client

Tutte le ambulanze all'interno del sistema trasporti 116-117 sono dotate di un tablet, dunque si suppone che questo client sia quello presente all'interno del veicolo di trasporto, quindi utilizzato dall'operatore che guida il veicolo. Esso permette di eseguire le relazioni tra operatore trasporto e il sistema riportate nel domain storytelling i figura 4.7 e 4.8. Nello specifico può compiere le seguenti attività:

- visualizzare i trasporti da eseguire nella giornata;

- **visualizzare in dettaglio il prossimo trasporto da eseguire:** l'operatore può quindi visualizzare la partenza e la destinazione ed i dati del paziente da trasportare;
- **confermare l'inizio di un trasporto:** confermando l'inizio di un trasporto si cambia lo stato del trasporto in "in corso" e quello dell'ambulanza in "occupata". Inoltre si sostituisce l'ora di inizio trasporto programmata con quella effettiva;
- **confermare la conclusione di un trasporto:** confermando l'inizio di un trasporto si cambia lo stato del trasporto in "completato" e quello dell'ambulanza in "libera", poiché ora può compiere nuovi trasporti. Inoltre si sostituisce l'ora di fine trasporto programmata con quella effettiva;
- **cambiare lo stato dell'ambulanza:** l'operatore può anche cambiare lo stato dell'ambulanza in "manutenzione" durante il rifornimento o qualora questa presenti guasti;
- **segnalare un emergenza:** all'interno del sistema è presente un tasto di emergenza, con la pressione di questo tasto viene in automatico allertata la centrale operativa che potrà individuare immediatamente la posizione dell'ambulanza. La segnalazione di emergenza parte anche qualora l'ambulanza non risulti più online e l'operatore non è raggiungibile, in questo caso si prenderà in considerazione l'ultima posizione rilevata.

8.2.2 User client

Questo client fa riferimento ad una piattaforma nel quale l'utente può autonomamente prenotare un trasporto. Questa piattaforma permette di ottimizzare la prenotazione lato utente e di ridurre il flusso di chiamate verso la centrale operativa. Tramite questa piattaforma l'utente può:

- **autenticarsi;**
- **registrarsi:** nel caso il paziente non sia presente all'interno del sistema;
- **prenotare un trasporto;**

- **anullare un trasporto prenotato;**
- **controllare le disponibilità delle ambulanze.**

8.2.3 Sistema 118

Il sistema 118, come riportato nel capitolo 4, necessita di uno stretto contatto con il sistema 116-117. Una necessità individuata durante l'analisi è che le ambulanze tra i due sistemi devono poter essere intercambiabili in caso di necessità. Tramite le API fornite, il 118 può controllare direttamente all'interno del sistema 116-117 la presenza di ambulanze disponibili, senza la necessità di contattare la centrale operativa, inoltre può autonomamente prenotarla per un certo lasso di tempo, ottimizzando i tempi. Inoltre il 118 potrebbe aver bisogno di accedere ai trasporti relativi ad un determinato paziente.

8.3 Ulteriori query sul Digital Twin Graph

Come detto in precedenza, il set di API sviluppate riportato nel capitolo 6.3 è limitato. Le API non sono l'unico punto di ingresso all'ecosistema di digital twin creato, Azure infatti mette a disposizione vari modi per l'accesso a quest'ultimo. In particolare permette di eseguire qualsiasi tipo di query sul digital twin graph, a patto di essere in possesso dei giusti livelli di accesso.

Per ottenere informazioni sui twins o sulle loro relazioni è possibile scrivere query in un linguaggio SQL-like personalizzato chiamato **Azure Digital Twins query language**. Per eseguire queste query si può utilizzare l'Azure Digital Twins Query API, un modo per utilizzare questa API è tramite l'utilizzo di un SDKs per Azure Digital Twins, come è stato fatto per le API sviluppate, ma in alternativa è possibile eseguirle tramite la CLI di Azure o l'ADTExplorer citato in precedenza.

Grazie a questo è possibile ricavare qualsiasi tipo di informazione necessaria dal digital twin graph, anche se questa non è recuperabile tramite le API Trasporti 116-117.

8.4 Scelta e costi della piattaforma

Per lo sviluppo prototipale del progetto Trasporti 116-117 si è scelto di utilizzare la piattaforma Azure. Ciò non preclude la scelta futura di migrare l'ecosistema sviluppato su altre piattaforme cloud o su una costruita ad-hoc. Come riportato nel libro *“Building Industrial Digital Twins”* [22], per costruire una piattaforma IoT sono necessarie alcune funzionalità di base, come la capacità di elaborazione, di archiviazione, di rete e la connettività, queste funzionalità sono fornite da molte piattaforme cloud, il che renderebbe più semplice acquistare questi servizi invece che costruirli. Ovviamente le due soluzioni hanno pro e contro, dunque è necessario fare una valutazione accurata delle due opzioni prima di costruire una soluzione basata su digital twin.

Per quanto riguarda le piattaforme cloud ci sono varie alternative tra cui è possibile scegliere, ad esempio, Microsoft Azure, Amazon Web Services (AWS), Google Cloud Platform (GCP), Oracle Cloud Infrastructure (OCI) o IBM Cloud. Prendiamo in esempio Azure, il suo costo è relativo al flusso di dati in entrata ed in uscita, dunque durante la fase di analisi serve stimare la mole di dati che interessa il sistema, in questo modo sarà possibile capire se è meglio scegliere una soluzione cloud a consumo, come nel caso di Azure, oppure se conviene la costruzione di una soluzione ad-hoc con alti costi iniziali ma con un futuro risparmio.

Le piattaforme cloud mettono a disposizione molti altri servizi di storage, sicurezza, analisi ed altro. In Azure questi servizi sono modulari ed è possibile aggiungerli alla propria soluzione in base alle esigenze, ogni servizio ha le proprie tariffe.

In definitiva quindi non esiste una soluzione migliore rispetto ad un'altra, tutto dipende dalle necessità che emergono durante la fase di analisi.

Capitolo 9

Conclusioni

Nonostante i digital twin siano una tecnologia applicata concretamente solamente negli ultimi anni, il mondo tecnologico si sta mettendo in moto per applicarla negli ambiti più disparati. Durante il lavoro svolto è risultata evidente l'importanza e la rilevanza che i digital twin possono assumere nell'ambito healthcare, è possibile, infatti, trovare in letteratura progetti che fanno riferimento a vari contesti all'interno della sanità digitale.

Lo scopo di questo lavoro era capire le potenzialità dell'utilizzo di questa tecnologia applicata al caso di studio Trasporti 116-117. Dopo lo sviluppo di un primo prototipo si può confermare il potenziale che i digital twin possono avere applicati a questo dominio; questo approccio può essere un punto di svolta per la risoluzione dei problemi sollevati dall'esperto di dominio e approfonditi durante la fase di analisi.

Dato che l'analisi del dominio è stata effettuata in parte su informazioni reali e in parte su informazioni ipotizzate, si suppone che ne sia uscita una rappresentazione del dominio semplificata rispetto alla realtà. Si prevede che dopo una complicazione del dominio derivata da ulteriori interazioni con l'esperto i vantaggi offerti dal sistema realizzato saranno ancora più evidenti.

Durante il processo di sviluppo del caso di studio si è scelto un approccio domain driven, questo ha permesso di analizzare gradualmente il dominio così da individuare i suoi punti salienti. Questo tipo di approccio ha permesso anche di produrre un'analisi e una progettazione ben definita e facilmente estensibile, questo ha portato di conseguenza ad un'implementazione pulita e veloce, senza incorrere in problemi consistenti durante questa fase.

Anche la scelta dello sviluppo in cloud, per la precisazione Azure, si è rivelata una scelta corretta. Inizialmente è stata necessaria una rilevante fase di apprendimento che permettesse di comprendere i servizi necessari per lo sviluppo del progetto; una volta chiari però la piattaforma permette di velocizzare in maniera significativa l'implementazione. Un punto di forza dell'utilizzo di una piattaforma cloud è sicuramente la scalabilità. Inoltre Azure ha un occhio di riguardo per la sicurezza, punto fondamentale per l'ambito healthcare.

9.1 Miglioramenti futuri

Il caso di studio è molto ampio e ci sono molte questioni meritevoli di approfondimento. La prima è sicuramente quella relativa alla sicurezza e privacy del sistema sviluppato che per la sua messa in opera necessita di un sistema di autenticazione con diversi livelli per gli utenti finali che lo andranno ad utilizzare, come detto nel capitolo 5.3.

Il prototipo sviluppato vuole essere una base per molti ampliamenti futuri. Un miglioramento importante è il passaggio da un'assegnazione statica dei trasporti alle ambulanze ad un'assegnazione dinamica calcolata in base alla posizione dei mezzi e alla centrale di appartenenza. Questo permetterebbe di ottimizzare i percorsi, con un conseguente risparmio di tempi e risorse.

Miglioramenti futuri possono essere fatti in ambito di analisi, simulazioni e predizioni. Per definizione i digital twin sono strumenti potenti per la simulazione e predizione di scenari futuri. Per l'organizzazione all'interno del dominio questa è una funzionalità importantissima, in quanto permetterebbe di mantenere ed allocare le giuste risorse, sia in termini di personale sia in termini di mezzi. Tramite tecniche di intelligenza artificiale il sistema potrebbe consigliare quali sono le risorse necessarie in determinate fasce orarie, evitando sia situazioni di penuria sia situazioni di esubero di queste. Il sistema è anche utile per individuare punti critici e risolvere eventuali problemi.

Bibliografia

- [1] Fhir (fast healthcare interoperability resources). <http://hl7.org/fhir/>.
- [2] Numero unico per cure non urgenti 116117. <https://www.salute.gov.it/>.
- [3] Snomed ct. <https://www.snomed.org/>.
- [4] Swedish digital twin consortium. <https://www.sdtc.se/>.
- [5] B. Björnsson, C. Borrebaeck, N. Elander, T. Gasslander, D. R. Gawel, M. Gustafsson, R. Jörnsten, E. J. Lee, X. Li, S. Lilja, D. Martínez-Enguita, A. Matussek, P. Sandström, S. Schäfer, M. Stenmarker, X. F. Sun, O. Sysoev, H. Zhang, M. Benson, and . S. D. T. Consortium. Digital twins to personalize medicine. *Genome medicine*, 2019.
- [6] Y. C. Caiming Zhang. A review of research relevant to the emerging industry trends: Industry 4.0, iot, blockchain,and business analytics. 2020.
- [7] J. Corral-Acero, F. Margara, M. Marciniak, C. Rodero, F. Loncaric, Y. Feng, A. Gilbert, J. F. Fernandes, H. A. Bukhari, A. Wajdan, M. V. Martinez, M. S. Santos, M. Shamohammdi, H. Luo, P. Westphal, P. Leeson, P. DiAchille, V. Gurev, M. Mayr, L. Geris, P. Pathmanathan, T. Morrison, R. Cornelussen, F. Prinzen, T. Delhaas, A. Doltra, M. Sitges, E. J. Vigmond, E. Zacur, V. Grau, B. Rodriguez, E. W. Remme, S. Niederer, P. Mortier, K. McLeod, M. Potse, E. Pueyo, A. Bueno-Orovio, and P. Lamata. The Digital Twin to enable the vision of precision cardiology. *European Heart Journal*, 2020.

-
- [8] A. Crooks, K. Schechtner, A. K. Dey, and A. Hudson-Smith. Creating smart buildings and cities. *IEEE Pervasive Computing*, 2017.
 - [9] F. Dembski, U. Wössner, M. Letzgus, M. Ruddat, and C. Yamu. Urban digital twins for smart cities and citizens: The case study of herrenberg, germany. *Sustainability*, 2020.
 - [10] A. Fuller, Z. Fan, C. Day, and C. Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 2020.
 - [11] E. H. Glaessgen and D. Stargel. The digital twin paradigm for future nasa and u.s. air force vehicles. 2012.
 - [12] M. Grieves. Digital twin: manufacturing excellence through virtual factory replication. 2014.
 - [13] M. Lehne, J. Sass, A. Essenwanger, J. Schepers, and S. Thun. Why digital medicine depends on interoperability. *npj Digital Medicine*, 2019.
 - [14] Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. J. Deen. A novel cloud-based framework for the elderly healthcare services using digital twin. *IEEE Access*, 2019.
 - [15] A. M. Madni, C. C. Madni, and S. D. Lucero. Leveraging digital twin technology in model-based systems engineering. *Systems*, 2019.
 - [16] A. F. Mendi, T. Erol, and D. Dogan. Digital twin in the military field. *IEEE Internet Computing*, 2021.
 - [17] N. Mohammadi, J. Taylor, and A. Vimal. Knowledge discovery in smart city digital twins. *53rd Hawaii International Conference on System Sciences*, 2020.
 - [18] S. Montagna, A. Croatti, A. Ricci, V. Agnoletti, V. Albarello, and E. Gamberini. Real-time tracking and documentation in trauma management. *Health informatics journal*, 2019.
 - [19] F. Pires, A. Cachada, J. Barbosa, A. P. Moreira, and P. Leitão. Digital twin in industry 4.0: Technologies, applications and challenges. *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, 2019.

-
- [20] A. Ricci, A. Croatti, and S. Montagna. Pervasive and connected digital twins - a vision for digital health. *IEEE Internet Computing*, 2021.
- [21] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, and D. Devine. Digital twin: Origin to future. *Applied System Innovation*, 2021.
- [22] S. Varan Nath and P. van Schalkwyk. *Building Industrial Digital Twins. Design, develop, and deploy digital twin solutions for real-world industries using Azure Digital Twins*. Packt Publishing, 2021.
- [23] Q. Wang, W. Jiao, P. Wang, and Y. Zhang. Digital twin for human-robot interactive welding and welder behavior analysis. *IEEE/CAA Journal of Automatica Sinica*, 2021.